Technische Universität München

TUM School of Computation, Information and Technology

Median-of-Means for Sparse Models: Fast Sparsifying Transforms and Recovery from Heavy-Tailed Measurements

Tim Fuchs

Vollständiger Abdruck der von der TUM School of Computation, Information and Technology der Technischen Universität München zur Erlangung eines

Doktors der Naturwissenschaften (Dr. rer. nat.)

genehmigten Dissertation.

Vorsitz: Prof. Dr. Christina Kuttler

Prüfende der Dissertation:

1. Prof. Dr. Felix Krahmer

2. Prof. Dr. Karin Schnass

3. Prof. Dr. Richard Kueng

Die Dissertation wurde am 24.06.2024 bei der Technischen Universität München eingereicht und durch die TUM School of Computation, Information and Technology am 02.01.2025 angenommen.

Abstract

The exponential growth in data volume, higher-resolution imaging, and the expansion of wireless communication have underscored the importance of sparse models, which enable efficient data representation and recovery by leveraging inherent sparsity.

This thesis introduces a novel algorithm for fast sparse transforms applicable to arbitrary transformation matrices and demonstrates an adaptation of compressed sensing algorithms for sparse recovery from heavy-tailed measurements. The key innovation is the use of a median-of-means estimator, which provides robustness against outliers and ensures strong concentration results.

Fast sparse transforms are essential as data dimensions and resolutions increase, enabling efficient representation of large datasets. Traditional algorithms often require specific matrix structures. In contrast, this work presents a randomized algorithm based on spherical designs that significantly reduces computational demands – even for unstructured matrices.

Wireless communication and sensor networks demand robust signal recovery from limited measurements. Traditional compressed sensing algorithms rely on the Restricted Isometry Property (RIP), which may not hold for heavy-tailed data. This thesis successfully adapts such an algorithm using the median-of-means estimator and discusses potential expansions for other recovery algorithms.

Zusammenfassung

Das exponentielle Wachstum des Datenvolumens, höher auflösende Bildgebung und die kontinuierliche Verbreitung drahtloser Kommunikation haben die Bedeutung von dünnbesetzten Modellen unterstrichen, die durch Ausnutzen dieser inhärenten Eigenschaft eine effiziente Datenrepräsentation und -wiederherstellung ermöglichen.

Diese Dissertation stellt einen neuartigen Algorithmus zur schnellen Transformation in dünnbesetzte Darstellungen für beliebige Transformationsmatrizen vor und demonstriert eine Adaption von Compressed-Sensing-Algorithmen für die Rekonstruktion dünnbesetzter Vektoren aus endlastig verteilten Messungen. Die Schlüsselinnovation ist die Verwendung eines Median-von-Mittelwerten-Schätzers, der Robustheit gegenüber Ausreißern bietet und starke Konzentrationsresultate gewährleistet.

Algorithmen zur schnellen Transformation in dünnbesetze Darstellungen sind entscheidend, da Datenmengen und -auflösungen zunehmen und sie eine effiziente Darstellung großer Datensätze ermöglichen. Traditionelle Algorithmen erfordern oft spezifische Matrixstrukturen. Im Gegensatz dazu wird in dieser Arbeit ein randomisierter Algorithmus vorgestellt, der auf sphärischen Designs basiert und die Rechenanforderungen selbst für unstrukturierte Matrizen erheblich reduziert.

Drahtlose Kommunikation und Sensornetzwerke erfordern eine robuste Signalwiederherstellung aus begrenzten Messungen. Traditionelle Compressed-Sensing-Algorithmen basieren auf der Restricted Isometry Property (RIP), die bei endlastig verteilten Daten möglicherweise nicht erfüllt ist. Diese Dissertation demonstriert die erfolgreiche Anpassung eines solchen Algorithmus mithilfe des Median-von-Mittelwerten-Schätzers und diskutiert mögliche Erweiterungen für andere Rekonstruktionsalgorithmen.

Acknowledgements

First and foremost, I need to express my deep gratitude and respect for my supervisor, Felix Krahmer. The results presented in this thesis would not have been possible without your guidance and our continuous discussions. Your infectious excitement and steady search for the next challenge helped to form both my professional but also personal development during this time. I already miss all our deep conversations and really hope we will stay in touch!

Further, I want to thank my mentor Henning Christ. Throughout my entire PhD journey, you have been a constant source of support and guidance. Knowing now your calendar, I simply do not understand how you found the time for our regular meetings. Your different perspective, both as a physicist and also as a manager at my employer of choice, was invaluable and helped dispel any concerns and find new motivation whenever it was needed.

I am also very grateful to my collaborators for their scientific expertise, valuable advice and numerous insightful discussions and ideas. In particular, to David Gross who seems to simply never stop thinking about a problem until it is solved and to Richard Kueng on his mission to spread appreciation for median-of-means – as one can see from the title of this thesis, I am fully converted!

My gratitude extends to the members of the committee: My thanks to Karin Schnass and Richard Kueng for agreeing to devote their time to be members of the committee and reviewing this thesis and to Christina Kuttler for chairing the committee. Additionally, I would like to thank all my colleagues at our research unit. I am very grateful for all the good times and social gatherings we spent together during our PhD time.

Finally, I would like to express my appreciation and gratitude to my parents, Ute and Robert, my sister, Nina, and grandparents, Hilde and Gert, for their unconditional love and support not only throughout the challenging times of my PhD but throughout my entire life. I thank you for believing in me and never doubting my career choice, even though it might not have seemed understandable to you.

Most importantly, of course, I need to thank my wife, Markéta. Your unparalleled support and never-ending belief in me is what keeps me going every day. I am so incredibly grateful to have you in my life. This thesis is dedicated to you.

Publications by the Author

[32] T. Fuchs, D. Gross, P. Jung, F. Krahmer, R. Kueng, and D. Stöger. "Proof methods for robust low-rank matrix recovery". In: *Compressed Sensing in Information Processing.* Springer, 2022, pp. 37–75.

[33] T. Fuchs, D. Gross, F. Krahmer, R. Kueng, and D. Mixon. "Sketching with Kerdock's Crayons: Fast Sparsifying Transforms for Arbitrary Linear Maps". In: *SIAM Journal on Matrix Analysis and Applications* 43.2 (2022), 939–952, © 2022 SIAM.

[34] T. Fuchs, F. Krahmer, and R. Kueng. "Greedy-type sparse recovery from heavy-tailed measurements". In: *2023 International Conference on Sampling Theory and Applications (SampTA).* 2023, 1–5, © 2023 IEEE.

[79] C. M. Verdun*, T. Fuchs*, et al. "Group testing for SARS-CoV-2 allows for up to 10-fold efficiency increase across realistic scenarios and testing strategies". In: *Frontiers in Public Health* 9 (2021), p. 583377.

Contents

1 Introduction 13

1.1 Problem Setting . . . 16

1.2 Outline . . . 18

2 Probability Theory 21

2.1 Probability Measure & Events . . . 21

2.2 Random Variables . . . 23

2.3 Concentration Bounds . . . 26

2.4 Median-of-Means . . . 29

3 Compressed Sensing 35

3.1 ℓ_0-minimization & Restricted Isometry Property . . . 35

3.2 Recovery Methods . . . 39

4 Fast Linear Transforms Using Spherical Designs 45

4.1 Two-Stage Matrix-Vector Multiplication . . . 45

4.2 Approximate Spherical Designs . . . 49

5 Sketching with Kerdock's Crayons: Fast Sparsifying Transforms for Arbitrary Linear Maps 57

5.1 Introduction . . . 57

5.1.1 Our Approach . . . 61

5.1.2 Outline . . . 65

5.2 Projective Designs and the Median-of-Means . . . 66

5.3 Fast Preprocessing with Kerdock Sets . . . 71

5.4 Numerical Results . . . 77

6 Greedy-Type Sparse Recovery from Heavy-Tailed Measurements 81

6.1 Introduction . . . 81

6.2 Median-of-Means 84

6.3 Iterative Median-of-Means Algorithm 89

6.4 Numerical Analysis 93

6.5 Conclusion 96

6.6 Iterative Median-of-Means for CoSaMP 97

7 Conclusion and Outlook **105**

Bibliography **107**

1 Introduction

In today's rapidly evolving world, mathematics is the foundation of various technological and scientific advancements. Its influence extends across a multitude of disciplines, including physics, engineering, economics, and computer science, underscoring its essential role in solving complex problems and driving innovation. From the algorithms that power search engines and social media platforms to the intricate models that predict weather patterns and financial markets, the understanding of very fundamental mathematical concepts, today, can be leveraged to significant impact.

One of the most profound contributions of mathematics is its ability to abstract and generalize real-world phenomena into comprehensible models. This allows building on well-understood theoretical results to obtain solutions that are not only theoretically sound but also practically implementable. For instance, mathematical optimization techniques have revolutionized industries by enhancing operational efficiency, while statistical methods have become indispensable for the formulation of informed decisions based on data analysis [5]. Also, very recent events as the worldwide COVID-19 pandemic could be impacted by well-established mathematical concepts, such as pandemic modeling or efficient group testing procedures [1, 79].

