Université Sorbonne Paris Nord

Laboratoire d'Informatique de Paris Nord

Faster Algorithms for Approximating Combinatorial and Geometric Data

Thèse de doctorat présentée par Alexandre Louvet

pour l'obtention du grade de Docteur en Informatique

soutenue le 16 juillet 2025 devant le jury d'examen constitué de

Victor-Emmanuel Brunel	École Nationale de la Statistique et de l'Administration Economique	Directeur
Mónika Csikós	Université Paris Cité	Examinatrice
Yan Gérard	Université Clermont Auvergne	Examinateur
Guillaume Lecué	École Supérieure des Sciences Economiques et Commerciales	Examinateur
Nabil Mustafa	Université Sorbonne Paris Nord	Directeur
János Pach	Rényi Institute of Mathematics	Examinateur
Sophie Toulouse	Université Sorbonne Paris Nord	Examinatrice

Rapporteurs

Sergio Cabello	University of Ljubljana	Rapporteur
Jeff M. Phillips	University of Utah	Rapporteur

Acknowledgments

I am really grateful to have received the guidance of my advisors Nabil and Victor. Thanks to both of you for your continuous support throughout this journey. Thank you for introducing me to many fascinating problems. Lastly, thank you for the captivating discussions and really helpful advice both on and off research.

I would like to thank the ANR for the ADDS grant that funded the first three years of this thesis and Villetaneuse's IUT who enabled me to pursue a 4th year on my PhD. I am also really grateful to János Pach who hosted me at the Renyi Institute of Mathematics during a semester and to all the people I met and worked with there.

I would like to thank Sergio Cabello and Jeff Phillips for accepting to review my manuscript as well as my jury: Mónika Csikós, Yan Gérard, Guillaume Lecué, János Pach and Sophie Toulouse for participating to my defense.

Of course this adventure would not have been possible without my family and friends. Thank you all for the bike trips, the walks, the meals, the games, the chats and all these other ordinary but wonderful times spent together that helped me to push through the hardships.

Abstract

High-dimensional data are common in fields like genomics, image processing, and social network analysis. They present significant challenges in computational and statistical analysis. Datasets of larger and larger size in these different fields increases the need for efficient data approximation tools. Even if the dimension of data might be large, their intrinsic dimension, that is, the minimal dimension of the information they bear might be small. This is represented through concepts such as VC-dimension that we discuss in this work. We focus our work on finding efficient algorithm for large datasets of fixed intrinsic dimension.

In this thesis, we present algorithms to compute some fundamental structures of combinatorial data reduction: ε-approximations, δ-coverings, low-discrepancy colorings and low-crossing partitions. In particular we study the following three problems.

In the first part, we present a new two-player game and show the existence of an almost optimal strategy using the celebrated Lovett-Meka discrepancy algorithm. We then present a multiplicative weights algorithms to compute a family of small average discrepancy colorings that uses our game's value in its analysis.

In the second part, we present a new low-crossing partition algorithm for general set systems. This is the first instance of a practically fast algorithm to compute low-crossing partitions for general set systems, as previous results are limited to set systems spanned by halfspaces in low dimension.

Finally, we present new algorithms to compute near-minimal δ-coverings of finite VC-dimension set systems. These algorithms are the first non-trivial algorithms to obtain δ-coverings of minimal size. We present applications of it to low-discrepancy coloring and ε-approximation, deriving faster algorithms matching the optimal discrepancy of finite VC-dimension set systems: $O\left(n^{\frac{1}{2}-\frac{1}{2d}}\right)$.

Résumé en français

Les données de grande dimension sont courantes dans des domaines tels que la génomique, le traitement d'images et l'analyse des réseaux sociaux. Elles posent des défis importants en matière d'analyse computationnelle et statistique. L'augmentation de la taille des ensembles de données dans ces différents domaines accroît le besoin d'outils efficaces d'approximation des données. Même si la dimension des données peut être grande, leur dimension intrinsèque, c'est-à-dire la dimension minimale de l'information qu'elles contiennent, peut être réduite. Cela est représenté à travers des concepts tels que la dimension VC, que nous discutons dans cette thèse. Nous concentrons notre étude sur la recherche d'algorithmes efficaces pour les grands ensembles de données de dimension intrinsèque fixe.

Dans cette thèse, nous présentons des algorithmes pour calculer des structures fondamentales de la réduction combinatoire des données : ε-approximations, δ-recouvrements, colorations à faible discrépance et partitions à faibles croisements. En particulier, nous étudions les trois problèmes suivants.

Dans la première partie, nous présentons un nouveau jeu à deux joueurs et montrons l'existence d'une stratégie quasi-optimale en utilisant le célèbre algorithme de discrépance de Lovett et Meka. Nous présentons ensuite un algorithme à poids multiplicatifs pour calculer une famille de colorations à faible discrépance moyenne, utilisant la valeur de notre jeu dans son analyse.

Dans la deuxième partie, nous présentons un nouvel algorithme de partition à faibles croisements pour les systèmes d'ensembles généraux. Il s'agit de la première instance d'un algorithme rapide en pratique pour calculer des partitions à faibles croisements de systèmes d'ensembles généraux, les résultats précédents utilisant étant limités aux systèmes engendrés par des demi-espaces en dimension faible.

Enfin, nous présentons de nouveaux algorithmes pour calculer des δ-recouvrements quasi minimaux de systèmes d'ensembles de dimension VC finie. Ces algorithmes sont les premiers algorithmes non triviaux permettant d'obtenir des δ-recouvrements de taille minimale. Nous en présentons des applications à la coloration à faible discrépance et aux ε-approximation, en dérivant des algorithmes plus rapides atteignant la discrépance optimale pour les systèmes d'ensembles de dimension VC finie : $O\left(n^{\frac{1}{2}-\frac{1}{2d}}\right)$.

Contents

Organization of the manuscript

In Chapter 1, we give an overview of the different problems that will be discussed throughout this manuscript, as well as our main results on these problems.

In Chapter 3, we discuss the works in the literature related to the problems we discuss in the manuscript.

In Chapter 4,5 and 6, we give the detailed proofs of our main results. The three chapters can be read independently as they discuss problems that do not rely on each other.

In Chapter 4, we present a two-player game related to discrepancy. To illustrate its purpose, we show algorithms to compute a family of small average discrepancy colorings. This chapter relies on the following publication.

"Brunel, V., Louvet, A., & Mustafa, N. H. (2025). A new discrepancy game. https://hal.science/hal-05064760"

In Chapter 5, we show the results of a new low-crossing partition algorithm for general set system. This chapter is based on the following publication.

"Csikós, M., Louvet, A., & Mustafa, N. H. (2025). A Greedy Algorithm for Low-Crossing Partitions for General Set Systems. In 2025 Proceedings of the Symposium on Algorithm Engineering and Experiments (ALENEX) (pp. 209-220). Society for Industrial and Applied Mathematics."

In Chapter 6, we present new algorithms to compute near-minimal δ-coverings of finite VC-dimension set systems and the application of it to discrepancy.

Not included in the manuscript: "Bhore, S., Keszegh, B., Kupavskii, A., Le, H., Louvet, A., Pálvölgyi, D., & Tóth, C. D. (2025). Spanners in Planar Domains via Steiner Spanners and non-Steiner Tree Covers. In Proceedings of the 2025 Annual ACM-SIAM Symposium on Discrete Algorithms (SODA) (pp. 4292-4326). Society for Industrial and Applied Mathematics."

Throughout this manuscript, we present statements in color boxes. Definitions are in red boxes, corollary in green boxes and remarks in blue boxes. To present lemmas, theorems and applications we use purple boxes for our contributions and yellow boxes for results of the literature.

Chapter 1

Context and contributions

Each chapter in this thesis will present problems on *set systems*. We denote by X the finite ground set of n elements, most often a subset of $\mathbb{R}^d$, and by $\mathcal{F}$ the collection of m subsets of X. We refer to the elements of $\mathcal{F}$ as *ranges*. We present problems on abstract as well as geometric set systems.

1.1 Preliminaries

We first introduce some background concepts. We give more details on them in Chapter 3.

VC-dimension was first introduced by Vapnik and Chervonenkis [VC71] as a measure of the complexity of a set of functions that can be learned by a statistical binary classification algorithm. It is used to represent the complexity of a set system.

Definition 1.1. The VC-dimension of a set system $(X, \mathcal{F})$, denoted by VC-dim($\mathcal{F}$), is the size of the largest $Y \subseteq X$ for which $|\mathcal{F}|_Y| = |\{F \cap Y \mid F \in \mathcal{F}\}| = 2^{|Y|}$, where $|\cdot|$ is the cardinality of a finite set. We say that such a Y is shattered by $\mathcal{F}$.

An important operation on sets is the set symmetric difference. Let S, S' be two finite sets; we denote their *symmetric difference* $\Delta(S, S')$ by,

$$\Delta(S, S') = (S \cup S') \setminus (S \cap S').$$

We call δ-*separated* set systems such that for all distinct $F, F' \in \mathcal{F}$, $|\Delta(F, F')| \geq \delta$.

Finally, we introduce combinatorial discrepancy.

Definition 1.2. (Discrepancy) Let $(X, \mathcal{F})$ be a set system. We call *coloring* a function $\chi : X \to \{-1, 1\}$. We call discrepancy of $(X, \mathcal{F})$ with respect to a coloring χ the value

$$\operatorname{disc}_\chi(X, \mathcal{F}) = \max_{F \in \mathcal{F}} \chi(F)$$

where $\chi(F) = \left|\sum_{x \in F} \chi(x)\right|$.

We call the discrepancy of $(X, \mathcal{F})$ the value

$$\min_{\chi : X \to \{-1,1\}} \operatorname{disc}_\chi(X, \mathcal{F}).$$

Computing colorings with small discrepancy has been widely studied, as it has applications in a wide range of domains such as machine learning, optimization or mathematical finance[1].

In the next three sections, we present the main contributions in this thesis.

[1]See chapter 1 of [Mat99].

1.2 Combinatorial discrepancy games (Chapter 4)

In Chapter 4, we introduce a new two player game related to discrepancy called `Discrepancy Learning Game` (DLG). The two players, Alice and Bob, compete in $T = \Theta(n)$ rounds. They play on the same set system $(X, \mathcal{F})$. Alice is only given the ground set X of the set system and Bob the full set system $(X, \mathcal{F})$.

At each round $t = 1$ to T,

- Alice chooses a coloring χ_t of X and transmits it to Bob.
- On receiving χ_t, Bob chooses a range $F_t \in \mathcal{F}$ and transmits it to Alice.

The goal of Alice is to minimize $\frac{1}{T}\sum_{t=1}^{T}|\chi_t(F_t)|$ and for Bob to maximize the same expression. Alice will discover the set system one range at a time as they are revealed to her by Bob.

The game looks hopeless for Alice, as she is expected to find a coloring with low discrepancy on a set system she does not know. At first glance, it seems that the only viable strategy is to pick a random coloring each iteration, which would result in $\frac{1}{T}\sum_{t=1}^{T}|\chi_t(F_t)| = O\left(\sqrt{n\ln(m)}\right)$. It turns out, surprisingly, that Alice can obtain a much better bound in this blind setting.

We will prove the following theorem in Chapter 4.

Contribution 1. (DLG) Let Alice and Bob play a T rounds game of `Discrepancy Learning Game` on $(X, \mathcal{F})$ with finite VC-dimension d.

There exists a strategy for Alice such that regardless of Bob's choice of F_t,

$$\frac{1}{T}\sum_{t=1}^{T}|\chi_t(F_t)| = O\left(\max\left(T^{-\frac{1}{2d}}\sqrt{n}, n^{\frac{1}{2}-\frac{1}{2d}}\right)\log^2(n)\sqrt{\ln(T\log(n))}\right)$$

with probability at least $\frac{1}{2}$.

In particular, for $T = \Omega(n)$, we obtain

$$\frac{1}{T}\sum_{t=1}^{T}|\chi_t(F_t)| = O\left(n^{\frac{1}{2}-\frac{1}{2d}}\log^{\frac{5}{2}}(Tn)\right) \text{ w.h.p.}$$

We show that there exists finite VC-dimension set systems where, regardless of Alice's strategy, Bob can make sure that $\frac{1}{T}\sum_{t=1}^{T}|\chi_t(F_t)| = \Omega\left(n^{\frac{1}{2}-\frac{1}{2d}}\right)$. That is, this bound is optimal in the worst case up to a polylog(mn) factor.

The strategy of Alice relies on the discrepancy algorithm of Lovett and Meka [LM15] that we present in detail in Chapter 3.

Application of Contribution 1. Using the result on DLG, we present an algorithmic version of DLG that computes a family of low average discrepancy colorings.

Given $(X, \mathcal{F})$ a set system with finite VC-dimension d, our algorithm returns $\frac{n}{16}$ colorings $x^{(1)}, \ldots, x^{(n)}$ such that:

$$\forall k \leq m, \frac{16}{n}\mathbb{E}\left[\sum_{t=1}^{\frac{n}{16}} |v_k \cdot x^{(t)}|\right] = O\left(n^{\frac{1}{2}-\frac{1}{2d}} \log^{\frac{5}{2}}(mn \log(n))\right)$$

where for all $t \in \left[1, \frac{n}{16}\right]$, the coloring $x^{(t)}$ is computed using only one additional range of $(X, \mathcal{F})$ that was not used to compute $x^{(t-1)}$.

The algorithm succeeds with probability at least $\frac{1}{4}$ in expected time $\tilde{O}\left(n^4 + mn^{\frac{3}{2}+\frac{1}{2d}}\right)$.

1.3 Low-crossing partitions (Chapter 5)

Let $(X, \mathcal{F})$ be a set system. We say that a range $F \in \mathcal{F}$ *crosses* a set $P \subseteq X$ if and only if there exist two elements $x, y \in P$ such that $x \in F$ and $y \notin F$. Let $I(P, F)$ be the indicator function that is 1 iff F crosses P, and 0 otherwise. Given a set system $(X, \mathcal{F})$, our goal is to construct a partition $\mathcal{P}$ of X such that each set is approximately the same size, and each range in $\mathcal{F}$ crosses a sublinear number of parts of $\mathcal{P}$. These two properties are formalized in the following definitions.

Definition 1.3. (Crossing Number) Given $(X, \mathcal{F})$ and a partition $\mathcal{P} = \{P_1, \ldots, P_t\}$ of X, the crossing number of $\mathcal{F}$ with respect to $\mathcal{P}$, denoted by $\kappa_{\mathcal{F}}(\mathcal{P})$, is the maximum number of sets of $\mathcal{P}$ that are crossed by a range in $\mathcal{F}$. Formally,

$$\kappa_{\mathcal{F}}(\mathcal{P}) = \max_{F \in \mathcal{F}} \sum_{i=1}^{t} I(P_i, F).$$

Definition 1.4. ((τ, κ)-Partitions) Given a set system $(X, \mathcal{F})$, $n = |X|$, a (t, κ)-partition of $(X, \mathcal{F})$ is a partition of X into disjoint sets $P_1, \ldots, P_t$ such that

(i) for all $i \in [t-1]$, we have $|P_i| = \lfloor \frac{n}{t} \rfloor$,
(ii) $\frac{n}{t} \leq |P_t| \leq \frac{2n}{t}$, and
(iii) $\kappa_{\mathcal{F}}(\mathcal{P}) \leq \kappa$.

We refer to the P_i's as parts of the partition $\mathcal{P}$.

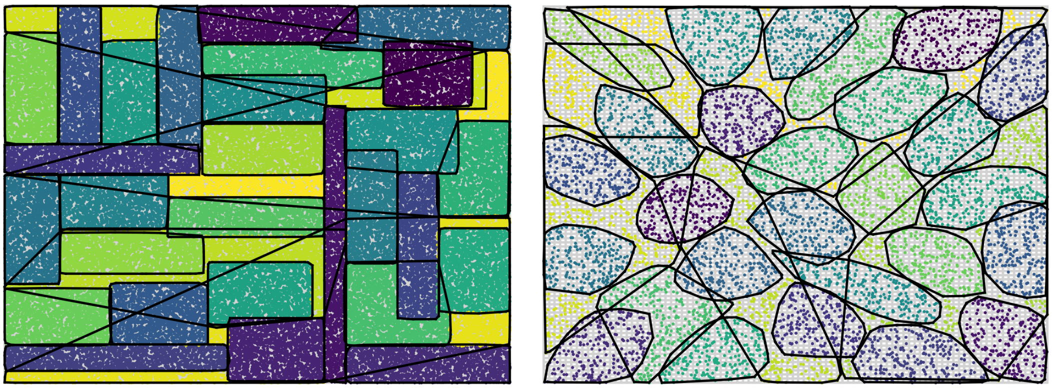
Simplicial partitions are a key tool for the geometric range searching problem, where the aim is to preprocess a set of points X (in $\mathbb{R}^d$) to be able to answer simplex containment queries on X. That is, each query specifies a simplex $Q \subseteq \mathbb{R}^d$, and the goal is to compute the set $Q \cap X$ quickly. The current-best way to solve this problem is via hierarchical data-structures (e.g., partition trees), which are derived from recursive applications of simplicial partitions [Mat92, Mat93]. This construction was further improved and simplified by Chan [Cha12].

Another application of simplicial partitions is set system approximation. [STZ06] showed that low-crossing partitions imply the existence of small-error ε-approximation[2].

On the experimental side, for the purposes of computing ε-approximations, Matheny and Phillips [MP18] constructed simplicial partitions for the specific case of set systems induced by half-spaces in $\mathbb{R}^2$, via methods inspired by the algorithms of Matoušek [Mat92] and Chan [Cha12]. However these methods rely on cuttings; therefore all previous experimental studies were limited to half-spaces in $\mathbb{R}^2$.

In Chapter 5, we present our work on this problem. We consider the problem of computing partitions with low crossing numbers for general set systems. We aimed for a simple algorithm that is fast in practice. In particular, our method does not rely on cuttings and so it works for general set systems, as well as geometric set systems in higher dimensions.

Contribution 2. Our algorithm constructs a low-crossing partition iteratively, building one part at a time by greedily extending it with one element which does not introduce too many new crossings. As a motivation for this greedy approach, we prove, under some hereditary assumptions, that any $\left(t, \tilde{O}\left(t^{1-\frac{1}{d}}\right)\right)$ partition can be created using this method. The algorithm we present has time complexity $O(nmt)$. We also give a faster variant of this algorithm with running time $\tilde{O}(mn + nt^2)$ that can, further, be partially parallelized on as many as m cores.



We present the evaluation of our algorithm on a variety of set systems, including abstract and high-dimensional geometric ones. As an illustration, we compare the results of a uniform random ε-approximation to an ε-approximation computed with our algorithm, for the geometric set system induced by disks.

In Figure 2, we present the approximations of a set system with 8192 elements on concentric circles and ranges induced by disks. On top, partitions computed with our algorithm (with 1024 random disks) and on the bottom with a uniform random sample. We see that the approximation we obtain with a partition is visibly better, leaving fewer gaps on the concentric circles, than a uniform random sample of the same size.

[2]See Section 3.6.1 for details on this construction.

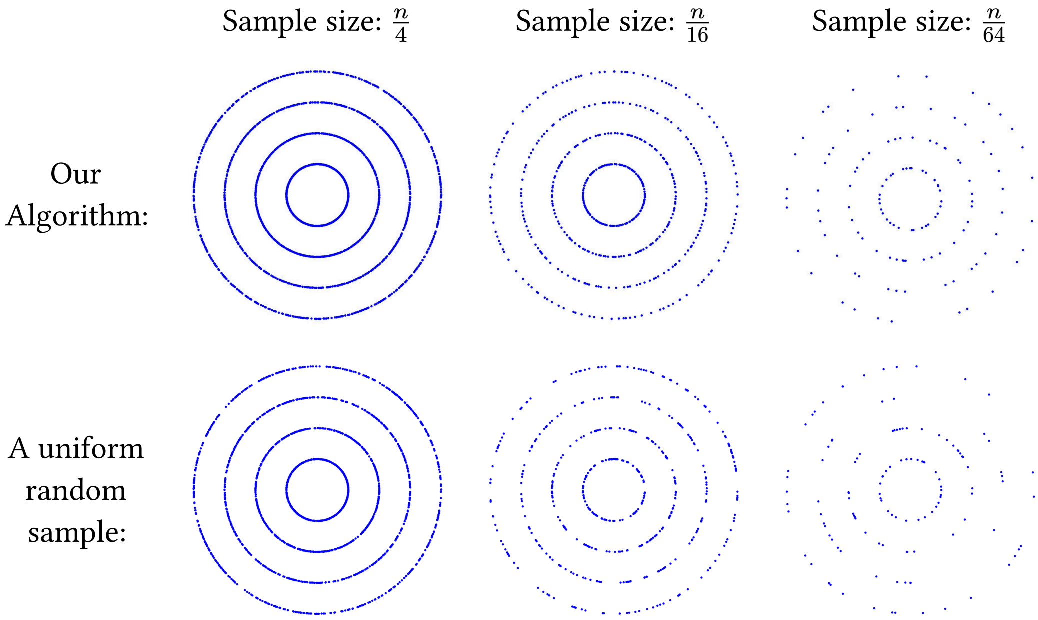


Figure 2: Approximations of disks

1.4 δ-coverings and δ-packings (Chapter 6)

In this chapter, we combine several tools and techniques to give new algorithms to compute low-discrepancy colorings of finite VC-dimension set systems. We aim to give a self-contained picture of the state-of-the-art algorithms to compute optimal discrepancy colorings of finite VC-dimension set systems.

Let $(X, \mathcal{F})$ be a finite set system and $\delta \in [1, n]$.

A $\boldsymbol{\delta}$ **—packing** over $(X, \mathcal{F})$ is $\mathcal{P} \subseteq 2^X$ such that for all $P_1, P_2 \in \mathcal{P}, |\Delta(P_1, P_2)| > \delta$. We say that $\mathcal{P}$ is **maximal** if $\forall F \in \mathcal{F}, \exists P \in \mathcal{P}$ s.t. $|\Delta(F, P)| \leq \delta$.

A $\boldsymbol{\delta}$ **—covering** over $(X, \mathcal{F})$ is $\mathcal{C} \subseteq 2^X$ such that for all $F \in \mathcal{F}, \exists C \in \mathcal{C}$ s.t. $|\Delta(F, C)| \leq \delta$. We say that $\mathcal{C}$ is **minimal** if $\forall C \in \mathcal{C}, \exists F \in \mathcal{F}$ s.t. $|\Delta(F, C)| \leq \delta$ and $\forall C' \in \mathcal{C} \setminus \{C\}, |\Delta(F, C')| > \delta$.

By definition, a maximal δ-packing is also a δ-covering. However, a minimal δ-covering might not be a δ-packing as two sets of a minimal δ-covering can be arbitrarily close.

Haussler [Hau95] showed that the size of δ-packings of finite VC-dimension set systems are bounded by $O\left(\left(\frac{n}{\delta}\right)^d\right)$. δ-coverings and δ-packings are an important part of many results improving general bounds in finite VC-dimension[3]. For some of these applications, the computation of δ-coverings is not necessary as they are only a tool of analysis of algorithms and not part of the algorithms themselves. This is the case for ε-approximations. However, in some cases, they are needed. For instance, to compute optimal discrepancy colorings of finite VC-dimension set systems, one needs to compute δ-coverings of size matching the bound of Haussler's lemma.

[3]See chapter 13 of [BLM13] for applications in statistics and chapter 13 of [Mus22] for application to ε-approximation.

A simple method to obtain minimal δ-coverings and maximal δ-packings is to construct them greedily[4]. This method leads to δ-coverings of size $O\left(\left(\frac{n}{\delta}\right)^d\right)$ in time $O\left(\frac{mn^{d+1}}{\delta^d}\right)$.

For some particular set systems, fast algorithms to compute small δ-coverings are known. This is, in particular, the case of set systems spanned by halfspaces where δ-coverings can be constructed using cuttings [Mat92]. The first non-trivial algorithm to compute δ-coverings of any set system of finite VC-dimension is due to Matoušek, Welzl and Wernisch [MWW93].

However this result is not optimal as set systems with finite VC-dimension admit δ-coverings of size at most $O\left(\left(\frac{n}{\delta}\right)^d\right)$ [Hau95].

In Chapter 6, we prove the following statement.

Contribution 3. Given a set system $(X, \mathcal{F})$ with VC-dimension at most d and $\delta \in [4, n]$, there exists an algorithm that computes a δ-covering of size $O\left(\left(\frac{n}{\delta}\right)^d\right)$ with probability at least $\frac{1}{2}$. The algorithm has time complexity $O\left(\frac{mdn}{\delta}\log\left(\frac{n}{\delta}\right) + \left(\frac{n}{\delta}\right)^{2d+2}\log^d\left(\frac{n}{\delta}\right)\right)$.

This improves the result of [MWW93] by removing the log factor from [MWW93] bound with an increase in time complexity of $O\left(\left(\frac{n}{\delta}\right)^{d+2}\right)$ for $\delta = \Omega\left(\sqrt{n\log\left(\frac{n}{\delta}\right)}\right)$ and the same time complexity otherwise.

Our algorithm is the first instance of a non-trivial algorithm to compute δ-coverings of any finite VC-dimension set systems of size matching Haussler's bound.

In the same chapter, we also give variants of this algorithm for specific types of finite VC-dimension set systems.

Matoušek [Mat95] proved, following on Matoušek, Welzl and Wernisch's work [MWW93], that finite VC-dimension set systems admit colorings with discrepancy of order $O\left(n^{\frac{1}{2}-\frac{1}{2d}}\right)$ which is optimal [Ale90]. However, Matoušek's result is not constructive.

Application of Contribution 3. Combining Matoušek's work, our δ-covering algorithm and the work of Lovett and Meka on discrepancy [LM15], we present in Chapter 6 an algorithm that computes a coloring with discrepancy $O\left(n^{\frac{1}{2}-\frac{1}{2d}}\right)$. The algorithm has time complexity $\tilde{O}\left(mn^{\frac{1}{d}} + n^{2+\frac{2}{d}}\log^d(n) + n^3\log^{3d}(mn)\right)$.

This improves on both best previous results. On the one hand, it improves the discrepancy bound obtained by combining the covering algorithm of [MWW93] with the discrepancy algorithm of [LM15] that produces a coloring with discrepancy $O\left(n^{\frac{1}{2}-\frac{1}{2d}}\sqrt{\log(\log(n))}\right)$ in time $\tilde{O}\left(mn^{\frac{1}{d}} + n^3\log^{3d}(mn)\right)$. On the other hand, it improves the runtime of combining greedy covering with the algorithm of [LM15] for $d > 1$. This method produces a coloring with discrepancy $O\left(n^{\frac{1}{2}-\frac{1}{2d}}\right)$ in time $\tilde{O}\left(mn^2 + n^3\log^{3d}(mn)\right)$.

[4]See details in Section 3.3.1.1.

Chapter 2

Contexte et contributions

Chaque chapitre de cette thèse présente des problèmes sur les *systèmes d'ensembles*. On note X l'ensemble de base fini de n éléments, le plus souvent un sous-ensemble de $\mathbb{R}^d$, et $\mathcal{F}$ la collection de m sous-ensembles de X. On appelle les éléments de $\mathcal{F}$ des *ensembles d'étendues*. Nous étudions des problèmes sur des systèmes d'ensembles abstraits ainsi que géométriques.

2.1 Connaissances préliminaires

Nous introduisons d'abord quelques notions de base, détaillées davantage dans le Chapitre 3.

La dimension VC a été introduite par Vapnik et Chervonenkis [VC71] comme mesure de la complexité d'un ensemble de fonctions pouvant être appris par un algorithme de classification binaire statistique. Elle sert à représenter la complexité d'un système d'ensembles.

Definition 2.1. La dimension VC d'un système d'ensembles $(X, \mathcal{F})$, notée VC-dim($\mathcal{F}$), est la taille du plus grand $Y \subseteq X$ tel que $|\mathcal{F}|_Y| = |\{F \cap Y \mid F \in \mathcal{F}\}| = 2^{|Y|}$, où $|\cdot|$ désigne la cardinalité d'un ensemble fini. Un tel Y est dit être éclaté (shattered) par $\mathcal{F}$.

Une opération importante sur les ensembles est la différence symétrique, qui peut être utilisée comme une métrique entre ensembles d'étendues. Soient S, S' deux ensembles finis, on note leur *différence symétrique* $\Delta(S, S')$ définie comme

$$\Delta(S, S') = (S \cup S') \setminus (S \cap S').$$

On dit qu'un système d'ensembles est δ-*séparé* si pour tous $F, F' \in \mathcal{F}$ distincts, $|\Delta(F, F')| \geq \delta$.

Enfin, nous présentons la notion de discrépance combinatoire.

Definition 2.2. (Discrépance) Soit $(X, \mathcal{F})$ un système d'ensembles. On appelle *coloration* une fonction $\chi : X \to \{-1, 1\}$. La discrépance de $(X, \mathcal{F})$ par rapport à une coloration χ est définie par

$$\text{disc}_\chi(X, \mathcal{F}) = \max_{F \in \mathcal{F}} \chi(F)$$

où $\chi(F) = \left|\sum_{x \in F} \chi(x)\right|$.

La discrépance de $(X, \mathcal{F})$ est alors donnée par

$$\min_{\chi : X \to \{-1,1\}} \text{disc}_\chi(X, \mathcal{F}).$$

Le calcul de colorations à faible discrépance a été largement étudié car il a des applications dans de nombreux domaines tels que l'apprentissage automatique, l'optimisation ou la finance mathématique[5].

Dans les trois sections suivantes, nous présentons les principales contributions de cette thèse.

2.2 Jeux de discrépance combinatoire (Chapitre 4)

Dans le Chapitre 4, nous introduisons un nouveau jeu à deux joueurs lié à la discrépance appelé `Jeu de discrépance par apprentissage` (`DLG`). Les deux joueurs, Alice et Bob, s'affrontent pendant $T = \Theta(n)$ tours. Ils jouent sur le même système d'ensembles $(X, \mathcal{F})$. Alice ne connaît que l'ensemble de base X du système d'ensembles, tandis que Bob connaît l'ensemble complet $(X, \mathcal{F})$.

À chaque iteration $t = 1$ à T,

- Alice choisit une coloration χ_t de X et la transmet à Bob.
- À la réception de χ_t, Bob choisit une étendue $F_t \in \mathcal{F}$ et la transmet à Alice.

L'objectif d'Alice est de minimiser $\frac{1}{T}\sum_{t=1}^{T}|\chi_t(F_t)|$, tandis que celui de Bob est de maximiser cette même expression. Alice découvre le système d'ensembles une étendue à la fois, au fur et à mesure que Bob les lui révèle.

Le jeu semble perdu d'avance pour Alice, car on attend d'elle qu'elle trouve une coloration de faible discrépance sur un système d'ensembles qu'elle ne connaît pas. À première vue, il semble que la seule stratégie viable soit de choisir une coloration aléatoire à chaque itération, ce qui mènerait à $\frac{1}{T}\sum_{t=1}^{T}|\chi_t(F_t)| = O\left(\sqrt{n \ln(m)}\right)$. Il s'avère, de manière surprenante, qu'Alice peut obtenir une bien meilleure borne même dans ce cadre aveugle.

Nous démontrerons le théorème suivant dans le Chapitre 4.

Contribution 1. (DLG)

Soient Alice et Bob jouant un jeu de T tours du `DLG` sur $(X, \mathcal{F})$ ayant une dimension VC finie d.

Il existe une stratégie pour Alice telle que, quel que soit le choix de F_t par Bob,

$$\frac{1}{T}\sum_{t=1}^{T}|\chi_t(F_t)| = O\left(\max\left(T^{-\frac{1}{2d}}\sqrt{n}, n^{\frac{1}{2}-\frac{1}{2d}}\right)\log^2(n)\sqrt{\ln(T\log(n))}\right)$$

avec une probabilité d'au moins $\frac{1}{2}$.

En particulier, pour $T = \Omega(n)$, on obtient

$$\frac{1}{T}\sum_{t=1}^{T}|\chi_t(F_t)| = O\left(n^{\frac{1}{2}-\frac{1}{2d}}\log^{\frac{5}{2}}(Tn)\right).$$

Nous montrons qu'il existe des systèmes d'ensembles de dimension VC finie pour lesquels, quelle que soit la stratégie d'Alice, Bob peut s'assurer que $\frac{1}{T}\sum_{t=1}^{T}|\chi_t(F_t)| = \Omega\left(n^{\frac{1}{2}-\frac{1}{2d}}\right)$. Autrement dit, cette borne est optimale dans le pire des cas à un facteur polylog(mn) près.

[5] Voir le chapitre 1 de [Mat99].

La stratégie d'Alice repose sur l'algorithme de discrépance de Lovett et Meka [LM15], que nous détaillons dans le Chapitre 3.

Application de Contribution 1. En utilisant le résultat sur le DLG, nous présentons un algorithme à poids multiplicatifs pour calculer une famille de colorations à faible discrépance moyenne.

Étant donné $(X, \mathcal{F})$ un système d'ensembles de dimension VC finie d, notre algorithme renvoie $\frac{n}{16}$ colorations $x^{(1)}, ..., x^{(n)}$ telles que :

$$\forall k \leq m, \frac{16}{n}\mathbb{E}\left[\sum_{t=1}^{\frac{n}{16}} |v_k \cdot x^{(t)}|\right] = O\left(n^{\frac{1}{2}-\frac{1}{2d}} \log^{\frac{5}{2}}(mn\log(n))\right)$$

avec une probabilité d'au moins $\frac{1}{4}$ en temps espéré $\tilde{O}\left(n^4 + mn^{\frac{3}{2}+\frac{1}{2d}}\right)$.

2.3 Partitions à faibles croisements (Chapitre 5)

Soit $(X, \mathcal{F})$ un système d'ensembles. On dit qu'un ensemble $F \in \mathcal{F}$ *coupe* un ensemble $P \subseteq X$ s'il existe deux éléments $x, y \in P$ tels que $x \in F$ et $y \notin F$. On note $I(P, F)$ la fonction indicatrice qui vaut 1 si F coupe P, et 0 sinon. Étant donné un système d'ensembles $(X, \mathcal{F})$, notre objectif est de construire une partition $\mathcal{P}$ de X telle que chaque ensemble soit de taille approximativement égale, et chaque ensemble d'étendues dans $\mathcal{F}$ coupe un nombre sous-linéaire de parties de $\mathcal{P}$. Ces deux propriétés sont formalisées par les définitions suivantes.

Definition 2.3. (Nombre de croisements) Étant donné $(X, \mathcal{F})$ et une partition $\mathcal{P} = \{P_1, ..., P_t\}$ de X, le nombre de croisements de $\mathcal{F}$ par rapport à $\mathcal{P}$, noté $\kappa_{\mathcal{F}}(\mathcal{P})$, est le nombre maximal de parties de $\mathcal{P}$ coupées par un ensemble d'étendues dans $\mathcal{F}$. Formellement,

$$\kappa_{\mathcal{F}}(\mathcal{P}) = \max_{F \in \mathcal{F}} \sum_{i=1}^{t} I(P_i, F).$$

Definition 2.4. ((τ, κ)-Partitions) Soit $(X, \mathcal{F})$ un système d'ensembles, $n = |X|$, une (t, κ)-partition de $(X, \mathcal{F})$ est une partition de X en ensembles disjoints $P_1, ..., P_t$ telle que

(i) pour tout $i \in [t-1]$, on a $|P_i| = \lfloor \frac{n}{t} \rfloor$,
(ii) $\frac{n}{t} \leq |P_t| \leq \frac{2n}{t}$, et
(iii) $\kappa_{\mathcal{F}}(\mathcal{P}) \leq \kappa$.

On appelle les P_i les parties de la partition $\mathcal{P}$.

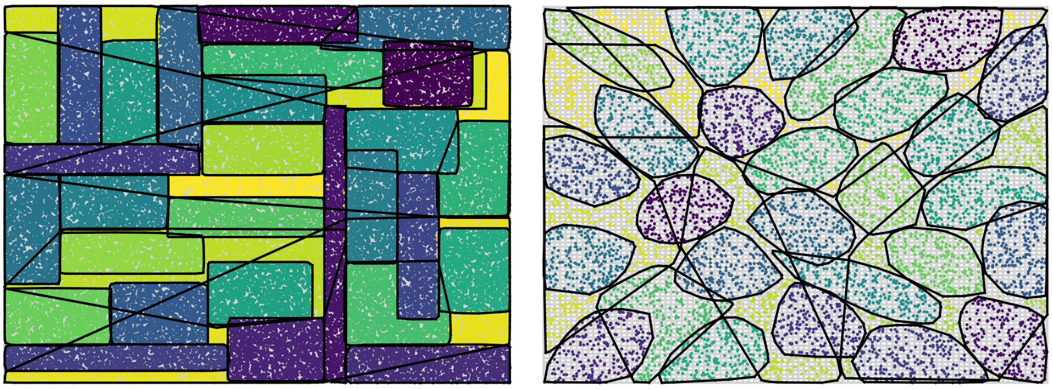
Les partitions simpliciales sont un outil clé pour le problème de recherche d'étendues géométrique, où le but est de prétraiter un ensemble de points X (dans $\mathbb{R}^d$) afin de pouvoir répondre efficacement à des requêtes d'inclusion dans un simplexe. Chaque requête spécifie un simplexe $Q \subseteq \mathbb{R}^d$ et le but est de calculer rapidement l'ensemble $Q \cap X$. La meilleure approche actuelle repose sur des structures de données hiérarchiques (par exemple, les arbres de partition), issues d'applications récursives de partitions simpliciales [Mat92, Mat93]. Cette construction a été améliorée et simplifiée par Chan [Cha12].

Une autre application des partitions simpliciales est l'approximation de systèmes d'ensembles. [STZ06] a montré que les partitions à faibles croisements impliquent l'existence d'ε-approximations à faible erreur[6].

Du point de vue expérimental, pour le calcul des ε-approximations, Matheny et Phillips [MP18] ont construit des partitions simpliciales pour le cas spécifique des systèmes d'ensembles induits par des demi-espaces dans $\mathbb{R}^2$, via des méthodes inspirées des algorithmes de Matoušek [Mat92] et Chan [Cha12]. Cependant, ces méthodes reposent sur les découpages, donc toutes les études expérimentales antérieures étaient limitées aux demi-espaces dans $\mathbb{R}^2$.

Dans le Chapitre 5, nous présentons nos travaux sur ce problème. Nous considérons le problème de calcul de partitions à faible nombre de croisements pour des systèmes d'ensembles généraux. Nous visons un algorithme simple et rapide en pratique. En particulier, notre méthode ne repose pas sur des découpages et fonctionne donc pour les systèmes d'ensembles généraux, ainsi que pour les systèmes géométriques en dimension élevée.

Contribution 4. Notre algorithme construit une partition à faibles croisements de manière itérative, en construisant une partie à la fois en l'agrandissant gloutonnement avec un élément qui n'introduit pas trop de nouveaux croisements. Pour motiver cette approche gloutonne, nous démontrons, sous certaines hypothèses d'hérédité, que toute partition $\left(t, \tilde{O}\left(t^{1-\frac{1}{d}}\right)\right)$ peut être créée ainsi. L'algorithme a une complexité en temps de $O(nmt)$. Nous présentons aussi une variante plus rapide avec une complexité $\tilde{O}(mn + nt^2)$, parallélisable sur jusqu'à m cœurs.



Nous présentons l'évaluation de notre algorithme sur une variété de systèmes d'ensembles, y compris des systèmes abstraits et géométriques en grande dimension. À titre d'illustration, nous comparons les résultats d'une ε-approximation aléatoire uniforme avec une ε-approximation calculée à l'aide de notre algorithme, pour le système d'ensembles géométrique induit par des disques.

Dans la Figure 4, nous présentons les approximations d'un système d'ensembles avec 8192 éléments disposés sur des cercles concentriques et des étendues induites par des disques. En

[6] Voir Section 3.6.1 pour plus de détails sur cette construction.

haut, les partitions calculées avec notre algorithme (avec 1024 disques aléatoires) et en bas, avec un échantillon aléatoire uniforme. On observe que l'approximation obtenue avec une partition est visiblement meilleure, laissant moins de trous sur les cercles concentriques, qu'un échantillon aléatoire uniforme de même taille.

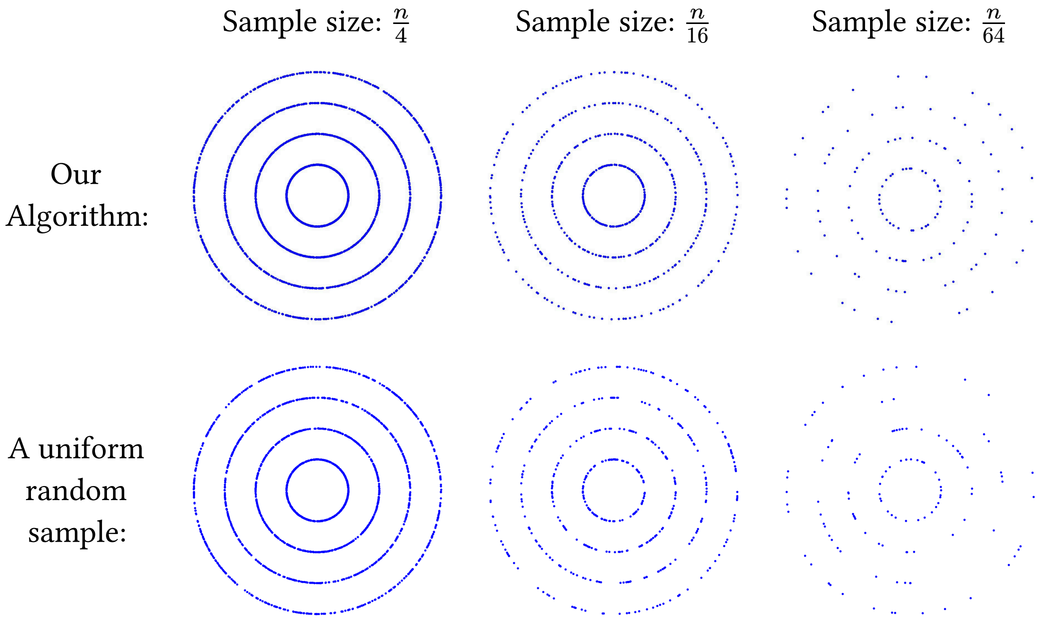


Figure 4: Approximations de disques

2.4 δ-recouvrement et δ-paquet (Chapitre 6)

Dans ce chapitre, nous combinons plusieurs outils et techniques afin de proposer de nouveaux algorithmes pour calculer des colorations de faible discrépance dans des systèmes d'ensembles de dimension VC finie. Notre objectif est de fournir une vue d'ensemble autonome des algorithmes les plus avancés permettant de calculer des colorations optimales en termes de discrépance pour ces systèmes. Soit $(X, \mathcal{F})$ un système d'ensembles fini et $\delta \in [1, n]$.

Un **δ-paquet** sur $(X, \mathcal{F})$ est un ensemble $\mathcal{P} \subseteq 2^X$ tel que $\forall P_1, P_2 \in \mathcal{P}, |\Delta(P_1, P_2)| > \delta$. On dit que $\mathcal{P}$ est **maximal** si $\forall F \in \mathcal{F}, \exists P \in \mathcal{P}$ t.q. $|\Delta(F, P)| \leq \delta$.

Un **δ-recouvrement** sur $(X, \mathcal{F})$ est un ensemble $\mathcal{C} \subseteq 2^X$ tel que $\forall F \in \mathcal{F}, \exists C \in \mathcal{C}$ tel que $|\Delta(F, C)| \leq \delta$. On dit que $\mathcal{C}$ est **minimal** si $\forall C \in \mathcal{C}, \exists F \in \mathcal{F}$ t.q. $|\Delta(F, C)| \leq \delta$ et $\forall C' \in \mathcal{C} \setminus \{C\}, |\Delta(F, C')| > \delta$.

Par définition, un δ-paquet maximal est aussi un δ-recouvrement. Cependant, les δ-recouvrements minimaux ne sont pas nécessairement des δ-paquets, car deux ensembles d'un δ -recouvrement minimal peuvent être arbitrairement proches.

Haussler [Hau95] a montré que la taille des δ-paquets pour des systèmes d'ensembles de dimension VC finie est bornée par $O\left(\left(\frac{n}{\delta}\right)^d\right)$. Les δ-recouvrements et δ-paquets jouent un rôle important dans de nombreux résultats permettant d'améliorer les bornes générales en dimension VC finie[7]. Pour certaines de ces applications, il n'est pas nécessaire de calculer explicitement les δ-recouvrements, car ils servent uniquement d'outil d'analyse des

[7] Voir le chapitre 13 de [BLM13] pour des applications en statistique et le chapitre 13 de [Mus22] pour des applications aux ε-approximations.

algorithmes et ne font pas partie des algorithmes eux-mêmes. C'est notamment le cas des ε-approximations. Cependant, dans certains cas, ils sont nécessaires. Par exemple, pour calculer des colorations de discrépance optimales de systèmes d'ensembles à dimension VC finie, il faut calculer des δ-recouvrements dont la taille respecte la borne du lemme de Haussler.

Une méthode simple pour obtenir des δ-recouvrements minimaux et des δ-paquets maximaux consiste à les construire de manière gloutonne[8]. Cette méthode donne des δ-recouvrements de taille $O\left(\left(\frac{n}{\delta}\right)^d\right)$ en temps $O\left(\frac{mn^{d+1}}{\delta^d}\right)$.

Pour certains systèmes d'ensembles particuliers, des algorithmes rapides pour calculer de petits δ-recouvrements sont connus. C'est notamment le cas des systèmes induits par des demi-espaces, où les δ-recouvrements peuvent être construits à l'aide de découpages [Mat92]. Le premier algorithme non trivial permettant de calculer des δ-recouvrements pour n'importe quel système d'ensembles de dimension VC finie est dû à Matoušek, Welzl et Wernisch [MWW93].

Cependant, ce résultat n'est pas optimal, car les systèmes d'ensembles à dimension VC finie admettent des δ-recouvrements de taille au plus $O\left(\left(\frac{n}{\delta}\right)^d\right)$ [Hau95].

Dans le Chapitre 6, nous prouvons l'énoncé suivant.

Contribution 2. Étant donné un système d'ensembles $(X, \mathcal{F})$ de dimension VC au plus d et $\delta \in [4, n]$, il existe un algorithme qui calcule un δ-recouvrement de taille $O\left(\left(\frac{n}{\delta}\right)^d\right)$ avec une probabilité d'au moins $\frac{1}{2}$. L'algorithme a une complexité en temps $O\left(\frac{mdn}{\delta}\log\left(\frac{n}{\delta}\right) + \left(\frac{n}{\delta}\right)^{2d+2}\log^d\left(\frac{n}{\delta}\right)\right)$.

Cela améliore le résultat de [MWW93] en supprimant le facteur log de leur borne, au prix d'une complexité en temps augmentée de $O\left(\left(\frac{n}{\delta}\right)^{d+2}\right)$ pour $\delta = \Omega\left(\sqrt{n\log\left(\frac{n}{\delta}\right)}\right)$, et de complexité identique sinon.

Notre algorithme est la première instance d'un algorithme non trivial permettant de calculer des δ-recouvrements de taille correspondant à la borne de Haussler pour tout système d'ensembles de dimension VC finie.

Dans le même chapitre, nous présentons aussi des variantes de cet algorithme pour des types spécifiques de systèmes à dimension VC finie.

Matoušek [Mat95] a démontré, à la suite des travaux de Matoušek, Welzl et Wernisch [MWW93], que les systèmes à dimension VC finie admettent des colorations de discrépance d'ordre $O\left(n^{\frac{1}{2}-\frac{1}{2d}}\right)$, ce qui est optimal [Ale90]. Cependant, le résultat de Matoušek n'est pas constructif.

[8] Voir les détails dans Section 3.3.1.1.

Application de Contribution 2. En combinant le travail de Matoušek, notre algorithme de δ-recouvrement et celui de Lovett et Meka sur la discrépance [LM15], nous présentons dans Chapitre 6 un algorithme qui calcule une coloration avec une discrépance de $O\left(n^{\frac{1}{2}-\frac{1}{2d}}\right)$. L'algorithme a une complexité en temps $\tilde{O}\left(mn^{\frac{1}{d}} + n^{2+\frac{2}{d}} \log^d(n) + n^3 \log^{3d}(mn)\right)$.

Cela améliore les deux meilleurs résultats précédents. D'une part, cela améliore la borne de discrépance obtenue en combinant l'algorithme de recouvrement de [MWW93] avec celui de [LM15], qui donne une coloration de discrépance $O\left(n^{\frac{1}{2}-\frac{1}{2d}}\sqrt{\log(\log(n))}\right)$ en temps $\tilde{O}\left(mn^{\frac{1}{d}} + n^3 \log^{3d}(mn)\right)$. D'autre part, cela améliore le temps d'exécution obtenu en combinant les recouvrements gloutons avec l'algorithme de [LM15] pour $d > 1$. Cette méthode produit une coloration avec discrépance $O\left(n^{\frac{1}{2}-\frac{1}{2d}}\right)$ en temps $\tilde{O}\left(mn^2 + n^3 \log^{3d}(mn)\right)$.

Chapter 3

Previous work

3.1 VC-dimension

We give some intuition on the definition of VC-dimension introduced in Chapter 1.

Let the following ground set consisting of 4 points in the plane as presented in Figure 5. Consider the set system with that ground set and every range formed by the intersection between the ground set and a disk.

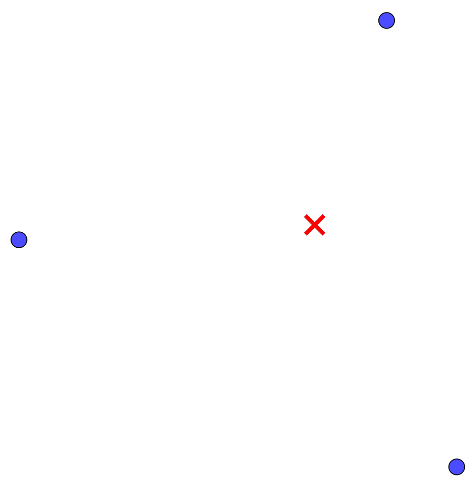


Figure 5: A ground set of 4 points in the plane

We can observe that the VC-dimension of this set system is at most 3 as the range that contains all three blue points without containing the red point can not be achieved by a disk as any disk containing the blue points will also contain the red one. In fact, the VC-dimension of such a set system is exactly 3. We present more results on VC-dimension of some particular set systems in the next section.

VC-dimension represents the combinatorial complexity of a set system. This combinatorial limit also implies a bound on the size of set systems with finite VC-dimension. Sauer and Shelah [Sau72, She72] independently showed that if $(X, \mathcal{F})$ has finite VC-dimension d then $|\mathcal{F}| = O(n^d)$.

Lemma 3.1. ([Sau72, She72]) Let $(X, \mathcal{F})$ be a set system with finite VC-dimension d, then

$$\forall Y \subseteq X \text{ s.t. } |Y| \geq d, |\mathcal{F}|_Y| \leq \sum_{i=0}^{d} \binom{|Y|}{i} = O(|Y|^d).$$

In fact, this bound an almost be seen as the definition of VC-dimension itself. In particular, the reciprocal of Lemma 3.1 is also true up to some logarithmic factor, that is

Lemma 3.2. ([Mus22] Lemma 4.6) Let $(X, \mathcal{F})$ be a set system such that

$$\forall Y \subseteq X \text{ with } |Y| \geq d, |\mathcal{F}|_Y| \leq |Y|^d$$

then the VC-dimension of $(X, \mathcal{F})$ is at most $d \log(d)$.

Finite VC-dimension set systems have been also widely studied in statistical learning theory for classification problems as they represent the idea that the information one aims to classify is contained in a smaller space than the input size. For instance in image classification problems this represents that only a subset of the pixels contain all the sufficient information to solve the classification problem[9].

3.1.1 VC-dimension of some common set systems

VC-dimension has been widely studied and the VC-dimension of some common set system are well-known. We present some of these results.

First we discuss set systems *spanned* by some geometric object such as halfspaces or disks. In that case we mean that the ranges are defined by all possible intersections between the geometric objects and the ground set. Formally, given a set X of elements of $\mathbb{R}^d$, the set system spanned by halfspaces in $\mathbb{R}^d$ is $(X, \mathcal{F})$ where

$$\mathcal{F} = \{X \cap \mathcal{H}, \mathcal{H} \text{ halfspace of } \mathbb{R}^d\}.$$

VC-dimension of halfspaces. The VC-dimension of set systems spanned by halfspaces in $\mathbb{R}^d$ is $d + 1$. This result comes from the fact that any set of $d + 1$ points is shattered by halfspaces: any set of $d + 1$ points in general position is linearly dependent in $\mathbb{R}^d$. The upper bound on the VC-dimension comes from radon's lemma.

Lemma 3.3. (Radon's lemma [Rad21]) Any set of $d + 2$ points in $\mathbb{R}^d$ can be partitioned into two sets whose convex hulls intersect.

This lemma implies that, calling A and B the two partitions of a set of size $d + 2$ given by Lemma 3.3, it is not possible to find a halfspace containing exclusively the element of A as there will be elements of B on both sides of the halfspaces. We illustrate this in $\mathbb{R}^2$ in Figure 6, where it is not possible to find a halfspace containing all the red dots and no blue cross.

[9] See [DGL13] chapter 12 and 13 for more on applications of VC-dimension to statistical learning theory.

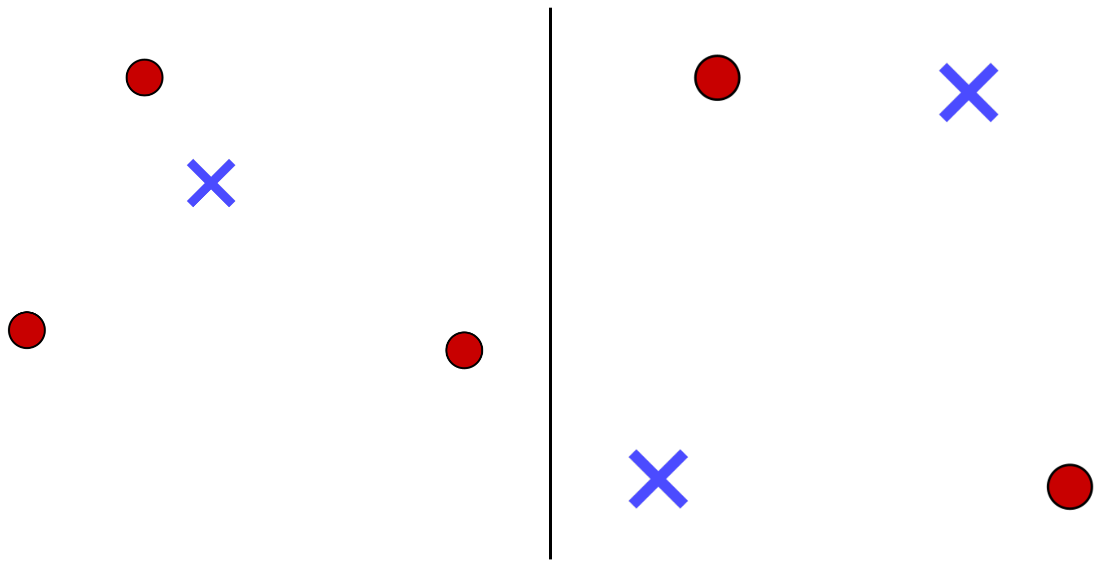


Figure 6: The red points cannot be separated from the blue points

Bounds have been shown to exist for a variety of set systems spanned by geometric objects. For instance the VC-dimension of set systems spanned by balls in $\mathbb{R}^d$ is $d + 1$ (see proof in [Mus22], Lemma 4.14).

VC-dimension of dual set systems.

Definition 3.4. The dual set system of a set system $(X, \mathcal{F})$ is defined as $(Y, \mathcal{G})$ where

$$Y = \{y_F : F \in \mathcal{F}\} \text{ and } \mathcal{G} = \{\{y_s : F \in \mathcal{F} \text{ s.t. } x \in F\} : x \in X\}.$$

If $(X, \mathcal{F})$ has VC-dimension d, then the VC-dimension of $(Y, \mathcal{G})$ is at most 2^{d+1}. The proof of this claim can be found in [Mat13] (Lemma 10.3.4).

VC-dimension of set systems formed by set operations. Finally we restate a result of Dudley [Dud78] about the VC-dimension of set systems obtained with set operations. A concise proof of this result can be found in [Mat13] (Proposition 10.3.3).

Lemma 3.5. ([Dud78, Mat13]) Let $(X, \mathcal{F})$ be a set system with finite VC-dimension d and $k \geq 2$ be a fixed positive integer. Let $\psi : (2^X)^k \to 2^X$ be a function uses only the union, intersection and symmetric difference operations.

Let $\mathcal{T} = \{\psi(F_1, ..., F_k) : F_1, ..., F_k \in \mathcal{F}\}$. Then $(X, \mathcal{T})$ has VC-dimension $= O(kd \ln(k))$.

This result directly implies a bound on the VC-dimension of the symmetric difference set system.

Corollary 3.6. Let $(X, \mathcal{F})$ be set system with VC-dimension $\leq d$, Lemma 3.5 implies that $(X, \Delta(\mathcal{F}))$ where:

$$\Delta(\mathcal{F}) = \{\Delta(F_1, F_2) : F_1, F_2 \in \mathcal{F}\}$$

has VC-dimension $= O(d)$.

This result follows from applying Lemma 3.5 with $k = 2$.

3.2 Approximations of set systems

3.2.1 ε-nets

An ε-net is a subset of X such that any set containing more than an ε fraction of the points contains at least one point of the net. Formally,

Definition 3.7. A set $N \subseteq X$ is an ε-net of $\mathcal{F}$ iff

$$\forall F \in \mathcal{F} \text{ s.t. } |F| \geq \varepsilon n, F \cap N \neq \emptyset.$$

Haussler and Welzl showed that in set systems with finite VC-dimension, ε-net can be constructed by uniformly sampling X.

Lemma 3.8. ([HW87]) Let $(X, \mathcal{F})$ be a set system with finite VC-dimension d and $\varepsilon \in \left[0, \frac{1}{2}\right]$, a uniform random sample of X of size $\frac{8d}{\varepsilon} \log\left(\frac{8d}{\varepsilon}\right)$ is an ε-net of X with probability at least $\frac{1}{\sqrt{2}}$.

Pach and Woeginger [PW90] showed the existence of abstract set systems with VC-dimension 2 where any ε-nets has size $\Omega\left(\frac{1}{\varepsilon} \log\left(\frac{1}{\varepsilon}\right)\right)$.

Later, Pach and Tardos [PT13] even showed that this lower bound holds for geometric set systems.

An application of ε-nets is presented in Section 3.3.1.2.

3.2.2 ε-approximations

An ε-approximation is a subset of X such that the set system obtained by projecting all ranges of $\mathcal{F}$ on the approximation maintains proportionally the same size up to an error depending on ε. Formally,

Definition 3.9. A set $A \subseteq X$ is an ε-approximation of $\mathcal{F}$ if

$$\left| \frac{|F|}{|X|} - \frac{|F \cap A|}{|A|} \right| \leq \varepsilon.$$

One can obtain an ε-approximations from a uniform random sample of X w.h.p. The simple analysis of uniform sampling demonstrate that sampling uniformly $O\left(\frac{1}{\varepsilon^2} \ln(m)\right)$ elements of X gives an ε-approximation[10]. The following result is from the work of Li, Long and Srinivasan [LLS01] building on Talagran's result [Tal94]. They show that this bound can be made completely independent of m and n for finite VC-dimension set systems. In that case the number of elements to sample only depends on ε and the VC-dimension of the set system.

[10] See Theorem 12.2 of [Mus22] for a proof of this result.

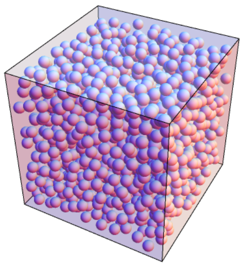
Lemma 3.10. ([LLS01, Tal94]) Let $(X, \mathcal{F})$ be a set system with finite VC-dimension d and $\varepsilon \in \left[0, \frac{1}{2}\right]$. There exists a constant $c \in \mathbb{R}$ such that a uniform random sample of X of size $\frac{cd}{\varepsilon^2}$ is an ε-approximation of X with probability at least $\frac{1}{\sqrt{2}}$ for some absolute constant c.

A simple proof of this result can be found in [CM22].

This bound is not optimal and it is possible to improve it further by sampling points in a non-uniform manner. In Section 3.6, we explain two methods to obtain ε-approximation of size $O\left(\frac{d}{\varepsilon^{\frac{2d}{d+1}}}\right)$.

3.3 Packings in finite VC-dimension

A small digression.



Question 3.11. (Sphere packing problem) For ε in $]0, 1[$, how many spheres of radius ε can be packed in a sphere of radius 1 of $\mathbb{R}^d$?

The sphere packing problem has been widely studied[11] and is still a very open problem. Despite the problem having been studied for more than 400 years the optimality proof in dimension 8 is less than 10 years old [Via17] and the optimal packing is unknown in almost all dimensions greater than 8. We do not extensively discuss the sphere packing problem but we present a simple upper bound for the problem.

The volume of a ball of radius r in $\mathbb{R}^d$ is

$$\frac{\pi^{\frac{d}{2}}}{\Gamma\left(\frac{d}{2}+1\right)} r^d = f(d) r^d$$

where Γ is Euler's gamma function.

A simple bound for the sphere packing problem is that the total volume occupied by packed sphere can not exceed the volume of the sphere of radius 1. This gives that the number of spheres packed $\mathcal{S}$ is bounded by the inequality:

$$\mathcal{S} \times f(d)\varepsilon^d \leq f(d)1^d \Leftrightarrow \mathcal{S} \leq \frac{f(d)1^d}{f(d)\varepsilon^d} = \left(\frac{1}{\varepsilon}\right)^d.$$

Haussler's packing lemma.

Haussler shows that packing ranges at distance δ in set systems with n elements and finite VC-dimension d roughly behave like packing spheres of radius $\frac{\delta}{n}$ in $\mathbb{R}^d$. The bound that Haussler

[11]See [CS13] for more information the sphere packing problem.

proved is in fact equal, up to some multiplicative factor, to the bound presented for the sphere packing problem.

Haussler's packing lemma ([Hau95]) Let $(X, \mathcal{F})$ be a set system with VC-dimension $\leq d$. If there exists $\delta \in [1, n]$ such that for all distinct $F, F' \in \mathcal{F}$, $|\Delta(F, F')| \geq \delta$, then there exists a constant c_{H} only depending on d s.t.

$$|\mathcal{F}| \leq c_{\mathrm{H}} \left(\frac{n}{\delta}\right)^d.$$

Haussler also proved in [Hau95] that for all $\delta \in [1, n]$, there exists a δ-separated set system with VC-dimension d of size $\Theta\left(\left(\frac{n}{\delta}\right)^d\right)$ making this bound tight.

His construction is the set system $(X, \mathcal{F})$ with $X = [1, n]$ and

$$\mathcal{F} = W\left(1, \frac{n}{d}\right) \times \ldots \times W\left(n - \frac{n}{d} + 1, n\right).$$

This represents all possible unions of exactly d sets of $W\left(1, \frac{n}{d}\right), \ldots, W\left(n - \frac{n}{d} + 1, n\right)$ where

$$\forall i \in [1, d-1], W\left(\frac{in}{d} + 1, \frac{(i+1)n}{d}\right) = \left\{\emptyset, \left\{\frac{in}{d} + 1\right\}, \ldots, \left\{\frac{in}{d} + 1, \ldots, \frac{(i+1)n}{d}\right\}\right\}.$$

The packing lemma is often used when one wants to enforce some property on all ranges of the set system. This method usually consists of separating the ranges in two groups.

- A collection $\mathcal{C}$ of ranges far apart from each other which size will be bounded using the packing lemma. Enforcing some property on $\mathcal{C}$ will then be "easier" than on the whole collection $\mathcal{F}$.
- The remaining ranges $\mathcal{F} \setminus \mathcal{C}$ where the property will be guaranteed with a small error proportional to the symmetric difference with a range in $\mathcal{C}$.

This method can lead to time complexity improvements as the property is enforced on a small sub-collection of $\mathcal{F}$ only. It can even lead to improvements in the bound when the property can be guaranteed with some total budget as the budget per range increases since the total number of range to enforce the property on decreases. An example of such improvement will be presented in Section 3.4.

Haussler's bound was later refined by Ezra [Ezr16] for set systems with (d, d_1)-property. These set systems are such that for any sequence $I \subseteq X$ with $|I| = l$ and any parameter $1 \leq k \leq l$, the number of sets of size at most k in $\mathcal{F}|_I = \{F \cap I, F \in \mathcal{F}\}$ is of size at most $O\left(n^{d_1} k^{d-d_1}\right)$. Then δ-separated sets of such set system have size:

$$O\left(\frac{n^{d_1} k^{d-d_1}}{\delta^d}\right)$$

where $k = \max_{F \in \mathcal{F}} |F|$. It is interesting to notice that for set systems with no particular bound on the size of the biggest set, that is $k = n$, Ezra's result gives Haussler's bound.

The idea of parameterizing the VC-dimension to more finely describe how range are distributed depending on their size has been generalized as the notion of *shallow-cell complexity*. This notion was first introduced by Chan, Grant, Könemann, and Sharpe [Cha+12][12]. The generalization of Haussler's packing lemma to set systems with bounded shallow-cell complexity is from Mustafa [Mus16].

3.3.1 δ-coverings and δ-packings

We restate the definition of δ-coverings and δ-packings.

Definition 3.12. (δ-packings and δ-coverings) A **δ —packing** over X is $\mathcal{P} \subseteq 2^X$ such that for all $P_1, P_2 \in \mathcal{P}, |\Delta(P_1, P_2)| > \delta$. We say that $\mathcal{P}$ is **maximal** for $(X, \mathcal{F})$ if $\forall F \in \mathcal{F}, \exists P \in \mathcal{P}$ s.t. $|\Delta(F, P)| \leq \delta$.

A **δ —covering** over $(X, \mathcal{F})$ is $\mathcal{C} \subseteq 2^X$ such that for all $F \in \mathcal{F}, \exists C \in \mathcal{C}$ s.t. $|\Delta(F, C)| \leq \delta$. We say that $\mathcal{C}$ is **minimal** if $\forall C \in \mathcal{C}, \exists F \in \mathcal{F}$ s.t. $|\Delta(F, C)| \leq \delta$ and $\forall C' \in \mathcal{C} \setminus \{C\}, |\Delta(F, C')| > \delta$.

We present different methods to obtain δ-coverings of various size.

Greedy construction of maximal δ-packings and minimal δ-coverings.

A simple method to obtain minimal δ-coverings and maximal δ-packings is to construct them greedily.

Algorithm 1: Greedy maximal δ-packings

```
Input: (X, F), δ
P ← ∅
for F ∈ F do
    if min_{P∈P} |Δ(F, P)| > δ then
        P ← P ∪ {F}
return P
```

Lemma 3.13. Given $(X, \mathcal{F})$ a set system and $\delta \in [1, ..., n]$, Algorithm 1 returns a maximal δ-packing of $(X, \mathcal{F})$ in time $O(m|P|n)$.

Proof. The algorithm builds a maximal δ-packing by adding iteratively any range with symmetric difference greater than δ to the packing. At the end of Algorithm 1 for any range F not added to the packing, there exists a range $P \in \mathcal{P}$ s.t. $\Delta(F, P) \leq \delta$.

The time complexity comes from the fact that for each $F \in \mathcal{F}$, we compute its symmetric difference with at most $|\mathcal{P}|$ ranges which can be done in $O(n)$ operations. This gives a time complexity of:

[12] The reader may refer to Section 4.2 of [Mus22] for more information on shallow-cell complexity.

$$\underbrace{m}_{|\mathcal{F}|} \times |P| \times n = O(m|P|n).$$

□

Corollary 3.14. Given a set system $(X, \mathcal{F})$ of finite VC-dimension d and $\delta \in [1, \ldots, n]$, Haussler's packing lemma implies that Algorithm 1 builds a maximal δ-packing of $(X, \mathcal{F})$ of size $O\left(\left(\frac{n}{\delta}\right)^d\right)$ in time $O\left(\frac{mn^{d+1}}{\delta^d}\right)$.

In the same way, we present a greedy algorithm to construct minimal δ-coverings.

Algorithm 2: Greedy minimal δ-coverings

Input: $(X, \mathcal{F})$, δ

$\mathcal{C} \leftarrow \mathcal{F}$
for $F \in \mathcal{F}$ **do**
  if $\min_{C \in \mathcal{C}} |\Delta(F, C)| \leq \delta$ **then**
    $\mathcal{C} \leftarrow \mathcal{C} \setminus \{F\}$
return $\mathcal{C}$

Lemma 3.15. Given $(X, \mathcal{F})$ a set system and $\delta \in [1, \ldots, n]$, Algorithm 2 returns a minimal δ-packing of $(X, \mathcal{F})$ in time $O(m|C|n)$.

We omit the proof of Lemma 3.15 as it is very similar to the proof of Lemma 3.13.

Fast constructions of δ-coverings.

[MWW93] showed that δ-coverings can be constructed efficiently using ε-nets. The resulting algorithm is presented below.

Algorithm 3: Efficient δ-covering algorithm [MWW93]

Input: $(X, \mathcal{F})$, δ

$\mathcal{S} \leftarrow \emptyset$
$\mathcal{C} \leftarrow \emptyset$
$N \leftarrow \frac{\delta}{n}$-net of $(X, \Delta(\mathcal{F}))$
for $F \in \mathcal{F}$ **do**
  $Q \leftarrow F \cap N$
  if $Q \notin \mathcal{S}$ **then**
    $\mathcal{S} \leftarrow \mathcal{S} \cup \{Q\}$
    $\mathcal{C} \leftarrow \mathcal{C} \cup \{F\}$
return $\mathcal{C}$

They proved the following theorem.

Theorem 3.16. **([MWW93])** Given $(X, \mathcal{F})$ a set system with finite VC-dimension d and $\delta \in [1, ..., n]$, Algorithm 3 returns a δ-covering of size $O\left(\left(\frac{n}{\delta}\log\left(\frac{n}{\delta}\right)\right)^d\right)$ in time $O\left(\frac{mdn}{\delta}\log\left(\frac{n}{\delta}\right) + \frac{dn^d}{\delta^d}\log^{d+1}\left(\frac{n}{\delta}\right)\right)$ with probability at least $\frac{1}{\sqrt{2}}$.

Proof. Lemma 3.8 and Corollary 3.6 shows that N can be constructed by sampling $O\left(\frac{n}{\delta}\log\left(\frac{n}{\delta}\right)\right)$ elements of X. Since N is a $\frac{\delta}{n}$-net with probability at least $\frac{1}{\sqrt{2}}$:

$$\forall R \in \Delta(\mathcal{F}) \text{ s.t. } |R| \geq \delta, R \cap N \neq \emptyset.$$

In particular, this means that for two ranges $F, F' \in \mathcal{F}$:

$$\Delta(F, F') \geq \delta \Rightarrow \Delta(F, F') \cap N \neq \emptyset \Leftrightarrow F \cap N \neq F' \cap N$$

as there must exist at least one element of N that is in $F \setminus F'$ or $F' \setminus F$.

For each $F \in \mathcal{F}$, Algorithm 3 computes iteratively the intersection of F with N. Each iteration where $F \cap N$ has not been obtained from some other $F' \cap N$, F is added to $\mathcal{C}$, the cover that will be returned. Its intersection is added to $\mathcal{S}$, the list of intersection that had not been obtained.

This ensures that,

$$\forall F \in \mathcal{F}, \exists C \in \mathcal{C} \text{ s.t. } F \cap N = C \cap N \Rightarrow \Delta(F, F') < \delta.$$

Therefore $\mathcal{C}$ is a δ-cover with size $|\mathcal{S}|$.

We now compute the size of $\mathcal{S}$.

By Lemma 3.1,

$$|\mathcal{S}| = |\mathcal{F}|_N| \leq \sum_{i=0}^{d} \binom{|N|}{i} = O(|N|^d) = O\left(\left(\frac{n}{\delta}\log\left(\frac{n}{\delta}\right)\right)^d\right).$$

The runtime comes from the fact that for each $F \in \mathcal{F}$, we need to determine whether its intersection $F \cap N \in \mathcal{S}$. By maintaining a sorted tree structure on $\mathcal{S}$, we can ensure that the number of comparison to do is $O(\log(|\mathcal{S}|))$ where each comparison can be done in $O(|N|)$ operations. If $F \cap N \notin \mathcal{S}$, we need to insert $F \cap N$ in $\mathcal{S}$. The sorted tree structure guarantees this insertions to be possible in $O(\log(|S|))$ operations. This gives a runtime of:

$$\underbrace{\underbrace{m}_{|\mathcal{F}|} \times \underbrace{d\log\left(\frac{n}{\delta}\log\left(\frac{n}{\delta}\right)\right)}_{\log(|S|)} \times \underbrace{\frac{n}{\delta}\log\left(\frac{n}{\delta}\right)}_{|N|}}_{\text{Determining whether } F\cap N \in \mathcal{S}} + \underbrace{\underbrace{\left(\frac{n}{\delta}\log\left(\frac{n}{\delta}\right)\right)^d}_{|S|} \times \underbrace{d\log\left(\frac{n}{\delta}\log\left(\frac{n}{\delta}\right)\right)}_{\log(|S|)}}_{\text{Adding } F\cap N \text{ to } \mathcal{S} \text{ if } F\cap N \notin \mathcal{S}}$$

$$= O\left(\frac{mdn}{\delta}\log\left(\frac{n}{\delta}\right) + \frac{dn^d}{\delta^d}\log^{d+1}\left(\frac{n}{\delta}\right)\right).$$

□

The construction provided by Haussler to prove the optimality of Haussler's packing lemma exhibits the existence of VC-dimension d set systems where minimal δ-coverings have size $\Theta\left(\left(\frac{n}{\delta}\right)^d\right)$.