The exponential growth in data volume, higher-resolution imaging, and the expansion of wireless communication have underscored the importance of sparse models. These advancements necessitate efficient methods for data processing and signal recovery, as traditional approaches struggle with the increasing scale and complexity of contemporary data-driven applications. Sparse models, leveraging inherent data sparsity, offer promising solutions by enabling efficient data representation and recovery. Sparse representation techniques have become a cornerstone in many areas of data science and engineering, offering a compact and efficient means of representing large datasets. This is particularly vital in applications like medical imaging, video

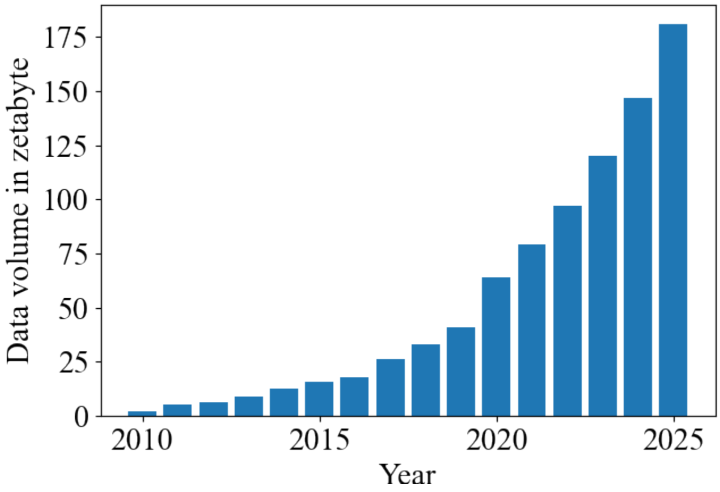

Figure 1: "Volume of data/information created, captured, copied, and consumed worldwide from 2010 to 2020, with forecasts from 2021 to 2025" [75].

processing, and sensor networks, where data volumes are vast, and computational resources are often limited [8, 55, 57]. By focusing on the significant components of data and ignoring redundant information, sparse models facilitate faster processing and more efficient storage, making them indispensable in modern computational frameworks.

A key innovation of this thesis is the use of a median-of-means estimator, which provides robustness against outliers and ensures strong concentration results. The median-of-means approach is a powerful statistical tool that offers a robust alternative to the sample mean, particularly in the presence of heavy-tailed distributions. This robustness is crucial for applications where data is noisy or contains outliers. Considering that median-of-means has been known at least since the 1980s [3, 45, 68], it appears underused for the applications outlined above.

This thesis introduces a novel algorithm for fast sparse transforms applicable to arbitrary transformation matrices, addressing the limitations of existing methods that often require specific matrix structures. Traditional algorithms like the Discrete Fourier Transform (DFT) and Chebyshev Transform are efficient but restricted to specific types of structured matrices [35, 70]. In contrast, the proposed randomized algorithm, based on spherical designs and median-of-means, significantly reduces computational demands even for unstructured matrices. This advancement allows for more flexible and efficient data processing, accommodating the diverse and often unstructured nature of real-world data.

Wireless communication and sensor networks require robust signal recovery from limited measurements. In compressed sensing, the goal is to recover sparse signals from a small number of linear measurements. Traditional compressed sensing algorithms rely on the Restricted Isometry Property (RIP), which ensures that the measurement matrix preserves the geometry of sparse signals [30]. However, RIP may not hold for heavy-tailed data. This thesis demonstrates the successful adaptation of compressed sensing algorithms using the median-of-means estimator, offering robust recovery with high probability even for heavy-tailed matrices and requiring only a bounded fourth moment. This adaptation extends the applicability of compressed sensing techniques to a broader range of practical scenarios where traditional assumptions do not hold.

Overall, this thesis advances methodologies for fast sparsifying transforms and signal recovery for heavy-tailed measurements, addressing the critical needs of modern applications. The proposed approaches are poised to have a significant impact on various fields, ensuring efficient data processing and robust recovery in increasingly complex and large-scale settings.

1.1 Problem Setting

The core of this thesis is two publications addressing an extension of recovery algorithms for compressed sensing [33] and a randomized algorithm for fast sparsifying transforms [34]. Both problem settings address the issue of obtaining a sparse vector (i.e., a vector containing only a small number of non-zero entries) based on a given matrix and vector – once through solving an underdetermined linear system, once through efficient randomized matrix-vector multiplication.

Compressed Sensing

The basic underlying problem of compressed sensing can be formulated in the following way: An unknown vector $x \in \mathbb{C}^n$ has to be recovered from the m-dimensional measurement vector $y := Ax$. In most applications, the dimension of the measurement vector is significantly smaller than the dimension of x, i.e., $m < n$. In general, such an under-determined system has an infinite number of solutions which makes the recovery of the original x impossible. This changes if x is sparse.

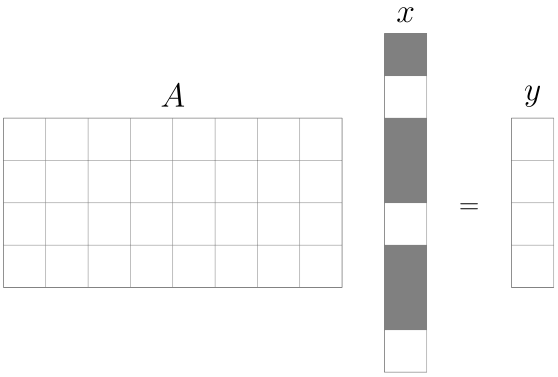

Figure 2: Example of a typical compressed sensing sparse recovery problem. The filled grey tiles symbolize zero entries.

Applications have one thing in common: measuring and directly storing the large vector x is either not possible (e.g., medical imaging) or not desired (e.g., storing photos). For completeness, there are various other – more complex – problem settings that belong to the broad field of compressed sensing (e.g., [30, 32]). This thesis focuses on the most fundamental case outlined above.

Sparsifying Transforms

The problem setting of developing an algorithm for a fast sparsifying transform seems very similar to compressed sensing. Given a matrix $A \in \mathbb{R}^{m \times n}$ and a stream of vectors $x_1, x_2, \dots \in \mathbb{R}^n$ for which $y_i = Ax_i$ is s-sparse for every i, an efficient algorithm for computing those sparse vectors has to be constructed.

In contrast to compressed sensing, it is not necessary to solve an (underdetermined) linear equation, but instead perform simple matrix-vector multiplication. As this requires significant computational effort for very large matrices, various publications are establishing fast algorithms that take advantage of a specific required structure of the underlying matrix A (e.g., Fourier or Chebyshev transforms [35, 70]).

When considering important constructions of such matrices, e.g., sparse dictionary learning, it becomes clear that such a specific structure of A cannot be assumed in general [2]. Our publication [33] addresses this problem of introducing a fast sparsifying transform that does not require any specific structure and demonstrates superior efficiency compared to the standard matrix-vector multiplication.

1.2 Outline

Probability Theory

In this section, basic concepts of probability theory are recalled and the most important concentration inequalities are introduced. Additionally, the concept of median-of-means, which is the main underlying tool of this thesis, is explained in detail.

Compressed Sensing

Compressed sensing is the main motivation behind publication [34] and related ideas which are discussed in section 6. Therefore, an overview of the problem setting with a focus on already-established recovery methods is provided.

Fast Linear Transforms Using Spherical Designs

This section introduces the initial underlying idea of a two-stage algorithm based on spherical designs which was later revised leading to [33]. The main contribution is the proof of a randomized construction of approximate spherical designs with arbitrary order and dimension.

Sketching with Kerdock's Crayons: Fast Sparsifying Transforms for Arbitrary Linear Maps

This section is based on publication [33] with appropriate adaptations. The main goal of this work was to establish a fast sparsifying transform that not only shows convincing theoretical results but also exhibits faster computational runtime for real-world scenarios. This was achieved by applying the strong concentration properties of the median-of-means estimator and efficiently utilizing the sparsity.

Greedy-Type Sparse Recovery from Heavy-Tailed Measurements

This section is based on publication [34] with appropriate adaptations. Commonly used recovery algorithms in compressed sensing are typically based on iteratively computing a sample mean to obtain an approximation of the sparse original vector. Consequently, for weakly concentrating measurement matrices (e.g., sampled from a heavy-tailed distribution), it is difficult to establish strong recovery guarantees. Leveraging insights gained into the median-of-means estimator, we constructed an iterative algorithm allowing the recovery from heavy-tailed matrices. In addition to the results of the publication, an adapted version of the Compressive Sampling Matching Pursuit (CoSaMP) algorithm based on median-of-means is discussed.

2 Probability Theory

In the following, some selected concepts of probability theory will be introduced. Sections 2.1-2.3 are based on [82] which is a good source for a more extensive overview. If not cited otherwise, the presented results have been adapted from there. In section 2.4, the concept of median-of-means is explained via a generalized form of the formulation and proofs in [33] and [34].

2.1 Probability Measure & Events

Prior to delving into specific results, appropriate wording has to be established. The set containing all possible outcomes of a random experiment is called sample space and is denoted by Ω. Subsets $A, B \subseteq \Omega$ are called events.

Definition 2.1 (Probability measure)**.** *A probability measure $\mathbb{P}$ is a real function assigning probabilities in $[0,1]$ to any event $A \subseteq \Omega$ and further fulfills $P(\Omega) = 1$ and*

$$\mathbb{P}\left(\bigcup_{i=1}^{m} A_i\right) = \sum_{i=1}^{m} \mathbb{P}(A_i) \qquad \forall A_1, \ldots, A_m \subseteq \Omega \text{ with } A_i \cap A_j = \emptyset \quad \forall i \neq j.$$

This directly implies how to obtain the probability of the union of non-disjoint sets. The union of A and B can simply be split into three disjoint sets

$$A \cup B = [A \setminus (A \cap B)] \quad \cup \quad [B \setminus (A \cap B)] \quad \cup \quad [A \cap B]\,.$$

As $\mathbb{P}(A) = \mathbb{P}(A \setminus (A \cap B)) + \mathbb{P}(A \cap B)$, this results in the general formula

$$\mathbb{P}(A \cup B) = \mathbb{P}(A) + \mathbb{P}(B) - \mathbb{P}(A \cap B).$$

A direct consequence of this result is the union bound which is a fundamental tool used throughout this thesis:

Corollary 2.2 (Union bound for events)**.** *For* $A_1, \dots, A_m \subseteq \Omega$,

$$\mathbb{P}\left(\bigcup_{i=1}^{m} A_i\right) \leq \sum_{i=1}^{m} \mathbb{P}(A_i)$$

The last important concept which has to be introduced is the independence of events:

Definition 2.3. *Two events A and B are called independent if*

$$\mathbb{P}(A \cap B) = \mathbb{P}(A)\mathbb{P}(B).$$

Intuitively, two events are independent if the occurrence of one event does not impact the probability of the occurrence of the other event.

2.2 Random Variables

For most applications, it is beneficial to consider random variables, which are mappings between sample space and $\mathbb{R}$ or $\mathbb{C}$. To keep this introduction focused on the relevant basics, technicalities like measurability will be assumed and not explained in this context. Further, all random variables are considered to be real for simplicity. An extension of the relevant concepts to the complex space will be shown in section 6.

Each random variable X has a probability distribution defined by a non-negative density function $f_X(x)$, fulfilling $\mathbb{P}(X = x) = f_X(x)$ for discrete random variables and $\mathbb{P}(a \leq X \leq b) = \int_a^b f_X(x)dx$ if X is continuous.

Expectation

As with the density, the expectation of a random variable is defined separately for discrete and continuous random variables:

Definition 2.4 (Expectation)**.** *Let X be a random variable with density $f_X(x)$. Then, the expectation of the discrete/continuous random variable is defined as*

$$\mathbb{E}[X] = \begin{cases} \sum_x x f_X(x) & \textit{(discrete)} \\ \int_{-\infty}^{\infty} x f_X(x)dx & \textit{(continuous).} \end{cases}$$

Similarly, the expectation of $g(X)$ can be computed as

$$\mathbb{E}[g(X)] = \begin{cases} \sum_x g(x) f_X(x) & \textit{(discrete)} \\ \int_{-\infty}^{\infty} g(x) f_X(x)dx & \textit{(continuous).} \end{cases}$$

Based on this definition, known properties of the integral (and countable sum) can be applied directly to the expectation:

Property 2.5 (Linearity)**.** *Let $a_1, \dots, a_m$ be constants and $X_1, \dots, X_m$ random variables. Then,*

$$\mathbb{E}\left[\sum_{i=1}^{m} a_i X_i\right] = \sum_{i=1}^{m} a_i \mathbb{E}[X_i].$$

Property 2.6 (Jensen's inequality)**.** *Let g be a convex function. Then,*

$$\mathbb{E}[g(X)] \geq g(\mathbb{E}[X]).$$

If two random variables X and Y are independent, their joint density function fulfills $f_{X,Y}(x, y) = f_X(x) f_Y(y)$ which implies:

Property 2.7. *Let $X_1, \dots, X_m$ be independent. Then,*

$$\mathbb{E}[\Pi_{i=1}^{m} X_i] = \Pi_{i=1}^{m} \mathbb{E}[X_i].$$

Variance

Definition 2.8 (Variance)**.** *The variance of a random variable X with finite expectation is defined as*

$$\mathbb{V}(X) = \mathbb{E}[(X - \mathbb{E}[X])^2] = \mathbb{E}[X^2] - \mathbb{E}[X]^2.$$

$\mathbb{V}(X) \geq 0$ holds for any random variable, as $(X - \mathbb{E}[X])^2$ is obviously non-negative.

Property 2.9. *Let $a_1, \dots, a_m$ be constants and $X_1, \dots, X_m$ independent random variables. Then,*

$$\mathbb{V}\left[\sum_{i=1}^{m} a_i X_i\right] = \sum_{i=1}^{m} a_i^2 \mathbb{V}[X_i].$$

Heavy-Tailed Distributions

For further information about the distributions mentioned in this thesis, [82] remains a good reference. Nevertheless, due to its importance for the presented work and its less-known definition, the concept of heavy-tailed distributions is briefly introduced.

Definition 2.10 (Heavy-tailed distribution [54])**.** *A distribution is called heavy-tailed if* $\mathbb{E}[e^{tX}] = \infty$ *for any* $t > 0$.

Common examples of this type of distribution are the Student's t or log-normal distribution. As the name 'heavy-tailed' suggests, it exhibits very slowly decaying tails of the density function and is, therefore, prone to outliers when sampling from this distribution. A density plot demonstrating this behavior can be seen in Figure 3.

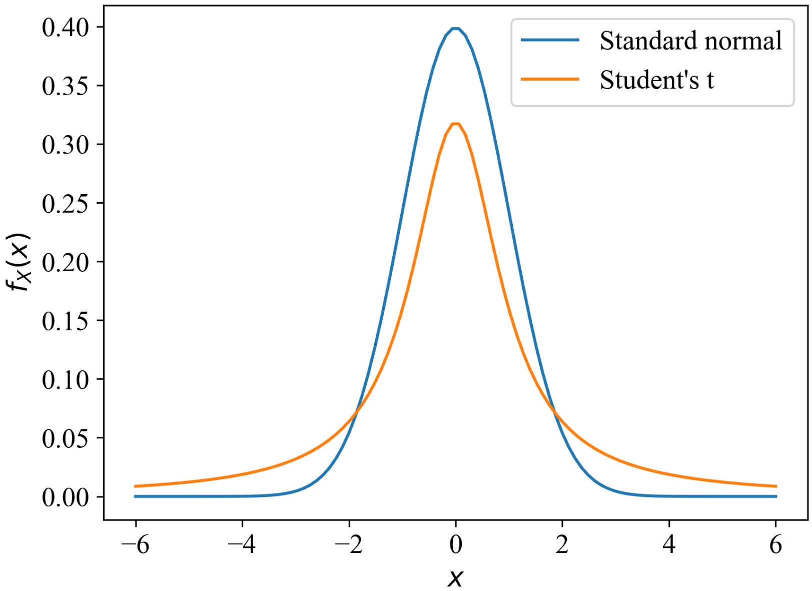

Figure 3: Comparison of the density function of a standard normal and Student's t ($df = 1$) random variable.

2.3 Concentration Bounds

In this subsection, various concentration bounds are discussed which are the foundation of nearly every result of this thesis. The quantity of interest is $\mathbb{P}(X \geq \gamma)$ which has to be bounded from above. Depending on the level of knowledge about the distribution of X, different strategies for bounding the tails of this distribution are available. Typically, stricter requirements on the distribution allow for the construction of a tighter bound. As there are exceptions to this rule and as the availability of information about the underlying distribution can be limited, a broad range of bounds finds application in this thesis.