As we can see this algorithm is significantly faster than Algorithm 2. However the δ-covering it constructs might not be minimal. This is because the $\frac{\delta}{n}$-net ensures that distant ranges will have different intersections with the net. However, the converse is not true and pairs of ranges close to each others might also have different intersections with the net. This leads to adding pairs of ranges close to each other to the cover computed making it non-minimal.

Fast δ-coverings for set systems spanned by halfspaces using cuttings.

In this section we present cuttings and how they can be used as an efficient tool to compute δ-coverings of small size. This is a result from Matoušek [Mat92].

Definition 3.17. (Cuttings) Given a set H of n hyperplanes in $\mathbb{R}^d$, a $\frac{1}{r}$**-cutting** for H is a collection of possibly unbounded d-dimensional closed simplices with disjoint interiors, which together cover $\mathbb{R}^d$ and such that the interior of each simplex intersects at most $\frac{n}{r}$ hyperplanes.

Theorem 3.18. ([Cha93]) There exists an algorithm that computes a $\frac{1}{r}$-cutting of size $O(r^d)$ in time $O(nr^{d-1})$.

We introduce the set system *corresponding in the dual vector space* to a set system spanned by halfspaces. This construction used on par with cuttings will give δ-covering of small size.

Let $(X, \mathcal{F})$ be a set system spanned by halfspaces in $\mathbb{R}^d$ and let $(Y, \mathcal{G})$ be the corresponding set system in the dual vector space of $(X, \mathcal{F})$[13].

The duality transform, that we denote dual(), is the function that, to a point $x \in \mathbb{R}^d_*$, associates the hyperplane $\{x' \in \mathbb{R}^d : x' \cdot x = 1\}$ and,, to the hyperplane not passing through the origin of the form $\{x' \in \mathbb{R}^d : x' \cdot x = 1\}$, the point $x \in \mathbb{R}^d_*$.

$(Y, \mathcal{G})$ is defined as $Y = \{\text{dual}(H_F) : F \in \mathcal{F}\}$ where H_F is the hyperplane bounding the halfspace F and $\mathcal{G} = \{\text{dual}(x) : x \in X\}$.

Let H be a hyperplane defined as $\{x' \in \mathbb{R}^d : x' \cdot x = 1\}$ for $x \in \mathbb{R}^d_*$. We call H^+ the halfspace bounded by H that does not contain the origin i.e. $\{x' \in \mathbb{R}^d : x' \cdot x > 1\}$. Similarly, we call H^- the halfspace bounded by H that contains the origin: $\{x' \in \mathbb{R}^d : x' \cdot x \leq 1\}$.

The corresponding dual vector space admit some well-known properties that will be necessary for the proof of the covering algorithm that we present below.

[13]This is a different notion than the dual set system presented in Definition 3.4.

Lemma 3.19. Let $x \in \mathbb{R}^d$ and H be a hyperplane of $\mathbb{R}^d$:

- $x \in H \Leftrightarrow \mathrm{dual}(H) \in \mathrm{dual}(x)$,
- $x \in H^+ \Leftrightarrow \mathrm{dual}(H) \in \mathrm{dual}(x)^+$,
- $x \in H^- \Leftrightarrow \mathrm{dual}(H) \in \mathrm{dual}(x)^-$.

For the sake of completeness, we restate the proof of these properties.

Proof of Lemma 3.19. We prove the first assertion.

Let $H = \{x' \in \mathbb{R}^d : x' \cdot y = 1\}$, since $x \in H, x \cdot y = 1$.

In the dual space, $\mathrm{dual}(x) = \{x' \in \mathbb{R}^d : x' \cdot y = 1\}$ and $\mathrm{dual}(H) = y$.

As $x \cdot y = 1$, $\mathrm{dual}(H) \in \mathrm{dual}(x)$.

The other two assertions are proved the same way. □

We now present the algorithm to compute δ-coverings using cuttings first present by Matoušek [Mat92].

Algorithm 4: Fast δ-covering algorithm for halfspaces

Input: $(X, \mathcal{F})$, δ

$(Y, \mathcal{G}) \leftarrow$ dual set system of $(X, \mathcal{F})$

Compute $\{C_1, \dots, C_r\}$ a $\frac{\delta}{n}$-cutting of $(Y, \mathcal{G})$ using Theorem 3.18

return $\left\{\mathrm{dual}(c_1)^-, \mathrm{dual}(c_1)^+, \dots, \mathrm{dual}(c_r)^-, \mathrm{dual}(c_r)^+\right\}$ where $c_1 \in C_1, \dots, c_r \in C_r$

Lemma 3.20. (**[Mat92]**) Let be $(X, \mathcal{F})$ be a set system spanned by halfspaces in $\mathbb{R}^d$ and $\delta \in [1, n]$. Algorithm 4 returns a δ-covering of $(X, \mathcal{F})$ of size $O\left(\left(\frac{n}{\delta}\right)^d\right)$ in time $O\left(d(n + m) + n^d + m\log(n)\right)$.

Proof. Let $(X, \mathcal{F})$ be a set system spanned by halfspaces in $\mathbb{R}^d$ and let $(Y, \mathcal{G})$ be the corresponding set system in the dual vector space of $(X, \mathcal{F})$.

Let two points $y, y' \in Y$ and a hyperplane $G \in \mathcal{G}$. Suppose y and y' are on the two different sides of G. By Lemma 3.19, we have $x \in \mathrm{dual}(y)^-$ and $x' \notin \mathrm{dual}(y')^-$ or $x \in \mathrm{dual}(y')^-$ and $x' \notin \mathrm{dual}(y)^-$ (the same goes for $\mathrm{dual}(y)^+$ and $\mathrm{dual}(y')^+$). This mean that the number of hyperplanes intersecting the segment between y and y' is the symmetric difference between $\mathrm{dual}(y)^+$ and $\mathrm{dual}(y')^+$ or $\mathrm{dual}(y)^-$ and $\mathrm{dual}(y')^-$.

By Theorem 3.18, we can compute $\{C_1, \dots, C_r\}$ a $\frac{n}{\delta}$-cutting of $(Y, \mathcal{G})$ with $r = O\left(\left(\frac{n}{\delta}\right)^d\right)$. We explain why $\left\{\mathrm{dual}(c_1)^-, \mathrm{dual}(c_1)^+, \dots, \mathrm{dual}(c_r)^-, \mathrm{dual}(c_r)^+\right\}$ where $c_1 \in C_1, \dots, c_r \in C_r$ is a δ-covering of $(X, \mathcal{F})$.

By Definition 3.17, for all $k \in [1, r]$, C_k is intersected by at most δ hyperplanes of $\mathcal{G}$. This means that the segment between any two points inside C_k can not be intersected by more than δ hyperplanes. Therefore, the symmetric difference between two ranges $F, F' \in \mathcal{F}$ s.t. $\text{dual}(H_F), \text{dual}(H_{F'}) \in C_k$ is bounded by δ if they have the same orientation.

Finally, for all $F \in \mathcal{F}, \exists k \in [1, r]$ s.t. $\text{dual}(H_F) \in C_k$. That is, since c_k and $\text{dual}(H_F) \in C_k$, $\Delta\left(F, \text{dual}(c_k)^+\right) \leq \delta$ or $\Delta(F, \text{dual}(c_k)^-) \leq \delta$ depending of H_F's orientation.

Runtime analysis. Computing the dual vector space can be done in time $d(n+m)$. Theorem 3.18 gives that computing a $\frac{n}{\delta}$-cutting can be done in time $O\left(\frac{n^d}{\delta^{d-1}}\right)$. To be able to perform point query in the cutting computed, one must perform a preprocessing with time complexity $O\left(n^d\right)$ [Cha93]. Then each query takes $O(\log(n))$ operations and might have to be performed $|Y| = m$ times bringing the total time complexity required to compute $c_1, ..., c_r$ from the cutting to $O\left(n^d + m\log(n)\right)$. Adding the three steps together gives the time complexity stated in the theorem. □

Remark 3.21. We can use two optimizations to reduce the time complexity of Algorithm 4.

First, we do not need to compute the dual of $\mathcal{F}$ as we do not need it to compute the cutting: it is only required for the point query step.

Then instead of performing the query step with $\text{dual}(\mathcal{F})$, we can select c_k to be an arbitrary point of C_k say the corner or the middle point of the cell. This requires to solve a linear system of d equations with d unknowns that is $O(d^3)$ operations. This gives an improved total time complexity of $O\left(dn + \frac{n^d}{\delta^{d-1}} + d^3\left(\frac{n}{\delta}\right)^d\right)$.

However the covering computed has no guarantee to be a sub-collection of $\mathcal{F}$ but it will be a collection of halfspaces.

The covering constructed might not be minimal. For instance some cells of the cutting might be crossed by less than δ hyperplanes making the total number of hyperplanes cutting multiple adjacent cells less than δ. In that case selecting only one range for all these cells would cover all the ranges in it. Nonetheless, the δ-covering computed by Algorithm 2 has size $O\left(\left(\frac{n}{\delta}\right)^d\right)$ matching the size of minimal δ-coverings in the worst case.

Cuttings are an efficient tool to compute small δ-coverings however their implementation [Har00] is non-trivial, and is currently only available in $\mathbb{R}^2$. This heavily limits the scope of applications of algorithms using cuttings.

Weak ε-coverings.

A weak δ-covering $\mathcal{C}$ of $(X, \mathcal{F})$ is a sub-collection of $\mathcal{F}$ such that for any one specific range $F \in \mathcal{F}, \exists C \in \mathcal{C}$ such that $\Delta(F, C) \leq \delta$ however this might not be the case for all ranges $F \in \mathcal{F}$ simultaneously.

Matheny and Phillips studied weak δ-coverings [MP19] and showed that it had application to approximate discrepancy computation. This is different from the applications we present in the next section as, in that case, one wants to approximate the discrepancy of only one range: the range with maximum discrepancy w.r.t. a given coloring.

The main application we will present of δ-coverings is the computation of low-discrepancy colorings. For that problem, it is not possible to use weak δ-coverings as all we will be able to ensure low-discrepancy only on ranges that are covered. That is, we need all ranges to be covered simultaneously.

3.4 Combinatorial discrepancy

As presented in Chapter 1, combinatorial discrepancy is an important problem of combinatorial data approximation. In this section, we will present algorithms on this problem. In particular, we will detail the result of Lovett and Meka [LM15] that compute a low discrepancy coloring of general set systems in polynomial time.

To start of, we show a well-known result: a random bound on the discrepancy of set systems with a random coloring.

Lemma 3.22. Let $(X, \mathcal{F})$ be a set system, and let χ be a uniform random coloring of X that is for all $x \in X$,

$$\chi(x) = \begin{cases} 1 \text{ with probability } \frac{1}{2} \\ -1 \text{ with probability } \frac{1}{2} \end{cases}$$

then

$$\mathrm{disc}_\chi(X, \mathcal{F}) = O\left(\sqrt{n \log(m)}\right)$$

with probability at least $\frac{1}{2}$.

Proof. Let $(X, \mathcal{F})$ be a set system and χ be a uniform random coloring of X. Let $F \in \mathcal{F}$,

$$\mathbb{E}[\chi(F)] = \sum_{x \in F} \mathbb{E}[\chi(x)] = \sum_{x \in F} P(\chi(x) = 1) - P(\chi(x) = -1) = 0.$$

By Hoeffding's inequality:

$$P\left(\chi(F) \leq -\sqrt{\frac{|F|}{2} \log(4)}\right) < \frac{1}{4}$$

and

$$P\left(\chi(F) \geq \sqrt{\frac{|F|}{2} \log(4)}\right) < \frac{1}{4}.$$

Using a union bound and using the bound for all $F \in \mathcal{F}$, $|F| \leq n$, we obtain:

$$P\left(\mathrm{disc}_\chi(X, \mathcal{F}) \geq \sqrt{\frac{n}{2} \log(4m)}\right) < \frac{1}{2}.$$

□

Another formalization of the combinatorial discrepancy problem. A common way to formalize the combinatorial discrepancy problem on finite set systems is to represent the coloring χ as a vector $x \in \{-1,1\}^n$. In that case, denoting $X = \{x_1, ..., x_n\}$, we represent the ranges $F \in \mathcal{F}$ by their indicator vectors $v_F \in \{0,1\}^n$ s.t.

$$v_F[i] = \begin{cases} 1 \text{ if } x_i \in F \\ 0 \text{ otherwise} \end{cases}$$

where $v[i]$ represents the i-th element of a vector v.

The discrepancy problem can then be written as finding $x \in \{-1,1\}^n$ such that

$$x = \min_{y \in \{-1,1\}^n} \max_{F \in \mathcal{F}} |v_F \cdot y|.$$

This formalization of combinatorial discrepancy, allows for a more intuitive explanation of partial colorings. Suppose a coloring x where for all $i \in [1,n]$, $|x[i]| = 1 - \varepsilon$ for some small $\varepsilon > 0$ achieves a small discrepancy D. Then the coloring y with

$$y[i] = \begin{cases} 1 \text{ if } x[i] = 1 - \varepsilon \\ -1 \text{ otherwise} \end{cases}$$

will have discrepancy at most $D + \varepsilon n$[14]. We call these colorings, with some of their coefficients in $]-1,1[$, partial colorings and we refer to the n-dimensional hypercube as the space of colorings as it represents all possible colorings (partial and complete). This type of properties have been used to compute small discrepancy colorings either by combining multiple partial colorings or using them as intermediate colorings to obtain information on candidate final colorings.

3.4.1 Low-discrepancy colorings for general set systems

For a fixed set system, finding a coloring with minimal discrepancy is NP-hard, in fact, simply determining whether the discrepancy of a set system is 0 or not is NP-hard [CNN11]. The main line of work on discrepancy has therefore been to find small but not necessarily minimal discrepancy colorings. In particular Spencer [Spe85] proved the existence of colorings with discrepancy $O(\sqrt{n})$ for set systems with $|\mathcal{F}| = O(n)$ and more generally, set systems admit colorings with discrepancy $O\left(\sqrt{n \log\left(\frac{m}{n}\right)}\right)$. Spencer's result uses iterative partial coloring of the ground set. However Spencer's result is not constructive and, whether it was possible to find a polynomial algorithm computing such coloring, remained an open problem for nearly 25 years until Bansal's breakthrough [Ban10]. Using Spencer's partial coloring idea, he proposed an algorithm that solves semi definite programs to guide a random walk in the space of colorings.

The Lovett-Meka discrepancy algorithm.

[14] Using a more advanced probabilistic bounding techniques called "randomized rounding" we can even reduce that error (see [RT87, You95]).

Following up Bansal's breakthrough, Lovett and Meka [LM15] designed a faster and simpler algorithm to compute low-discrepancy colorings. The Lovett-Meka algorithm is a central tool used in the contribution we present in Chapter 4 so we restate their result formally.

Algorithm 5: [LM15] small discrepancy coloring

Input: $(X, \mathcal{F})$; $c_1, ..., c_m \in \mathbb{R}_+^*$; $x_0 \in [-1,1]^n$

for $t \leftarrow 1$ to $T = 8\log^2(m)$ **do**

$\quad C_t^{\text{var}} \leftarrow \left\{ i \in [1,n] : |x_{t-1}[i]| \geq 1 - \frac{1}{8\log(m)} \right\}$

$\quad C_t^{\text{disc}} \leftarrow \left\{ F \in \mathcal{F} : |(x_{t-1} - x_0) \cdot v_F| \geq c_j - \frac{1}{8\log(m)} \right\}$

$\quad V_t \leftarrow \{ u \in \mathbb{R}^n : \forall i \in C_t^{\text{var}}, u[i] = 0 \text{ and } \forall F \in C_t^{\text{disc}}, u \cdot v_F = 0 \}$

$\quad x_t \leftarrow x_{t-1} + \frac{U_t}{16\log(m)}$ where U_t is a standard Gaussian random variable in the space V_t

$\quad$ **if** $\exists i \in [1,n] : |x_{t-1}[i]| > 1$ or $F \in \mathcal{F} : |(x_{t-1} - x_0) \cdot v_F| \geq c_j$ **then**

$\quad\quad$ The algorithm fails

return x_T

This algorithm is surprisingly simple as it consists in a random walk where each additive step is a random Gaussian vector in the space of colorings. The directions that the random walk can have are subjected to some constraints. First there are constraints to ensure that the coloring will remain in the space of coloring, that is the n-dimensional hypercube, (C_t^{var}). Then, additional constraints are needed to ensure that no range will have too large discrepancy (C_t^{disc}). To prevent the walk from violating any constraints, if the partial coloring approaches a constraint, the algorithm enforces that the following steps are drawn from the space orthogonal to that constraint (V_t). This leads to a walk where the dimension of the space from which the Gaussian vectors are drawn reduces over the course of the walk. In addition to that there is a *danger zone* for each constraint. The Gaussian vectors are only drawn from the space orthogonal to a constraint when it is in its danger zone rather than once it reaches or exceeds it. The danger zone of a constraint i the strip of the coloring space close to the constraint. If a random step makes the walk overtake a constraint then the algorithm fails, however the probability of this happening is small.

We illustrate the walk from Algorithm 5 in Figure 7. We draw a walk in $\mathbb{R}^2$ starting at the origin with one range. The constraints associated with the hypercube are drawn in blue and the danger zones associated are delimited by dashes and filled in blue. The constraint associated with the range is drawn in red. We see that the walk first hit the red danger zone after what it starts moving orthogonally to the constraint before hitting a blue danger zone and finishing its course because it has no more dimension to move in.

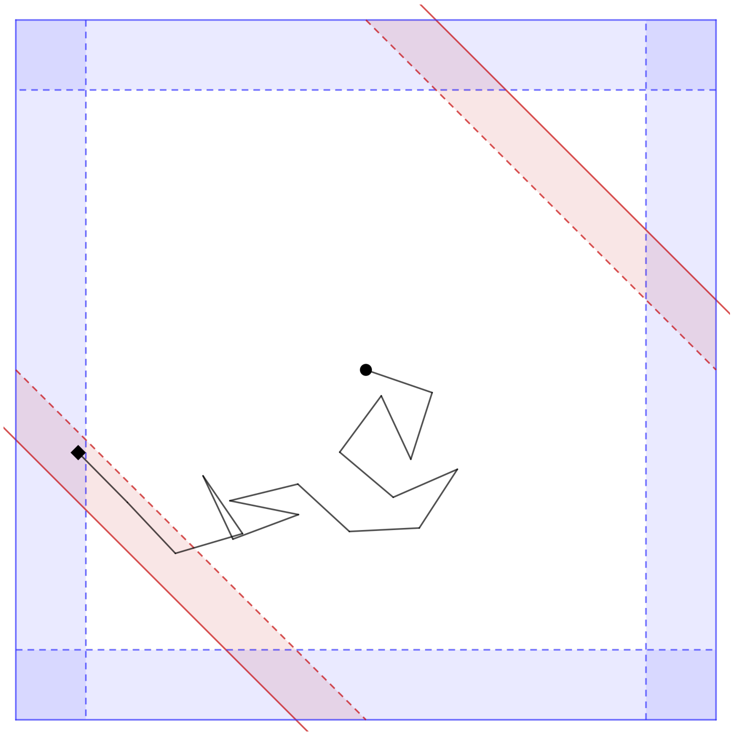


Figure 7: An illustration of the random walk from [LM15]

The following theorem is the main coloring theorem of [LM15].

Theorem 3.23. ([LM15] partial coloring theorem) Let $v_1, ..., v_m \in \mathbb{R}^n$ and $x_0 \in [-1,1]^n$. Let $c_1, ..., c_m \geq 0$ such that $\sum_{j=1}^{m} \exp\left(-\frac{c_j^2}{16}\right) \leq \frac{n}{16}$. Algorithm 5 finds, with probability at least 0.1, $x \in [-1,1]^n$ such that:

(i) $\forall j \leq m, |(x - x_0) \cdot v_j| \leq c_j \|v_j\|_2$,

(ii) $\left|\left\{1 \leq i \leq n : |x_i| \geq 1 - \frac{1}{8\log(m)}\right\}\right| \geq \frac{n}{2}$.

Moreover this algorithm has runtime $O((n+m)^3 \log^3(m))$.

We detail this theorem as we will use some of its properties in Chapter 4 and Chapter 6.

Property (i) states that, given a set system, a starting point and a budget for each range of the set system, Algorithm 5 returns a coloring of X with discrepancy proportional to the budget for each range. Naturally, there is a total budget limit and it is not possible to have a large budget for each range. This is represented by the inequality

$$\sum_{j=1}^{m} \exp\left(-\frac{c_j^2}{16}\right) \leq \frac{n}{16}$$

that we call *the entropy condition* of Theorem 3.23. However if all ranges get the same budget, we see that we can have a budget of $O\left(\log\left(\frac{m}{n}\right)\right)$ per range which means that the coloring returned has discrepancy $O\left(\sqrt{n \log\left(\frac{m}{n}\right)}\right)$.

Property (ii) means that after running the algorithm, at least half of the elements of X will be colored. This means that Algorithm 5 might need to be ran $\log(n)$ times to obtain a complete coloring. This is possible as the starting point of the walk is flexible and we can restart each run where the previous one ended.

We detail this process. Algorithm 5 with $x_0^1 = 0_{\mathbb{R}^n}$ and constraints $c_1^1, \ldots, c_m^1$ returns a partial coloring x_1. Then, running Algorithm 5 again on only the remaining—at most $\frac{n}{2}$—uncolored points and starting from $x_0^2 = x_1$ with constraints $c_1^2, \ldots, c_m^2$ returns a second coloring x_2. This is done by effectively setting the indices of colored elements to 0 in all v_j. Doing this $\log(n)$ times returns $x_{\log(n)}$, a complete coloring, and we have:

$$\begin{aligned}
\forall j, |x_{\log(n)} \cdot v_j| &= |\left(x_{\log(n)} - 0_{\mathbb{R}^n}\right) \cdot v_j| \\
&= |\left(x_{\log(n)} - x_{\log(n)-1} + x_{\log(n)-1} - x_{\log(n)-2} + \ldots + x_1 - 0_{\mathbb{R}^n}\right) \cdot v_j| \\
&\leq |\left(x_{\log(n)} - x_{\log(n)-1}\right) \cdot v_j| + \ldots + |(x_1 - 0_{\mathbb{R}^n}) \cdot v_j|. \qquad (1)
\end{aligned}$$

However at each step, we remove at least half of the elements of X that have been colored during this step, which means that for all j, at most $\frac{n}{2^k}$ elements of v_j are non-zero, thus $\left\|v_j\right\|_2 \leq \sqrt{\frac{n}{2^k}}$. Using this result with property (i) of Theorem 3.23 gives

$$\forall k, |(x_k - x_{k-1}) \cdot v_j| \leq c_j^k \sqrt{\frac{n}{2^k}}.$$

Using this in (1) gives:

$$\forall j, |x_{\log(n)} \cdot v_j| = \sum_{k=1}^{\log(n)} c_j^k \sqrt{\frac{n}{2^k}}.$$

In particular when setting for all $j \in [1, m]$, $c_j^k = O\left(\sqrt{\log\left(\frac{m2^k}{n}\right)}\right)$, we obtain:

$$\forall j, |x_{\log(n)} \cdot v_j| = \sum_{k=1}^{\log(n)} O\left(\sqrt{\frac{n \log\left(\frac{m2^k}{n}\right)}{2^k}}\right) = O\left(\sqrt{n \log\left(\frac{m}{n}\right)}\right).$$

With this technique, we obtain a complete coloring with discrepancy of the same order as the first partial coloring obtained with Algorithm 5. This generalizes to any choice of $c_1, \ldots, c_m$.

Next we present a lemma that shows how Algorithm 5 colors subsets of X that are not used as constraints. In fact any subset of X will be colored and it turns out the discrepancy of these subsets will be the same as if they had been colored randomly, that is discrepancy $O\left(\sqrt{n \log(mn)}\right)$ with constant probability.

Lemma 3.24. ([LM15] random coloring) For all $v \in \{0,1\}^n$, $c > 0$, let x be the coloring returned by Algorithm 5($(X, \mathcal{F})$,x_0) for $x_0 \in [-1,1]^n$, then

$$P\left(|(x - x_0) \cdot v| \geq \sqrt{8} c \|v\|_2 \log(m)\right) \leq 2 \exp\left(-\frac{c^2}{2}\right).$$

Proof. Bansal shows in his work [Ban10] a key result on martingale that we restate.

Lemma ([Ban10]) Let $X_0, X_1, \ldots, X_T$ be a martingale with increments $Y_i = X_i - X_{i-1}$. Suppose for $i \in [1, T]$, we have that the conditional distribution of Y_i given $(X_{i-1}, \cdots, X_0)$ is Gaussian with mean 0 and variance at most 1. Then for all $c > 0$,

$$P\left(|X_T - X_0| \geq c\sqrt{T}\right) \leq 2\exp\left(-\frac{c^2}{2}\right).$$

In Algorithm 5, at all iterations t, $x_t - x_{t-1} = \frac{U_t}{16\log(m)}$ and the conditional distribution of $\frac{U_t}{16\log(m)}$ given $(x_{t-1}, \cdots, x_0)$ is Gaussian with mean 0 and variance at most 1.

Let $v \in \{0,1\}^n$. At all iterations t, $(x_t - x_{t-1}) \cdot \frac{v}{\|v\|_2} = \frac{U_t \cdot v}{16\|v\|_2 \log(m)}$ and the conditional distribution of $\frac{U_t \cdot v}{16\|v\|_2 \log(m)}$ given $(x_{t-1}, \cdots, x_0)$ is Gaussian with mean 0 and variance at most 1 as it is a normalized sum of Gaussian random variables with mean 0 and variance at most 1.

Using Bansal's result, we obtain for all $c > 0$,

$$\begin{aligned} &P\left(\left|\left(x_{\sqrt{8}\log(m)} - x_0\right) \cdot v\right| \geq \sqrt{8}c\|v\|_2 \log(m)\right) \\ &= P\left(\left|\left(x_{\sqrt{8}\log(m)} - x_0\right) \cdot \frac{v}{\|v\|_2}\right| \geq \sqrt{8}c\log(m)\right) \\ &\leq 2\exp\left(-\frac{c^2}{2}\right). \end{aligned}$$

□

Corollary 3.25. For all $v \in \{0,1\}^n$, let $x_1, \ldots, x_{\log(n)}$ be the colorings returned by Algorithm 5 by running it iteratively, then for all $\nu > 0$,

$$P\left(|x_{\log(n)} \cdot v| \geq 4\|v\|_2 \log(m)\log(n)\sqrt{\ln\left(mn\frac{\log(n)}{\nu}\right)}\right) \leq \frac{2\nu}{mn}.$$

Proof. The algorithm is ran $\log(n)$ times to obtain a complete coloring $x_{\log(n)}$. The final discrepancy becomes:

$$\begin{aligned} |x_{\log(n)} \cdot v| &= |\left(x_{\log(n)} - 0_{\mathbb{R}^n}\right) \cdot v| \\ &= |\left(x_{\log(n)} - x_{\log(n)-1} + x_{\log(n)-1} - x_{\log(n)-2} + \ldots + x_1 - 0_{\mathbb{R}^n}\right) \cdot v| \\ &\leq |\left(x_{\log(n)} - x_{\log(n)-1}\right) \cdot v| + |\left(x_{\log(n)-1} - x_{\log(n)-2}\right) \cdot v| + \ldots + |(x_1 - 0_{\mathbb{R}^n}) \cdot v|. \end{aligned}$$

Lemma 3.24 with a family of coefficients $(c_k)_{1 \leq k \leq \log(n)} > 0$ gives,

$$P\left(|x_{\log(n)} \cdot v| \geq \sum_{k=1}^{\log(n)} \sqrt{8}c_k\|v\|_2 \log(m)\right) \leq 2\sum_{k=1}^{\log(n)} \exp\left(-\frac{c_k^2}{2}\right).$$

In particular, with for all $k \in [1, \log(n)]$, $c_k = \sqrt{2\ln\left(mn\frac{\log(n)}{\nu}\right)}$ in the previous equation, we obtain:

$$P\left(|x_{\log(n)} \cdot v| \geq 4\|v\|_2 \log(m)\log(n)\sqrt{\ln\left(mn\frac{\log(n)}{\nu}\right)}\right) \leq \frac{2\nu}{mn}.$$

□

Remark 3.26. It is important for this result to depend on the norm of v as the applications we present in Chapter 4 and Chapter 6 will require this. This is why we can not apply the same technique to obtain a geometric sum when summing over the different runs of the algorithm. This would result in a dependence in n instead of $\|v\|_2$ as it might be that, in between two runs, no non-zero coefficient of v is colored in particular if $\|v\|_2$ is small.

Despite Algorithm 5 being stochastic, in the next sections, for simplification purpose, we do not handle its success probability and assume that it succeeds all the times. This is a standard practice that can be applied to any stochastic algorithm. Suppose a stochastic algorithm has a probability of success of $\frac{1}{a}$ for some $a > 1$. Then in expectation, we obtain a successful run of such algorithm for every a runs of the algorithm. Therefore running the algorithm until obtaining a successful run of the algorithm has an expected time complexity of a times the time complexity of the algorithm.

This type of algorithms are called *Monte Carlo algorithm* as they return the right solution with a given probability. The procedure described above is what is called a *Las Vegas algorithm*. This means that the procedure returns the right solution with probability 1 but the resources that will be used are unknown. The procedure is also the natural way to transform a Monte Carlo algorithm into a Las Vegas algorithm. By replacing a Monte Carlo algorithm by its Las Vegas counterpart we can ensure a probability of success of 1 but the time complexity given will be the expected time complexity and will be multiplied by the inverse of the success probability as explained above.[15]

Algorithm 5's success probability is $O\left(\frac{1}{\log(n)}\right)$, therefore we can expect to obtain a successful run in a logarithmic number of runs. Therefore, by using its Las Vegas counterpart, we have an algorithm that returns a coloring with the discrepancy chosen as input for its constraint range and expected time complexity $\tilde{O}((n+m)^3)$. However, this does not change the probability of Corollary 3.25 that we will bound and will play a role in the success probability of the algorithms we present.

Other related works on low-discrepancy colorings algorithms.

Levy, Ramadas and Rothvoss [LRR17] proposed a derandomization of Lovett and Meka's work using the Multiplicative Weight Update technique with runtime $\tilde{O}(n^4 m)$. The major issue with these algorithms is their dependence on m, as in general $m \gg n$. This field of research is very active and has seen various improvements such as a no partial coloring algorithm

[15] See section 3.5 of [MU17] for a more detailed example.

[HSS14] and algorithms for different type of set systems such as low-degree set systems [BM20] or sparse set systems [JSS23].

A related problem of the combinatorial discrepancy problem is the l_∞-discrepancy problem. A is a given $m \times n$ matrix. The goal is to find $x \in \{-1, 1\}^n$ so as to minimize $\|Ax\|_\infty$. We recover the set discrepancy problem when the coefficients of A are either 0 or 1. In that case the columns of A are the indicator vectors of each set. Some work has been done on solving this problem with random walks [BDG19, Ban+19]. Recently some combinatorial algorithm [Gre23] and some optimization algorithm [DSW22] lowered runtime on that line of study to $O(\text{nnz}(A) + n^{2.53})$.

3.4.2 Low-discrepancy coloring colorings of finite VC-dimension set systems

The results presented so far apply to general set systems. However, when working with finite VC-dimension set systems, it is possible to improve them.

Matoušek, Welzl and Wernisch proved in [MWW93] that finite VC-dimension set system admit colorings with discrepancy of order $O\left(n^{\frac{1}{2}-\frac{1}{2d}} \log(m)\right)$. This result was further improved in [Mat95] to $O\left(n^{\frac{1}{2}-\frac{1}{2d}}\right)$ which is optimal [Ale90].

The first result rely on the partial coloring lemma due to Beck [Bec81], that we state below.

Theorem 3.27. **([Bec81])** Let $(X, \mathcal{F})$ be a set system and $\mathcal{G}$ a collection of subsets of X s.t. for all $G \in \mathcal{G}, |G| \leq s$ and

$$\prod_{F \in \mathcal{F}} (|F| + 1) \leq 2^{\frac{n-1}{5}}$$

then there exists a partial coloring $\chi : X \to \{-1, 0, 1\}$ such that at least $\frac{n}{10}$ elements of X are colored and for all $F \in \mathcal{F}, \chi(F) = 0$ and for all $G \in \mathcal{G}, |\chi(G)| \leq \sqrt{2s \ln(4|\mathcal{G}|)}$.

We first explain how to obtain the bound of [MWW93] using [Bec81].

Let $(X, \mathcal{F})$ be a set system with finite VC-dimension d and K a large constant in $\mathbb{R}$. By Haussler's packing lemma, there exists a $Kn^{1-\frac{1}{d}} \log^{\frac{1}{d}}(n)$-covering $\mathcal{C}$ of $(X, \mathcal{F})$ with size

$$O\left(\frac{n^d}{\left(n^{1-\frac{1}{d}} \log^{\frac{1}{d}}(n)\right)^d}\right) = O\left(\frac{n}{\log(n)}\right).$$

Let $\mathcal{G} = \{F \setminus C_F : F \in \mathcal{F}\} \cup \{C_F \setminus F : F \in \mathcal{F}\}$ where

$$C_F \in \mathcal{C} \text{ s.t. } |\Delta(F, C_F)| \leq Kn^{1-\frac{1}{d}} \log^{\frac{1}{d}}(n).$$

It is possible to apply [Bec81] with $\mathcal{C}$ and $\mathcal{G}$ as $n^{\frac{n}{\log(n)}} = O(2^n)$ which means that for a large enough K, the condition of [Bec81] is respected. This gives a coloring χ such that for all $C \in \mathcal{C}, \chi(C) = 0$ and for all $G \in \mathcal{G}, |\chi(G)| \leq \sqrt{2Kn^{1-\frac{1}{d}} \log^{\frac{1}{d}}(n) \ln(4|\mathcal{G}|)} \leq \sqrt{4Kn^{1-\frac{1}{d}} \ln(8m)}$. This means that for all $F \in F$,

$$\chi(F) \leq \underbrace{\chi(C_F)}_{=0} + \chi\left(\underbrace{F \setminus C_F}_{\in \mathcal{G}}\right) + \chi\left(\underbrace{C_F \setminus F}_{\in \mathcal{G}}\right) \leq 2\sqrt{4Kn^{1-\frac{1}{d}} \ln(8m)}.$$

To remove the log factor from the result of [MWW93], Matoušek used a technique called *chaining*. The idea behind this technique is to compute a family of coverings of different size to bound the discrepancy of the *error sets*, i.e., the difference between the sets from $\mathcal{C}$ and the ranges of $\mathcal{F}$.

To do so, we need to overcome a limitation of [Bec81]: it only uses two collections of sets. Among these two collections, one of them will have discrepancy 0 with respect to the coloring. This can be generalized to multiple collections of sets that obtain a discrepancy determined by some budget. This idea resulted in a theorem called the entropy method and was suggested by Boppana to simplify [Spe85] result.

Theorem 3.28. (Entropy method) Let $(X, \mathcal{F})$ be a set system and for all $F \in \mathcal{F}$, let $c_F > 0$. If, for some absolute constant $K \in \mathbb{R}$,

$$\sum_{F \in \mathcal{F}} K \exp\left(-\frac{c_F^2}{4|F|}\right) \log\left(2 + \frac{\sqrt{|F|}}{c_F}\right) < \frac{n}{5},$$

then there exists a coloring χ of X s.t. for all $F \in \mathcal{F}, |\chi(F)| < c_F$.

This theorem is very similar to Theorem 3.23. In fact, in a way, Theorem 3.23 is the constructive version of Theorem 3.28 that is purely existential. In the same way than Theorem 3.23, there is a total budget that depends on a function of the discrepancy that we want in the coloring and the size of each set.

We explain the result of [Mat95] to improve the discrepancy bound from $O\left(n^{\frac{1}{2}-\frac{1}{2d}} \log^{\frac{1}{2d}}(m)\right)$ to $O\left(n^{\frac{1}{2}-\frac{1}{2d}}\right)$.

Let $\mathcal{C}_1, ..., \mathcal{C}_{\log(n)}$ where for all $i \in [1, \log(n)]$, $\mathcal{C}_i$ is a maximal $\frac{n}{2^i}$-packing of $(X, \mathcal{F})$ and $\mathcal{C}_i \subseteq \mathcal{F}$. In particular, $C_{\log(n)} = \mathcal{F}$ and we set $C_0 = \emptyset$. By Haussler's packing lemma, for all

$$i \in [1, \log(n)], |\mathcal{C}_i| = O\left(2^{di}\right).$$

We have

$$\forall C \in \mathcal{C}_i, \exists C' \in \mathcal{C}_{i-1} \text{ s.t. } |\Delta(C, C')| \leq \frac{n}{2^i}..$$

since $\mathcal{C}_{i-1}$ is a $\frac{n}{2^{i-1}}$-covering of $\mathcal{F}$ and $\mathcal{C}_i \subseteq \mathcal{F}$.