Lemma 2.11 (Markov's inequality)**.** *Let X be a non-negative random variable. Then,*

$$\mathbb{P}(X \geq \gamma) \leq \frac{\mathbb{E}[X]}{\gamma} \qquad \forall \gamma > 0.$$

Markov's inequality is the basis for a variety of tail bounds, as any (even non-negative) random variable X can be substituted by $Y = \phi(X)$ for any non-negative, strictly increasing function ϕ resulting in

$$\mathbb{P}(X \geq \gamma) = \mathbb{P}(\phi(X) \geq \phi(\gamma)) \leq \frac{\mathbb{E}[\phi(X)]}{\phi(\gamma)} \qquad \forall \gamma \text{ s.t. } \phi(\gamma) > 0.$$

Some notable examples are Chebyshev's inequality

$$\mathbb{P}(|X - \mathbb{E}[X]| \geq \gamma) = \mathbb{P}(|X - \mathbb{E}[X]|^2 \geq \gamma^2) \leq \frac{\mathbb{V}(X)}{\gamma^2},$$

and the Chernoff bound

$$\mathbb{P}(X \geq \gamma) = \mathbb{P}(e^{tX} \geq e^{t\gamma}) \leq \frac{\mathbb{E}[e^{tX}]}{e^{t\gamma}} \qquad t > 0.$$

A major advantage of the aforementioned results above is that they are universally applicable – requiring only the expectation or variance of the underlying distribution. The Chernoff bound already allows for possibly tighter results by constructing a bound based on the moment generating function $\mathbb{E}[e^{tX}]$ of the distribution of X. Focusing now on the sum of Bernoulli random variables and using their specific structure, an even tighter bound can be established:

Lemma 2.12 (Multiplicative Chernoff bound [64])**.** *Let $X_1, \dots, X_m$ be independent Bernoulli random variables with $\mathbb{P}(X_i = 1) =: p_i := 1 - \mathbb{P}(X_i = 0)$. Then, for all $\lambda > 0$*

$$\mathbb{P}\left(\sum_{i=1}^{m} X_i \geq (1+\lambda) \sum_{i=1}^{m} p_i\right) \leq \left(\frac{e^{\lambda}}{(1+\lambda)^{1+\lambda}}\right)^{\sum_{i=1}^{m} p_i}.$$

Proof. The moment generating function of $\sum_{i=1}^{m} X_i$ can be calculated directly. The bound is then a simple property of the exponential function:

$$\begin{aligned}\mathbb{E}[e^{t\sum_{i=1}^{m} X_i}] &= \prod_{i=1}^{m} (1 - p_i + p_i e^t) \\ &\leq \prod_{i=1}^{m} \left(e^{-p_i + p_i e^t}\right) = \prod_{i=1}^{m} \left(e^{(e^t - 1)}\right)^{p_i} = \left(e^{(e^t-1)}\right)^{\sum_{i=1}^{m} p_i}.\end{aligned}$$

Applying now the Chernoff bound and the inequality above, the proof is nearly completed:

$$\mathbb{P}\left(\sum_{i=1}^{m} X_i \geq (1+\lambda) \sum_{i=1}^{m} p_i\right) \leq \frac{\mathbb{E}[e^{t\sum_{i=1}^{m} X_i}]}{e^{t(1+\lambda)\sum_{i=1}^{m} p_i}} \leq \left(\frac{e^{(e^t-1)}}{e^{t(1+\lambda)}}\right)^{\sum_{i=1}^{m} p_i}.$$

By choosing $t = \log(1+\lambda)$, the result follows. □

Hereafter, typically the formulation for independent identically distributed (i.i.d.) Bernoulli random variables will be used:

Corollary 2.13. *Let $X_1, \dots, X_m$ be i.i.d. Bernoulli random variables with $\mathbb{P}(X_i = 1) =: p := 1 - \mathbb{P}(X_i = 0)$. Then, for all $\lambda > 0$*

$$\mathbb{P}\left(\sum_{i=1}^{m} X_i \geq (1+\lambda)mp\right) \leq \left(\frac{e^{\lambda}}{(1+\lambda)^{1+\lambda}}\right)^{mp}.$$

2.4 Median-of-Means

As presented in the following sections, various problem settings can be reduced to efficiently estimating the mean of a random variable. A common approach involves taking multiple independent copies of this random variable and computing their sample mean. In comparison, the median-of-mean estimator is introduced, which appears to have been discovered independently in the following three works [3, 45, 68]. However, for consistency, a generalized form of the formulation and proofs in [33] and [34] is shown.

Definition 2.14 (Sample Mean)**.** *Let $X_1, \dots, X_m$ be independent identically distributed copies of a random variable X with finite mean and variance. The sample mean is defined as*

$$\bar{X} = \frac{1}{m}\sum_{i=1}^{m} X_i.$$

Remark 2.15. *The expectation and variance of $\bar{X}$ can be computed as follows:*

- $\mathbb{E}[\bar{X}] = \mathbb{E}[X]$
- $\mathbb{V}(\bar{X}) = \frac{1}{m^2}\mathbb{V}(\sum_{i=1}^{m} X_i) = \frac{1}{m^2}\sum_{i=1}^{m}\mathbb{V}(X_i) = \frac{1}{m}\mathbb{V}(X)$

Summarized, $\bar{X}$ preserves the mean of X while reducing the variance. This scaling in $\frac{1}{m}$ can be beneficial when using tail bounds like Chebyshev's inequality (which, as mentioned before, follows directly from Markov's inequality):

Lemma 2.16 (Chebyshev's inequality)**.** *Let $X_1, \dots, X_m$ be independent identically distributed copies of a random variable X with finite mean and variance and $\bar{X}$ their sample mean. Then, for all $\gamma > 0$*

$$\mathbb{P}(|\bar{X} - \mathbb{E}[X]| \geq \gamma) \leq \frac{\mathbb{V}(\bar{X})}{\gamma^2} = \frac{\mathbb{V}(X)}{m\gamma^2}.$$

Depending on the distribution of X, this bound might not be sufficient for the corresponding problem setting, e.g., if a bound of the type

$$\mathbb{P}(|\bar{X} - \mathbb{E}[X]| \geq \gamma) \leq \frac{\eta}{n}$$

is desired. To achieve this with Chebyshev's inequality,

$$m \in \mathcal{O}(\frac{n}{\gamma^2})$$

would be required, assuming $\mathbb{V}(X)$ does not scale with n.

For well-concentrating distributions (e.g., the normal distribution), stronger concentration bounds than Chebyshev's inequality can be considered. For heavy-tailed distributions or in case of limited knowledge about the distribution, the sample mean might simply not concentrate well enough around its mean. For such situations, substituting the sample mean with the more robust median-of-means approach might be advisable.

Definition 2.17 (Median). *Let $x_1, \ldots, x_m \in \mathbb{R}$ be a sorted set of real numbers (i.e., $x_1 \leq \cdots \leq x_m$), then the median is defined as*

$$median(x_1, \ldots, x_m) = \begin{cases} x_{\frac{m+1}{2}} & \text{if } m \text{ is odd} \\ \frac{1}{2}(x_{\frac{m}{2}} + x_{\frac{m}{2}+1}) & \text{if } m \text{ is even.} \end{cases}$$

The general idea is quite intuitive. In contrast to the sample mean, the median is very robust against outliers. Nevertheless, the median does not need to converge towards the mean of the distribution for $m \to \infty$ (e.g., the Bernoulli distribution has a mean of p, yet an odd number of samples will always be 0 or 1).

Combining the advantages of sample mean and median, one splits the m samples into K subsamples of size J (i.e., $m = KJ$). First, the sample mean over the J

samples is calculated for each of the K subsets.

$$\underbrace{\begin{bmatrix} X_1 \\ \vdots \\ X_J \end{bmatrix}}_{\Rightarrow \bar{X}_1} \quad \underbrace{\begin{bmatrix} X_{J+1} \\ \vdots \\ X_{2J} \end{bmatrix}}_{\Rightarrow \bar{X}_2} \quad \cdots \quad \underbrace{\begin{bmatrix} X_{m-J+1} \\ \vdots \\ X_m \end{bmatrix}}_{\Rightarrow \bar{X}_K}$$

The median-of-means estimator is then defined as the median of those K means:

$$\hat{\mu} := \text{median}\left(\bar{X}_1, \ldots, \bar{X}_K\right).$$

Using a generalized version of the result and proof in [34], it can be shown that the median-of-means estimator exhibits a significantly stronger scaling compared to Lemma 2.16.

Lemma 2.18. *Let $X_1, \ldots, X_m$ be independent identically distributed copies of a random variable X with finite mean and variance. Then, the median-of-means estimator $\hat{\mu}$, defined as*

$$\hat{\mu} = \textit{median}\{\bar{X}_1, \ldots, \bar{X}_K\} \quad \textit{with } \bar{X}_k = \frac{1}{J}\sum_{i=1}^{J} X_{i+(k-1)J} \textit{ for } k \in [K],$$

fulfills

$$\mathbb{P}(|\hat{\mu} - \mathbb{E}[X]| \geq \gamma) \leq e^{-K/2}$$

if $J \geq \frac{2e^2\mathbb{V}(X)}{\gamma^2}$.

Corollary 2.19. *Given the assumptions of Lemma 2.18,*

$$\mathbb{P}(|\hat{\mu} - \mathbb{E}[X]| \geq \gamma) \leq \frac{\eta}{n}$$

holds for $m \in \mathcal{O}(\frac{\log(n)}{\gamma^2})$.