Remark 3.29. The fact that every covering is a sub collection of $\mathcal{F}$ is essential here as it ensures that no new range to cover is introduced at any level of the chaining process. It also ensures that the i^{th} covering of the chaining process will be a covering of all other coverings computed in previous steps. In particular this exclude computing fast packings from cuttings as explained in Remark 3.21.

Let $\mathcal{G}_i = \{C \setminus C' : C \in \mathcal{C}_i\} \cup \{C' \setminus C : C \in \mathcal{C}_i\}$ where

$$C' \in \mathcal{C}_{i-1} \text{ s.t. } |\Delta(C, C')| \leq \frac{n}{2^{i-1}}.$$

We apply Theorem 3.28 with

$$\forall C \in \mathcal{C}_i, c_C = K_1 \frac{n^{\frac{1}{2}-\frac{1}{2d}}}{\left(1 + |i - \frac{\log(n)}{d}|\right)^2}$$

for some large constant K_1 to obtain a coloring χ s.t.

$$\forall C \in \mathcal{C}_i, |\chi(C)| < K_1 \frac{n^{\frac{1}{2}-\frac{1}{2d}}}{\left(1 + |i - \frac{\log(n)}{d}|\right)^2}.$$

The computational details of why this choice of c_C satisfies the condition of Theorem 3.28 can be read in Section 5.5 of [Mat99].

We compute the discrepancy of this coloring on the ranges. Every range can be written as the sum of the errors between sets from the packings:

$$\forall F \in \mathcal{F}, \exists C_1, C_1' \in \mathcal{C}_1, \ldots, C_{\log(n)}, C'_{\log(n)} \in \mathcal{C}_{\log(n)} \text{ s.t.}$$
$$F = \left(\ldots(((C_1 \setminus C_1') \cup C_2) \setminus C_2') \cup \ldots \cup C_{\log(n)}\right) \setminus C'_{\log(n)}.$$

This gives:

$$\begin{aligned}
\chi(F) &\leq \sum_{i=1}^{\log(n)} \chi(C_i) + \chi(C_i') \\
&\leq 2K_1 n^{\frac{1}{2}-\frac{1}{2d}} \sum_{i=1}^{\log(n)} \frac{1}{\left(1 + |i - \frac{\log(n)}{d}|\right)^2} \\
&\leq 4K_1 n^{\frac{1}{2}-\frac{1}{2d}} \sum_{i=1}^{\infty} \frac{1}{(1+i)^2} \\
&= \frac{2K_1 \pi^2 n^{\frac{1}{2}-\frac{1}{2d}}}{3}.
\end{aligned}$$

3.4.3 Combinatorial games about discrepancy

In Chapter 4, we introduce a new combinatorial game related to discrepancy. Combinatorial games is an important topic with applications to many problems[16]. In particular the work of Alon, Krivelevich, Spencer and Szabó [Alo+05] studies the combinatorial discrepancy problem as a game. In their game the two players will chose alternatively an element of the ground set of a set system.

The goal for the first player, called the *balancer*, is to finish the game with approximately half of the elements of each range of the set system. The goal for the other player, called the

[16] See [Fra12] for a general survey on combinatorial games.

unbalancer, is to avoid this situation. The authors show the existence of a winning strategy for the balancer if the approximation error allowed to win the game is large enough.

This game relates to discrepancy as the two players can be seen as the two colors to define a coloring. The approximation error obtained by the balancer will be the discrepancy of the coloring defined by the choices of the two players.

The game we present in Chapter 4 is very different from this one as the two players are not in competition to chose their respective objects. In [Alo+05], the two players chose elements from the same ground set and the choice of one element will prevent the other player from choosing the same element at a later stage of the game. In our game, the two players are choosing different types of objects: the first player chooses coloring and the second player ranges. This leads to an easier analysis of strategies as the strategies of our two players do not necessarily depend on each others.

3.5 Simplicial partitions

In this section, we present some work on simplicial partitions that we introduced in Chapter 1 under their parameterized name: (t, κ)-partitions. The study of (t, κ)-partitions originated in computational geometry in the late 1980s, under the name *simplicial partitions*. A breakthrough result that established a key bound, as well as their significance, is the result from Matoušek [Mat92] who gave a construction of simplicial partitions using cuttings (cf Definition 3.17).

Algorithm 6: [Mat92] simplicial partition algorithm

Input: $(X, \mathcal{H}), t \in [2, \frac{n}{2}]$

$\forall H \in \mathcal{H}, \pi_0(H) \leftarrow 1$

for $i \leftarrow 0$ to $t-1$ **do**

  $Q_i \leftarrow X \setminus \left(\cup_{k=1}^{i} P_k\right)$

  $\mathcal{H}^{\pi_i} \leftarrow \bigcup_{H \in \mathcal{H}} \left\{ \underbrace{H, \ldots, H}_{\pi_i(H) \text{ times}} \right\}$ (this is a multiset)

  $C_i \leftarrow$cutting of $(Q^i, \mathcal{H}^{\pi_i})$ with at most $\frac{t|Q^i|}{n}$ cells

  $P_{i+1} \leftarrow \frac{n}{t}$ elements of a cell of C

  $\forall H \in \mathcal{H}, \pi_{i+1}(H) \leftarrow \pi_i(H) \times 2^{I(P_{i+1}, H)}$

return $P_1, \ldots P_t$

Overview of the algorithm. This algorithm successively constructs the parts of the partition. It relies on Multiplicative Weight Update (MWU). This means maintaining weights on each range that evolves exponentially with the crossing number of the range. After each part is constructed, it doubles the weight of ranges that cross it. This idea allows to maintain all ranges' crossing number of similar order as if one range was to be intersected more than the others it would be avoided in the next iterations even at the cost of intersecting many other range. This idea has been applied to design a wide variety of algorithms (see [AHK12] for more details).

To construct the i-th part, the algorithm will compute a cutting with at most $\frac{t|Q^i|}{n}$ cells, that is an $O\left(\left(\frac{n}{t|Q^i|}\right)^{\frac{1}{d}}\right)$-cutting of $(Q^i, \mathcal{H}^{\pi_i})$ where Q_i contains the elements of X that have not been assigned to parts 1 to $i-1$ and $\mathcal{H}^{\pi_i}$ is the multiset of all hyperplanes $H \in \mathcal{H}$ repeated $\pi_i(H)$ times. The part will then be constructed by choosing $\frac{n}{t}$ arbitrary elements in one cell of the cutting.

Theorem 3.30. ([Mat92]) Given a set X of n elements in $\mathbb{R}^d$, let $\mathcal{H}$ denote the family of hyperplanes induced by X. That is, the collection of all hyperplanes defined by at least $d+1$ elements of X. Then for any integer parameter $t \in \left[2, \frac{n}{2}\right]$, Algorithm 6 computes a $\left(t, O\left(\log(m) + t^{1-\frac{1}{d}}\right)\right)$-partition of the set system on X induced by the halfspaces defined by $\mathcal{H}$.

Proof. Suppose we have constructed parts $1, ..., i-1$.

The cutting C_i has size $\frac{t|Q^i|}{n}$ therefore, by the pigeonhole principle, there exists at least one cell of C_i with at least $\frac{n}{t}$ points.

for all $H \in \mathcal{H}, \pi_i(H) = 2^{\sum_{k=1}^{i} I(P_k, H)}$, thus:

$$\max_{H \in \mathcal{H}} \sum_{k=1}^{i} I(P_k, H) \le \log\left(\sum_{H \in \mathcal{H}} \pi_i(H)\right). \qquad (2)$$

We want to evaluate the increase factor of $\pi_i(\mathcal{H}) := \sum_{H \in \mathcal{H}} \pi_i(H)$ between each iteration. The hyperplanes which weight increases between two iterations are the hyperplanes intersecting the cell from which we select the $\frac{n}{t}$ points. This means that their number is bounded by Theorem 3.18: C_i is an $O\left(\left(\frac{n}{t|Q^i|}\right)^{\frac{1}{d}}\right)$-cutting, therefore, its cells are intersected by at most $\frac{C|\mathcal{H}^{\pi_i}|n^{\frac{1}{d}}}{t^{\frac{1}{d}}|Q^i|^{\frac{1}{d}}}$ hyperplanes for some constant C. This gives the following induction formula on $\pi_i(\mathcal{H})$.

$$\begin{aligned}
\pi_{i+1}(\mathcal{H}) &\le \pi_i(\mathcal{H}) + \frac{C|\mathcal{H}^{\pi_i}|n^{\frac{1}{d}}}{t^{\frac{1}{d}}|Q^i|^{\frac{1}{d}}} \\
&= \pi_i(\mathcal{H}) + \frac{C\pi_i(\mathcal{H})n^{\frac{1}{d}}}{t^{\frac{1}{d}}\left(n - \frac{in}{t}\right)^{\frac{1}{d}}} \\
&= \pi_0(\mathcal{H}) \prod_{k=0}^{i} \left(1 + \frac{Cn^{\frac{1}{d}}}{t^{\frac{1}{d}}\left(n - \frac{kn}{t}\right)^{\frac{1}{d}}}\right) \\
&= \pi_0(\mathcal{H}) \prod_{k=0}^{i} \left(1 + \frac{C}{(t-k)^{\frac{1}{d}}}\right).
\end{aligned}$$

Using the inequality $\log(1+x) \le x$, we obtain

$$\log\left(\pi_{t(\mathcal{F})}\right) \leq \log(\pi_0(\mathcal{F})) + \sum_{k=1}^{t} \frac{1}{k^{\frac{1}{d}}}$$
$$= O\left(\log(m) + t^{1-\frac{1}{d}}\right).$$

Using this result in (2) finishes the proof. □

An important open problem related of simplicial partitions is that the exact class of set systems admitting low-crossing partitions is unknown. Their existence have been demonstrated for complex set systems such as semi-algebraic set systems [AMS13].

However, not all set systems—even very simple geometric ones—admit partitions for all parameters t. Alon, Haussler and Welzl [AHW87] showed that for $t = O(\sqrt{n})$, finite projective planes do not admit partitions of size t with sublinear crossing number even though these set systems have VC-dimension 2.

3.6 Computing ε-approximation with sub quadratic size in finite VC-dimension

As presented in Section 3.2.2, it is possible to construct an ε-approximation of a set system $(X, \mathcal{F})$ with finite VC-dimension by simply sampling uniformly X. This results in ε-approximation of size $O\left(\frac{d}{\varepsilon^2}\right)$ where d is the VC-dimension of $(X, \mathcal{F})$. It is possible to achieve smaller ε-approximations by sampling X non-uniformly. In this section, we present results showing how to guide sampling to achieve ε-approximation of size $O\left(\frac{d}{\varepsilon^{\frac{2d}{d+1}}}\right)$. First we use simplicial partitions and second combinatorial discrepancy.

3.6.1 Simplicial partitions

Simplicial partitions can be used to compute sub-quadratic sized ε-approximations (see Section 3.2.2 for the definition and more details).

The following theorem from Suri, Toth and Zhou shows that the existence of low-crossing partition implies the existence of small ε-approximations.

Lemma 3.31. (**[STZ06]**) Let $\mathcal{P} = \{P_1, ..., P_t\}$ be a $\left(t, t^{1-\frac{1}{d}}\right)$-partition of a sets system $(X, \mathcal{F})$, then a set $A = \{x_1, ..., x_r\} \subseteq X$ such that $x_1, ..., x_t$ are selected uniformly at random respectively in $P_1, ..., P_t$ is a $O\left(\left(\frac{1}{t}\right)^{\frac{d+1}{2d}} \sqrt{\ln(m)}\right)$-approximation of $(X, \mathcal{F})$ with probability at least $\frac{1}{2}$.

Proof. Let $A \in X$ constructed as in Lemma 3.31's statement. Let $F \in \mathcal{F}$. A part that does not intersect F does not contribute to the approximation error as the possible choices of elements from A:

$$\left|\frac{|F\cap A|}{|A|}-\frac{|F|}{|X|}\right|=\left|\frac{\sum_{i=1}^{t}\mathbb{1}_{x_i\in F}}{|A|}-\frac{|F\cap P_1|+\ldots+|F\cap P_t|}{|X|}\right|$$
$$=\left|\sum_{i\in I}\frac{\mathbb{1}_{x_i\in F}}{|A|}-\frac{|F\cap P_i|}{|X|}+\sum_{i\in[1,t]\setminus I}\frac{\mathbb{1}_{x_i\in F}}{|A|}-\frac{|F\cap P_i|}{|X|}\right|.$$

where I is the set of indices $i\in[1,t]$ such that F intersects P_i.

Parts P_i such that $i\notin I$ are of two types:

- Either $P_i\subseteq F$, in that case $\mathbb{1}_{x_i\in F}=1$ and $F\cap P_i=P_i$, that is: $\frac{\mathbb{1}_{x_i\in F}}{|A|}-\frac{|F\cap P_i|}{|X|}=\frac{1}{t}-\frac{\frac{n}{t}}{n}=0$.
- Either $P_i\cap F=\emptyset$, in that case $\mathbb{1}_{x_i\in F}=0$, that is: $\frac{\mathbb{1}_{x_i\in F}}{|A|}-\frac{|F\cap P_i|}{|X|}=0$.

Thus:

$$\left|\frac{|F\cap A|}{|A|}-\frac{|F|}{|X|}\right|=\left|\sum_{i\in I}\frac{\mathbb{1}_{x_i\in F}}{|A|}-\frac{|F\cap P_i|}{|X|}\right|.$$

Since $\mathcal{P}$ is a $\left(t,t^{1-\frac{1}{d}}\right)$-partition, $|I|\leq t^{1-\frac{1}{d}}$.

The error coming from each part P_i with $i\in I$ are independent bounded random variables in the range $\left[0,\frac{2n}{t}\right]$. That is, by Hoeffding's inequality, we have:

$$P\left(\left|\sum_{i\in I}\frac{\mathbb{1}_{x_i\in F}|X|}{|A|}-|F\cap P_i|\right|\geq\varepsilon n\right)\leq 2\exp\left(-\frac{2\varepsilon^2n^2}{t^{1-\frac{1}{d}}\left(\frac{2n}{t}\right)^2}\right)$$
$$=2\exp\left(-\frac{\varepsilon^2t^{1+\frac{1}{d}}}{2}\right).$$

That is with $\varepsilon=\frac{\sqrt{2\ln(4m)}}{t^{\frac{d+1}{2d}}}$:

$$P\left(\left|\sum_{i\in I}\frac{\mathbb{1}_{x_i\in F}|X|}{|A|}-|F\cap P_i|\right|\geq\varepsilon n\right)\leq\frac{1}{2m}.$$

A union bound over all ranges of $\mathcal{F}$ finishes the proof. □

3.6.2 Low-discrepancy colorings

In this section we present how to compute small ε-approximation from low discrepancy colorings. In particular we will examine the following algorithm that computes an ε-approximation of a set system $(X,\mathcal{F})$ given a coloring algorithm `algo`. This result is a generalization of ideas presented by Matoušek and Chazelle in [CM96].

Algorithm 7: Iterated Halving Algorithm

Input: $(X, \mathcal{F})$, t, $\texttt{algo} : (Y, \mathcal{G}) \to (\chi : Y \to \{-1, 1\})$
$(X_0, \mathcal{F}_0) = (X, \mathcal{F})$
for $i \leftarrow 0$ to $t - 1$ **do**
  $\chi_i \leftarrow \texttt{algo}(X_i, Y_i)$
  $X_i^+ \leftarrow \{x \in X_i : \chi_i(x) = +1\}$
  $X_i^- \leftarrow \{x \in X_i : \chi_i(x) = -1\}$
  if $|X_i^+| \geq \frac{|X_i|}{2}$ **then**
    $X_{i+1} \leftarrow$ arbitrary subset of X_i^+ of size $\frac{|X_i|}{2}$
  else
    $X_{i+1} \leftarrow$ arbitrary subset of X_i^- of size $\frac{|X_i|}{2}$
  $\mathcal{F}_{i+1} \leftarrow \mathcal{F}_i|_{X_{i+1}}$
return $X_{\log(t)}$

At each iteration, this algorithm computes a coloring using `algo`. Then it selects half of the points that have been colored with the majority color. If the coloring returned by `algo` has small discrepancy, the points with the same colors will almost represent half of the points of X_i for each range. Therefore by applying this procedure iteratively, we ensure that the number of points in each range in the subset we build is close to $\frac{|X|}{2^i}$.

Theorem 3.32. Let $(X, \mathcal{F})$ be a set system with $X \in \mathcal{F}$ and suppose `algo` is an algorithm that produces a coloring $\chi : X \to \{-1, 1\}$ s.t.

$$\text{disc}_\chi(X, \mathcal{F}) = f(n, m)$$

then for all $t \geq 0$, Algorithm 7 computes an ε-approximation of size $\lceil \frac{n}{2^t} \rceil$ of $(X, \mathcal{F})$ with

$$\varepsilon = \frac{2}{n}\left(f(n, m) + 2f\left(\left\lceil \frac{n}{2} \right\rceil, m\right) + \ldots + 2^{t-1} f\left(\left\lceil \frac{n}{2^{t-1}} \right\rceil, m\right)\right).$$

Proof. For simplicity, we assume that $|X|$ is a power of 2.

Suppose we have finished i iteration of the loop and that $|X_i^+| \geq \frac{|X_i|}{2}$, we have

$$|X_i^+| \leq \frac{|X_i|}{2} + \frac{f(|X_i|, m)}{2}.$$

We also have $X \in \mathcal{F} \Rightarrow X_i \in \mathcal{F}_i$ and therefore $\chi_i(X_i) \leq \text{disc}_{\chi_i}(X_i, \mathcal{F}_i) = f(|X_i|, m)$.

Let X_{i+1} be an arbitrary subset of X_i^+ of size $\frac{|X_i|}{2}$, then for all $F \in \mathcal{F}$ we also have

$$|F \cap X_i^+| \leq \frac{|F \cap X_i|}{2} + \frac{f(|X_i|, m)}{2}$$

as all elements of X_i^+ are colored $+1$.

Therefore by applying this iteratively, after t iterations, Algorithm 7 returns X_t of size $\frac{n}{2^t}$ with,

$$\forall F \in \mathcal{F}, |F \cap X_t| \leq \frac{|F|}{2^t} + \sum_{k=0}^{t-1} \frac{f\left(\frac{n}{2^k}\right)}{2^{t-1-k}} = \frac{|F\|X_t|}{n} + \left(\frac{2}{n}\sum_{k=0}^{t-1} 2^k f\left(\frac{n}{2^k}\right)\right).$$

The lower bound follows from similar calculations and gives:

$$\forall F \in \mathcal{F}, |F \cap X_t| \geq \frac{|F\|X_t|}{n} - \left(\frac{2}{n}\sum_{k=0}^{t-1} 2^k f\left(\frac{n}{2^k}\right)\right)$$

which finishes the proof. □

Chapter 4

A discrepancy learning game

In this chapter, we will present a two-player game on discrepancy. We show that there exists an almost optimal strategy to this game. Finally, we also show that this game can be used in a MWU algorithm to compute a family of low-average discrepancy colorings.

4.1 Discrepancy Learning Game

`Discrepancy Learning Game` is a two-player game where the two players, Alice and Bob, compete in T rounds. Bob chooses a set system $(X, \mathcal{F})$ with finite VC-dimension d and discloses the ground set to Alice.

At each round t,

- Alice chooses a coloring χ_t of X and discloses it to Bob.
- After that, Bob chooses a range $F_t \in \mathcal{F}$ and sends it to Alice.

The goal of Alice is to minimize $\sum_{t=1}^{T} |\chi_t(F_t)|$ and for Bob to maximize the same expression. Alice and Bob are allowed to select, respectively, χ_t and F_t in a stochastic manner.

This game is of independent interest. We also present an application of the game to an algorithm to compute a family of colorings with low average discrepancy.

> **Remark 4.1.** This game is not trivial as choosing random colorings is not a good strategy for Alice. Since Bob chooses the range knowing the coloring chosen by Alice, he can simply choose the range with the largest discrepancy. This guarantees Bob a game value of at least $\Omega\left(T\sqrt{n\ln(m)}\right)$ as the discrepancy of random coloring is tight[17].

In the next section, we present a simple algorithm that uses the discrepancy of ranges w.r.t. a coloring computed by Algorithm 5 to improve itself. This algorithm shows that using Algorithm 5 to infer the ranges to use as constraints is a valid approach. We will exploit this idea in Alice's strategy that we present in Section 4.3.

4.2 Low discrepancy coloring guided by the Lovett-Meka algorithm

Algorithm 8, presented below, uses the discrepancy of ranges w.r.t. a coloring computed by Algorithm 5 to successively improve the choice of constraints to use to compute a low-discrepancy coloring.

[17] See Chapter 1 of [Cha00] for details.

Algorithm 8: Low discrepancy coloring guided by [LM15]

Input: $(X, \mathcal{F})$ with finite VC-dimension d

1 $\chi_0 \leftarrow$ random coloring of X

2 $t \leftarrow 1$

3 **while** true **do**

4 $\quad F_t \leftarrow \text{argmax}_{F \in \mathcal{F}} \ |\chi_{t-1}(F)|$

5 $\quad$ **If** $|\chi_{t-1}(F_t)| \leq 4^{1+\frac{1}{d}} c_{\mathrm{H}}^{\frac{1}{2d}} n^{\frac{1}{2}-\frac{1}{2d}} \log(m) \log(n) \sqrt{\ln(8mn \log(n))}$ **then**[18]

6 $\quad\quad$ **return** χ_{t-1}

7 $\quad \chi_t \leftarrow$ the complete coloring obtained by iteratively running Algorithm 5 on $(X, \{F_1, \ldots, F_t\})$ with constraints $\forall k \leq t, c_k = 0$

8 $\quad t \leftarrow t + 1$

We prove the following theorem.

Lemma 4.2. Given $(X, \mathcal{F})$ of VC-dimension $\leq d$, Algorithm 8 returns a coloring $\chi : X \to \{-1, 1\}$ such that

$$\text{disc}_\chi(X, \mathcal{F}) = O\Big(n^{\frac{1}{2}-\frac{1}{2d}} \log^{\frac{5}{2}}(mn)\Big)$$

with probability at least $\frac{1}{2}$. Algorithm 8's expected time complexity is $\tilde{O}(n^2 m + n^4)$.

Proof. The condition on line 5 guarantees that if Algorithm 8 returns a coloring then the coloring returned satisfies the theorem statement. The only thing to verify is whether Algorithm 8 terminates in a finite number of steps and that the entropy condition of Algorithm 5 are satisfied at all times.

Claim 4.3. At iteration t, $\min_{1 \leq i \leq t-1} |\Delta(F_i, F_t)| > (16 c_{\mathrm{H}})^{\frac{1}{d}} n^{1-\frac{1}{d}}$ with probability at least $1 - \frac{1}{2n}$.

We assume the claim above. By applying the claim's assumption over t iterations, we obtain that $F_1, \ldots, F_t$ forms a $(16c_{\mathrm{H}})^{\frac{1}{d}} n^{1-\frac{1}{d}}$-packing with probability at least $1 - \frac{t}{2n}$. By Haussler's packing lemma, the size of a $(16c_{\mathrm{H}})^{\frac{1}{d}} n^{1-\frac{1}{d}}$-packings can not exceed $\frac{n}{16}$. Therefore at iteration $\frac{n}{16}$, if the algorithm has not already stopped, $F_1, \ldots, F_{\frac{n}{16}}$ is a $(16c_{\mathrm{H}})^{\frac{1}{d}} n^{1-\frac{1}{d}}$-packing with probability at least $\frac{31}{32}$. That is, either the algorithm has already terminated. Or, with probability at least $\frac{31}{32}$, the algorithm terminates as $\left\{F_1, \ldots, F_{\frac{n}{16}}\right\}$ is a maximal $(16c_{\mathrm{H}})^{\frac{1}{d}} n^{1-\frac{1}{d}}$-packing.

Since $\sum_{j=1}^{\frac{n}{16}} \exp(0) \leq \frac{n}{16}$, the entropy condition of Algorithm 8 is satisfied.

Time complexity analysis. If the algorithm succeeds, the number of iterations is bounded by $\frac{n}{16}$ and line 4 has time complexity $O(mn)$ as we need to compute the discrepancy of each range. Line 6 has time complexity $\tilde{O}(n^3)$ as we use Algorithm 5 on a set system with n elements and at most $\frac{n}{16}$ ranges. □

[18] c_{H} denotes the constant in Haussler's packing lemma.

Algorithm 8 constructs a covering using information obtained from the discrepancy of a coloring. This is the opposite of the classical paradigm that builds a covering, then computes a coloring. However, reversing the paradigm comes at a cost as the time complexity of coloring algorithms is large, having to run these algorithms multiple times induces a large time complexity.

We now come back to the proof of Claim 4.3.

Proof of Claim 4.3. Let F be such that there exists $i < t$ with $|\Delta(F, F_i)| \leq (16c_{\mathrm{H}})^{\frac{1}{d}} n^{1-\frac{1}{d}}$. Then since F_i is a constraint of Algorithm 5 for the computation of χ_{t-1},

$$|\chi_{t-1}(F)| \leq \underbrace{|\chi_{t-1}(F_i)|}_{=0} + |\chi_{t-1}(F \setminus F_i)| + |\chi_{t-1}(F_i \setminus F)| = |\chi_{t-1}(F \setminus F_i)| + |\chi_{t-1}(F_i \setminus F)|.$$

By Corollary 3.25 (with $\nu = \frac{1}{8}$), with probability at least $1 - \frac{1}{4mn}$,

$$|\chi_{t-1}(F \setminus F_i)| \leq 4(16c_{\mathrm{H}})^{\frac{1}{2d}} n^{\frac{1}{2}-\frac{1}{2d}} \log(m)\log(n)\sqrt{\ln(8mn\log(n))}.$$

The same holds for $|\chi_{t-1}(F \setminus F_i)|$. That is, the probability for the discrepancy of ranges close to a range used as a constraint to be large is at most $1 - \frac{1}{2mn}$.

By applying the union bound over all ranges, we obtain that, with probability at least $1 - \frac{1}{2n}$, any range closer than $(16c_{\mathrm{H}})^{\frac{1}{d}} n^{1-\frac{1}{d}}$ to at least one of the F_t would not be an exception to the condition of line 5. That is, with probability at least $1 - \frac{1}{2n}$, at iteration t, any range that doesn't meet the condition of line 5 must be at distance at least $(16c_{\mathrm{H}})^{\frac{1}{d}} n^{1-\frac{1}{d}}$ from $F_1, \dots, F_{t-1}$. This finishes the proof of the claim. □

Remark 4.4. It is not possible to reduce the first term of the time complexity of Algorithm 8 by computing the approximate discrepancy by computing it on an ε-approximation instead of on the whole ground set. That is, for $A \subseteq X$, computing the discrepancy on $(A, \mathcal{F}|_A)$.

This would introduce an additive error term in the discrepancy bound equal to $\varepsilon|X|$. To improve the time complexity whilst preserving the current discrepancy bound, we would need the error to be at most of order $O\left(n^{\frac{1}{2}-\frac{1}{2d}} \log^{\frac{5}{2}}(mn)\right)$. That is, we would need $\varepsilon|X| = \Theta\left(n^{\frac{1}{2}-\frac{1}{2d}}\right) \Leftrightarrow \varepsilon = \Theta\left(n^{-\frac{1}{2}-\frac{1}{2d}}\right)$.

To obtain such epsilon with a random sample, which is the only known method that would lead to a time complexity improvement, we would require to sample $\Omega(\varepsilon^{-2}) = \Omega\left(n^{1+\frac{1}{d}}\right)$ elements from X. However $|X| = n$, therefore, approximate discrepancy over a random sample is not possible.

Selecting the range with maximum discrepancy to use as a constraint is a natural approach. A range with large discrepancy is also a range far away from ranges with small discrepancy w.r.t. the symmetric difference. This is due to the fact that the discrepancy of a range A is bounded by the discrepancy of a range B plus the discrepancy of their symmetric difference $\Delta(A, B)$. Furthermore, the discrepancy of a set can not exceed its cardinality. Thus, if the discrepancy of B and the cardinality of the symmetric difference $\Delta(A, B)$ are small, the discrepancy of A will be small.

We build on this idea to present an almost optimal strategy for Alice in the next section.

4.3 An almost optimal stochastic strategy for Alice

In this section we will prove the following theorem.

Theorem 4.5. Let Alice and Bob play a T rounds game of DLG on $(X, \mathcal{F})$ with finite VC-dimension d.

There exists a strategy for Alice such that regardless of Bob's choice of F_t,

$$\frac{1}{T}\sum_{t=1}^{T}|\chi_t(F_t)| = O\Big(\max\Big(T^{-\frac{1}{2d}}\sqrt{n}, n^{\frac{1}{2}-\frac{1}{2d}}\Big)\log^2(n)\sqrt{\ln(T\log(n))}\Big)$$

with probability at least $\frac{1}{2}$.

In particular, for $T = \Omega(n)$, we obtain

$$\frac{1}{T}\sum_{t=1}^{T}|\chi_t(F_t)| = O\Big(n^{\frac{1}{2}-\frac{1}{2d}}\log^{\frac{5}{2}}(Tn)\Big) \text{ w.h.p.}$$

We show that there exists finite VC-dimension set systems where, regardless of Alice's strategy, Bob can make sure that $\frac{1}{T}\sum_{t=1}^{T}|\chi_t(F_t)| = \Omega\Big(n^{\frac{1}{2}-\frac{1}{2d}}\Big)$. That is, this bound is optimal in the worst case up to a polylog(mn) factor.

Proof of Theorem 4.5. Let $t \le T$ and denote $k \in \mathbb{N}$ such that $\frac{kn}{16} < t \le \frac{(k+1)n}{16}$. To simplify the notations, we assume that $\frac{n}{16}$ is an integer. Alice's strategy will be to choose the coloring χ_t to be the coloring returned by Algorithm 5 with constraints $\chi_t(F_i) = 0$ for all $i \in \left[\frac{kn}{16} + 1, t-1\right]$.

The entropy condition of Theorem 3.23 is satisfied at each round of the game as we use less than $\frac{n}{16}$ ranges as constraints.

Alice applies a cyclic strategy. That is, every $\frac{n}{16}$ rounds, Alice acts as if the game just started and forgets the ranges given by Bob in the previous iterations. We now explain what happens during on of these cycles by supposing that $T \le \frac{n}{16}$. We will then generalize this result at the end of the proof.

We will show that a limited number of ranges given by Bob can have a large discrepancy w.r.t. the colorings chosen by Alice. To do that, we will assign the ranges given by Bob to buckets depending on their symmetric difference with the ranges given in previous iterations. Let $r > 0$ and $P_0, \dots, P_r, P_{r+1}$ be empty sets. We sequentially sort the ranges $F_1, \dots, F_T$ given by Bob to one of $P_0, \dots, P_r, P_{r+1}$ by adding F_t to the smallest $j \le r$ such that for all $P \in P_j$, $|\Delta(F_t, P)| \ge \frac{n}{2^j}$. If F_t can not be added to any $P_j, j \le r$, we add F_t to P_{r+1}.

By construction, at all iterations, P_j is a $\frac{n}{2^j}$-packing for all $j \in [1, r]$. Therefore, Haussler's packing lemma gives that, $|P_j| \le c_{\mathrm{H}} 2^{jd}$.

Denote $j_1, \dots, j_t$ the indices of the buckets to which $F_1, \dots, F_T$ were added. We bound the contribution to the game value that each range chosen by Bob have solely depending on the bucket this range gets assigned to.

Let $t \leq T$, suppose that F_t has been added to the bucket with index j_t. That is, there exists $t' < t$ such that $F_{t'} \in P_{j_t - 1}$ and $|\Delta(F_{t'}, F_t)| \leq \frac{n}{2^{j_t-1}}$. Thus we have:

$$\begin{aligned}|\chi_t(F_t)| &\leq |\chi_t(F_{t'})| + |\chi_t(\Delta(F_{t'}, F_t))| \\ &\leq 0 + 4\sqrt{|\Delta(F_{t'}, F_t)|}\log(T)\log(n)\sqrt{\ln(4T\log(n))} \\ &\leq 4\sqrt{\frac{n}{2^{j_t-1}}}\log(T)\log(n)\sqrt{\ln(4T\log(n))}.\end{aligned}$$

The first summand of the bound is a direct application of Theorem 3.23 as $F_{t'}$ is used as a constraint to compute χ_t. The second comes from applying Corollary 3.25 (with $\nu = \frac{mn}{4T}$ and $m \leq T$) on the symmetric difference set which size is bounded because P_{j_t-1} is a packing. Thus we have,

$$P\left(|\chi_t(F_t)| \geq 4\sqrt{\frac{n}{2^{j-1}}}\log(T)\log(n)\sqrt{\ln(4T\log(n))} \;\middle|\; F_t \in P_{j_t}\right) < \frac{1}{2T}. \tag{3}$$

By using a union bound over all iterations $t \leq T$, we obtain that, with probability at most $\frac{1}{2}$, this property holds for each pair composed of a range and a bucket.

This means that we have both a bound on the discrepancy of the ranges added to a given bucket and a bound on the size of every bucket. To obtain a bound on the game value, we sum up the maximum contribution of ranges depending on which bucket they get sorted to. Note that we bound the size of the last bucket, P_{r+1}, by T.

$$\begin{aligned}\sum_{t=1}^{T}|\chi_t(F_t)| &\leq \sum_{j=0}^{r} c_{\mathrm{H}} 2^{jd} \underbrace{\left(4\sqrt{\frac{n}{2^{j-1}}}\log(T)\log(n)\sqrt{\ln(4T\log(n))}\right)}_{\text{contributions from ranges in } P_j} \\ &\quad \underbrace{+4T\sqrt{\frac{n}{2^r}}\log(T)\log(n)\sqrt{\ln(4T\log(n))}}_{\text{contributions from ranges in } P_{r+1}} \\ &= 4\sqrt{2}c_{\mathrm{H}}\sqrt{n}\log(T)\log(n)\sqrt{\ln(4T\log(n))}\sum_{j=0}^{r}2^{j(d-\frac{1}{2})} \\ &\quad +4T\sqrt{\frac{n}{2^r}}\log(T)\log(n)\sqrt{\ln(4T\log(n))} \\ &\leq c_{\mathrm{H}}\sqrt{n}2^{r(d-\frac{1}{2})+\frac{7}{2}}\log(T)\log(n)\sqrt{\ln(4T\log(n))} \\ &\quad +4T\sqrt{\frac{n}{2^r}}\log(T)\log(n)\sqrt{\ln(4T\log(n))}.\end{aligned} \tag{4}$$

In (4), the first term increases with r whilst the second one decreases. To make these three terms of the same order, we set $r = \frac{\log(T)}{d} \Leftrightarrow 2^{rd} = T$. Then, (4) becomes

$$\begin{aligned}\sum_{t=1}^{T}|\chi_t(F_t)| &\leq 2^{\frac{7}{2}} c_{\mathrm{H}} \sqrt{n} T^{1-\frac{1}{2d}} \log(T)\log(n)\sqrt{\ln(4T\log(n))} \\ &\quad + 4T^{1-\frac{1}{2d}} \sqrt{n}\log(T)\log(n)\sqrt{\ln(4T\log(n))} \\ &= T^{1-\frac{1}{2d}}\sqrt{n}\log(T)\log(n)\sqrt{\ln(4T\log(n))}(2c_{\mathrm{H}}+4).\end{aligned}$$

That is,

$$\frac{1}{T}\sum_{t=1}^{T}|\chi_t(F_t)| \leq T^{-\frac{1}{2d}}\sqrt{n}\log(T)\log(n)\sqrt{\ln(4T\log(n))}(2c_{\mathrm{H}}+4).$$

Therefore, for $T = \frac{n}{16}$, we obtain:

$$\frac{1}{T}\sum_{t=1}^{T}|\chi_t(F_t)| = O\Big(n^{\frac{1}{2}-\frac{1}{2d}}\log^{\frac{5}{2}}(mn)\Big).$$

Analysis of cycling. Now suppose that there exists $k \in \mathbb{N}^*$ s.t. $\frac{kn}{16} < T \leq \frac{(k+1)n}{16}$. By applying the same strategy every $\frac{n}{16}$ rounds, we obtain

$$\begin{aligned}\frac{1}{T}\sum_{t=1}^{T}|\chi_t(F_t)| &\leq \frac{1}{T}\sum_{i=0}^{k-1}\sum_{t=\frac{in}{16}+1}^{\frac{(i+1)n}{16}}|\chi_t(F_t)| + \frac{1}{T}\sum_{t=\frac{kn}{16}+1}^{T}|\chi_t(F_t)| \\ &\leq \frac{1}{T}\sum_{i=0}^{k}\left(\frac{n}{16}\right)^{1-\frac{1}{2d}}\sqrt{n}\log\left(\frac{n}{16}\right)\log(n)\sqrt{\ln(4T\log(n))}(2c_{\mathrm{H}}+4) \\ &\quad + \frac{1}{T}\left(T-\frac{kn}{16}\right)^{1-\frac{1}{2d}}\sqrt{n}\log\left(T-\frac{kn}{16}+1\right)\log(n)\sqrt{\ln(4T\log(n))}(2c_{\mathrm{H}}+4) \\ &\leq \frac{16}{kn}\sum_{i=0}^{k}\left(\frac{n}{16}\right)^{1-\frac{1}{2d}}\sqrt{n}\log^2(n)\sqrt{\ln(4T\log(n))}(2c_{\mathrm{H}}+4) \\ &\quad + \frac{16}{kn}\left(T-\frac{kn}{16}\right)^{1-\frac{1}{2d}}\sqrt{n}\log^2(n)\sqrt{\ln(4T\log(n))}(2c_{\mathrm{H}}+4) \\ &\leq 16^{\frac{1}{2d}}n^{-\frac{1}{2d}}\sqrt{n}\log^2(n)\sqrt{\ln(4T\log(n))}(2c_{\mathrm{H}}+4) \\ &\quad + \frac{16}{kn}\left(\frac{n}{16}\right)^{1-\frac{1}{2d}}\sqrt{n}\log^2(n)\sqrt{\ln(4T\log(n))}(2c_{\mathrm{H}}+4) \\ &\leq 16^{\frac{1}{2d}}n^{\frac{1}{2}-\frac{1}{2d}}\log^2(n)\sqrt{\ln(4T\log(n))}(2c_{\mathrm{H}}+4) \\ &\quad + \frac{16^{\frac{1}{2d}}}{k}n^{\frac{1}{2}-\frac{1}{2d}}\log^2(n)\sqrt{\ln(4T\log(n))}(2c_{\mathrm{H}}+4)\end{aligned}$$

The second line is obtained by applying the same proof that we use for all cycles of $\frac{n}{16}$ iterations. However, to obtain this result with probability at least $\frac{1}{2}$, the union bound of (3) has to be applied over all iterations of all cycles.

Therefore, we also obtain that:

$$\frac{1}{T}\sum_{t=1}^{T}|\chi_t(F_t)| = O\left(n^{\frac{1}{2}-\frac{1}{2d}}\log^{\frac{5}{2}}(Tn)\right).$$

Optimality. [Ale90] showed that there exists set systems with finite VC-dimension where any coloring has discrepancy $\Omega\left(n^{\frac{1}{2}-\frac{1}{2d}}\right)$. Thus, if the game is played on such set system, by choosing the range with maximum discrepancy at each iteration, Bob can ensure that, regardless of Alice's strategy, $\frac{1}{T}\sum_{t=1}^{T}|\chi_t(F_t)| = \Omega\left(n^{\frac{1}{2}-\frac{1}{2d}}\right)$. □

4.4 MWU Algorithm

We now present an algorithm to construct a family of low average discrepancy colorings using MWU. We use the result we proved on DLG in the previous section to bound the discrepancy of the family of colorings.

Algorithm 9: MWU Algorithm guided by [LM15]

Input: $(X, \mathcal{F})$

$\forall F \in \mathcal{F}, y_1(F) \leftarrow 1$

for $t \leftarrow 1$ to $\frac{n}{16}$ **do**

  $\chi_t \leftarrow$ coloring obtained with Algorithm 5 on $(X, \{F_1, \ldots, F_{t-1}\})$ with constraints $c_1, \ldots, c_{t-1} = 0$

  $F_t \leftarrow$ randomly sampled range according to y_t

  $\forall F \in \mathcal{F}, y_{t+1}(F) = y_t(F)(1 + \eta\ |\chi_t(F)|)$ where $\eta = \frac{\sqrt{\ln(m)}}{n\sqrt{\left(\ln\left(mn^{\frac{\log(n)}{4}}\right)\right)\log(m)\log(n)}}$

return $\chi_1, \ldots, \chi_{\frac{n}{16}}$

Algorithm 9 computes $\frac{n}{16}$ colorings that we will show have small average discrepancy. The algorithm emulates an $\frac{n}{16}$-rounds game of DLG. That is, colorings are picked according to the strategy we proved to be good for Alice and we emulate Bob's choice of range with the multiplicative weight update. We maintain weights on ranges that will increase with the discrepancy that a range has w.r.t the colorings we construct. This will ensure that range that have large discrepancy w.r.t. the colorings have higher probability to be picked.

In the proof, we will show that the average discrepancy of the colorings returned can be bounded by the value of DLG and an error term that we will control.

We will prove the following statement.

Application 4.6. (of Theorem 4.5) Given $(X, \mathcal{F})$ a set system with finite VC-dimension d, Algorithm 9 returns $\frac{n}{16}$ colorings $\chi_1, \ldots, \chi_{\frac{n}{16}}$ such that:

$$\forall F \in \mathcal{F}, \frac{1}{T} \sum_{t=1}^{\frac{n}{16}} |\chi(F)| = O\left(n^{\frac{1}{2} - \frac{1}{2d}} \log^{\frac{5}{2}}(n)\right)$$

where for all $t \in \left[1, \frac{n}{16}\right]$, the coloring $x^{(t)}$ is computed using only one additional range of $(X, \mathcal{F})$ that was not used to compute $x^{(t-1)}$.

The algorithm succeeds with probability at least $\frac{1}{4}$ in expected time $\tilde{O}(n^4 + n^2 m)$.

Remark 4.7. This application of DLG does not directly improve the state of the art in terms of discrepancy algorithms. As presented in Section 3.4 and Chapter 6, it is possible to obtain algorithm achieving a better discrepancy bound with a single coloring. The goal of this section is to show a simple application of the game.

Proof of Application 4.6. Let $T < \frac{n}{16}$, we first compute a bound on $Y_{T+1} := \sum_{F \in \mathcal{F}} y_{T+1}(F)$.

$$\begin{aligned} Y_{T+1} &= \sum_{F \in \mathcal{F}} y_{T+1}(F) \\ &= \sum_{F \in \mathcal{F}} y_T(F)(1 + \eta |\chi_T(F)|) \\ &= Y_T + \sum_{F \in \mathcal{F}} \eta y_T(F) |\chi_T(F)| \\ &= Y_T + Y_T \sum_{F \in \mathcal{F}} \eta \frac{y_T(F)}{Y_T} |\chi_T(F)| \\ &= Y_T \left(1 + \sum_{F \in \mathcal{F}} \eta \frac{y_T(F)}{Y_T} |\chi_T(F)|\right). \end{aligned}$$

Applying these calculations recursively, we obtain:

$$Y_{T+1} = Y_1 \prod_{t=1}^{T} \left(1 + \sum_{F \in \mathcal{F}} \eta \frac{y_t(F)}{Y_t} |\chi_t(F)|\right).$$

Following the same induction calculation, we obtain that for all $F, T < \frac{n}{16}$, we have $y_t(F) = \prod_{t=1}^{T} (1 + \eta |\chi_t(F)|)$. Any weight of a range is naturally smaller than the sum of all weights, that is for all $F \in \mathcal{F}, T \le \frac{n}{16}$:

$$
\begin{aligned}
& y_T(F) \leq Y_T \\
\Leftrightarrow & \prod_{t=1}^{T}(1+\eta|\chi_t(F)|) \leq Y_1 \prod_{t=1}^{T}\left(1+\sum_{F'\in\mathcal{F}} \eta \frac{y_t(F')}{Y_t}\, |\chi_t(F')|\right) \\
\Leftrightarrow \ln & \left(\prod_{t=1}^{T}(1+\eta|\chi_t(F)|)\right) \leq \ln(Y_1) + \ln\left(\prod_{t=1}^{T}\left(1+\sum_{F'\in\mathcal{F}} \eta \frac{y_t(F')}{Y_t}\, |\chi_t(F')|\right)\right) \\
\Leftrightarrow & \sum_{t=1}^{T} \ln(1+\eta|\chi_t(F)|) \leq \ln(Y_1) + \sum_{t=1}^{T} \ln\left(1+\sum_{F'\in\mathcal{F}} \eta \frac{y_t(F)}{Y_t}\, |\chi_t(F')|\right).
\end{aligned}
$$

We use the inequality for all $x \leq \frac{1}{2}, \ln(1+x) \geq x - x^2$ on the l.h.s and $1+x \leq \exp(x)$ on the r.h.s.

$$
\begin{aligned}
\Leftrightarrow & \sum_{t=1}^{T}(\eta|\chi_t(F)| - \eta^2|\chi_t(F)|^2) \leq \ln(Y_1) + \sum_{t=1}^{T} \ln\left(\exp\left(\sum_{F'\in\mathcal{F}} \eta \frac{y_t(F')}{Y_t}\, |\chi_t(F')|\right)\right) \\
\Leftrightarrow & \sum_{t=1}^{T} \eta|\chi_t(F)| - \sum_{t=1}^{T} \eta^2|\chi_t(F)|^2 \leq \ln(Y_1) + \sum_{t=1}^{T}\sum_{F'\in\mathcal{F}} \eta \frac{y_t(F')}{Y_t}\, |\chi_t(F')| \\
\Leftrightarrow & \sum_{t=1}^{T}|\chi_t(F)| \leq \frac{\ln(Y_1)}{\eta} + \sum_{t=1}^{T} \eta|\chi_t(F)|^2 + \sum_{t=1}^{T}\sum_{F'\in\mathcal{F}} \frac{y_t(F')}{Y_t}\, |\chi_t(F')|.
\end{aligned}
$$

Substituting the initial weight $Y_1 = m$, we obtain

$$
\sum_{t=1}^{T}|\chi_t(F)| \leq \frac{\ln(m)}{\eta} + \sum_{t=1}^{T} \eta|\chi_t(F)|^2 + \sum_{t=1}^{T}\sum_{F'\in\mathcal{F}} \frac{y_t(F')}{Y_t}\, |\chi_t(F')|. \tag{5}
$$

By Corollary 3.25 ($\nu = 4$), we obtain that for fixed F, t, we have

$$
|\chi_t(F)| \leq 4\sqrt{n \ln\left(mn\frac{\log(n)}{4}\right)} \log(m)\log(n)
$$

with probability at least $1 - \frac{8}{mn}$.

That is, using a union bound on the m ranges at each of the $\frac{n}{16}$ iterations of the loop, we obtain that

$$
\forall F, t, |\chi_t(F)| \leq 4\sqrt{n \ln\left(mn\frac{\log(n)}{4}\right)} \log(m)\log(n)
$$

with probability at least $\frac{1}{2}$.

(5) becomes:

$$
\sum_{t=1}^{T}|\chi_t(F)| \leq \frac{\ln(m)}{\eta} + 16T\eta n \ln\left(mn\frac{\log(n)}{4}\right)\log^2(m)\log^2(n) + \sum_{t=1}^{T}\sum_{F'\in\mathcal{F}} \frac{y_t(F')}{Y_t}\, |\chi_t(F')|. \tag{6}
$$

In particular for $T = \frac{n}{16}$, by substituting $\eta = \frac{\sqrt{\ln(m)}}{n\sqrt{\ln\left(mn\frac{\log(n)}{4}\right)}\log(m)\log(n)}$, (6) becomes

$$\sum_{t=1}^{\frac{n}{16}} |\chi_t(F)| \leq 4n\sqrt{\ln(m)\ln\left(mn\frac{\log(n)}{4}\right)} \log(n)\log(m) + \sum_{t=1}^{\frac{n}{16}} \sum_{F' \in \mathcal{F}} \frac{y_t(F)}{Y_t} |\chi_t(F')|. \quad (7)$$

We now bound the second term of the right hand side of (7). For a fixed t, this term's maximum is attained when $y_t(F') = 1$ for $F' = \text{argmax}_{G \in \mathcal{F}} |\chi_t(G)|$ and 0 for all other weights. This means that:

$$\sum_{t=1}^{\frac{n}{16}} \sum_{F' \in \mathcal{F}} \frac{y_t(F')}{Y_t} |\chi_t(F')| \leq \sum_{t=1}^{\frac{n}{16}} \max_{F' \in \mathcal{F}} |\chi_t(F')|.$$

This is a particular case of `DLG` where Bob's strategy is to select each round the range with maximum discrepancy w.r.t. the coloring chosen by Alice. Using Theorem 4.5, we can bound this expression:

$$\sum_{t=1}^{\frac{n}{16}} \sum_{F' \in \mathcal{F}} \frac{y_t(F')}{Y_t} |\chi_t(F')| = O\left(n^{1-\frac{1}{2d}} \sqrt{n} \log^{\frac{5}{2}}(n)\right)$$

as $T = \frac{n}{16}$. Using this result in (7) gives the bound of the theorem.

Time complexity analysis. Algorithm 5 is ran on set systems with n elements and at most $\frac{n}{16}$ ranges which has time complexity $\tilde{O}(n^3)$. The weight update step on line 5 has time complexity $O(mn)$ giving the time complexity of the theorem statement. □

4.5 Improvements of the MWU Algorithm using sampling

As shown in the proof of Application 4.6, the error term from the classical MWU method is negligible compared to the value of `DLG`. In this section we exploit this result to present time complexity improvements to Algorithm 9 that can be obtained by performing the weight update step only on a random sample of $\mathcal{F}$. This will increase the value of the error term. That is why we will choose the size of the random sample to keep this increase under control. Our goal is for the order of the error term to match the order of the value of `DLG`. The proof uses the analysis of the MWU algorithm for low-crossing matchings from Csikós and Mustafa [CM21].

Algorithm 10: MWU Algorithm guided by [LM15] with sampling

Input: $(X, \mathcal{F})$

$\forall F \in \mathcal{F}, y_1(F) \leftarrow 1$

for $t \leftarrow 1$ to $\frac{n}{16}$ **do**

  $\chi_t \leftarrow$ coloring obtained with Algorithm 5 on $(X, \{F_1, \ldots, F_{t-1}\})$ with constraints $c_1, \ldots, c_t = 0$

  $F_t \leftarrow$ randomly sampled range according to y_t

  $\mathcal{S}_t \leftarrow$ sample of $\mathcal{F}$ where each range is sampled with probability $\boldsymbol{q} = \min\left(1, \frac{\sqrt{\ln(m)}}{n^{\frac{1}{2}-\frac{1}{2d}}}\right)$

  $\forall F \in \mathcal{S}_t, y_t(F) = y_{t-1}(F)\left(1 + \frac{\eta\,|\chi_t(F)|}{D}\right)$ where $\eta = \sqrt{\frac{8\ln(m)}{n}}$ and $D = 4\sqrt{n\ln\left(mn^{\frac{\log(n)}{4}}\right)}\log(m)\log(n)$

return $\chi_1, \ldots, \chi_{\frac{n}{16}}$

This algorithm is very similar to Algorithm 9, as the only differences are that the set of ranges which weight is updated at each iteration of the loop and that the weight update function is normalized. The normalization step is important in order to control the error factor. Our analysis will broadly follow the same steps as the one of Application 4.6 with some additional probability considerations to handle the error term due to the weight update sampling. In particular the proof will use the following lemma from Koufogiannakis and Young.

Lemma 4.8. ([KY14], lemma 10) Let $X = \sum_{i=1}^{T} x_i$ and $Y = \sum_{i=1}^{T} y_i$ be the sum of non-negative random variables where T is a random stopping time with finite expectation, and, for all, $|x_i - y_i| < 1$ and

$$\mathbb{E}\left[x_i - y_i \mid \sum_{s<i} x_s, \sum_{s<i} y_s\right] \leq 0$$

Let $\varepsilon \in [0, 1]$ and $A \in \mathbb{R}$, then

$$P((1-\varepsilon)X \geq Y + A) \leq \exp(-\varepsilon A)$$

We will prove the following statement.

Application 4.9. (of Theorem 4.5) Let $(X, \mathcal{F})$ be a set system with finite VC-dimension d, Algorithm 10 returns $\frac{n}{16}$ colorings $\chi_1, \ldots, \chi_{\frac{n}{16}}$ such that:

$$\forall F \in \mathcal{F}, \frac{1}{T}\mathbb{E}\left[\sum_{t=1}^{\frac{n}{16}} |\chi(F)|\right] = O\left(n^{\frac{1}{2}-\frac{1}{2d}} \log^{\frac{5}{2}}(n)\right)$$

where for all $t \in \left[1, \frac{n}{16}\right]$, the coloring $x^{(t)}$ is computed using only one additional range of $(X, \mathcal{F})$ that was not used to compute $x^{(t-1)}$.

The algorithm succeeds with probability at least $\frac{1}{4}$ in expected time $\tilde{O}\left(n^4 + mn^{\frac{3}{2}+\frac{1}{2d}}\right)$.

Proof. Let $T < \frac{n}{16}$, we first compute a bound on $Y_{T+1} := \sum_{F\in\mathcal{F}} y_{T+1}(F)$.

$$\begin{aligned}
Y_{T+1} &= \sum_{F\in\mathcal{F}} y_{T+1}(F) \\
&= \sum_{F\in\mathcal{F}} y_T(F)\left(1+\frac{\eta|\chi_T(F)|}{D}\mathbb{1}_{F\in\mathcal{S}_t}\right) \\
&= Y_T + \sum_{F\in\mathcal{F}} \frac{\eta y_T(F)|\chi_T(F)|}{D}\mathbb{1}_{F\in\mathcal{S}_t} \\
&= Y_T + Y_T\sum_{F\in\mathcal{F}} \eta\frac{y_T(F)}{Y_T}\frac{|\chi_T(F)|}{D}\mathbb{1}_{F\in\mathcal{S}_t} \\
&= Y_T + Y_T\sum_{F\in\mathcal{F}} \eta\frac{y_T(F)}{Y_T}\frac{|\chi_T(F)|}{D}\mathbb{1}_{F\in\mathcal{S}_t} \\
&= Y_T\left(1+\sum_{F\in\mathcal{F}} \eta\frac{y_T(F)}{Y_T}\frac{|\chi_T(F)|}{D}\mathbb{1}_{F\in\mathcal{S}_t}\right).
\end{aligned}$$

Applying this calculations recursively, we obtain:

$$Y_{T+1} = Y_1\prod_{t=1}^{T}\left(1+\sum_{F\in\mathcal{F}} \eta\frac{y_t(F)}{Y_t}\frac{|\chi_t(F)|}{D}\mathbb{1}_{F\in\mathcal{S}_t}\right).$$

Following the same induction calculation, we obtain that for all $F, T < \frac{n}{16}$, we have $y_t(F) = \prod_{t=1}^{T}\left(1+\frac{\eta|\chi_t(F)|}{D}\mathbb{1}_{F\in\mathcal{S}_t}\right)$. Any weight of a range is naturally smaller than the sum of weights, that is for all $F\in\mathcal{F}, T\leq\frac{n}{16}$:

$$\begin{aligned}
& y_T(F) \leq Y_T \\
\Leftrightarrow & \prod_{t=1}^{T}\left(1+\frac{\eta\,|\chi_t(F)|}{D}\mathbb{1}_{F\in\mathcal{S}_t}\right) \leq Y_0\prod_{t=1}^{T}\left(1+\sum_{F'\in\mathcal{F}}\eta\frac{y_t(F')}{Y_t}\frac{|\chi_t(F')|}{D}\mathbb{1}_{F'\in\mathcal{S}_t}\right) \\
\Leftrightarrow & \ln\left(\prod_{t=1}^{T}\left(1+\frac{\eta|\chi_t(F)|}{D}\mathbb{1}_{F\in\mathcal{S}_t}\right)\right) \leq \ln(Y_0) + \ln\left(\prod_{t=1}^{T}\left(1+\sum_{F'\in\mathcal{F}}\eta\frac{y_t(F')}{Y_t}\frac{|\chi_t(F')|}{D}\mathbb{1}_{F'\in\mathcal{S}_t}\right)\right) \\
\Leftrightarrow & \sum_{t=1}^{T}\ln\left(1+\frac{\eta|\chi_t(F)|}{D}\mathbb{1}_{F\in\mathcal{S}_t}\right) \leq \ln(Y_0) + \sum_{t=1}^{T}\ln\left(1+\sum_{F'\in\mathcal{F}}\eta\frac{y_t(F)}{Y_t}\frac{|\chi_t(F')|}{D}\mathbb{1}_{F'\in\mathcal{S}_t}\right).
\end{aligned}$$

We use the inequality for all $x\leq\frac{1}{2}, \ln(1+x)\geq x-x^2$ on the l.h.s and $1+x\leq\exp(x)$ on the r.h.s.

$$\begin{aligned}
\Leftrightarrow & \sum_{t=1}^{T}\left(\frac{\eta|\chi_t(F)|}{D}\mathbb{1}_{F\in\mathcal{S}_t} - \frac{\eta^2|\chi_t(F)|^2}{D^2}\mathbb{1}_{F\in\mathcal{S}_t}\right) \leq \ln(Y_0) + \sum_{t=1}^{T}\ln\left(\exp\left(\sum_{F'\in\mathcal{F}}\eta\frac{y_t(F')}{Y_t}\frac{|\chi_t(F')|}{D}\mathbb{1}_{F'\in\mathcal{S}_t}\right)\right) \\
\Leftrightarrow & \sum_{t=1}^{T}\frac{\eta|\chi_t(F)|}{D}\mathbb{1}_{F\in\mathcal{S}_t} - \sum_{t=1}^{T}\frac{\eta^2|\chi_t(F)|^2}{D^2}\mathbb{1}_{F\in\mathcal{S}_t} \leq \ln(Y_0) + \sum_{t=1}^{T}\sum_{F'\in\mathcal{F}}\eta\frac{y_t(F')}{Y_t}\frac{|\chi_t(F')|}{D}\mathbb{1}_{F'\in\mathcal{S}_t} \\
\Leftrightarrow & \sum_{t=1}^{T}\frac{|\chi_t(F)|}{D}\mathbb{1}_{F\in\mathcal{S}_t} \leq \frac{\ln(Y_0)}{\eta} + \sum_{t=1}^{T}\frac{\eta|\chi_t(F)|^2}{D^2}\mathbb{1}_{F\in\mathcal{S}_t} + \sum_{t=1}^{T}\sum_{F'\in\mathcal{F}}\frac{y_t(F')}{Y_t}\frac{|\chi_t(F')|}{D}\mathbb{1}_{F'\in\mathcal{S}_t}.
\end{aligned}$$

As in the proof of Application 4.6, using Lemma 3.24 we obtain that for fixed F, t,

$$|\chi_t(F)| \leq 4\sqrt{n \ln\left(mn\frac{\log(n)}{4}\right)} \log(m)\log(n)$$

with probability at least $1 - \frac{8}{mn}$.

That is, using a union bound on the m ranges at each of the $\frac{n}{16}$ iterations of the loop, we obtain that

$$\forall F, t, |\chi_t(F)| \leq 4\sqrt{n \ln\left(mn\frac{\log(n)}{4}\right)} \log(m)\log(n)$$

with probability at least $\frac{1}{2}$.

We will again use this result to bound the square term of the error and we simply bound $\mathbb{1}_{F\in\mathcal{S}_t}$ by 1. We also substitute the initial weight Y_1 by m. The previous equation becomes:

$$\sum_{t=1}^{T} \frac{|\chi_t(F)|}{D}\mathbb{1}_{F\in\mathcal{S}_t} \leq \frac{\ln(m)}{\eta} + 2\eta T + \sum_{t=1}^{T}\sum_{F'\in\mathcal{F}} \frac{y_t(F')}{Y_t}\frac{|\chi_t(F')|}{D}\mathbb{1}_{F'\in\mathcal{S}_t}.$$

This equation is true for all ranges, that is, in particular, it is true for the range that maximizes the l.h.s.

$$\max_{F\in\mathcal{F}}\sum_{t=1}^{T} \frac{|\chi_t(F)|}{D}\mathbb{1}_{F\in\mathcal{S}_t} \leq \frac{\ln(m)}{\eta} + 2\eta T + \sum_{t=1}^{T}\sum_{F'\in\mathcal{F}} \frac{y_t(F')}{Y_t}\frac{|\chi_t(F')|}{D}\mathbb{1}_{F'\in\mathcal{S}_t}.$$

If $\boldsymbol{q} = 1$ then $\mathbb{1}_{F\in\mathcal{S}_t} = 1$, thus taking the total expectation gives:

$$\mathbb{E}\left[\max_{F\in\mathcal{F}}\sum_{t=1}^{T} \frac{|\chi_t(F)|}{D}\mathbb{1}_{F\in\mathcal{S}_t}\right] \leq \frac{\ln(m)}{\eta} + 2\eta T + \mathbb{E}\left[\sum_{t=1}^{T}\sum_{F'\in\mathcal{F}} \frac{y_t(F')}{Y_t}\frac{|\chi_t(F')|}{D}\mathbb{1}_{F'\in\mathcal{S}_t}\right].$$

If $\boldsymbol{q} < 1$ then, using the inequality for all functions (f, g), $\max f(x) - \max g(x) \leq \max(f(x) - g(x))$, we obtain:

$$\frac{3}{4}\max_{F\in\mathcal{F}}\sum_{t=1}^{T} \frac{|\chi_t(F)|}{D}\boldsymbol{q} \leq \frac{\ln(m)}{\eta} + 2\eta T + \max_{F\in\mathcal{F}}\sum_{t=1}^{T} \frac{|\chi_t(F)|}{D}\left(\frac{3\boldsymbol{q}}{4} - \mathbb{1}_{F\in\mathcal{S}_t}\right)$$
$$+\sum_{t=1}^{T}\sum_{F'\in\mathcal{F}} \frac{y_t(F')}{Y_t}\frac{|\chi_t(F')|}{D}\mathbb{1}_{F'\in\mathcal{S}_t} \quad (8)$$

with $f(x) = \max_{F\in\mathcal{F}}\sum_{t=1}^{T} \frac{|\chi_t(F)|}{D}\frac{3\boldsymbol{q}}{4}$ and $g(x) = \max_{F\in\mathcal{F}}\sum_{t=1}^{T} \frac{|\chi_t(F)|}{D}\mathbb{1}_{F\in\mathcal{S}_t}$.

Taking the expectation, (8) becomes:

$$\frac{3}{4}\mathbb{E}\left[\max_{F\in\mathcal{F}}\sum_{t=1}^{T}\frac{|\chi_t(F)|}{D}\boldsymbol{q}\right] \leq \frac{\ln(m)}{\eta} + 2\eta T + \mathbb{E}\left[\max_{F\in\mathcal{F}}\sum_{t=1}^{T}\frac{|\chi_t(F)|}{D}\left(\frac{3\boldsymbol{q}}{4} - \mathbb{1}_{F\in\mathcal{S}_t}\right)\right] + \mathbb{E}\left[\sum_{t=1}^{T}\sum_{F'\in\mathcal{F}}\frac{y_t(F')}{Y_t}\frac{|\chi_t(F')|}{D}\mathbb{1}_{F'\in\mathcal{S}_t}\right]. \quad (9)$$

Since $\mathcal{S}_t$ and F_t are independent, (9) becomes:

$$\frac{3}{4}\mathbb{E}\left[\max_{F\in\mathcal{F}}\sum_{t=1}^{T}\frac{|\chi_t(F)|}{D}\boldsymbol{q}\right] \leq \frac{\ln(m)}{\eta} + 2\eta T + \mathbb{E}\left[\max_{F\in\mathcal{F}}\sum_{t=1}^{T}\frac{|\chi_t(F)|}{D}\left(\frac{3\boldsymbol{q}}{4} - \mathbb{1}_{F\in\mathcal{S}_t}\right)\right] + \boldsymbol{q}\sum_{t=1}^{T}\sum_{F'\in\mathcal{F}}\mathbb{E}\left[\frac{y_t(F')}{Y_t}\frac{|\chi_t(F')|}{D}\right]. \quad (10)$$

Lemma 4.8 with $x_t = \frac{|\chi_t(F)|}{D}\frac{3\boldsymbol{q}}{4}, y_t = \frac{|\chi_t(F)|}{D}\mathbb{1}_{F\in\mathcal{S}_t}, \varepsilon = \frac{1}{4}$ and $A = 4\ln(mT)$ gives:

$$P\left(\max_{F\in\mathcal{F}}\sum_{t=1}^{T}\frac{|\chi_t(F)|}{D}\left(\frac{3\boldsymbol{q}}{4} - \mathbb{1}_{F\in\mathcal{S}_t}\right) \geq 4\ln(mT)\right) \leq \frac{1}{T}.$$

Using the total probability formula, we obtain:

$$\mathbb{E}\left[\max_{F\in\mathcal{F}}\sum_{t=1}^{T}\frac{|\chi_t(F)|}{D}\left(\frac{3\boldsymbol{q}}{4} - \mathbb{1}_{F\in\mathcal{S}_t}\right)\right] \leq 4\ln(mT)\left(1-\frac{1}{T}\right) + T\times\frac{1}{T} = 4\ln(mT)+1.$$

Using this inequality, (10) becomes:

$$\frac{3}{4}\mathbb{E}\left[\max_{F\in\mathcal{F}}\sum_{t=1}^{T}\frac{|\chi_t(F)|}{D}\boldsymbol{q}\right] \leq \frac{\ln(m)}{\eta} + 2\eta T + 4\ln(mT) + 1 + \boldsymbol{q}\sum_{t=1}^{T}\sum_{F'\in\mathcal{F}}\mathbb{E}\left[\frac{y_t(F')}{Y_t}\frac{|\chi_t(F')|}{D}\right]$$

$$\Leftrightarrow \mathbb{E}\left[\max_{F\in\mathcal{F}}\sum_{t=1}^{T}|\chi_t(F)|\right] \leq \frac{4D}{3\boldsymbol{q}}\left(\frac{\ln(m)}{\eta} + 2\eta T + 4\ln(mT) + 1\right) + \frac{4}{3}\sum_{t=1}^{T}\sum_{F'\in\mathcal{F}}\mathbb{E}\left[\frac{y_t(F')}{Y_t}|\chi_t(F')|\right]. \quad (11)$$

In particular for $T = \frac{n}{16}$ and substituting the values of η and $\boldsymbol{q}$, (11) becomes:

$$\mathbb{E}\left[\max_{F\in\mathcal{F}}\sum_{t=1}^{\frac{n}{16}}|\chi_t(F)|\right] \leq \Theta\left(n^{\frac{3}{2}-\frac{1}{2d}}\log^{\frac{5}{2}}(mn)\right) + \frac{4}{3}\sum_{t=1}^{T}\sum_{F'\in\mathcal{F}}\mathbb{E}\left[\frac{y_t(F')}{Y_t}|\chi_t(F')|\right].$$

Again with the same arguments as for the proof of Application 4.6, the second term of the r.h.s. of this equation can be bounded by $O\left(n^{1-\frac{1}{2d}}\sqrt{n}\log^{\frac{5}{2}}(n)\right)$ using Theorem 4.5 with probability at least $\frac{1}{2}$.

This gives the bound of the theorem statement.

Time complexity analysis. the time complexity analysis is also very similar to the one of Algorithm 9: Algorithm 5 is ran on set systems with n elements and at most n ranges which has time complexity $\tilde{O}(n^3)$. The weight update step on line 6 has time complexity $O(\boldsymbol{q}mn)$ instead of simply $O(mn)$ giving the time complexity of the theorem statement. $\square$

The following corollary shows that this result implies the existence of a coloring in the family returned that has small discrepancy w.r.t. a majority of the ranges.

Corollary 4.10. Denote B the bound obtained in Application 4.9. There exists $i \in \left[1, \frac{n}{16}\right]$ such that $\left|\left\{F \in \mathcal{F} : \mathbb{E}[|\chi_i(F)|] \leq \frac{32B}{n}\right\}\right| \geq \frac{m}{2}$.

Proof. Application 4.9 gives for all $F \in \mathcal{F}$, $\sum_{t=1}^{\frac{n}{16}} \mathbb{E}[|\chi_t(F)|] \leq B$. Thus we have:

$$\sum_{F \in \mathcal{F}} \sum_{t=1}^{\frac{n}{16}} \mathbb{E}[|\chi_t(F)|] \leq mB$$
$$\Leftrightarrow \sum_{t=1}^{\frac{n}{16}} \sum_{F \in \mathcal{F}} \mathbb{E}[|\chi_t(F)|] \leq mB.$$

By the pigeonhole principle, $\exists i \leq \frac{n}{16}$ s.t. $\sum_{F \in \mathcal{F}} \mathbb{E}[|\chi_i(F)|] \leq \frac{16mB}{n}$.

Let Y be the random variable representing the event $\mathbb{E}[|\chi_t(F)|] \leq B$ a uniformly selected range $F \in \mathcal{F}$. By Markov's inequality:

$$P\left(Y \geq \frac{32B}{n}\right) \leq \frac{E[Y]n}{32B} = \frac{\frac{n}{m} \sum_{F \in \mathcal{F}} \mathbb{E}[|\chi_t(F)|]}{32B} \leq \frac{1}{2}.$$

$\square$

This corollary shows that there exists a coloring $\chi_i \in \left\{\chi_1, ..., \chi_{\frac{n}{16}}\right\}$ where $\chi_1, ..., \chi_{\frac{n}{16}}$ are the colorings returned by Algorithm 10 such that:

$$\left|\left\{F \in \mathcal{F} : \chi_i(F) = O\left(n^{\frac{1}{2} - \frac{1}{2d}} \log^{\frac{5}{2}}(mn)\right)\right\}\right| \geq \frac{m}{2}.$$

Chapter 5

A greedy algorithm for low-crossing partitions for general set systems

In this chapter, we present new algorithms to compute low-crossing partitions. Unlike previous work presented in Section 1.3 and Section 3.5, our algorithm can compute partitions on any set system. We first present a theorem to motivate a greedy approach to the low-crossing partition problem. In fact we show that, with some hereditary assumptions, set systems that admit low-crossing partitions admit an ordering within each of their parts such that the crossing number of a prefix part w.r.t. the ordering is a function of the size of the prefix part.

5.1 The Ordering Theorem

Let $(X, \mathcal{F})$ be a set system for which there exists a partition $\mathcal{P}$ of size t that has a low crossing number with respect to $\mathcal{F}$. Our main insight is that, for each $P_i \in \mathcal{P}$, there exists a permutation of the elements of P_i that can be added in sequence, *iteratively*, such that the crossing number increase for each addition is upper-bounded by a specific function. Thus, following such a sequence of additions results in a set of $\frac{n}{t}$ points that is crossed by few ranges. We refer to this function, derived below, as the *potential function*. The only requirement that we need for the existence of such a good permutation is the following hereditary property: there is a constant $d \geq 1$ such that

$$\forall Y \subseteq X \text{ and all } s \in [|Y|], (Y, \mathcal{F}|_Y) \text{ admits an } \left(s, s^{1-\frac{1}{d}}\right)\text{-partition} \tag{13}$$

where $\mathcal{F}|_Y = \{F \cap Y : F \in \mathcal{F}\}$.

Note that (13) is satisfied for those geometric set systems where partitions of sub-linear crossing numbers are proven to exist (e.g., geometric set systems induced by semialgebraic sets [AMS13]).

Theorem 5.1. Let $(X, \mathcal{F})$ be a set system satisfying (13) and $\mathcal{P} = \{P_1, \dots, P_r\}$ be r disjoint subsets of X, where $|P_i| = \frac{n}{t}$ for all $i \in [r]$. Let $\mathcal{R}$ be a family of subsets of X with crossing number at most κ with respect to $\mathcal{P}$. Let $P_l \in \mathcal{P}$ be any part and let $\mathcal{R}_l$ denote the set of ranges crossing P_l. Then there exists an ordering of the elements of P_l, say $\langle x_1, x_2, \dots, x_{\frac{n}{t}} \rangle$, such that:

$$\forall k \in \left[\frac{n}{t}\right], \text{the prefix set } \{x_1, \dots, x_k\} \text{ is crossed by at most } \frac{4|\mathcal{R}_l| t^{1/d} k^{1/d}}{n^{1/d}} \text{ sets of } \mathcal{R}. \tag{14}$$

Moreover, if P_l is chosen uniformly at random from $\{P_1, \dots, P_r\}$, then with probability at least $\frac{1}{2}$, there exists an ordering $\langle x_1, x_2, \dots, x_{\frac{n}{t}} \rangle$ of the elements of P_l such that

$$\forall k \in \left[\frac{n}{t}\right], \text{the prefix set } \{x_1, \dots, x_k\} \text{ is crossed by at most } \frac{4|\mathcal{R}| \kappa t^{1/d} k^{1/d}}{r n^{1/d}} \text{ sets of } \mathcal{R}. \tag{15}$$

We will use Theorem 5.1 to define a suitable potential function: each partition will be constructed greedily by adding elements to it that satisfy the upper bound in (15). Towards this goal, in Theorem 5.1, we can re-try with several random starting elements, in case the algorithm fails to compute an ordering satisfying the bound of (15), see Section 5.2 for more details.

We remark that the ordering provided by Theorem 5.1 is not trivial, even in simple set systems. For instance consider the set system on the n elements of the integer grid $[0, \sqrt{n}] \times [0, \sqrt{n}]$ in $\mathbb{R}^2$ and ranges defined by half-planes bounded by $\sqrt{n} - 1$ evenly-spaced horizontal lines and $\sqrt{n} - 1$ evenly-spaced vertical lines over the grid. We consider two orderings starting from the origin, in the construction of a single partition; see Figure 8. On the left, the first k elements of the ordering are always contained within a box of side-length $\sqrt{k}$. Thus, the number of ranges crossed by the first k elements is proportional to $\sqrt{k}$, as desired. This demonstrates that the origin is a good starting element of the ordering. On the other hand, the right-side drawing demonstrates a *bad* ordering where the number of lines crossed increases linearly with the number of elements in the prefix of the ordering.