Proof. Choosing $K = 2\log(\frac{n}{\eta})$ results in a bound $e^{-K/2} = e^{\log(\frac{\eta}{n})} = \frac{\eta}{n}$.
As $J \in \mathcal{O}(\frac{1}{\gamma^2})$ by assumption and $K \in \mathcal{O}(\log(n))$, $m = JK$ grants the required scaling of the number of measurements. □

Proof of Lemma 2.18. As the median-of-means combines sample mean and median, the proof is based on combining two separate bounds on both estimators:

- Chebyshev's inequality for a tail bound of the sample mean
- The multiplicative Chernoff bound for a tail bound of the median

By the properties of the variance,

$$\mathbb{V}[\bar{X}] = \frac{1}{J^2}\sum_{j=1}^{J}\mathbb{V}[X_j] = \frac{\mathbb{V}(X)}{J}.$$

Applying Chebyshev's inequality yields

$$p_J := \mathbb{P}(|\bar{X} - \mathbb{E}[X]| \geq \gamma) \leq \frac{\mathbb{V}[\bar{X}]}{\gamma^2} \leq \frac{\mathbb{V}(X)}{J\gamma^2}.$$

Based on this bound of the sample means, the median can now be bounded and both results combined. For every $k \in [K]$, define the Bernoulli random variable

$$I_k := \mathbf{1}\{|\bar{X}_k - \mathbb{E}[X]| \geq \gamma\}$$

with parameter p_J. By the definition of the median, $|\hat{\mu} - \mathbb{E}[X]| \geq \gamma$ can only occur if $|\bar{X}_k - \mathbb{E}[X]| \geq \gamma$ for at least half of the $\bar{X}_k$. Therefore,

$$\mathbb{P}\left(|\hat{\mu} - \mathbb{E}[X]| \geq \gamma\right) \leq \mathbb{P}\left(\sum_{k=1}^{K} I_k \geq \frac{K}{2}\right).$$

Applying the multiplicative Chernoff bound yields $\lambda > 0$

$$\begin{aligned}\mathbb{P}\left(\sum_{k=1}^{K} I_k \geq (1+\lambda)Kp_J\right) &\leq \left(\frac{e^{\lambda}}{(1+\lambda)^{1+\lambda}}\right)^{Kp_J} \\ &= e^{-Kp_J}\left(\frac{e}{1+\lambda}\right)^{(1+\lambda)Kp_J}.\end{aligned}$$

By assumption, $J \geq \frac{2e^2\mathbb{V}(X)}{\gamma^2}$ and therefore $p_J \leq \frac{1}{2e^2}$. Choosing $(1+\lambda)Kp = K/2$ concludes the proof:

$$\begin{aligned}\mathbb{P}(|\hat{\mu}_i - x_i| \geq \gamma) &\leq e^{-Kp_J}(2ep_J)^{K/2} \leq (2ep_J)^{K/2} \\ &\leq \left(\frac{2e\mathbb{V}(X)}{J\gamma^2}\right)^{K/2} \leq e^{-K/2}.\end{aligned}$$

□

The strong concentration of the median-of-means estimator for the example above ($\mathcal{O}(\frac{\log(n)}{\gamma^2})$ compared to $\mathcal{O}(\frac{n}{\gamma^2})$ for the sample mean) is the foundation of publications [33] and [34].

3 Compressed Sensing

Compressed sensing is a signal recovery technique with various applications, including image processing, medical imaging, and telecommunication. It tackles the problem of recovering a large amount of data from a comparably small amount of measurements. This requirement naturally arises whenever a precise measurement is impossible or if a small, compressed file should be stored to optimize storage space. This section will give a rough overview of the problem and existing recovery methods. For a more extensive overview, [30] can be recommended which serves as the groundwork of this section. If not cited otherwise, the presented results have been adapted from there.

3.1 ℓ_0-minimization & Restricted Isometry Property

Arguably, the most simple form of such a problem is recovering an unknown vector $x \in \mathbb{C}^n$ from a measurement vector $y := Ax \in \mathbb{C}^m$ with a known measurement matrix $A \in \mathbb{C}^{m \times n}$. In general, for $m < n$, this problem is underdetermined. In such a case, there is not exactly one unique solution $x^{\#}$ fulfilling the equation and, consequently, the recovery of the original unknown vector is impossible (visualized in Figure 4).

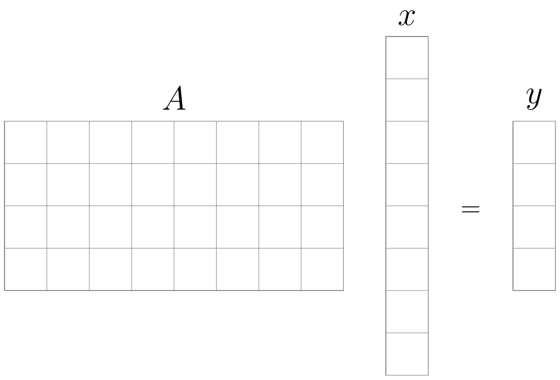

Figure 4: Example of an underdetermined recovery problem.

This potentially changes if x is a sparse vector:

Definition 3.1 (Sparsity)**.** *A vector $x \in \mathbb{C}^n$ is called s-sparse if at most s entries are not zero. The support (i.e., the index set of all non-zero entries) is usually denoted by S and fulfilles $|S| =: \|x\|_0 \leq s$.*

If the vector x is s-sparse with a known support set S and $s \leq m$, $y = A_S x_S$ can be efficiently solved with basic techniques (visualized in Figure 5).

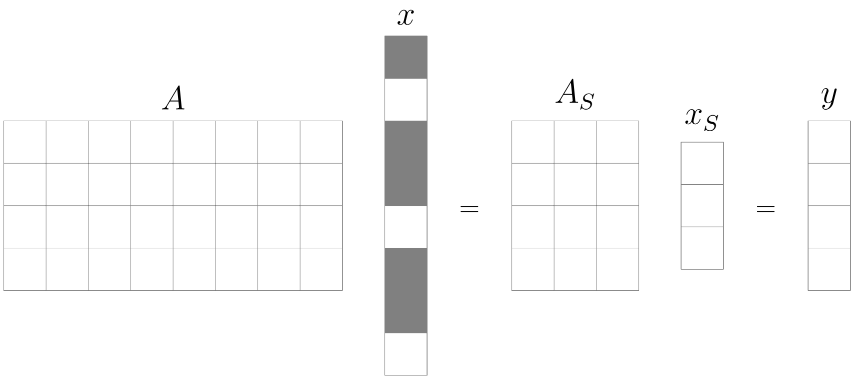

Figure 5: Example of a typical compressed sensing sparse recovery problem.
The filled grey tiles symbolize zero entries. A_S (resp. x_S) is the matrix A (resp. x) restricted to the columns (resp. entries) of the support set S of x.

In the field of compressed sensing, typically, only the sparsity level s but not the corresponding index set S are known. However, solving $y = A_S x_S$ for all possible sets $|S| \leq s$ is computationally expensive. In addition, there may be multiple solutions for different (or even the same) support sets with $|S| \leq s$.

While the problem of identifying **all** s-sparse vectors x fulfilling $Ax = y$ is also of interest, in this thesis, there is a unique s-sparse vector x that needs to be recovered. This task can be broken down into two fundamental questions:

- Which measurement matrices A allow for unique sparse recovery?

- How can an efficient recovery algorithm be constructed?

ℓ_0-minimization

As mentioned, there are potentially infinitely many vectors x fulfilling $Ax = y$. Intuitively, unique sparse recovery requires the ground truth $x^{\#}$ to be the only feasible solution which is s-sparse (i.e., all other feasible solutions have more than s non-zero entries).
In other words, the optimization problem

$$\begin{aligned} &\min \|x\|_0 \\ s.t. \qquad &Ax = y \end{aligned} \tag{P_0}$$

is required to have the unique solution $x^{\#}$, where $\|x\|_0$ denotes the number of non-zero entries of x.

Requiring a specific property of A which guarantees a unique solution of P_0 is, unfortunately, not a promising approach. Instead, it can be shown that, despite its notation, $\|x\|_0$ is not only no norm but, more importantly, non-convex. Further, it can be shown that solving P_0 is NP-hard (i.e., there potentially is no polynomial-time algorithm solving this problem). Therefore, the construction of an efficient algorithm for solving P_0 does not seem feasible and, therefore, is not the basis of well-known compressed sensing algorithms.

Before diving into such established recovery methods, the main condition for answering the first question, 'Which measurement matrices A allow for unique sparse recovery', has to be stated.

Restricted Isometry Property

The arguably most important property in this field is the so-called Restricted Isometry Property (RIP):

Definition 3.2 (Restricted Isometry Property)**.** *The RIP-constant δ_s of a matrix $A \in \mathbb{C}^{m\times n}$ is the smallest $\delta_s \geq 0$ fulfilling*

$$(1-\delta_s)\|x\|_2^2 \leq \|Ax\|_2^2 \leq (1+\delta_s)\|x\|_2^2$$

for all s-sparse vectors $x \in \mathbb{C}^n$.

As demonstrated in the following subsection, a sufficiently small RIP constant δ_{ks} allows establishing recovery guarantees for various methods.

Considering, that measurement matrices are typically assumed to be randomly chosen, there are various publications with results for different distributions (referring again to [30] and references therein for an overview). For orientation, a simplified version of the result for Gaussian matrices is stated.

Theorem 3.3. *Let $A \in \mathbb{R}^{m\times n}$ be a Gaussian matrix with $m < n$. Then, A fulfills the RIP with $\delta_s \leq \delta$ with probability at least $1-\epsilon$ if*

$$m \geq C\delta^{-2}\left(s\log\left(\frac{en}{s}\right) + \log\left(\frac{2}{\epsilon}\right)\right).$$

Similar, slightly weaker, results could be established for the more general group of subgaussian distributions. For heavy-tailed distributions, on the other hand, the proof of the RIP is difficult and does not exhibit a comparable scaling [52]. This is the motivation for the work presented in section 6.