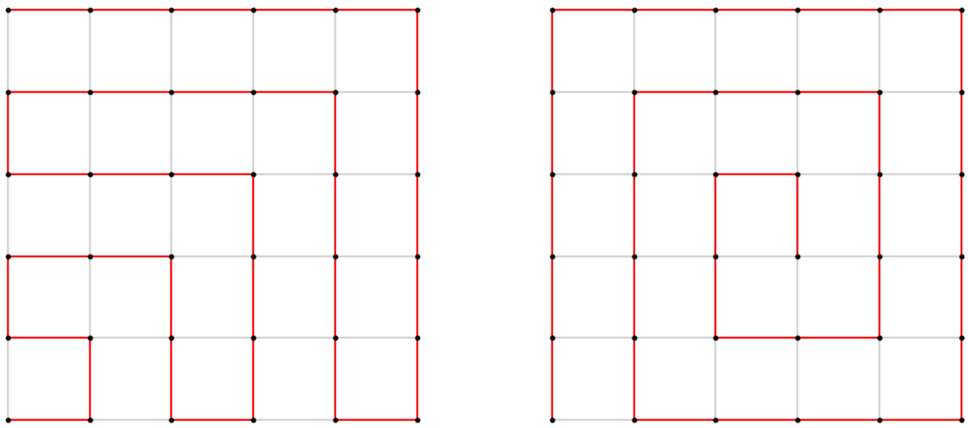


Figure 8: On the left, a *good* ordering with a number of crossings proportional to $\sqrt{k}$. On the right, a *bad* ordering where the crossing number evolves linearly with k, for $1 \le k \le 2\sqrt{k}$.

We return to the proof of Theorem 5.1.

Proof of Theorem 5.1. Set $Q^0 = P_l$ and $\mathcal{S}^0 = \{R \cap Q^0 : R \in \mathcal{R}_l\}$. By applying (13) with $s = 2^d$, $(Q^0, \mathcal{S}^0)$ has a simplicial partition $\mathcal{P}^1$, of size 2^d, with crossing number at most

$$\left(2^d\right)^{1-\frac{1}{d}} = 2^{d-1}.$$

By the pigeonhole principle, there exists a part, say $Q^1 \in \mathcal{P}^1$, that is crossed by at most

$$\frac{|\mathcal{S}^0| \cdot 2^{d-1}}{2^d} = \frac{|\mathcal{S}^0|}{2} \le \frac{|R_l|}{2}$$

ranges of $\mathcal{R}$. Denote the set of ranges that cross Q^1 by $\mathcal{S}^1$. Note that $\frac{n}{2^d t} \le |Q^1| \le \frac{n}{2^{d-1} t}$. Now we repeat the same process with $(Q^1, \mathcal{S}^1)$. That is, at the j-th step, we compute a $\left(2^d, 2^{d-1}\right)$-partition, denoted by $\mathcal{P}^{j+1}$, of $(Q^j, \mathcal{S}^j)$. Then by pigeonhole principle, there exists a set $Q^{j+1} \in \mathcal{P}^{j+1}$ that is crossed by at most

$$\frac{|S^j| \cdot 2^{d-1}}{2^d} = \frac{|S^j|}{2} \leq \frac{|R_l|}{2^{j+1}}$$

ranges of $\mathcal{R}$. Further, we have $\frac{n}{2^{(j+1)d}t} \leq |Q^{j+1}| \leq \frac{n}{2^{(j+1)(d-1)}t}$. We continue as long as $|Q_j| < 2^d$ and denote T, the first index where this is true. This results in a sequence

$$Q^T \subseteq Q^{T-1} \subseteq \ldots \subseteq Q^0 = P_l.$$

Our final ordering, denoted by π, is as follows:

- elements of Q^T (in any order),
- the elements of $Q^{T-1} \setminus Q^T$ (in any order),
- then the elements of $Q^{T-2} \setminus Q^{T-1}$ (in any order),

$\vdots$

- finally, the elements of $Q^0 \setminus Q^1$.

Now fix any $k \in \left[\frac{n}{t}\right]$, and let $j \in [T]$ be the largest index such that $|Q^j| > k$. That is,

$$\frac{n}{2^{(j+1)d}t} \leq |Q_{j+1}| \leq k < |Q_j| \leq \frac{n}{2^{j(d-1)}t}. \tag{17}$$

The first k elements in our ordering π all lie in Q^j, and by our construction, Q^j is crossed by at most $\frac{|R_l|}{2^j}$ sets of $\mathcal{R}$. That is, the set formed by the k first elements in our ordering is crossed by at most these many sets of $\mathcal{R}$:

$$\begin{aligned}\frac{|R_l|}{2^j} &= \frac{2|R_l|}{\left(2^{(j+1)d}\right)^{1/d}} \frac{t^{1/d}n^{1/d}}{t^{1/d}n^{1/d}} \\ &= \frac{2|R_l|t^{1/d}}{n^{1/d}} \left(\frac{n}{2^{(j+1)d}t}\right)^{1/d} \\ &\leq \frac{2|R_l|t^{1/d}}{n^{1/d}} \cdot k^{1/d},\end{aligned}$$

where the last step follows from (17).

Finally if P_l is a part picked uniformly at random, we have

$$\begin{aligned}\mathbb{E}[|\mathcal{R}_l|] &= \sum_{i=1}^{r} \frac{1}{r} |\mathcal{R}_i| \\ &= \frac{1}{r} \sum_{R \in \mathcal{R}} \sum_{i=1}^{r} I(P_i, R) \\ &\leq \frac{1}{r} \sum_{R \in \mathcal{R}} \kappa \\ &= \frac{|\mathcal{R}|\kappa}{r}.\end{aligned}$$

By Markov's inequality, we have

$$\Pr\left[|\mathcal{R}_l| \leq \frac{2|\mathcal{R}|\kappa}{r}\right] \leq \frac{1}{2}, \tag{19}$$

Thus with probability at least $\frac{1}{2}$, the k first elements in our ordering are crossed by at most

$$\frac{2|\mathcal{R}_l|t^{\frac{1}{d}}}{n^{\frac{1}{d}}} \cdot k^{\frac{1}{d}} \leq \frac{4|\mathcal{R}|\kappa t^{\frac{1}{d}} k^{\frac{1}{d}}}{r n^{\frac{1}{d}}}$$

ranges of R. □

5.2 Our Greedy Algorithm Using the Potential Function

Classical methods for building simplicial partitions use the multiplicative weight update (MWU) framework to maintain a weight function on each $F \in \mathcal{F}$ that evolves with the number of parts crossed by F. This is combined with the key step of finding a good set of $\frac{n}{t}$ elements of X (which constitutes the next part) that is crossed by ranges of low total weight in each iteration.

Greedy Potential. In our method, we keep the MWU framework to ensure low crossing number, but take a different approach for constructing the parts, inspired by Theorem 5.1. At iteration i, we sample a random element $x_0 \in X \setminus (P_1 \cup \ldots \cup P_{i-1})$ to be the starting element of P_i. The algorithm proceeds by greedily adding elements to P_i so that the total weight of ranges crossing P_i stays below the potential function bound of Theorem 5.1. To this end, we maintain a function $\omega(\cdot)$ which stores, for each $x \in X \setminus (P_1 \cup \ldots \cup P_i)$, the cost of adding x to P_i. In other words, $\omega(x)$ is the total weight of ranges in

$$\mathcal{C}(P_i, x) := \text{ranges in } \mathcal{F} \text{ that do not cross } \mathcal{P}_i \text{ but cross } P_i \cup \{x\}.$$

Initially, for any x, $\omega(x)$ is equal to the total weight of ranges crossing the edge $\{x_0, x\}$. Note that each time we pick an element x' to be added to P_i, we need to adjust, for each $x \in X \setminus P_i$, its weight $\omega(x)$ by removing the weight of those ranges that are both in $\mathcal{C}(P_i, x)$ and $\mathcal{C}(P_i, x')$.

Formally, at step k of the construction of P_i, with $x_0, \ldots, x_{k-1}$ the elements of X selected in the first $k-1$ steps of the construction of P_i and $\pi(F) = 2^{\sum_{j=1}^{i-1} I(P_j, F)}$.

$$\omega(x) = \sum_{F \in \mathcal{F}} \pi(F) I(\{x_0, x\}, F)(1 - I(\{x_0, \ldots, x_{k-1}\}, F)).$$

The resulting algorithm is presented in Greedy Potential. When set systems admit partitions with sublinear crossing number, d is immediately deduced from the crossing number. Otherwise, it is possible to run the algorithm $\log(n)$ times to search the value of d in $[1, n]$ that gives the best crossing number.

Algorithm 11: GreedyPotential

$n \leftarrow |X|, m \leftarrow |\mathcal{F}|, \mathcal{P} \leftarrow \emptyset$
$\forall F \in \mathcal{F}, \pi(F) \leftarrow 1$
for $i \leftarrow 1$ to t **do**
    $x_0 \leftarrow$ a random element of X
    $P_i \leftarrow \{x_0\}$
    cost $\leftarrow 0$
    for $F \in \mathcal{F}$ **do**
        foreach $x \in X$ with F crossing $\{x_0, x\}$ **do**
            $\omega(x) \leftarrow \omega(x) + \pi(F)$
    for $k \leftarrow 2$ to $\frac{n}{t}$ **do**
        $y_k \leftarrow$ any element of X s.t. cost + $\omega(y_k) \leq \frac{2k^{1/d} \sum_{F \in \mathcal{F}} \pi(F)}{|X|^{1/d}}$
        $X \leftarrow X \setminus \{y_k\}$
        cost $\leftarrow$ cost + $\omega(y_k)$
        foreach $F \in \mathcal{C}(P_i, y_k)$ **do**
            foreach $x \in X$ with F crossing $\{x_0, x\}$ **do**
                $\omega(x) \leftarrow \omega(x) - \pi(F)$
            $P_i \leftarrow P_i \cup \{y_k\}$
    $\forall F \in \mathcal{F}, \pi(F) \leftarrow \pi(F) \cdot 2^{I(P_i, F)}$
    $\mathcal{P} \leftarrow \mathcal{P} \cup \{P_i\}$
    $X \leftarrow X \setminus P_i$
return $\mathcal{P}$

Interestingly, this algorithmic idea is already present in Chan's paper [Cha12], though it is used *between* partitions. Chan's algorithm starts from X and then iteratively *refines* the initial partition with the use of cuttings. The order in which the partitions are refined is via a random permutation. Chan's algorithm is top-down, requiring cuttings to do the refinement from one level to the next. The heuristic we propose constructs the parts bottom-up via the existence of a potential function that guides our greedy algorithm.

The classical proofs proceed by upper-bounding, using cuttings or packing lemmas, the number of ranges crossing the i-th constructed part. This upper-bound is an absolute bound depending only on n, m, d, and i. For general set systems, we do not have access to cuttings or packing lemmas, and we only rely on the fact that the input set system satisfies (13). Thus we take a different strategy in the proof: we derive the upper bound on the crossing number for the i-th part by applying (13) iteratively to the remaining elements. Note that this upper bound depends also on the crossing number of the parts constructed so far. However, the theorem below shows that this additional term adds only a logarithmic factor to the crossing number.

Theorem 5.2. Overall crossing number bound. Let $(X, \mathcal{F})$ be a set system satisfying (13). Assuming Greedy Potential is always able to pick an element satisfying (15), then Greedy Potential constructs a $\left(t, O\left(\ln m + t^{1-\frac{1}{d}} \ln t\right)\right)$-partition w.r.t. $\mathcal{F}$.

Proof. Let X' be a subset of X of size exactly $t\lfloor \frac{n}{t} \rfloor$. Let $\mathcal{P}_0$ be a partition of X' into t equal-sized subsets, with crossing number $t^{1-\frac{1}{d}}$. Applying Theorem 5.1, we pick a random element from X'—which is equivalent to picking a random part of $\mathcal{P}_0$ as they all have the same size —and construct an $\frac{n}{t}$-sized set, say S_1, containing it using a greedy algorithm. Since (15) is satisfied for S_1, the number of ranges of $\mathcal{F}$ crossing S_1 is at most

$$\frac{4|\mathcal{F}|\ t^{1-1/d} t^{1/d} (n/t)^{1/d}}{tn^{1/d}} = \frac{4|\mathcal{F}|}{t^{1/d}}.$$

Next, we construct a new family of multisets $\mathcal{F}_1$ by duplicating the ranges crossing S_1:

$$\mathcal{F}_1 = \mathcal{F} \cup \{F \in \mathcal{F} : F \text{ crosses } S_1\}.$$

By our assumption on $(X, \mathcal{F})$, there exists a partition $\mathcal{P}_1$ of size $t-1$ for $X' \setminus S_1$ with crossing number $(t-1)^{1-1/d}$. Apply Theorem 5.1 and the greedy algorithm to get a set S_2 of $\frac{n}{t}$ elements from $X' \setminus S_1$. Since (15) is again satisfied for S_2, the number of ranges of $\mathcal{F}_1$ crossing S_2 is at most

$$\begin{aligned} &\frac{4|\mathcal{F}_1|\ (t-1)^{1-1/d} t^{1/d} \left(\frac{n}{t}\right)^{1/d}}{(t-1)n^{1/d}} \\ &\le \frac{|\mathcal{F}|\ \left(1 + \frac{4}{t^{1/d}}\right)\left(4t^{1-1/d}\right)}{t-1}. \end{aligned}$$

We again duplicate the sets of $\mathcal{F}_1$ crossing S_2, to get the next multiset $\mathcal{F}_2$.

Continuing on for t steps, we get a partition $\{S_1, ..., S_t\}$ of X', and $|\mathcal{F}_t|$ can be bounded as

$$\begin{aligned} |\mathcal{F}_t| &\le |\mathcal{F}| \prod_{i=1}^{t} \left(1 + \frac{4t^{1-1/d}}{i}\right) \\ &\le m \exp\left(4t^{1-1/d} \sum_{i=1}^{t} \frac{1}{i}\right) \\ &= O\left(m \exp\left(t^{1-1/d} \ln t\right)\right). \end{aligned}$$

On the other hand, a range crossing l parts in $\{S_1, ..., S_t\}$ appears at most 2^l times in $\mathcal{F}_t$, implying that

$$l \le \log(|\mathcal{F}_t|) = O\left(\ln m + t^{1-1/d} \ln t\right).$$

$\{S_1, ..., S_t \cup (X \setminus X')\}$ is a $\left(t, O\left(\ln m + t^{1-1/d} \ln t\right)\right)$-partition of X w.r.t. $\mathcal{F}$ because its crossing number is at most one more than the crossing number of $\{S_1, ..., S_t\}$ (the set with maximum crossing number might intersect $S_t \cup (X \setminus X')$ but not S_t). □

Note that the first set S_1 may contain elements from any mixture of the sets of $\mathcal{P}_0$. This is not a problem: the only property that we require is an upper bound on the total number of sets of $\mathcal{F}$ that cross S_1. The next step, for computing S_2, can take as input an arbitrary partition $\mathcal{P}_1$ on $X \setminus S_1$ with a suitably low crossing number. We do not require any "consistency" between the partitions $\mathcal{P}_0$ and $\mathcal{P}_1$.

The reader may notice that in our experiments we use the potential function

$$\frac{2k^{1/d} \sum_{F \in \mathcal{F}} \pi(F)}{|X|^{1/d}}, \tag{21}$$

which is slightly more restrictive than the one derived from Theorem 5.1. In the proof of Theorem 5.2, we apply successively Theorem 5.1 with decreasing t at each iteration which gives potential function: $i \cdot \frac{2k^{\frac{1}{d}} \sum_{F \in \mathcal{F}} \pi(F)}{|X|^{\frac{1}{d}}}$ for the elements of the i^{th} part. In our experiments, we use a potential function that does not depend on the number of parts already built. That is, the two potential functions are equal while building the first part but the experimental potential function remains constant when the theoretical one increases linearly after each part is built.

We also exclude some implementation details in the pseudo-code of Greedy Potential. For instance, after selecting y_k, we do not immediately update $\omega(x)$ with all ranges in $\mathcal{C}(P_i, y_k)$. We store these ranges in a queue and only update $\omega(x)$ in the next iteration, range by range, until we can find an element within the potential function rate for the next iteration.

The algorithmic bottleneck of Greedy Potential is the weight update operation (i.e. updating $\omega(\cdot)$) which gives an overall time complexity of $O(nmt)$, since it is only performed at most once per partition for each pair $(x, F) \in X \times \mathcal{F}$.

5.3 Variants

Min Weight In this variant of Greedy Potential, instead of selecting an arbitrary element satisfying the upper bound of the potential function, we pick the element with the lowest weight at the time. This variant has the same asymptotic time complexity of $O(nmt)$ as Greedy Potential.

Algorithm 12: MinWeight

$n \leftarrow |X|, m \leftarrow |\mathcal{F}|, \mathcal{P} \leftarrow \emptyset$
$\forall F \in \mathcal{F}, \pi(F) \leftarrow 1$
for $i \leftarrow 1$ to t **do**
    $x_0 \leftarrow$ a random element of X
    $P_i \leftarrow \{x_0\}$
    cost $\leftarrow 0$
    for $F \in \mathcal{F}$ **do**
        foreach $x \in X$ with F crossing $\{x_0, x\}$ **do**
            $\omega(x) \leftarrow \omega(x) + \pi(F)$
    for $k \leftarrow 2$ to $\frac{n}{t}$ **do**
        $y_k \leftarrow \text{argmin}_{x \in X}\, \omega(x)$
        $X \leftarrow X \setminus \{y_k\}$
        cost $\leftarrow$ cost + $\omega(y_k)$
        foreach $F \in \mathcal{C}(P_i, y_k)$ **do**
            foreach $x \in X$ with F crossing $\{x_0, x\}$ **do**
                $\omega(x) \leftarrow \omega(x) - \pi(F)$
            $P_i \leftarrow P_i \cup \{y_k\}$
    $\forall F \in \mathcal{F}, \pi(F) \leftarrow \pi(F) \cdot 2^{I(P_i, F)}$
    $\mathcal{P} \leftarrow \mathcal{P} \cup \{P_i\}$
    $X \leftarrow X \setminus P_i$
return $\mathcal{P}$

As we will see later, our experiments show that Min Weight generally finds partitions with lower crossing numbers and runs faster than Greedy Potential. It is beneficial to spend some extra time to search for the vertex with *the lowest* $\omega(\cdot)$ value at every iteration, as it then decreases the total number of crossings.

Part At Once Next, to improve the running time, we present a different approach where weight updates are done only when a part of the partition has been built. This has the added benefit that this can easily be parallelized.

Algorithm 13: PartAtOnce

$n \leftarrow |X|, m \leftarrow |\mathcal{F}|, \mathcal{P} \leftarrow \emptyset$
$\forall F \in \mathcal{F}, \pi(F) \leftarrow 1$
for $i \leftarrow 1$ to t **do**
  $x_0 \leftarrow$ a random element of X
  $\forall x \in X, \omega(x) \leftarrow 0$
  for w steps **do**
    $S \leftarrow$ random range with $\forall F \in \mathcal{F}, P(S = F) = \frac{\pi(F)}{\sum_{G \in \mathcal{F}} \pi(G)}$
    foreach $x \in X$ such that S crosses $\{x, x_0\}$ **do**
      $\omega(x) \leftarrow \omega(x) + \pi(S)$
  $P_i \leftarrow \{x_0\} \cup M_\omega$ where M_ω contains the $\frac{n}{t} - 1$ elements of X with the smallest weight w.r.t ω
  $\forall F \in \mathcal{F}, \pi(F) \leftarrow \pi(F) \cdot 2^{I(P_i, F)}$
  $\mathcal{P} \leftarrow \mathcal{P} \cup \{P_i\}$
  $X \leftarrow X \setminus P_i$
return $\mathcal{P}$

This algorithm builds each part of the partition by picking a random first element x_0 and then adding the $\frac{n}{t} - 1$ elements with the lowest ω values. Once a part is built, we update the weights of the ranges using the MWU rule as before. Furthermore, we only use an approximation of the cost of adding an element: we sample $w = \Theta(t)$ ranges according to their weights and compute elements' cost only with respect to these ranges. The resulting algorithm is given in Part At Once. The overall time complexity is $O(mn + twn)$, as the weight update takes time $O\left(\frac{mn}{t}\right)$. Another advantage of this algorithm is that the weight update step now can be parallelized to as many as m cores.

5.4 Experiments

We now turn to an evaluation of our algorithms on a variety of data.

Our initial experiments, including the ones for parameter setting are performed on geometric set systems induced by half-spaces in $\mathbb{R}^d$. There are two reasons for this:

Lower bound. The base set X can be constructed such that any partition into t parts has crossing number $\Omega\left(t^{1-\frac{1}{d}}\right)$.

Comparability. The only known implementation of low-crossing partitions is for half-space set systems in $\mathbb{R}^2$ [MP18].

In particular, we consider the following set system $(X, \mathcal{F})$.

Grid set system. Given two parameters $n, d \in \mathbb{N}$, the base set X consists of n points in the unit hypercube $[0,1]^d$ uniformly at random (each coordinate is set independently and uniformly). The range set $\mathcal{F}$ is induced by $dn^{1/d}$ grid-like halfspaces (for each of the standard basis vectors, we add $n^{1/d}$ evenly-spaced half-spaces orthogonal to this vector, each containing the point $(1, ..., 1)$). We also tested our algorithms on the variants of this set system with non-uniform point distributions and other types of halfspaces distribution such as non-uniform grids. We obtained similar results in terms of crossing number on non-uniform halfspaces-spanned set systems and therefore did not include these results in this work.

Regarding non-geometric set systems, we run our algorithms on different types of graphs (power-law neighborhood hypergraphs, Facebook social circles data, and ArXiv co-authorship data from the SNAP dataset collection [LK14]). We also provide in Section 5.4.3.3 experiments on lines in the projective planes where partitions have been proved to have linear crossing number for $t = O(\sqrt{n})$ [AHW87] and we obtain results in agreement to this.

5.4.1 Implementations

We provide two implementations for the partition algorithms, one in C++ and one in Rust. The Rust implementation is faster but more memory consuming. They are compatible as they use the same file format to store set systems and partitions. The Rust implementation does not implement all types of set systems generation as the set systems can be generated efficiently with the C++ implementation and partitioned afterwards with the Rust code. We also did not reimplement some evaluation and experimental functions such as violation number computation in Rust. The Rust implementation should mainly be used to quickly partition set systems stored in files. The runtimes we give in this work have been obtained with the Rust implementation.

We ran our experiments on a home computer with 16GB of RAM and an AMD Ryzen 7 5800X (16 cores) @ 4.85 GHz, all 16 cores were used for Part At Once.

5.4.2 Algorithm Parameters

Experiments on the number of samples to approximate the weights in Part At Once

Recall that the algorithm Part At Once has an input parameter $\boldsymbol{w}$ which determines how many weight-updates we do each iteration. That is, we want to estimate the sums $\sum_{F \in \mathcal{F}} \pi(F) I(F, \{x_0, x\})$ for all $x \in X$.

We study the influence of the parameters of set systems on the number of samples required to obtain a good approximation of the weight of each element. We draw in Figure 9 the evolution of the crossing number depending on $\boldsymbol{w}$ with varying parameters n, d, t. This experiment has been obtained by averaging 10 runs of Part At Once on grid set systems with $n = 8192, d = 2$ and $t = 128$ when they are not the varying parameter.

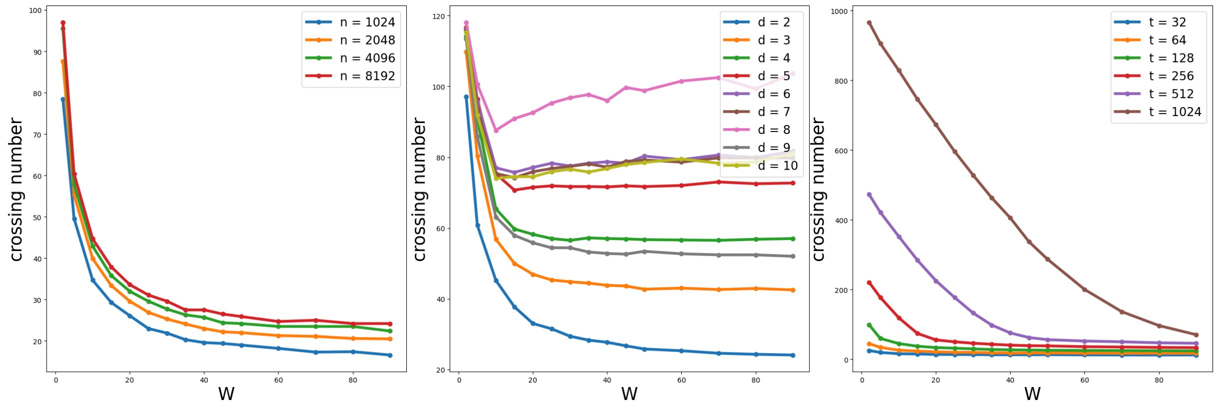


Figure 9: Evolution of the crossing number with $\boldsymbol{w}$ on grid set systems with varying n, d, t from left to right

As we can see on the figures, increasing $\boldsymbol{w}$ will decrease the crossing number. However after some point the decrease in the crossing number is small. At this point, further increasing $\boldsymbol{w}$ is not interesting as it increase runtime without significantly improving the crossing number.

We see that changing n and d does not change the number of samples required to obtain a small crossing number. The third graph reveals that the number of sample should be a function of t and interpolating the results gives that $\boldsymbol{w} \approx \frac{t}{2}$ is the point where the crossing number decrease is slow.

Thus we set $\boldsymbol{w} = \max(30, \frac{t}{2})$ for all the experiments with Part At Once on the grid set system.

We repeated this experiment on random halfspaces and random power-law graph set systems. We obtained that the number of samples to estimate well the element's weight similarly only depends on t. However, the number of samples to obtain a small crossing w.r.t. n and d is higher, therefore, we set $\boldsymbol{w} = \max(100, \frac{t}{2})$ for the other experiments.

Number of potential function violations

We have shown that the overall crossing number is bounded if we always extend parts with elements such that $\text{cost}(P_i) + \omega(x)$ is below the potential function of (15).

In Greedy Potential and Min Weight, we keep track of the number of iterations where there was no element such that $\text{cost}(P_i) + \omega(x)$ is below the upper bound provided by the potential function with $\text{cost}(P_i)$ the weight of the partial part (i.e. sum of the elements' weight added to it) during the course of the algorithm.

As we noted before, the potential function used in our implementations is stricter than the one provided by Theorem 5.1 (the theoretical potential is larger on all but the first part). We study whether it violates the experimental potential function as well as the theoretical potential function from Theorem 5.1.

Figure 10 visualizes how the number of violations evolve during the construction of each part of the partition. It consists of blocks of 4 lines, each block containing the data for 8192 points on the 2 −dimensional grid set system with, from top to bottom, 16, 32 and 64 parts. The 4 lines correspond (from top to bottom) to the following algorithms and violation measures:

(1) Greedy Potential , with violations according to the potential function of Theorem 5.1,
(2) Min Weight, with violations according to the potential function of Theorem 5.1,
(3) Greedy Potential, with violations according to the potential function of (21),
(4) Min Weight, with violations according to the potential function (21).

Each line consists of t many squares, where the color of the i^{th} square encodes the number of violations occurred during the construction of the i^{th} part.

These data have been obtained by averaging over the results of 100 runs of the algorithms. We observe that Min Weight rarely picks an element violating the potential function bound, even with respect to the practical, stricter potential function. However, when looking at Greedy Potential compared to the experimental potential function, we see an increase in the number of violations. Interestingly, in most of the cases, we still stay below the theoretical potential function bound.

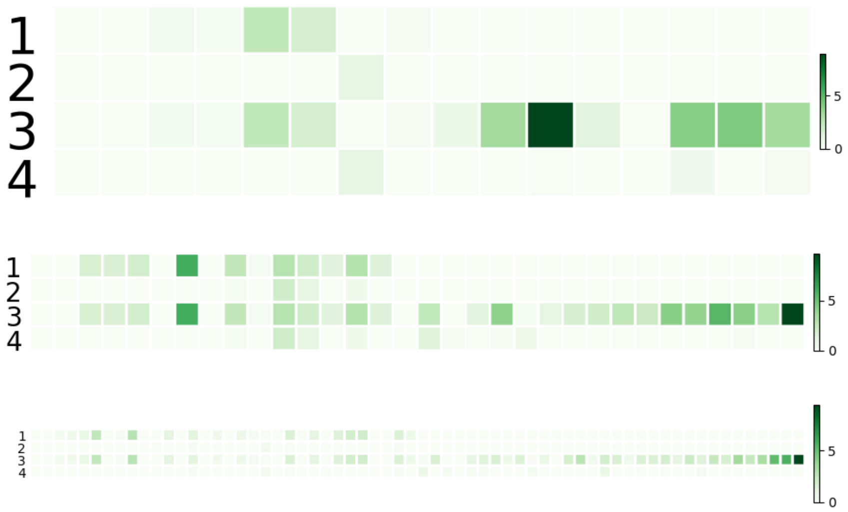


Figure 10: Number of potential function bound violations.

For reference, we also include numerical data on the number of potential function violations in Table 1 with bigger partition sizes.

input n,d,t	Greedy Potential # violations	Min Weight # violations
8192,2,128	159.4	2.1
8192,2,256	173.4	4.3
8192,2,512	121.9	1.1
8192,2,1024	84.0	21.3
8192,2,2048	60.9	11.6
8192,2,512	121.9	1.1
8192,3,512	146.7	47.2
8192,4,512	141.6	47.6
8192,5,512	96.4	33.5
8192,10,512	1.7	16.5
2048,2,512	20.9	0
4096,2,512	49.5	0
8192,2,512	121.9	0
16384,2,512	232.7	7.0

5.4.3 Performance evaluation

Grid set system

The grid set system has two parameters, $n, d \in \mathbb{N}$ and is constructed as follows. We take n points in the unit hypercube $[0,1]^d$, picking each coordinate independently and uniformly. For each of the standard basis vectors, add $n^{\frac{1}{d}}$ evenly-spaced halfspaces orthogonal to this vector, each containing the point $(1, ..., 1)$.

We studied the evolution of both the crossing number and runtime of our algorithms depending on the different variables n, d and t of the grid set system. Figure 11 illustrates the results of the algorithms averaged over 10 executions. We also include the raw data of experiments on grids and random halfspace in Table 2.

For $d = 2$, we compare our method to **MP-Matoušek**, which is the implementation of Matoušek's algorithm by Matheny and Phillips [MP18] in `Python`[19]. **MP-Matoušek** uses the branch factor of the polytree they build to construct cuttings as an optimization parameter: increasing it can reduce the crossing number but also increases runtime. We use the default branching factor provided by their implementation in our experiments.

[19]The code of Matheny is available on Github [Mat18], we modified it to be able to use it on our input data and made the modifications available on Github.

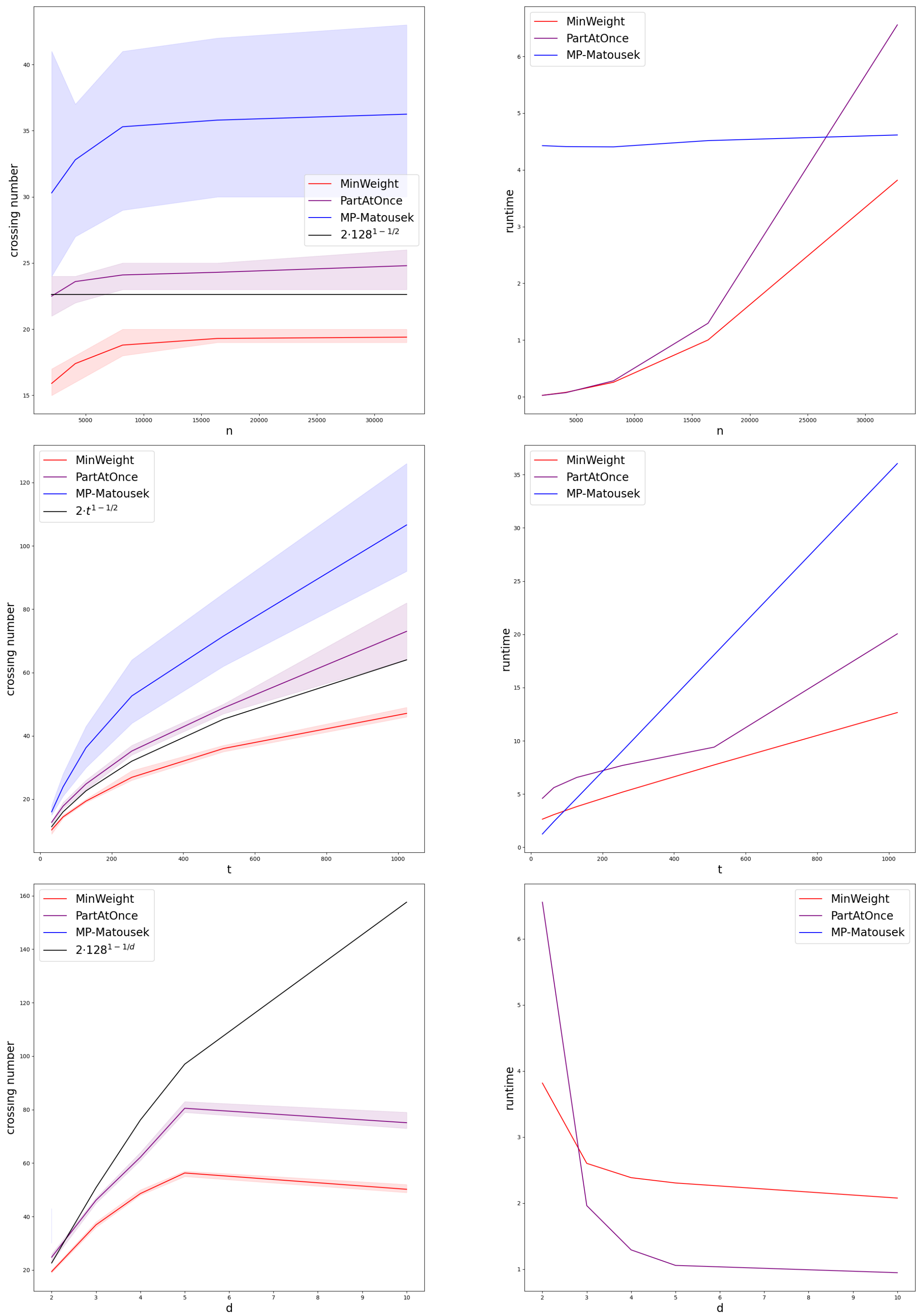


Figure 11: Average crossing numbers and runtimes of the 3 variations of the algorithm depending on the parameters n, d, t on the grid set system. The curves trace the averages, the shaded area corresponds to ± 1 standard deviation. Each parameter has been tested independently and we set $n = 8192$, $d = 2$ and $t = 512$ when they are not the parameter varying.

The results are presented in Figure 11. The left column represents the crossing numbers with, from top to bottom, varying n, d, and finally t. The lines represent $\kappa_{\mathcal{F}}$. The black line marks $2t^{1-\frac{1}{d}}$, that is the order of the crossing number that can be achieved for set systems induced by halfspaces. The red lines corresponds to Min Weight, the purple one to Part At Once and the blue lines to **MP-Matoušek**. The graphs on the right represent the runtimes with the same color code.

We see that Min Weight and Part At Once obtain crossing numbers close to the optimal bounds of $t^{1-\frac{1}{d}}$. Part At Once is significantly faster than Min Weight on large set systems even if on small ones the overhead of parallelization makes it slower. This is particularly visible on random halfspaces set system partition as $|\mathcal{F}|$ is larger (cf Table 2).

Greedy Potential performs worse than the other two methods on abstract systems as well, we omitted further data on this algorithm for readability purpose. However, we include it for reference on figures where it does not diminish readability (Table 2, Figure 12 and Table 7).

We even see that Min Weight and Part At Once consistently outperform **MP-Matoušek** in terms of crossing number. However, for large halfspaces-spanned set systems, **MP-Matoušek** is faster than our algorithms as it uses ranges sampling to compute cuttings (cf. random halfspaces results in Table 2). Sampling also allows their implementation to maintain a stable runtime when n and m increases when our runtimes increases with both. However our implementations' runtime doesn't increase as much when t increases. We also tried adding ranges sampling in our weight computation, the results we obtained were not conclusive.

Table 2: κ_F and runtime of our algorithms on the grid set system and a set system generated by halfspaces.

input n,d,t	$2t^{1-1/d}$	**MP-Matoušek** $\kappa_{\mathcal{F}}$	runtime (s)	Min Weight $\kappa_{\mathcal{F}}$	runtime (s)	Greedy Potential $\kappa_{\mathcal{F}}$	runtime (s)	Part At Once $\kappa_{\mathcal{F}}$	runtime (s)
Grid									
2048,2,128	22.6	30.3	4.43	15.9	0.0265	45.4	0.029	22.5	0.0253
4096,2,128	22.6	32.8	4.41	17.4	0.0789	50.0	0.0867	23.6	0.0734
8192,2,128	22.6	35.3	4.41	18.8	0.256	49.0	0.259	24.1	0.281
16384,2,128	22.6	35.8	4.52	19.3	1.0	52.3	0.848	24.3	1.3
32768,2,128	22.6	36.25	4.62	19.4	3.82	50.8	2.83	24.8	6.56
32768,2,32	11.3	16.0	1.24	10.3	2.64	22.6	1.59	12.7	4.59
32768,2,64	16.0	23.9	2.41	14.4	3.06	34.5	1.97	17.8	5.6
32768,2,128	22.6	36.25	4.62	19.4	3.82	50.8	2.83	24.8	6.56
32768,2,256	32.0	52.6	9.07	26.9	5.18	99.8	4.3	35.2	7.69
32768,2,512	45.3	71.5	18.1	36.0	7.74	211.8	7.45	48.8	9.4
32768,2,1024	64.0	106.6	36.0	47.1	12.7	566.6	14.7	73.0	20.1
32768,2,128	22.6	36.25	4.62	19.4	3.82	50.8	2.83	24.8	6.56
32768,3,128	50.8			36.9	2.6	91.1	1.12	46.1	1.96
32768,4,128	76.1			48.6	2.38	104.2	0.855	62.2	1.29
32768,5,128	97.0			56.3	2.3	128.0	0.657	80.5	1.06

input n,d,t	$2t^{1-1/d}$	**MP-Matoušek** $\kappa_{\mathcal{F}}$	runtime (s)	Min Weight $\kappa_{\mathcal{F}}$	runtime (s)	Greedy Potential $\kappa_{\mathcal{F}}$	runtime (s)	Part At Once $\kappa_{\mathcal{F}}$	runtime (s)
32768,10,128	158			50.2	2.08	128.0	0.311	75.1	0.946
Random Halfspaces									
2048,2,128	22.6	34.5	4.3	19.0	2.8	49.5	3.27	27.8	0.764
4096,2,128	22.6	37.2	4.35	20.5	12.0	54.4	13.9	28.7	5.52
8192,2,128	22.6	40.0	4.43	21.9	50.1	53.2	56.7	28.4	45.1
8192,2,32	11.3	17.6	1.13	12.5	18.0	24.6	19.6	14.2	34.6
8192,2,64	16.0	26.3	2.26	16.3	29.5	38.1	33.4	20.3	40.3
8192,2,128	22.6	40.0	4.43	21.9	50.1	53.2	56.7	28.4	45.1
8192,2,256	32.0	55.3	9.0	28.4	89.6	110.0	108	40.4	49.1
8192,2,512	45.3	79.5	17.6	36.2	166	240.4	214	46.3	61.8
8192,2,128	22.6	40.0	4.43	21.9	50.1	53.2	56.7	28.4	45.1
8192,3,128	50.8			44.5	66.1	92.8	76.9	55.9	37.4
8192,4,128	76.1			64.7	81.5	113.7	98.5	86.6	27.9
8192,5,128	97.0			82.3	94.4	122.4	86.7	107.0	20.3
8192,10,128	158			121.3	128	128.0	90.5	128.0	10.4

Abstract set systems induced by neighborhoods in graphs

Now we turn to experiments on abstract set systems. We will focus on set systems induced by closed neighborhoods in graphs, which allows us to perform experiments both on large-scale synthetic data (power-law random graphs), and on real-word network datasets. Given a graph $G = (V, E)$, the neighborhood set system of G is a set system $(X, \mathcal{F})$ with $X = V$, and $\mathcal{F}$ consists of closed neighborhoods of vertices, that is, $\mathcal{F} = \{\{y \in V : (x, y) \in E\} \cup \{x\} : x \in V\}$. Our implementation initiates the study of low-crossing partitions for these set systems; as there are no theoretical guarantee for their existence, we compare our results with $t^{1-\frac{1}{d}}$ where d is the VC-dimension.[20]

Power-law random graphs. A random graph generated with respect to the power-law distribution (n, β) is a graph where the probability for a vertex to have degree $c \geq 1$ is proportional to $\frac{1}{c^\beta}$ [ACL01]. Coudert et al. [Cou+24] studied the expected VC-dimension of power-law graphs. We use their average observed VC-dimension to evaluate the crossing numbers obtained by our algorithms. The results are presented in Table 3.

Table 3: Crossing number and runtime of our algorithms on the power-law graph neighborhood set system.

input n,β,t	VC-dim [Cou+24]	$t^{1-1/d}$	Min Weight $\kappa_{\mathcal{F}}$	runtime (s)	Part At Once $\kappa_{\mathcal{F}}$	runtime (s)
2000,2,32	5.2	16.43	10.4	0.0139	18.05	0.00837

[20] Note that the neighborhood set system of a graph is self-dual, thus d is equal to the dual VC-dimension as well.

input n,β,t	VC-dim [Cou+24]	$t^{1-1/d}$	Min Weight		Part At Once	
			$\kappa_{\mathcal{F}}$	runtime (s)	$\kappa_{\mathcal{F}}$	runtime (s)
2000,2.5,32	3.8	12.85	8.0	0.0132	15.5	0.00789
2000,3,32	3	10.08	6.1	0.0128	13.5	0.00762
2000,2,128	5.2	50.35	17.25	0.0206	28.5	0.0251
2000,2.5,128	3.8	35.7	10.3	0.0181	20.4	0.0237
2000,3,128	3	25.4	6.8	0.017	15.8	0.0227
2000,2,512	5.2	154.3	18.1	0.0398	31.35	0.0908
2000,2.5,512	3.8	99.15	10.1	0.0335	20.2	0.0865
2000,3,512	3	64.0	6.7	0.0293	14.8	0.0824
4000,2,32	5.8	17.61	10.95	0.0682	19.9	0.0171
4000,2.5,32	4.05	13.6	8.0	0.0682	17.1	0.0157
4000,3,32	3	10.08	6.3	0.0664	14.4	0.0152
4000,2,128	5.8	55.45	20.95	0.0845	34.55	0.0524
4000,2.5,128	4.05	38.63	12.2	0.0796	24.6	0.0478
4000,3,128	3	25.4	8.2	0.0812	18.1	0.0461
4000,2,512	5.8	174.6	26.4	0.145	51.0	0.189
4000,2.5,512	4.05	109.7	12.3	0.127	26.5	0.174
4000,3,512	3	64.0	8.1	0.116	18.2	0.169
30000,2,32	6.8	19.22	11.0	11.1	23.8	0.782
30000,2.5,32	4.75	15.43	8.2	11.1	21.2	0.727
30000,3,32	3	10.08	6.2	11.2	17.6	0.707
30000,2,128	6.8	62.71	24.3	12.0	51.7	1.52
30000,2.5,128	4.75	46.09	13.2	11.8	37.4	1.42
30000,3,128	3	25.4	8.8	11.8	24.2	1.35
30000,2,512	6.8	204.6	49.2	13.5	175.8	3.6
30000,2.5,512	4.75	137.7	18.0	13.2	51.2	3.54
30000,3,512	3	64.0	9.2	12.9	28.8	3.44

The crossing numbers that our algorithms obtain are comparable to $t^{1-\frac{1}{d}}$. This suggests that low-crossing partitions might exist in abstract set systems. Similarly to the geometric case, Min Weight obtains the best crossing number among the different variants and the best runtime is obtained with Part At Once due to its parallel execution.

Finally, we tested our algorithms on two *real world* network datasets.

Facebook social circles. We ran our algorithms on set systems generated from a graph representing the friendships between Facebook users. We use the data set from McAuley and Leskovec [ML12], which is composed of the network induced by the friends of 10 users. The graph is composed of 4039 nodes and 88234 edges. The VC-dimension of this graph is 6 [Cou+24]. Given a graph $G = (V, E)$ and a radius $r \in \mathbb{N}$, we now study the set system with base set V, where each vertex $v \in V$ defines a range containing all elements at distance at most r from v in G, that is, our set of ranges is defined as $\mathcal{F} = \{\{y \in V : \text{dist}_G(x, y) \leq r\} : x \in V\}$. We compile the results obtained for experiments on social network data in Table 4.

Table 4: Crossing number and runtime of our algorithms on the power-law graph neighborhood set system.

input	Min Weight		Part At Once	
r,t	$\kappa_{\mathcal{F}}$	runtime (s)	$\kappa_{\mathcal{F}}$	runtime (s)
1,10	6	0.0677	10	0.0213
1,20	7	0.0709	14	0.0286
1,40	9	0.0949	17	0.0364
2,10	7	0.115	7	0.324
2,20	6	0.188	10	0.431
2,40	10	0.334	14	0.468
3,10	6	0.146	6	0.487
3,20	7	0.247	8	0.618
3,40	9	0.422	9	0.683

The graph upon which we build our set system is relatively sparse: the average degree is 21.84, but the maximum degree is 1045. We observe that our crossing numbers are consistently below the $t^{1-\frac{1}{d}}$ bound, where d is the VC-dimension of the graph.

ArXiv co-authorship graph. We also did experiments on the neighborhood set system of the collaboration graph from the High Energy Physics - Phenomenology arXiV subject [LKF07]. It is composed of 12008 nodes and 118521 edges, with a maximum degree of 491 and VC-dimension 5 [Cou+24]. We ran experiments with higher values of t and observe that the crossing numbers remain low.

Table 5: Crossing number and runtime for our algorithms on the ArXiv co-authorship graph.

input	Min Weight		Part At Once	
r,t	$\kappa_{\mathcal{F}}$	runtime (s)	$\kappa_{\mathcal{F}}$	runtime (s)
1,50	12	0.973	21	0.182
1,100	16	1.08	28	0.256
1,200	19	1.17	34	0.373
1,500	26	1.46	41	0.74
2,50	30	1.6	42	0.885

input r,t	Min Weight $\kappa_{\mathcal{F}}$	Min Weight runtime (s)	Part At Once $\kappa_{\mathcal{F}}$	Part At Once runtime (s)
2,100	48	2.3	69	1.03
2,200	73	3.8	118	1.42
2,500	121	7.29	254	1.94
3,50	41	4.04	49	2.42
3,100	72	7.01	90	3.0
3,200	126	12.8	172	4.06
3,500	287	29.3	384	5.71

Finite projective planes

Projective planes of order $a \in \mathbb{N}$ are a set of points X and lines $\mathcal{F}$ with the following properties:

- $|X| = |\mathcal{F}| = a^2 + a + 1$,
- $\forall x \in X, |\{F \in \mathcal{F} : x \in F\}| = a + 1$,
- $\forall F \in \mathcal{F}, |F \cap X| = a + 1$.

Alon, Haussler and Welzl [AHW87] showed that for $t = O(\sqrt{n})$, finite projective planes do not admit partitions of size t with sublinear crossing number even though these set systems have VC-dimension 2. We tested our algorithms on the embedding of projective planes in dimension 3, that is $X \subseteq \mathbb{N}^3$ and lines are constructed to meet properties 2 and 3. We only implemented this sets system for prime orders as this embedding does not work with some non-primes order. The data for our experiments on projective planes of order 233 is in Table 6.

Table 6: Crossing number and runtime of our algorithms on the power-law graph neighborhood set system.

input n,t	Part At Once κ_F
54523,200	185
54523,500	229
54523,1000	234
54523,2000	234

As expected our algorithms obtains the maximum intersection number for high number of partitions.

ε-Approximations

As explained in Section 1.3, low-crossing partitions are a way to compute ε-approximation.

We compare the error factor $\varepsilon = \max_{F \in \mathcal{F}} \left| \frac{|F|}{|X|} - \frac{|F \cap A|}{|A|} \right|$ obtained with random sample of size t as well as those obtained with an approximation constructed from a t-partition on our different algorithms over the grid set system. The results are compiled in Figure 12. Talagrand [Tal94] showed that a uniform random sample of size t is an $O\left(\frac{1}{\sqrt{t}}\right)$-approximation.

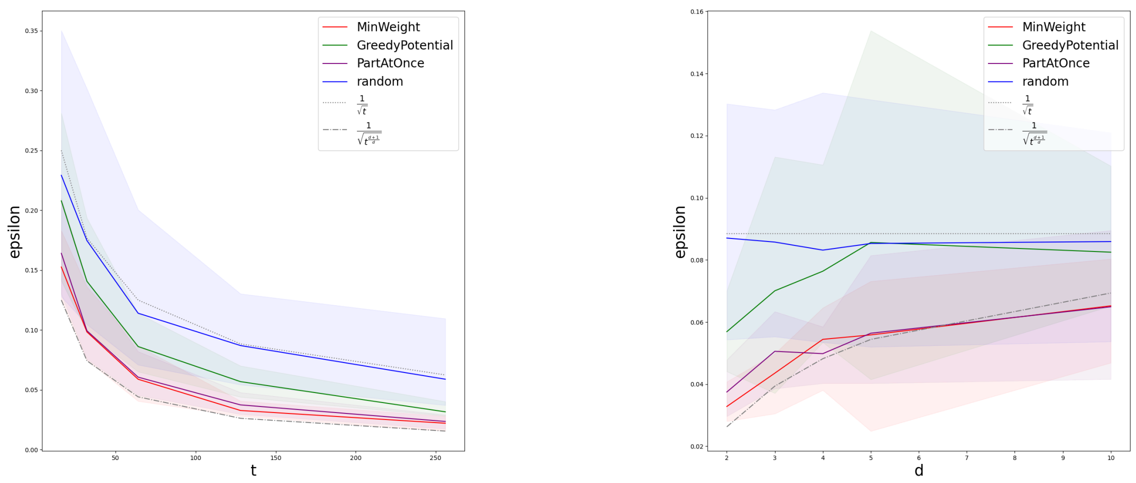


Figure 12: Value of the error factor ε for varying t on the 2-dimensional grid set system on top and d on 64 parts of the d-dimensional grid set system on the bottom. The grey curves represent $\frac{1}{\sqrt{t}}$ and $\frac{1}{\sqrt{t^{\frac{d+1}{d}}}}$ which are respectively the expected error factor for a uniform random sample of size t and the optimal error factor for the d-dimensional grid sets system

As we can see, our algorithms outperform the random sample (in blue) and, in particular, Min Weight and Greedy Potential obtain an error factor close to the optimal one $\frac{1}{\sqrt{t^{\frac{d+1}{d}}}}$ [Ale90] (see the known bounds section of [Mus22] for a sketch of the proof). The raw data for the experiment is available in Table 7 below.

Table 7: $\max_{F\in\mathcal{F}}\left|\frac{|F|}{|X|}-\frac{|F\cap A|}{|A|}\right|$ for our algorithms on the grid set system averaged over 10 runs.

input n,d,t	Random Sample Error Factor	Min Weight Error Factor	Greedy Potential Error Factor	Part At Once Error Factor
8192,2,16	0.2292	0.1526	0.2078	0.164
8192,2,32	0.1747	0.09839	0.1407	0.0994
8192,2,64	0.114	0.05889	0.08621	0.06064
8192,2,128	0.08704	0.03281	0.0569	0.03748
8192,2,256	0.05892	0.02216	0.0317	0.02366
8192,3,16	0.2463	0.1723	0.2419	0.1819
8192,3,32	0.1801	0.1191	0.1507	0.1264
8192,3,64	0.1177	0.07119	0.1021	0.07696
8192,3,128	0.08575	0.04354	0.07006	0.05057
8192,3,256	0.05213	0.03263	0.04526	0.03157
8192,4,16	0.2458	0.1634	0.2327	0.2001

input n,d,t	Random Sample Error Factor	Min Weight Error Factor	Greedy Potential Error Factor	Part At Once Error Factor
8192,4,32	0.1755	0.1268	0.158	0.1286
8192,4,64	0.1243	0.08322	0.1005	0.08251
8192,4,128	0.08316	0.05443	0.07637	0.04983
8192,4,256	0.05478	0.03888	0.04963	0.03499
8192,5,16	0.2475	0.1915	0.2046	0.2068
8192,5,32	0.1695	0.1139	0.177	0.1297
8192,5,64	0.116	0.07838	0.1028	0.08718
8192,5,128	0.08529	0.05586	0.08562	0.05641
8192,5,256	0.06015	0.04	0.04855	0.0391
8192,10,16	0.2305	0.1811	0.2308	0.2069
8192,10,32	0.1787	0.1383	0.1806	0.1352
8192,10,64	0.1165	0.1005	0.1301	0.1064
8192,10,128	0.08591	0.06522	0.0825	0.0649
8192,10,256	0.06065	0.0462	0.05548	0.03805

Chapter 6

Near-Minimal δ-Coverings of Finite VC-dimension Set Systems and Applications

This chapter, combines several tools and techniques to give new algorithms to compute low-discrepancy colorings of finite VC-dimension set systems. We aim to give a self-contained picture of the state-of-the-art algorithms to compute optimal discrepancy colorings of finite VC-dimension set systems.

The only method known to obtain low-discrepancy colorings of finite primal VC-dimension set systems is to compute small-size δ-coverings and use the chaining technique. We present improvements on the trivial algorithms to obtain small-size coverings and give a complete picture of the algorithms and their complexities to derive low-discrepancy colorings from small-size colorings.

We will present algorithms to compute near-minimal δ-coverings of finite VC-dimension set systems. We define them below.

Definition 6.1. ((δ_1, δ_2)-covering) Let $(X, \mathcal{F})$ be a finite set system and $\delta_2 \leq \delta_1 \leq n$. A (δ_1, δ_2)-covering of $(X, \mathcal{F})$ is a collection $\mathcal{C} \subseteq \mathcal{F}$ such that $\mathcal{C}$ is a δ_1-covering and a δ_2-packing of $(X, \mathcal{F})$.

We say that a (δ_1, δ_2)-covering is a *near-minimal δ-covering* iff $\delta_2 = \Theta(\delta_1)$.

Near-minimal coverings are practically equivalent to minimal coverings for most applications that require minimal coverings as the bound on their size is equal up to some constant.

In Section 6.1, we present our main algorithm that computes a $\left(\delta, \frac{\delta}{4}\right)$-covering of a set system with finite VC-dimension. This is the first instance of a non-trivial algorithm that computes δ-coverings of size matching Haussler's bound. In Section 6.3, we present variations of our main algorithm for specific types of finite VC-dimension set systems. We summarize the time complexity of our algorithms compared to the previous best known algorithms in the following table.

Algorithm	Size of δ-covering	Time complexity
VC-dimension d		
[MWW93]	$O\left(\left(\frac{n}{\delta}\log\left(\frac{n}{\delta}\right)\right)^d\right)$	$O\left(\frac{mdn}{\delta}\log\left(\frac{n}{\delta}\right) + \frac{dn^d}{\delta^d}\log^{d+1}\left(\frac{n}{\delta}\right)\right)$
Greedy covering (Section 3.3.1.1)	$O\left(\left(\frac{n}{\delta}\right)^d\right)$	$O\left(\frac{mn^{d+1}}{\delta^d}\right)$

Algorithm	Size of δ-covering	Time complexity
Algorithm 14	$O\left(\left(\frac{n}{\delta}\right)^d\right)$	$O\left(\frac{mdn}{\delta}\log\left(\frac{n}{\delta}\right)+\left(\frac{n}{\delta}\right)^{2d+2}\log^d\left(\frac{n}{\delta}\right)\right)$
Halfspaces		
[Mat92]	$O\left(\left(\frac{n}{\delta}\right)^d\right)$	$O(d(n+m)+n^d+m\log(n))$
[Mat92] (fast variation[21])	$O\left(\left(\frac{n}{\delta}\right)^d\right)$	$O\left(dn+\frac{n^d}{\delta^{d-1}}+d^3\left(\frac{n}{\delta}\right)^d\right)$
Algorithm 18	$O\left(\left(\frac{n}{\delta}\right)^d\right)$	$O\left(\frac{m\ln(m)n^d}{\delta^{2d+2}}\right)$
Balls		
[Mat92] + Veronese maps[22]	$O\left(\left(\frac{n}{\delta}\right)^{d+1}\right)$	$O(d(n+m)+n^{d+1}+m\log(n))$
Algorithm 17	$O\left(\left(\frac{n}{\delta}\right)^d\right)$	$O\left(\left(\frac{nd}{\delta}\right)^{2d+2}\log^d\left(\frac{nd}{\delta}\right)\right)$

In Section 6.2, we present two applications of our near-minimal covering algorithm. First we show that combining our algorithm with the Lovett-Meka discrepancy algorithm [LM15] and chaining as presented in [Mat95] gives the current fastest algorithm to obtain optimal discrepancy colorings for finite VC-dimension set systems. We also show that our algorithm can be used to improve the runtime of the naive algorithm to verify whether a given subset of X is an ε-approximation of it.

6.1 A near-minimal covering algorithm for finite VC-dimension set systems

In this section, we present our main covering algorithm for finite VC-dimension set systems. The algorithm we present computes a $\left(\delta, \frac{\delta}{4}\right)$-covering of a set system with finite VC-dimension. We recall that this means that the algorithm constructs a δ-covering that is also a $\frac{\delta}{4}$-packing. We first explain with a simple construction why we followed this approach.

6.1.1 Motivation of our approach

Maximal δ-packings are δ-coverings, that is, they are (δ, δ)-coverings. However there exists minimal δ-coverings that are not δ-packings. We show, in Claim 6.2, that there exists minimal δ-coverings with arbitrarily close ranges.

[21] See details in Remark 3.21

[22] See section 10.3 of [Mat13] for information on Veronese maps

Claim 6.2. Let $X = [1, ..., n]$, $\forall \varepsilon > 1$ and $\delta \geq 1$ s.t. $2\delta + \varepsilon \leq n$, there exists a collection of four ranges $\mathcal{F}$ of X s.t. $\mathcal{F}$ admits a minimal δ-covering $\mathcal{C}$ with

$$\min_{C_1 \neq C_2 \in \mathcal{C}} |\Delta(C_1, C_2)| = \varepsilon.$$

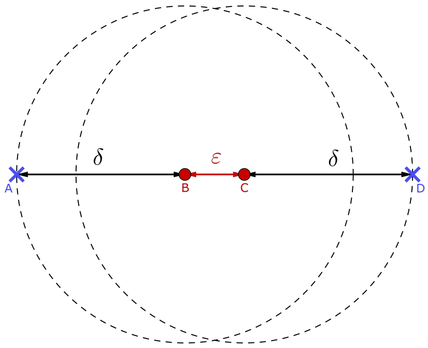


Figure 13: $\mathcal{F}$ as dots with euclidean distance representing symmetric difference

Proof. Consider the collection

$$\begin{aligned} \mathcal{F} = \{A &= \{1, ..., n\}, \\ B &= \{\delta + 1, ..., n\}, \\ C &= \{\delta + \varepsilon + 1, ..., n\}, \\ D &= \{2\delta + \varepsilon + 1, ..., n\}\}. \end{aligned}$$

We show that $\mathcal{C} = \{B, C\}$ satisfies Claim 6.2.

Since $|\Delta(A, B)| = |\Delta(C, D)| = \delta$, B covers A and itself and C covers D and itself. Thus $\mathcal{C}$ is a δ-covering of $\mathcal{F}$.

Since $|\Delta(B, D)| = |\Delta(A, C)| = \varepsilon + \delta$, if either B or C were to be removed from $\mathcal{C}$, A or D would not be covered anymore. Thus $\mathcal{C}$ is a minimal δ-covering.

Finally, $\min_{C_1 \neq C_2 \in \mathcal{C}} |\Delta(C_1, C_2)| = |\Delta(B, C)| = \varepsilon$. □

In the set system we detail in the proof of Claim 6.2, $\{A, C\}, \{A, D\}$ and $\{B, D\}$ are also minimal δ-coverings but, unlike $\{B, C\}$, are also δ-packings. Constructing minimal δ-coverings that are not δ-packings can prove to be rather difficult as they do not have as much structure as δ-packings. This is why our focus for efficiently building minimal δ-coverings is to find coverings with packing properties as well.

6.1.2 Our Algorithm

We now present our near-minimal covering algorithm. This algorithm uses the fast δ-covering construction of [MWW93] presented in Section 3.3.1.2. It then prunes the ranges that are redundant in the covering obtained.

Algorithm 14: $(\delta, \frac{\delta}{4})$-covering algorithm

Input: $(X, \mathcal{F}), \delta$
$\mathcal{C} \leftarrow \frac{\delta}{2}$-covering obtained with Algorithm 3
$\mathcal{R} \leftarrow \emptyset$
$A \leftarrow \frac{\delta}{8n}$-approximation of $(X, \Delta(\mathcal{C}))$
for $C \in \mathcal{C}$ **do**
  if $\forall R \in \mathcal{R}, |\Delta(C, R) \cap A| \geq \frac{3\delta|A|}{8n}$ **then**
    $\mathcal{R} \leftarrow \mathcal{R} \cup \{C\}$
return $\mathcal{R}$

This algorithm performs pruning in a greedy manner. Starting from the covering obtained from Section 3.3.1.2, we add a range to our final construction only if it is not already covered by some range previously added to our construction. In addition to simply verifying whether a range is covered by another one, we approximate the symmetric difference between ranges using an ε-approximation. We show the following theorem.

Theorem 6.3. Given a set system $(X, \mathcal{F})$ with VC-dimension at most d and $\delta \in [4, n]$, Algorithm 14 returns a near-minimal $(\delta, \frac{\delta}{4})$-covering with probability at least $\frac{1}{2}$. Moreover, it has time complexity $O\left(\frac{mdn}{\delta}\log(\frac{n}{\delta}) + (\frac{n}{\delta})^{2d+2}\log^d(\frac{n}{\delta})\right)$.

Proof. Since $\mathcal{C}$ is a sub-set system of $(X, \mathcal{F})$, the VC-dimension of $(X, \mathcal{C})$ is at most d. By definition of a $\frac{\delta}{8n}$-approximation, for all $C, C' \in \mathcal{C}$,

$$|\Delta(C, C')| \geq \frac{|\Delta(C, C') \cap A| n}{|A|} - \frac{\delta}{8}.$$

Thus any $C \in \mathcal{C}$ added to $\mathcal{R}$ is such that:

$$\forall R \in \mathcal{R}, |\Delta(R, C)| \geq \frac{3\delta}{8} - \frac{\delta}{8} = \frac{\delta}{4}$$

i.e. $\mathcal{R}$ is a $\frac{\delta}{4}$-packing.

The ε-approximation also guarantees that:

$$|\Delta(C, C')| \leq \frac{|\Delta(C, C') \cap A| n}{|A|} + \frac{\delta}{8}.$$

That is,

$$\forall C \in \mathcal{C}, \exists R \in R \text{ (eventually itself) s.t. } |\Delta(R, C)| \leq \frac{3\delta}{8} + \frac{\delta}{8} = \frac{\delta}{2}$$

i.e. $\mathcal{R}$ is a $\frac{\delta}{2}$-covering of $(X, \mathcal{C})$.

Let $F \in \mathcal{F}$, since $\mathcal{C}$ is a $\frac{\delta}{2}$-covering of $(X, \mathcal{F})$, $\exists C \in \mathcal{C}$ s.t. $|\Delta(F, C)| \leq \frac{\delta}{2}$. The triangular inequality implies that

$$\forall F \in \mathcal{F}, \exists C \in \mathcal{C}, R \in \mathcal{R} \text{ s.t. } |\Delta(F,R)| \leq |\Delta(F,C)| + |\Delta(C,R)| \leq \frac{\delta}{2} + \frac{\delta}{2} = \delta.$$

Thus, $\mathcal{R}$ is a δ-covering of $(X, \mathcal{F})$.

Runtime analysis. By Lemma 3.10 and Corollary 3.6, A can be constructed by sampling $O\left(\frac{dn^2}{\delta^2}\right)$ points from X with probability at least $\frac{1}{\sqrt{2}}$.

Theorem 3.16 states that computing a $\frac{\delta}{2}$-covering with Algorithm 3 has time complexity

$$O\left(\frac{mdn}{\delta}\log\left(\frac{n}{\delta}\right) + \frac{dn^d}{\delta^d}\log^{d+1}\left(\frac{n}{\delta}\right)\right) \tag{22}$$

and that $|\mathcal{C}| = O\left(\left(\frac{n}{\delta}\log\left(\frac{n}{\delta}\right)\right)^d\right)$. By Haussler's packing lemma, $|\mathcal{R}| = O\left(\left(\frac{n}{\delta}\right)^d\right)$. Therefore, the loop of Algorithm 14 has time complexity

$$O(|\mathcal{C}\|\mathcal{R}\|A|) = O\left(\left(\frac{n}{\delta}\log\left(\frac{n}{\delta}\right)\right)^d\left(\frac{n}{\delta}\right)^d d\left(\frac{n}{\delta}\right)^2\right) = O\left(d\left(\frac{n}{\delta}\right)^{2d+2}\log^d\left(\frac{n}{\delta}\right)\right). \tag{23}$$

Adding (22) and (23) finishes the proof. □

6.2 Applications of our δ-covering algorithm

6.2.1 Application to low-discrepancy colorings computation

We now explain some applications of the covering algorithm we presented. The first application that we present is low-discrepancy colorings for finite VC-dimension set systems. We will present two algorithms that use our covering algorithms to compute the ranges to use as constraints with Algorithm 5.

We first explain a simpler version of the complete algorithm that does not use chaining and obtains a coloring with discrepancy $O\left(n^{\frac{1}{2}-\frac{1}{2d}}\log^{\frac{5}{2}}(mn)\right)$.

Obtaining a polylog approximation of the optimal discrepancy for finite VC-dimension set systems

The following algorithm produces a coloring with polylog approximation of the optimal discrepancy of finite VC-dimension set systems, i.e. a coloring with for all $F \in \mathcal{F}, \chi(F) = \tilde{O}\left(n^{\frac{1}{2}-\frac{1}{2d}}\right)$.

Algorithm 15: Polylog approximation of low-discrepancy coloring for finite VC-dimension set systems

Input: $(X, \mathcal{F})$

$\mathcal{C} \leftarrow \left(2^{\frac{6}{d}+2}c_{\mathrm{H}}^{\frac{1}{d}}n^{1-\frac{1}{d}}, 2^{\frac{6}{d}}c_{\mathrm{H}}^{\frac{1}{d}}n^{1-\frac{1}{d}}\right)$-covering of $(X, \mathcal{F})$ computed with Algorithm 14

return complete coloring χ computed with Algorithm 5 with constraints for all $C \in \mathcal{C}, \chi(C) = 0$

This algorithm produces a near-minimal δ-covering of the set system using Algorithm 14 and then uses the ranges of the covering as constraints with Algorithm 5. We prove the following theorem.

Application 6.4. (of Theorem 6.3) Given a set system $(X, \mathcal{F})$, Algorithm 15 returns a coloring $\chi : X \to \{-1, 1\}$ such that:

$$\mathrm{disc}_\chi(X, \mathcal{F}) = O\left(n^{\frac{1}{2}-\frac{1}{2d}} \log^{\frac{5}{2}}(mn)\right)$$

with probability at least $\frac{1}{4}$. The algorithm has time complexity $\tilde{O}\left(mn^{\frac{1}{d}} + n^{2+\frac{2}{d}} \log^d(n) + n^3\right)$.

Proof. Haussler's packing lemma gives that $\mathcal{C}$ has size at most $\frac{n}{32}$. Therefore, the entropy condition of Theorem 3.23 is satisfied since:

$$\sum_{C \in \mathcal{C}} \exp\left(-\frac{0^2}{16}\right) = |\mathcal{C}| \leq \frac{n}{32} < \frac{n}{16}.$$

Let $F \in \mathcal{F}$, since $\mathcal{C}$ is a $2^{\frac{6}{d}+2} c_{\mathrm{H}}^{\frac{1}{d}} n^{1-\frac{1}{d}}$-covering, there exists $C \in \mathcal{C}$ s.t.

$$|\Delta(C, F)| \leq 2^{\frac{6}{d}+2} c_{\mathrm{H}} n^{1-\frac{1}{d}}.$$

That is, we have:

$$\begin{aligned}
\chi(F) &\leq \underbrace{\chi(C)}_{=0} + \chi(C \setminus F) + \chi(F \setminus C) \\
&\leq 8\sqrt{|\Delta(F, C)|} \log(m) \log(n) \sqrt{\ln(8m \log(n))} \\
&\leq 2^{\frac{3}{d}+4} c_{\mathrm{H}}^{\frac{1}{2d}} n^{\frac{1}{2}-\frac{1}{2d}} \log(m) \log(n) \sqrt{\ln(8m \log(n))}
\end{aligned}$$

with probability at least $1 - \frac{1}{2m}$.

The bound on $\chi(C \setminus F)$ and $\chi(F \setminus C)$ comes from Corollary 3.25 ($\nu = \frac{n}{8}$).

The union bound over all ranges of $\mathcal{F}$ gives the statement of the theorem and the time complexity simply follows from the complexities of Algorithm 5 and Algorithm 14. □

Obtaining optimal discrepancy for finite VC-dimension set systems using chaining

The construction from the previous section is not optimal as proven in [Ale90]. We see that some ranges have 0 discrepancy whilst all the other ranges have large discrepancy. We aim to reduce the gap between the discrepancy of ranges in the covering and the discrepancy of ranges outside of it.

We can compute a coloring with a discrepancy bound without the polylog factor. This is attained using the same method that enabled Matoušek [Mat95] to improve on Matoušek, Welzl and Wernisch's result [MWW93]: using chaining (see Section 3.4.2). This leads to Algorithm 16 presented below.

Algorithm 16: Low-discrepancy coloring for finite VC-dimension set systems

Input: $(X, \mathcal{F})$

$\forall i \in [0,r] : \mathcal{C}_i \leftarrow \left(2^{\frac{6}{d}+2-i} c_{\mathrm{H}}^{\frac{1}{d}} n^{1-\frac{1}{d}}, 2^{\frac{6}{d}-i} c_{\mathrm{H}}^{\frac{1}{d}} n^{1-\frac{1}{d}}\right)$-covering of $(X, \mathcal{F})$ computed with Algorithm 14, where $r = 2\log\left(\log(m)\log(n)\sqrt{\log(mn\log(n))}\right)$

$\forall i \in [0,r] : \mathcal{A}_i = \left\{ C \setminus C' : C \in \mathcal{C}_i, C' \in \mathcal{C}_{i+1} \text{ s.t. } \Delta(C,C') \leq 2^{\frac{6}{d}+2-i} c_{\mathrm{H}}^{\frac{1}{d}} n^{1-\frac{1}{d}} \right\}$

$\forall i \in [0,r] : \mathcal{B}_i = \left\{ C' \setminus C : C \in \mathcal{C}_i, C' \in \mathcal{C}_{i+1} \text{ s.t. } \Delta(C,C') \leq 2^{\frac{6}{d}+2-i} c_{\mathrm{H}}^{\frac{1}{d}} n^{1-\frac{1}{d}} \right\}$

return complete coloring χ computed with Algorithm 5 with constraints $\forall C \in C_0, \chi(C) = 0, \forall i \in [0, r-1], \chi(A_i) = \chi(B_i) = \frac{K 2^{\frac{i}{2}}}{(1+i)^2}$ for some large constant K.

This algorithm computes multiple packings in order to bound the discrepancy of smaller and smaller difference sets. This has to be done until the difference sets are small enough for their random discrepancy bound to be small enough. This means that we will compute a family of near-minimal coverings using our algorithm and will use these coverings with carefully chosen constraints in Algorithm 5. We prove the following theorem.

Application 6.5. (of Theorem 6.3) Given a set system $(X, \mathcal{F})$ with finite VC-dimension $\leq d$ such that $n \gg d$, Algorithm 16 returns a coloring $\chi : X \to \{-1, 1\}$ such that:

$$\mathrm{disc}_\chi(X, \mathcal{F}) = O\left(n^{\frac{1}{2}-\frac{1}{2d}}\right)$$

with probability at least $\frac{1}{4}$. The algorithm has time complexity

$$\tilde{O}\left(mn^{\frac{1}{d}} + n^{2+\frac{2}{d}} \log^d(n) + n^3 \log^{3d}(mn) \log^{3d}(\log(mn))\right).$$

Proof. Haussler's packing lemma gives that for all $i \in [0,r]$, $\mathcal{C}_i$ has size at most $\frac{n2^{id}}{32}$. Therefore,

$$\begin{aligned}
\sum_{i=0}^{r-1} (|\mathcal{A}_i| + |\mathcal{B}_i|) \exp\left(-\frac{K^2 2^i}{16(1+i)^4}\right) &= \sum_{i=0}^{r-1} 2|\mathcal{C}_i| \exp\left(-\frac{K^2 2^i}{16(1+i)^4}\right) \\
&= 2\sum_{i=0}^{r-1} \frac{n2^{id}}{32} \exp\left(-\frac{K^2 2^i}{16(1+i)^4}\right) \\
&= \frac{n}{16} \sum_{i=0}^{r-1} 2^{id} \exp\left(-\frac{K^2 2^i}{16(1+i)^4}\right) \\
&= \frac{n}{16} \sum_{i=0}^{r-1} \exp\left(id\ln(2) - \frac{K^2 2^i}{16(1+i)^4}\right). \qquad (24)
\end{aligned}$$

By setting K large enough, we can ensure that

$$\forall i \in [1,r], \exp\left(id\ln(2) - \frac{K^2 2^i}{16(1+i)^4}\right) \leq \frac{6}{\pi^2(1+i)^2}. \qquad (25)$$

Using (25) in (24) yields:

$$\begin{aligned}\sum_{i=0}^{r} |\mathcal{C}_i| \exp\left(-\frac{K^2 2^i}{16(1+i)^4}\right) &\leq \frac{n}{16} \sum_{i=0}^{r-1} \frac{6}{\pi^2(1+i)^2} \\ &\leq \frac{n}{16} \sum_{i=0}^{\infty} \frac{6}{\pi^2(1+i)^2} \\ &\leq \frac{n}{16}.\end{aligned}$$

that is, the entropy condition of Theorem 3.23 is satisfied.