3.2 Recovery Methods

Most recovery methods can be assigned to one of the following three categories:

- Basis Pursuit
- Greedy Algorithms
- Thresholding

As mentioned before, the results are based on [30] where detailed proofs and further references can be accessed.

Basis Pursuit

Candès, Romberg, and Tao [15] and Donoho [26] published two papers revolutionizing the field of compressed sensing by shifting the focus from P_0 to the optimization problem

$$\begin{aligned} &\min \|x\|_p \qquad\qquad (P_p) \\ s.t. \qquad &Ax = y. \end{aligned}$$

The underlying intuition is that $\|x\|_p^p \to \|x\|_0$ for $p \to 0$. Therefore, $\|x\|_0$ can be approximated by $\|x\|_p^p$. Yet, for $p < 1$, the optimization problem P_p is again non-convex. For $p > 1$, the problem is convex, but the sparsest feasible vector possibly is not a minimizer anymore. For example,

$$\begin{pmatrix} 1 & 2 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \end{pmatrix} = 2$$

can be solved by all vectors $x_\alpha := (\alpha, 1 - \frac{\alpha}{2})^T$ with $\alpha \in \mathbb{C}$. The vector $x^{\#} = (0,1)^T$ is unique minimizer for both P_0 and P_1. However, for $p > 1$, there are $\alpha > 0$ with $\|x_\alpha\|_p < \|x_0\|_p$. Consequently, there is a 2-sparse vector that is a minimizer of P_p

and not the 1-sparse vector x_0.

Based on this observation, the Basis Pursuit has been established which corresponds to solving the ℓ_1-minimization problem P_1 to recover x. Similarly, to the idea outlined for P_0, the following recovery guarantee in terms of RIP can be shown:

Theorem 3.4. *If a matrix $A \in \mathbb{C}^{m\times n}$ fulfills the RIP with $\delta_{2s} < \frac{1}{3}$, then any s-sparse vector $x \in \mathbb{C}^n$ is the unique solution of*

$$\begin{aligned} &\min \|x\|_1 \\ s.t. \qquad &Ax = y. \end{aligned} \tag{P_1}$$

In other words, by solving the convex ℓ_1-minimization problem, the s-sparse vector x can be recovered as its unique solution. Due to the convexity, a variety of optimization algorithms is available to efficiently solve P_1 and recover x.

Greedy Algorithms

A typically more efficient group of recovery algorithms is the so-called greedy algorithms. They are motivated by the idea that A^*A is sufficiently close to the identity I_n such that the largest entries of A^*Ax are a good indicator for non-zero entries of x. Typically, greedy algorithms start with an empty support set $S^{(0)}$ which is filled iteratively. After updating the support set, a new approximation is obtained via least squares minimization over the fixed support set. This is based on the observation stated at the beginning of this section: $A_S x_S = y$ is not underdetermined and can be solved via basic techniques – provided the support set S is known. Since $S^{(k)}$ may not be the correct support, $A_S x_S = y$ might potentially lack a solution and is, therefore, substituted by $\min \|A_S x_S - y\|_2$.

For OMP, first analyzed for sparse recovery in [77], a guaranteed recovery after a fixed number of iterations can be shown as long as a strong RIP is fulfilled.

Theorem 3.5. *If a matrix $A \in \mathbb{C}^{m \times n}$ fulfills the RIP with $\delta_{13s} < \frac{1}{6}$, then any s-sparse vector $x \in \mathbb{C}^n$ can be recovered by OMP within $12s$ iterations.*

Algorithm 1 Orthogonal Matching Pursuit (OMP)

```
Data: Matrix $A \in \mathbb{C}^{m \times n}$, measurement $y \in \mathbb{C}^m$
Result: $s$-sparse approximation of vector $x \in \mathbb{C}^n$
$S^{(0)} = \varnothing$
$x^{(0)} = 0$
for $k$ in $1, \ldots, \bar{n}$ do
    $S^{(k)} = S^{(k-1)} \cup \operatorname{argmax}_{i \in [n]} |A^*(y - Ax^{(k-1)})|$
    $x^{(k)} = \operatorname{argmin}_{x \in \mathbb{C}^n} \{(\|Ax - y\|)_2, \operatorname{supp}(x) \subseteq S^{(k)}\}$
end for
```

In every iteration, OMP only adds a single new index to the support set S^k which will never be removed again. That implies, that once an incorrect index gets added to the support set, a recovery of x is not possible within s iterations – explaining the large number of required iterations in the theorem.

CoSaMP, introduced in [67], is another greedy algorithm that circumvents this issue by constructing the new $S^{(k)}$ by combining the support of the last approximation $x^{(k)}$ with the $2s$ largest entries of $A^*(y - Ax^{(k-1)})$ (defined as $L_{2s}(A^*(y - Ax^{(k-1)}))$). In addition, a hard thresholding operator of order s (denoted by H_s) is applied at the end of each iteration, which sets all but the s largest entries of the approximation to 0 – guaranteeing the desired sparsity level and $|S^{(k)}| \leq 3s$. Therefore, incorrect indices do not necessarily remain in the support set.

Algorithm 2 Compressive Sampling Matching Pursuit (CoSaMP)

```
**Data:** Matrix $A \in \mathbb{C}^{m \times n}$, measurement $y \in \mathbb{C}^m$, sparsity level $s$
**Result:** $s$-sparse approximation of vector $x \in \mathbb{C}^n$
$S^{(0)} = \varnothing$
$x^{(0)} = 0$
**for** $k$ in $1, \ldots, \bar{n}$ **do**
    $S^{(k)} = \mathrm{supp}(x^{(k-1)}) \cup L_{2s}(A^*(y - Ax^{(k-1)}))$
    $\tilde{x}^{(k)} = \mathrm{argmin}_{x \in \mathbb{C}^n}\{(\|Ax - y\|)_2, \mathrm{supp}(x) \subseteq S^{(k)}\}$
    $x^{(k)} = H_s(\tilde{x}^{(k)})$
**end for**
```

Combined, a less strong RIP is required while still exhibiting a convincing rate of convergence:

Theorem 3.6. *If a matrix $A \in \mathbb{C}^{m \times n}$ fulfills the RIP with $\delta_{4s} < \frac{\sqrt{\sqrt{\frac{11}{3}}-1}}{2}$ and $x \in \mathbb{C}^n$ is a s-sparse vector. Then the approximation $x^{(k)}$ obtained after k iterations of the CoSaMP algorithm satisfies*

$$\|x^{(k)} - x\|_2 \leq \rho^k \|x^{(0)} - x\|_2$$

with $\rho = \sqrt{\frac{2\delta_{4s}^2(1+3\delta_{4s}^2)}{1-\delta_{4s}^2}}$.

Thresholding Algorithms

Thresholding algorithms are iterative procedures combined with a hard thresholding operator that enforces the desired sparsity level.

The arguably most straightforward approach is the Iterative Hard Thresholding, initially used for compressed sensing in [9]:

Algorithm 3 Iterative Hard Thresholding (IHT)

```
**Data:** Matrix $A \in \mathbb{C}^{m \times n}$, measurement $y \in \mathbb{C}^m$, sparsity level $s$
**Result:** $s$-sparse approximation of vector $x \in \mathbb{C}^n$
$x^{(0)} = 0$
**for** $k$ in $1, \dots, \bar{n}$ **do**
    $x^{(k)} = H_s(x^{(k-1)} + A^*(y - Ax^{(k-1)}))$
**end for**
```

A second, computationally more expensive, method is the Hard Thresholding Pursuit, analyzed for compressed sensing in [31]. The HTP uses the logic of the IHT to obtain a first s-sparse approximation $\tilde{x}^{(k)}$ and then uses its support set $S := \text{supp}(\tilde{x}^{(k)})$ for the least squares approach already discussed for the Greedy algorithms.

Algorithm 4 Hard Thresholding Pursuit (HTP)

```
**Data:** Matrix $A \in \mathbb{C}^{m \times n}$, measurement $y \in \mathbb{C}^m$, sparsity level $s$
**Result:** $s$-sparse approximation of vector $x \in \mathbb{C}^n$
$x^{(0)} = 0$
**for** $k$ in $1, \dots, \bar{n}$ **do**
    $\tilde{x}^{(k)} = H_s(x^{(k-1)} + A^*(y - Ax^{(k-1)}))$
    $x^{(k)} = \text{argmin}_{x \in \mathbb{C}^n}\{(\|Ax - y\|)_2, \text{supp}(x) \subseteq \text{supp}(\tilde{x}^{(k)})\}$
**end for**
```

Convergence of the IHT can already be shown for $\delta_{3s} < \frac{1}{2}$. For this work, only a more restrictive version with a stronger convergence result is stated.