Similarly to Matoušek, we decompose each range of $\mathcal{F}$ in a sum of disjoint error sets computed between the range and a range from each near-minimal covering i.e.

$$\forall F \in \mathcal{F}, \exists A_0 \in \mathcal{A}_0, B_0 \in \mathcal{B}_0, \ldots, A_r \in \mathcal{A}_r, B_r \in \mathcal{B}_r \text{ s.t.}$$
$$F = (\ldots(((A_1 \setminus B_1) \cup A_2) \setminus B_2) \cup \ldots A_r) \setminus B_r.$$

That is, we have:

$$\begin{aligned}\forall F \in \mathcal{F}, \chi(F) &\leq \sum_{i=0}^{r-1} [\chi(A_i) + \chi(B_i)] + \chi(A_r) + \chi(B_r) \\ &\leq 2\sum_{i=0}^{r-1}\left[\sqrt{|A_i|}\frac{K2^{\frac{i}{2}}}{(1+i)^2}\right] + 8\sqrt{|A_r|}\log(m)\log(n)\sqrt{\ln(mn\log(n))} \\ &\leq \sum_{i=0}^{r-1}\left[2^{\frac{3}{d}+1-\frac{i}{2}} c_{\mathrm{H}}^{\frac{1}{2d}} n^{\frac{1}{2}-\frac{1}{2d}} \frac{K2^{\frac{i}{2}}}{(1+i)^2}\right] + \frac{8n^{\frac{1}{2}-\frac{1}{2d}}\log(m)\log(n)\sqrt{\ln(mn\log(n))}}{\log(m)\log(n)\sqrt{\ln(mn\log(n))}} \\ &\leq 2^{\frac{3}{d}+1} c_{\mathrm{H}}^{\frac{1}{2d}} n^{\frac{1}{2}-\frac{1}{2d}} K \sum_{i=0}^{r-1}\left[\frac{1}{(1+i)^2}\right] + 8n^{\frac{1}{2}-\frac{1}{2d}} \\ &\leq 2^{\frac{3}{d}+1} c_{\mathrm{H}}^{\frac{1}{2d}} n^{\frac{1}{2}-\frac{1}{2d}} K \sum_{i=0}^{\infty}\left[\frac{1}{(1+i)^2}\right] + 8n^{\frac{1}{2}-\frac{1}{2d}} \\ &\leq 2^{\frac{3}{d}+1} c_{\mathrm{H}}^{\frac{1}{2d}} n^{\frac{1}{2}-\frac{1}{2d}} K \frac{\pi^2}{6} + 8n^{\frac{1}{2}-\frac{1}{2d}} = O\left(n^{\frac{1}{2}-\frac{1}{2d}}\right).\end{aligned}$$

The bound on $\chi(A_r) + \chi(B_r)$ is obtained with Corollary 3.25 ($\nu = 1$) with probability at least $1 - \frac{2}{nm}$.

With the union bound over all sets in $\mathcal{A}_r \cup \mathcal{B}_r$, we obtain the optimal discrepancy bound for all ranges $F \in \mathcal{F}$ with probability at least $1 - \frac{\left(2\log(m)\log(n)\sqrt{\ln(mn\log(n))}\right)^d}{8m}$. We obtain that this bound on the probability is at least $\frac{1}{2}$ when $m = \Theta\left(n^d\right)$ and $\frac{n}{\log(n)} > 2de^{-\frac{2}{5}}$.

Runtime analysis. Computing the $O(\log(\log(mn)))$ coverings using Algorithm 14 with probability $\frac{1}{2}$ has expected time complexity

$$O\left(\left[mn^{\frac{1}{d}} + n^{2+\frac{2}{d}}\log^d(n)\right]\log(mn)\right). \tag{26}$$

Computing for all $i \in [0, r-1]$, $\mathcal{A}_i$ and $\mathcal{B}_i$ has time complexity

$$O\Big(n^2\Big(\log(m)\log(n)\sqrt{\log(mn\log(n))}\Big)^{2d}\log\Big(\log(m)\log(n)\sqrt{\log(mn\log(n))}\Big)\Big).$$

This is negligible compared to computing the coverings.

Finally computing χ using Algorithm 5, has time complexity

$$O\big(n^3\log^{3d}(mn)\log^{3d}(\log(mn))\big) \tag{27}$$

as we compute a coloring on n points and $O\big(n\log^d(n)\log(\log(mn))\big)$ constraints. Adding (26) and (27) together gives the time complexity stated in Application 6.5. □

Remark 6.6. K will only depend on d and in fact solving (25) gives that $K = 100\sqrt{d}$ is sufficient to ensure the inequality.

Corollary 6.7. Using Algorithm 16 with Theorem 3.32 we obtain that, using the iterated halving technique, we can compute an ε-approximation of a set system with finite VC-dimension d of size $O\Big(\frac{1}{\varepsilon^{\frac{2d}{d+1}}}\Big)$ by setting t s.t. $2^t = \Theta\Big(\varepsilon^{\frac{2d}{d+1}}n\Big)$.

6.2.2 Verifying a random ε-approximation

Lemma 3.10 shows that one can obtain an ε-approximation of a finite VC-dimension set system w.h.p. by sampling uniformly elements of the ground set. In this section, we consider an algorithm to determine whether a given random sample A is an ε-approximation of a given set system.

The naive algorithm to achieve that task would be to verify for each range $F \in \mathcal{F}$ whether the approximation error $\left|\frac{|F|}{|X|} - \frac{|F\cap A|}{|A|}\right|$ is smaller than ε. This method has time complexity $O(m|A|)$ as verifying one range's error factor has time complexity $O(|A|)$ assuming the size of a range can be known in time $O(1)$.

The proof of Lemma 3.10 relies on chaining and coverings. They prove that a subset $A \subseteq X$ is an ε-approximation if it is an ε-approximation w.r.t. $\frac{n}{2^i}$-coverings of X with $0 \leq i \leq \log\big(\frac{1}{\varepsilon}\big)$. This directly implies that given a subset $A \subseteq X$, we can verify whether A is an ε -approximation of X by computing $(\mathcal{C}_i)_{1\leq i\leq \log(\frac{1}{\varepsilon})}$ a family of coverings where C_i is an $\frac{n}{2^i}$-covering of X and verifying that A is an ε-approximation on these coverings.

Algorithm 14 builds a covering of size $O\bigg(\Big(\frac{n}{\log(\frac{1}{\varepsilon})}\Big)^d\bigg)$ in time:

$$O\left(m\log\left(\frac{1}{\varepsilon}\right)\log\left(\log\left(\frac{1}{\varepsilon}\right)\right) + \log^{d+2}\left(\frac{1}{\varepsilon}\right)\log^d\left(\log\left(\frac{1}{\varepsilon}\right)\right)\right).$$

Verifying whether a set A is an approximation of each range of $\mathcal{C}_i$ has time complexity $O(|\mathcal{C}_i\|A|)$. This gives a total time complexity for the construction of the $O\big(\log\big(\frac{1}{\varepsilon}\big)\big)$ coverings and verification for each of them of:

$$O\left(m \log^2\left(\frac{1}{\varepsilon}\right) \log\left(\log\left(\frac{1}{\varepsilon}\right)\right) + \log^{d+3}\left(\frac{1}{\varepsilon}\right) \log^d\left(\log\left(\frac{1}{\varepsilon}\right)\right) + \left(\frac{n}{\log\left(\frac{1}{\varepsilon}\right)}\right)^d |A| \log\left(\frac{1}{\varepsilon}\right)\right).$$

This improves on the time complexity of the naive algorithm when $m = \Theta(n^d)$ and $|A| = O\left(\frac{1}{\varepsilon^2}\right)$ as we obtain a time complexity of $O\left(\frac{m|A|}{\log^{d-1}\left(\frac{1}{\varepsilon}\right)}\right)$.

6.3 Variations of our δ-covering algorithm for specific types of set systems

6.3.1 Geometric set systems with a conforming map

In this section, we discuss improvements on the runtime of Algorithm 14 for geometric set systems with a conforming map. These improvements use ideas presented in [MP19].

Conforming maps emerge naturally when working with geometric set systems. They represent the idea that the geometric objects spanning the set system are determined by a finite number of points. This concept is sometimes called the *degrees of freedom* of a geometric object. For instance, halfspaces in $\mathbb{R}^d$ are determined by at most $d+1$ points. Assuming general position, a consequence is that any halfspace that contains any $d+1$ points of X will also contain all the other points of X corresponding to the halfspace determined by these $d+1$ points[23].

Definition 6.8. A geometric conforming map $\psi : 2^X \to \mathcal{F}$ of size $d \in \mathbb{N}$ has the following properties:

(i) $\forall N \subset X, \forall F \in \mathcal{F}, \psi(F \cap N) \cap N = F \cap N$
(ii) $\forall F \in \mathcal{F}, \exists E \subseteq F, |E| \leq d$ s.t. $\psi(E) = F$

Remark 6.9. The computation of conforming map is, in general, non-trivial. We denote the runtime of its computation with $T_{|E|}$ as its runtime depends, in the cases we present, on the size of E.

To compute near-minimal δ-coverings of set systems admitting a conforming map, we will use the following algorithm.

[23] The reader may refer to [MP19] for more about conforming maps and examples.

Algorithm 17: $\left(\delta, \frac{\delta}{4}\right)$-covering algorithm using conforming maps

Input: $(X, \mathcal{F})$, δ, ψ

$\mathcal{C} \leftarrow \emptyset$

$N \leftarrow \frac{\delta}{2n}$-net of $(X, \Delta(\mathcal{F}))$

for $E \in \mathcal{E} = \{S \in 2^N \text{ s.t. } |S| \leq d\}$ **do**

    $\mathcal{C} \leftarrow \mathcal{C} \cup \{\psi(E)\}$

$A \leftarrow \frac{\delta}{8n}$-approximation of $(X, \Delta(\mathcal{C}))$

for $C \in \mathcal{C}$ **do**

    if $\forall R \in \mathcal{R}, |\Delta(C, R) \cap A| \geq \frac{3\delta|A|}{8n}$ **then**

        $\mathcal{R} \leftarrow \mathcal{R} \cup \{C\}$

return $\mathcal{R}$

In Algorithm 17, instead of building a ε-net to select candidate ranges for our near-minimal covering, we list all subsets of the ground set of size at most d and select the *unique* range determined by each subset. We will prove the following theorem.

Theorem 6.10. Given $(X, \mathcal{F})$ a set system with finite VC-dimension d such that there exists a conforming map $\psi(E) : 2^X \to \mathcal{F}$ of size d with computational time complexity T_d for $|E| \leq d$. Algorithm 17 returns a near-minimal $\left(\delta, \frac{\delta}{4}\right)$-covering of $(X, \mathcal{F})$ with probability at least $\frac{1}{2}$. The time complexity of the algorithm is

$$O\left(T_d \left(\frac{nd}{\delta}\right)^d \log^d\left(\frac{nd}{\delta}\right) + \left(\frac{nd}{\delta}\right)^{2d+2}\right).$$

Proof of Theorem 6.10. Algorithm 17 uses the same principle than Algorithm 14 as we first construct a covering and then prune it greedily whilst approximating the symmetric difference. This is why, in the proof, we simply show that the construction of $\mathcal{C}$ leads to a $\frac{\delta}{2}$-covering of the same size than Algorithm 4. Proving that Algorithm 17 constructs a $\left(\delta, \frac{\delta}{4}\right)$-covering from $\mathcal{C}$ follows the exact same steps than the proof of Theorem 6.3.

We first show that $\mathcal{C}$ is a $\frac{\delta}{2}$-covering of $(X, \mathcal{F})$.

Let $F \in \mathcal{F}$ be any range of $(X, \mathcal{F})$.

- If $|F \cap N| \leq d, \exists E \in \mathcal{E}$ s.t. $F \cap N = E$.

$$\begin{aligned} & F \cap N = E \\ \Rightarrow\ & \psi(F \cap N) = \psi(E) \\ \Rightarrow\ & \psi(F \cap N) \cap N = \psi(E) \cap N \\ \Rightarrow\ & F \cap N = \psi(E) \cap N. \end{aligned}$$

The last step follows from property (i) of conforming maps.

- If $|F \cap N| > d$, by property (ii) of conforming maps, there exists $E \in 2^X, |E| \leq d$ s.t. $\psi(E) = F \cap N$.

Then, by property (i) of ψ:

$$\psi(E) \cap N = \psi(F \cap N) \cap N = F \cap N.$$

In both cases, there exists $E \in \mathcal{E}$ such that $F \cap N = \psi(E) \cap N$. Therefore by definition of ε-nets, $|\Delta(F, \psi(E))| \leq \frac{\delta}{2}$. Otherwise, N would contain a point from $\Delta(F, \psi(E))$ and we would have $F \cap N \neq \psi(E) \cap N$. That is for all $F \in \mathcal{F}, \exists E \in \mathcal{E}$ s.t. $|\Delta(F, \psi(E))| \leq \frac{\delta}{2}$ i.e. $\mathcal{C}$ is a $\frac{\delta}{2}$ -covering of size $O(|N|^d) = O\left(\left(\frac{nd}{\delta}\right)^d \log^d\left(\frac{nd}{\delta}\right)\right)$.

Runtime analysis. Since N is a $\frac{\delta}{2n}$-net for $(X, \Delta(\mathcal{F}))$, by Lemma 3.8 and Corollary 3.6, N can be obtained by uniformly sampling $O\left(\left(\frac{nd}{\delta}\right) \log\left(\frac{nd}{\delta}\right)\right)$ points from X. $\mathcal{E}$ is composed of all sets of size at most d of the power set of N, that is:

$$|\mathcal{E}| = \sum_{i=0}^{d} \binom{\frac{nd}{\delta}\log\left(\frac{nd}{\delta}\right)}{i} = O\left(\left(\frac{nd}{\delta}\right)^d \log^d\left(\frac{nd}{\delta}\right)\right).$$

thus the first loop has time complexity

$$O\left(T_d \left(\frac{nd}{\delta}\right)^d \log^d\left(\frac{nd}{\delta}\right)\right).$$

Similarly to Algorithm 14, the second loop has time complexity $O(|\mathcal{C}\|\mathcal{P}\|A|) = O(|\mathcal{E}\|\mathcal{P}\|A|)$

$$= O\left(\left(\frac{n}{\delta}\right)^{2d+2} \log^d\left(\frac{n}{\delta}\right)\right).$$

We get a total runtime for Algorithm 17 of

$$O\left(T_d \left(\frac{nd}{\delta}\right)^d \log^d\left(\frac{nd}{\delta}\right) + \left(\frac{nd}{\delta}\right)^{2d+2} \log^d\left(\frac{nd}{\delta}\right)\right).$$

□

This algorithm has, for instance, a better time complexity than Algorithm 14 for set systems defined by balls in $\mathbb{R}^d$. Smallest enclosing disk of d points in dimension d can be computed in expected time complexity $O(d^{d+2})$ [Wel91]. Therefore, we have $T_d = O(d^{d+2} + nd^3)$ for set systems spanned by balls in $\mathbb{R}^d$. This means that for δ large and fixed d, the time complexity of Algorithm 17 is better than the time complexity of Algorithm 14.

6.3.2 Projection uniformity

In this section we will show an improvement to Algorithm 14 that can be obtained for set systems that have a uniform distribution of the pairwise symmetric difference between its ranges. This proof relies on a new concept that we call *Projection Uniformity* and define below.

Definition 6.11. Let $(X, \mathcal{F})$ be a set system and $\delta \leq n$. We call the *projection uniformity of radius* δ of $(X, \mathcal{F})$

$$\mathrm{PU}_\delta = \min_{F \in \mathcal{F}} |\{G \in \mathcal{F}, \text{ s.t. } |\Delta(F, G)| \leq \delta\}|.$$

Projection uniformity of radius δ is the cardinality of the smallest ball of radius δ centered at each range of $(X, \mathcal{F})$ w.r.t. the symmetric difference metric.

The algorithm we present selects candidate ranges for our near-minimal covering simply by randomly sampling ranges of $\mathcal{F}$. This idea leads to Algorithm 18, detailed below. We will analyze the number of draw required to obtain a δ-covering depending on the projection uniformity of radius δ.

Algorithm 18: $\left(\delta, \frac{\delta}{4}\right)$-covering algorithm using projection uniformity

Input: $(X, \mathcal{F})$, δ

$\mathcal{C} \leftarrow$ uniform random sample of $\mathcal{F}$ of size $\frac{m \ln(2m)}{\mathrm{PU}_{\frac{\delta}{2}}}$
$A \leftarrow \frac{\delta}{8n}$-approximation of $(X, \Delta(\mathcal{C}))$
for $C \in \mathcal{C}$ **do**
  if $\forall R \in \mathcal{R}, |\Delta(C, R) \cap A| \geq \frac{3\delta|A|}{8n}$ **then**
    $\mathcal{R} \leftarrow \mathcal{R} \cup \{C\}$
return $\mathcal{R}$

We will prove the following theorem.

Theorem 6.12. Given $(X, \mathcal{F})$ of VC-dimension d and $\delta \leq n$. Algorithm 18 returns a $\left(\delta, \frac{\delta}{4}\right)$-covering with probability at least $\frac{1}{2}$ in time $O\left(\frac{m}{\mathrm{PU}_{\frac{\delta}{2}}}\left(\frac{n}{\delta}\right)^{d+2} \ln(m)\right)$.

Overview of the proof. For a fixed δ, we can consider the collection containing all ranges covering any range $F \in \mathcal{F}$:

$$\{G \in \mathcal{F}, \text{ s.t. } |\Delta(F, G)| \leq \delta\}.$$

A collection of ranges containing at least one range from all such collections of a set system is a δ-covering of the set system as it covers all ranges by definition. We then bound the number of ranges to draw uniformly at random to obtain at least one range from each such collections of the set system.

Proof of Theorem 6.12. The random sampling process simply replaces the covering construction step of [MWW93] in Algorithm 14 and we again only show that $\mathcal{C}$ is a $\frac{\delta}{2}$-covering of $(X, \mathcal{F})$ as the rest of the proof follows the exact same steps as the proof of Theorem 6.3.

For all $F \in \mathcal{F}$, we denote $\mathcal{B}_{\frac{\delta}{2}}(F) = \left\{G \in \mathcal{F}, \text{s.t. } |\Delta(F, G)| \leq \frac{\delta}{2}\right\}$. By definition

$$\forall F \in \mathcal{F}, |\mathcal{B}_{\frac{\delta}{2}}(F)| \geq \mathrm{PU}_{\frac{\delta}{2}}.$$

For all ranges $F \in \mathcal{F}$, the probability for a range G picked uniformly at random in $\mathcal{F}$ to be in $\mathcal{B}_{\frac{\delta}{2}}(F)$ is

$$\frac{|\mathcal{B}_{\frac{\delta}{2}}(F)|}{m} \geq \frac{\mathrm{PU}_{\frac{\delta}{2}}}{m}.$$

Let $F \in \mathcal{F}$, the probability that, after $\frac{m\ln(2m)}{\mathrm{PU}_{\frac{\delta}{2}}}$ draws, $\mathcal{C}$ contains at least a range in $\mathcal{B}_{\frac{\delta}{2}}(F)$ is

$$\left(1 - \frac{|\mathcal{B}_{\frac{\delta}{2}}(F)|}{m}\right)^{\frac{m\ln(2m)}{\mathrm{PU}_{\frac{\delta}{2}}}} \leq \left(1 - \frac{\mathrm{PU}_{\frac{\delta}{2}}}{m}\right)^{\frac{m\ln(2m)}{\mathrm{PU}_{\frac{\delta}{2}}}} \leq e^{-\ln(2m)} = \frac{1}{2m}.$$

By using a union bound over all ranges $F \in \mathcal{F}$, we obtain that $\mathcal{C}$ contains at least a range from each of the $\mathcal{B}_{\frac{\delta}{2}}(F)$ with probability at least $\frac{1}{2}$. That is, $\mathcal{C}$ is a $\frac{\delta}{2}$-covering of $(X, \mathcal{F})$ of size $\frac{m\ln(2m)}{\mathrm{PU}_{\frac{\delta}{2}}}$ with probability at least $\frac{1}{2}$.

As for Algorithm 17, the conclusion follows the exact same steps as the proof of Theorem 6.3.

Time complexity analysis. The loop of the algorithm has time complexity $O(|\mathcal{C}\|\mathcal{P}\|A|) =$

$$O\left(\frac{m}{\mathrm{PU}_{\frac{\delta}{2}}}\ln(m)\left(\frac{n}{\delta}\right)^{d+2}\right).$$

□

Remark 6.13. In the worst case, $\mathrm{PU}_{\delta} = 1$, that is, Algorithm 18 has time complexity $O\left(m\left(\frac{n}{\delta}\right)^{d+2}\ln(m)\right)$. The time complexity of Algorithm 18 is smaller than the time complexity of Algorithm 14 when

$$\mathrm{PU}_{\frac{\delta}{2}} = \Omega\left(\max\left(\frac{n^{d+1}\ln(m)}{\delta^{d+1}\log\left(\frac{n}{\delta}\right)}, \frac{m\delta^d\ln(m)}{n^d\log^d\left(\frac{n}{\delta}\right)},\right)\right).$$

Projection uniformity has never received a systematic study. However a result of Chazelle and Welzl [CW89] showed that for a set system spanned by halfspaces in $\mathbb{R}^d$,

$$\mathrm{PU}_{\frac{\delta}{2}} \geq \frac{\binom{\frac{\delta}{2}}{d}}{d!} \geq \frac{\delta^d}{d!(2d)^d}.$$

This result implies that for set systems spanned by halfspaces, Algorithm 18 has time complexity

$$O\left(\frac{m\ln(m)n^d}{\delta^{2d+2}}\right)$$

which is smaller than the time complexity of Algorithm 14 for $\delta = \Omega\left(n^{\frac{d+1}{2d+1}}\right)$.

Chapter 7

Perspectives

In this chapter, we present some open problems that might be of interest to the reader.

Discrepancy learning game.

An interesting first problem would be to close the gap between the discrepancy obtained with the strategy that we present for Alice, $O\left(n^{\frac{1}{2}-\frac{1}{2d}}\log^{\frac{5}{2}}(mn)\right)$, and the optimal bound $O\left(n^{\frac{1}{2}-\frac{1}{2d}}\right)$. The goal would be to obtain it without computing the symmetric difference between ranges given by Bob. This would probably require new ideas to compute low-discrepancy colorings for finite VC-dimension set systems that do not rely on the computation of small δ-coverings. In fact, the chaining approach does not work straightforwardly as this technique require a fine knowledge of the symmetric difference between the ranges of the set system.

Small δ-coverings.

An important problem for small δ-coverings would be to be able to construct δ-coverings of finite VC-dimension set systems of size $\tilde{O}\left(\left(\frac{n}{\delta}\right)^d\right)$ in time with sublinear dependence in m. This would open the possibility to have near-minimal algorithms to compute δ-coverings in time sublinear in m as the pruning part of the algorithm we present is already independent of m when pruning coverings of this size.

A second line of improvement for the algorithms we present is to improve the pruning process of the algorithms. The pruning process we present is relatively simple and could maybe be improved by avoiding to compute the symmetric difference between every pair of range from the pruned covering.

Finally, performing a thorough study of projection uniformity of various types of set systems in particular geometric ones could be of interest. This could lead to complexity improvements of the current δ-coverings algorithms by constructing δ-coverings using sampling.

Low-discrepancy coloring algorithms.

Low-discrepancy coloring algorithms is the most widely studied problem discussed in this thesis. However, no general algorithm with time complexity below matrix inversion time complexity is known. It would be an important improvement on current algorithms to achieve such complexity. This could open the way to near-quadratic or even linear time discrepancy algorithms.

Low-crossing partitions.

A first major open problem regarding low-crossing partitions is to find an algorithm that computes with guarantees low-crossing partitions without relying on cuttings.

Low-crossing matchings can be computed efficiently without cuttings [CM21] and are low-crossing partitions of size n/2. It would be interesting to see whether the ideas that apply to low-crossing matchings can be applied more generally to partitions of any size. Another interesting idea to explore regarding the link between low-crossing matchings and low-crossing partitions is whether it is possible to efficiently obtain low-crossing partitions given a low-crossing matching.

Finally, improving the state of current cutting algorithms either by extending them to other types of geometric set systems and to higher dimensions would improve low-crossing partitions algorithms instantly.

Bibliography

[ACL01] Aiello, William ; Chung, Fan ; Lu, Linyuan: A Random Graph Model for Power Law Graphs. In: *Experimental Mathematics* vol. 10 (2001)

[AHK12] Arora, Sanjeev ; Hazan, Elad ; Kale, Satyen: The Multiplicative Weights Update Method: a Meta-Algorithm and Applications. In: *Theory of Computing* vol. 8, Theory of Computing (2012)

[AHW87] Alon, Noga ; Haussler, David ; Welzl, Emo: Partitioning and geometric embedding of range spaces of finite Vapnik-Chervonenkis dimension. In: *Proceedings of the third annual symposium on Computational geometry*, 1987

[Ale90] Alexander, Ralph: Geometric methods in the study of irregularities of distribution. In: *Combinatorica* vol. 10, Springer (1990)

[Alo+05] Alon, Noga ; Krivelevich, Michael ; Spencer, Joel ; Szabó, Tibor: Discrepancy games. In: *the electronic journal of combinatorics* vol. 12 (2005)

[AMS13] Agarwal, Pankaj K. ; Matoušek, Jiří ; Sharir, Micha: On Range Searching with Semialgebraic Sets. II. In: *SIAM Journal on Computing* vol. 42 (2013)

[Ban+19] Bansal, Nikhil ; Dadush, Daniel ; Garg, Shashwat ; Lovett, Shachar: The Gram–Schmidt Walk: A Cure for the Banaszczyk Blues. In: *Theory of Computing* vol. 15 (2019)

[Ban10] Bansal, Nikhil: Constructive algorithms for discrepancy minimization. In: *2010 IEEE 51st Annual Symposium on Foundations of Computer Science*, 2010, pp. 3–10

[BDG19] Bansal, Nikhil ; Dadush, Daniel ; Garg, Shashwat: An Algorithm for Komlós Conjecture Matching Banaszczyk's Bound . In: *SIAM Journal on Computing* vol. 48 (2019)

[Bec81] Beck, József: Roth’s estimate of the discrepancy of integer sequences is nearly sharp. In: *Combinatorica* vol. 1 (1981)

[BLM13] Boucheron, Stéphane ; Lugosi, Gábor ; Massart, Pascal: *Concentration inequalities: A nonasymptotic theory of independence* : Oxford university press, 2013

[BM20] Bansal, Nikhil ; Meka, Raghu: On the discrepancy of random low degree set systems. In: *Random Structures & Algorithms* vol. 57 (2020)

[Cha+12] Chan, Timothy M ; Grant, Elyot ; Könemann, Jochen ; Sharpe, Malcolm: Weighted capacitated, priority, and geometric set cover via improved quasi-uniform sampling. In: *Proceedings of the twenty-third annual ACM-SIAM symposium on Discrete Algorithms*, 2012, pp. 1576–1585

[Cha00] Chazelle, Bernard: *The Discrepancy Method: Randomness and Complexity* : Cambridge University Press, 2000 — ISBN 0-521-77093-9

[Cha12] Chan, Timothy M.: Optimal Partition Trees. In: *Discrete & Computational Geometry* vol. 47 (2012)

[Cha93] Chazelle, Bernard: Cutting hyperplanes for divide-and-conquer. In: *Discrete & Computational Geometry* vol. 9 (1993)

[CM21] Csikos, Monika ; Mustafa, Nabil H.: Escaping the Curse of Spatial Partitioning: Matchings with Low Crossing Numbers and Their Applications. In: *37th International Symposium on Computational Geometry (SoCG 2021).* vol. 189, 2021

[CM22] Csikós, Mónika ; Mustafa, Nabil H.: Optimal approximations made easy. In: *Information Processing Letters* vol. 176 (2022)

[CM96] Chazelle, Bernard ; Matoušek, Jiří: On Linear-Time Deterministic Algorithms for Optimization Problems in Fixed Dimension. In: *Journal of Algorithms* vol. 21 (1996)

[CNN11] Charikar, Moses ; Newman, Alantha ; Nikolov, Aleksandar: Tight hardness results for minimizing discrepancy. In: *Proceedings of the twenty-second annual ACM-SIAM symposium on Discrete Algorithms*, 2011

[Cou+24] Coudert, David ; Csikós, Mónika ; Ducoffe, Guillaume ; Viennot, Laurent: Practical Computation of Graph VC-Dimension. In: *Symposium on Experimental Algorithms (SEA)*, 2024

[CS13] Conway, John Horton ; Sloane, Neil James Alexander: *Sphere packings, lattices and groups.* vol. 290 : Springer Science & Business Media, 2013

[CW89] Chazelle, Bernard ; Welzl, Emo: Quasi-optimal range searching in spaces of finite VC-dimension. In: *Discrete & Computational Geometry* vol. 4 (1989)

[DGL13] Devroye, Luc ; Györfi, László ; Lugosi, Gábor: *A probabilistic theory of pattern recognition* : Springer Science & Business Media, 2013

[DSW22] Deng, Yichuan ; Song, Zhao ; Weinstein, Omri: Discrepancy Minimization in Input-Sparsity Time, arXiv (2022)

[Dud78] Dudley, Richard M: Central limit theorems for empirical measures. In: *The Annals of Probability* (1978)

[Ezr16] Ezra, Esther: A size-sensitive discrepancy bound for set systems of bounded primal shatter dimension. In: *SIAM Journal on Computing* vol. 45 (2016)

[Fra12] Fraenkel, Aviezri: Combinatorial games: selected bibliography with a succinct gourmet introduction. In: *The Electronic Journal of Combinatorics* (2012)

[Gre23] Green Larsen, Kasper: Fast discrepancy minimization with hereditary guarantees. In: *Proceedings of the 2023 Annual ACM-SIAM Symposium on Discrete Algorithms (SODA)*, 2023

[Har00] Har-Peled, Sariel: Constructing Planar Cuttings in Theory and Practice. In: *SIAM J. Comput.* vol. 29 (2000)

[Hau95] Haussler, David: Sphere packing numbers for subsets of the Boolean n-cube with bounded Vapnik-Chervonenkis dimension. In: *Journal of Combinatorial Theory, Series A* vol. 69 (1995)

[HSS14] Harvey, Nicholas JA ; Schwartz, Roy ; Singh, Mohit: Discrepancy without partial colorings. In: *Approximation, Randomization, and Combinatorial Optimization. Algorithms and Techniques (APPROX/RANDOM 2014)*, 2014

[HW87] Haussler, David ; Welzl, Emo: ε-nets and simplex range queries. In: *Discrete & Computational Geometry* vol. 2 (1987)

[JSS23] Jain, Vishesh ; Sah, Ashwin ; Sawhney, Mehtaab: Spencer's theorem in nearly input-sparsity time. In: *Proceedings of the 2023 Annual ACM-SIAM Symposium on Discrete Algorithms (SODA)*, 2023

[KY14] Koufogiannakis, Christos ; Young, Neal E: A nearly linear-time PTAS for explicit fractional packing and covering linear programs. In: *Algorithmica* vol. 70 (2014)

[LK14] Leskovec, Jure ; Krevl, Andrej: SNAP Datasets: Stanford Large Network Dataset Collection. In: *http://snap.stanford.edu/data* (2014)

[LKF07] Leskovec, Jure ; Kleinberg, Jon ; Faloutsos, Christos: Graph evolution: Densification and shrinking diameters. In: *ACM transactions on Knowledge Discovery from Data (TKDD)* vol. 1 (2007)

[LLS01] Li, Yi ; Long, Philip M. ; Srinivasan, Aravind: Improved Bounds on the Sample Complexity of Learning. In: *Journal of Computer and System Sciences* vol. 62 (2001)

[LM15] Lovett, Shachar ; Meka, Raghu: Constructive Discrepancy Minimization by Walking on the Edges. In: *SIAM Journal on Computing* vol. 44 (2015)

[LRR17] Levy, Avi ; Ramadas, Harishchandra ; Rothvoss, Thomas: Deterministic Discrepancy Minimization via the Multiplicative Weight Update Method. In: *Integer Programming and Combinatorial Optimization*, 2017

[Mat13] Matoušek, Jiří: *Lectures on discrete geometry* : Springer Science & Business Media, 2013

[Mat18] Matheny, Michael: pypartition. In: *https://github.com/michaelmathen/pypartition*, GitHub (2018)

[Mat92] Matoušek, Jiří: Efficient partition trees. In: *Discrete & Computational Geometry* vol. 8 (1992)

[Mat93] Matoušek, Jiří: Range searching with efficient hierarchical cuttings. In: *Discrete & Computational Geometry* vol. 10 (1993)

[Mat95] Matoušek, Jiří: Tight upper bounds for the discrepancy of half-spaces. In: *Discrete & Computational Geometry* vol. 13 (1995)

[Mat99] Matoušek, Jiří: *Geometric discrepancy: An illustrated guide.* vol. 18 : Springer Science & Business Media, 1999

[ML12] McAuley, Julian ; Leskovec, Jure: Learning to discover social circles in ego networks. In: *Proceedings of the 25th International Conference on Neural Information Processing Systems - Volume 1, NIPS'12*, 2012

[MP18] Matheny, Michael ; Phillips, Jeff M: Practical Low-Dimensional Halfspace Range Space Sampling. In: *26th Annual European Symposium on Algorithms (ESA)*, 2018

[MP19] Matheny, Michael ; Phillips, Jeff M: Computing Approximate Statistical Discrepancy. In: *29th International Symposium on Algorithms and Computation*, 2019

[MU17] Mitzenmacher, Michael ; Upfal, Eli: *Probability and computing: Randomization and probabilistic techniques in algorithms and data analysis* : Cambridge university press, 2017

[Mus16] Mustafa, Nabil H: A simple proof of the shallow packing lemma. In: *Discrete & Computational Geometry* vol. 55 (2016)

[Mus22] Mustafa, Nabil H: *Sampling in combinatorial and geometric set systems.* vol. 265 : American Mathematical Society, 2022

[MWW93] Matousek, Jiri ; Welzl, Emo ; Wernisch, Lorenz: Discrepancy and ε-approximations for bounded VC-dimension. In: *Combinatorica* vol. 13 (1993)

[PT13] Pach, János ; Tardos, Gábor: Tight lower bounds for the size of epsilon-nets. In: *Journal of the American Mathematical Society* vol. 26 (2013)

[PW90] Pach, János ; Woeginger, Gerhard: Some new bounds for epsilon-nets. In: *Proceedings of the sixth annual symposium on Computational geometry*, 1990, pp. 10–15

[Rad21] Radon, Johann: Mengen konvexer Körper, die einen gemeinsamen Punkt enthalten. In: *Mathematische Annalen* vol. 83 (1921)

[RT87] Raghavan, Prabhakar ; Tompson, Clark D: Randomized rounding: a technique for provably good algorithms and algorithmic proofs. In: *Combinatorica* vol. 7, Springer (1987)

[Sau72] Sauer, Noel: On the density of families of sets. In: *Journal of Combinatorial Theory, Series A* vol. 13 (1972)

[She72] Shelah, Saharon: A combinatorial problem; stability and order for models and theories in infinitary languages. In: *Pacific Journal of Mathematics* vol. 41 (1972)

[Spe85] Spencer, Joel: Six standard deviations suffice. In: *Transactions of the American Mathematical Society* vol. 289 (1985)

[STZ06] Suri, Subhash ; Toth, Csaba D ; Zhou, Yunhong: Range Counting over Multidimensional Data Streams. In: *Discrete & Computational Geometry* vol. 36 (2006)

[Tal94] Talagrand, Michel: Sharper Bounds for Gaussian and Empirical Processes. In: *The Annals of Probability* vol. 22 (1994)

[VC71] Vapnik, Vladimir ; Chervonenkis, Alexey: On the Uniform Convergence of Relative Frequencies of Events to Their Probabilities. In: *Theory of Probability and its Applications* vol. 16 (1971)

[Via17] Viazovska, Maryna S: The sphere packing problem in dimension 8. In: *Annals of mathematics* (2017)

[Wel91] Welzl, Emo: Smallest enclosing disks (balls and ellipsoids). In: *New Results and New Trends in Computer Science*, 1991

[You95] Young, Neal E: Randomized rounding without solving the linear program. In: *Proceedings of the sixth annual ACM-SIAM symposium on Discrete algorithms*, 1995