Theorem 3.7. *If a matrix $A \in \mathbb{C}^{m \times n}$ fulfills the RIP with $\delta_{3s} < \frac{1}{\sqrt{3}}$, then every s-sparse vector $x \in \mathbb{C}^n$ is approximated with the following error rates*

$$\|x^{(k)} - x\|_2 \leq \rho^k \|x^{(0)} - x\|_2$$

with

$$\rho = \begin{cases} \sqrt{3}\delta_{3s} & \textit{for IHT(Alg.3)} \\ \sqrt{2\delta_{3s}^2/(1-\delta_{2s}^2)} & \textit{for HTP(Alg.4).} \end{cases}$$

Intuitively, HTP shows a stronger convergence. This can also be seen in the results as

$$3\delta_{3s}^2 > 2\delta_{3s}^2/(1-\delta_{2s}^2) \quad \Leftrightarrow \quad 1-\delta_{2s}^2 > \frac{2}{3} \quad \Leftrightarrow \quad \delta_{2s} < \frac{1}{\sqrt{3}}.$$

The last inequality follows from the assumption $\delta_{3s} < \frac{1}{\sqrt{3}}$ as every $2s$-sparse vector is also $3s$-sparse (as it means 'at most 3s' non-zero entries). Therefore, $\delta_{2s} < \delta_{3s} < \frac{1}{\sqrt{3}}$.

4 Fast Linear Transforms Using Spherical Designs

A fundamental problem of numerical linear algebra is the efficient computation of matrix-vector products. Given a matrix $A \in \mathbb{R}^{m \times n}$ and a vector $x \in \mathbb{R}^n$, the matrix-vector product can be seen as m inner products of the type

$$(Ax)_i = a_i^T x \qquad \forall k \in [m],$$

where a_i^T denotes the ith row of A. As the inner product of two n-dimensional vectors can be obtained in $O(n)$ operations, the computation of Ax requires $O(mn)$ operations.

This can be sped up significantly if A has a specific beneficial structure as already indicated in section 1.1. An overview of such methods is given in the next section as part of publication [33] and, therefore, omitted here.

In this section, the initial idea for the aforementioned publication is outlined and a (to our knowledge) novel theory for a randomized construction of approximate spherical designs is established.

The underlying idea of a two-stage algorithm for fast matrix-vector multiplication based on spherical designs originated from early discussions of Felix Krahmer and Dustin Mixon in 2014 which has never been published. The construction of approximate spherical designs was my contribution and initiated further discussions with Felix Krahmer. There, we realized the importance of a sparsity assumption which initiated our work on [33].

4.1 Two-Stage Matrix-Vector Multiplication

For an arbitrary matrix and vector, there is little hope of obtaining their matrix-vector product using less than $O(mn)$ computations for realistic dimensions when calculating this product at once.

This changes if the matrix is known in advance and there is the possibility of performing a preprocessing:

Remark 4.1. *Assume there is a matrix $P \in \mathbb{R}^{N \times n}$ with $N < \min(m, n)$ such that*

$$AP^T Px \approx Ax.$$

Then, after a preprocessing step of calculating AP^T in $\mathcal{O}(mnN)$ computations, $AP^T Px$ can be obtained for any vector $x \in \mathbb{R}^n$ in only $\mathcal{O}(N(m+n))$ operations ($\mathcal{O}(Nn)$ for computing Px and $\mathcal{O}(Nm)$ multiplying the precomputed AP^T and the vector Px).

There are two important aspects to point out:

- As $N < \min(m, n)$, if A has full rank, there cannot be a single matrix P which fulfills $AP^T Px \approx Ax$ for every $x \in \mathbb{R}^n$. In fact, as the dimension of Px is smaller than the dimension of Ax, there would be vectors $x \in \ker(P) \setminus \ker(\mathrm{A})$, i.e., $Ax \neq 0$ but $AP^T Px = 0$.
- N should be as small as possible to increase the efficiency gain of the per-vector computation and, therefore, justify the additional effort of the preprocessing.

To avoid the first issue, a more extensive preprocessing is required where Aw has to be precomputed for a large number of vectors $w \in U \subset S^{n-1}$. To address the second issue, P is constructed by randomly sampling N vectors from U with N sufficiently small.

The natural choice for a suitable set $U \subset S^{n-1}$ is a spherical design:

Definition 4.2 ([24, 78])**.** *Let $t \in \mathbb{N}$. A non-empty, finite subset $U \subset S^{n-1}$ is called spherical t-design if the equality*

$$\int_{S^{n-1}} f(x) d\sigma(x) = \frac{1}{|U|} \sum_{x \in U} f(x)$$

holds for all homogenous polynomials f of degree at most t. $\sigma(x)$ denotes the normalized spherical measure.

In other words, spherical designs allow the computation of the integral over the sphere for any such function by simply computing the arithmetic mean of this function evaluated for all elements of the spherical design.

For constructing designs, it is helpful to consider the following equivalent formulation:

Lemma 4.3 ([78]). *Let $t = 2p$ for a $p \in \mathbb{N}$, then for all $n \geq 2$, a $U \subseteq S^{n-1}$ is a spherical t-design if*

$$\frac{1}{|U|}\sum_{x \in U}\langle w, x\rangle^{2p} = \int_{S^{n-1}} \langle w, x\rangle^{2p} d\sigma(x) \qquad \forall w \in S^{n-1}.$$

Corollary 4.4 ([78]). *The integral in Lemma 4.3 can be computed directly as*

$$\begin{aligned}\int_{S^{n-1}} \langle w, x\rangle^{2p} d\sigma(x) = \int_{S^{n-1}} x_1^{2p} d\sigma(x) &= \frac{1 \cdot 3 \cdot 5 \cdots (2p-1)}{n \cdot (n+2) \cdots (n+2p-2)} \\ &= \frac{1}{\sqrt{\pi}} \frac{\Gamma(p+\frac{1}{2})\Gamma(\frac{n}{2})}{\Gamma(p+\frac{n}{2})} =: \gamma_{p,n}\end{aligned}$$

which holds for all $w \in S^{n-1}$.

In summary, the idea is to perform a preprocessing based on a full spherical design, but to then only sample from the full design to obtain an efficient per-vector computation. By its definition, a spherical design seems suitable for preserving information of any vector and, therefore, a uniform sample from such a design should be sufficient for preserving information for a single fixed vector x with high probability.

Based on those results, one could now construct the algorithm and derive recovery guarantees. Unfortunately, first proof strategies pointed to the following issue:

- The higher the order t of the used spherical design, the lower the required sample size N and, therefore, computations.

- Already the theoretical lower bounds for the existence of spherical designs are scaling superlinearly in the dimension n (e.g., [24, 85]) and explicit constructions for higher-order designs in high dimensions seem either unknown or are showing an exponential scaling in n [6, 10]. This would defeat the purpose of the algorithm as speed benefits can only be expected for large dimensions and a preprocessing might not be feasible anymore if the design is too large.

4.2 Approximate Spherical Designs

The only need for the explicit construction of an exact spherical t-design was to ensure that a uniform sample of this set would preserve enough information when multiplied with a fixed arbitrary vector. Since there is only an interest in the properties of the uniform sample and not the full set, a natural consequence is relaxing this restriction and using an approximate spherical design as full set $U \subset S^{n-1}$ instead:

Definition 4.5. *[4] Let $t = 2p$ for a $p \in \mathbb{N}$. A non-empty, finite subset $U \subset S^{n-1}$ is called δ-approximate spherical t-design if*

$$(1-\delta)\gamma_{p,n} \leq \frac{1}{|U|}\sum_{x\in U}\langle w, x\rangle^{2p} \leq (1+\delta)\gamma_{p,n}$$

is fulfilled for all $w \in S^{n-1}$.

Motivated by the orthogonal invariance of the multivariate normal distribution, we define the following approach for constructing approximate spherical designs: Let $G_i \sim \mathcal{N}(0, I_n)$ be i.i.d. Gaussian vectors and $X_i = \frac{G^{(i)}}{\|G^{(i)}\|_2}$. It is known that X_i is uniformly distributed on S^{n-1} and, naturally, independent of its initial length $\|G^{(i)}\|_2$ (e.g., [53]).

The idea is to sample a set of such vectors X_i in order to obtain a δ-approximate spherical t-design $U_\delta^n := \{X_1, \dots, X_L\}$. Computing the expectation yields

$$\begin{aligned}\mathbb{E}\left[\frac{1}{L}\sum_{i=1}^{L}\langle w, X_i\rangle^{2p}\right] &= \mathbb{E}[\langle w, X_1\rangle^{2p}] = \mathbb{E}\left[\left\langle w, \frac{G_1}{\|G_1\|_2}\right\rangle^{2p}\right]\frac{\mathbb{E}[\|G_1\|_2^{2p}]}{\mathbb{E}[\|G_1\|_2^{2p}]}\\ &= \frac{\mathbb{E}\left[\langle w, G_1\rangle^{2p}\right]}{\mathbb{E}[\|G_1\|_2^{2p}]} \overset{(*)}{=} \frac{\frac{2^p}{\sqrt{\pi}}\Gamma(p+\frac{1}{2})}{2^p\frac{\Gamma(p+\frac{n}{2})}{\Gamma(\frac{n}{2})}} = \gamma_{p,n}.\end{aligned}$$

As $\langle w, G_1\rangle \sim \mathcal{N}(0,1)$ for every $w \in S^{n-1}$ and $\|G_1\|_2 \sim \chi_N$, $(*)$ follows by directly computing the $2p$-th moment of both distributions.

The theory of different types of convergence of a sequence of random variables would exceed the scope of this section and is not part of the construction or proof. Nevertheless, the Uniform Law of Large Numbers serves as motivation why the described construction can be successful:

Lemma 4.6 (Uniform Law of Large Numbers [69])**.** *Let X_i be i.i.d., W compact and $f(w, X_i)$ continuous and bounded by a function $g(X_i)$ with $\mathbb{E}[g(X_i)] < \infty$. Then,*

$$\sup_{w \in W} \left| \frac{1}{L} \sum_{i=1}^{L} f(w, X_i) - \mathbb{E}[f(w, X)] \right| \xrightarrow{P} 0,$$

where $Z_i \xrightarrow{P} 0$ denotes the convergence in probability, i.e., $\mathbb{P}(|Z_i| \geq \epsilon) \xrightarrow{i \to \infty} 0$.

As S^{n-1} is compact and the function $f(w, x) = \langle w, x \rangle^{2p}$ is continuous and bounded by 1 for all $w, x \in S^{n-1}$, Lemma 4.6 implies

$$V_L := \sup_{w \in S^{n-1}} \left| \frac{1}{L} \sum_{i=1}^{L} \langle w, X_i \rangle^{2p} - \gamma_{p,n} \right| \xrightarrow{\mathrm{P}} 0.$$

Intuitively, sampling a sufficient amount of vectors X_i leads to a δ-approximate spherical t-design with high probability.

For a given $\delta > 0$, the goal is to compute $\tilde{L}_\delta$ such that $V_L \leq \delta \gamma_{p,n}$ for all $L \geq \tilde{L}_\delta$. The proof idea is as follows: As V_L appears difficult to bound directly, instead, a bound for $\mathbb{E}[V_L]$ is proven first. The bound for V_L will then follow via a concentration result for empirical processes.

Bound for $\mathbb{E}[V_L]$

Lemma 4.7 ([80])**.** *Let $\mathcal{F}$ be a set of bounded functions. Then,*

$$\mathbb{E}\left[\sup_{f\in\mathcal{F}}\left|\frac{1}{L}\sum_{i=1}^{L} f(X_i)-\mathbb{E}[f(X_1)]\right|\right]\leq 2\mathbb{E}\left[\sup_{f\in\mathcal{F}}\left|\frac{1}{L}\sum_{i=1}^{L}\epsilon_i f(X_i)\right|\right]. \tag{4.1}$$

Here, ϵ_i are i.i.d. Rademacher random variables, i.e., $\mathbb{P}(\epsilon_i=-1)=\mathbb{P}(\epsilon_i=+1)=\frac{1}{2}$.

Proof The proof of this result follows by a symmetrization argument. Let $Y_1,\dots,Y_n$ be an i.i.d. copy of $X_1,\dots,X_n$. Then, we can bound

$$\begin{aligned}\mathbb{E}\left[\sup_{f\in\mathcal{F}}\left|\frac{1}{L}\sum_{i=1}^{L} f(X_i)-\mathbb{E}[f(X_1)]\right|\right] &= \mathbb{E}_X\left[\sup_{f\in\mathcal{F}}\left|\frac{1}{L}\sum_{i=1}^{L}\big(f(X_i)-\mathbb{E}_{Y_i}[f(Y_i)]\big)\right|\right]\\ &= \mathbb{E}_X\left[\sup_{f\in\mathcal{F}}\left|\mathbb{E}_Y\left[\frac{1}{L}\sum_{i=1}^{L}(f(X_i)-f(Y_i))\right]\right|\right]\\ &\leq \mathbb{E}_{X,Y}\left[\sup_{f\in\mathcal{F}}\left|\frac{1}{L}\sum_{i=1}^{L}(f(X_i)-f(Y_i))\right|\right]\\ &= \mathbb{E}_{X,Y,\epsilon}\left[\sup_{f\in\mathcal{F}}\left|\frac{1}{L}\sum_{i=1}^{L}\epsilon_i\,(f(X_i)-f(Y_i))\right|\right]\\ &\leq 2\mathbb{E}_{X,\epsilon}\left[\sup_{f\in\mathcal{F}}\left|\frac{1}{L}\sum_{i=1}^{L}\epsilon_i f(X_i)\right|\right].\end{aligned}$$

The last equality follows by the symmetrization argument. As ϵ is independent of X and Y, $(f(X_i)-f(Y_i))\overset{d}{=}\epsilon_i(f(X_i)-f(Y_i))$.

□

The quantity $\mathbb{E}\left[\sup_{f\in\mathcal{F}}\frac{1}{L}\left|\sum_{i=1}^{L}\epsilon_i f(X_i)\right|\right]$ is called the Rademacher complexity of $\mathcal{F}$. There are various approaches for bounding it.

Theorem 4.8 (Ledoux-Talagrand contraction [53])**.** *Let $g:\mathbb{R}_+\to\mathbb{R}_+$ be convex and increasing. Let $\phi_i:\mathbb{R}\to\mathbb{R}$ satisfy $\phi_i(0)=0$ and be Lipschitz continuous with* *constant λ. For any bounded $T \subseteq \mathbb{R}^n$,*

$$\mathbb{E}g\left(\frac{1}{2}\sup_{t\in T}\left|\sum_{i=1}^{L}\epsilon_i\phi_i(t_i)\right|\right) \leq \mathbb{E}g\left(L\sup_{t\in T}\left|\sum_{i=1}^{L}\epsilon_i t_i\right|\right).$$

By choosing $g(x) = x$ and $T = \{(f(X_1),\dots,f(X_n)) : f \in \mathcal{F}\}$ with $\mathcal{F}$ such that T remains bounded, one obtains the following corollary:

Corollary 4.9. *Let $\phi_i : \mathbb{R} \to \mathbb{R}$ satisfy $\phi_i(0) = 0$ and be Lipschitz continuous with constant λ. If $T = \{(f(X_1),\dots,f(X_n)) : f \in \mathcal{F}\} \subseteq \mathbb{R}^n$ is bounded almost surely, then,*

$$\mathbb{E}\sup_{f\in\mathcal{F}}\left|\sum_{i=1}^{L}\epsilon_i\phi_i(f(X_i))\right| \leq 2\lambda\mathbb{E}\sup_{f\in\mathcal{F}}\left|\sum_{i=1}^{L}\epsilon_i f(X_i)\right|.$$

It should be noted that it is sufficient if ϕ is λ-Lipschitz on the range of f.

Set $f(x) = \langle w, x\rangle$ and $\mathcal{F} := \{f(x) = \langle w, x\rangle : w \in S^{n-1}\}$ and $\phi(x) = x^{2p}$. As $f : S^{n-1} \to [-1,1]$ and $|\phi(x)-\phi(y)| = |x^{2p}-y^{2p}| = |x-y||\sum_{k=0}^{2p-1} x^k y^{2p-1-k}| \leq 2p|x-y|$ for all $x, y \in [-1,1]$, ϕ is Lipschitz continuous with constant $L = 2p$. Combining the results from above with this choice of $\mathcal{F}$ and ϕ, the following Lemma can be shown.

Lemma 4.10. *Let $X_1,\dots,X_L$ be i.i.d. random vectors uniformly distributed on S^{n-1}. Then,*

$$\mathbb{E}\left[\sup_{w\in S^{n-1}}\left|\frac{1}{L}\sum_{i=1}^{L}\langle w, X_i\rangle^{2p} - \gamma_{p,n}\right|\right] \leq \frac{8p}{\sqrt{L}}.$$

Proof. Applying first, Lemma 4.7 and, then, Corollary 4.9 allows using the basic property $\sup_{w\in S^{n-1}}\langle w, x\rangle = \|x\|_2$ to obtain

$$\begin{aligned}\mathbb{E}\left[\sup_{w\in S^{n-1}}\left|\frac{1}{L}\sum_{i=1}^{L}\langle w, X_i\rangle^{2p} - \gamma_{p,n}\right|\right] &\le 2\mathbb{E}\left[\sup_{w\in S^{n-1}}\left|\frac{1}{L}\sum_{i=1}^{L}\epsilon_i\langle w, X_i\rangle^{2p}\right|\right] \\ &\le 8p\mathbb{E}\left[\sup_{w\in S^{n-1}}\left|\frac{1}{L}\sum_{i=1}^{L}\epsilon_i\langle w, X_i\rangle\right|\right] = \frac{8p}{L}\mathbb{E}\left[\left\|\sum_{i=1}^{L}\epsilon_i X_i\right\|_2\right].\end{aligned}$$

Using Jensen's inequality and the independence of the Rademacher random variables, the proof can be concluded:

$$\begin{aligned}\frac{8p}{L}\mathbb{E}\left[\left\|\sum_{i=1}^{L}\epsilon_i X_i\right\|_2\right] &\le \frac{8p}{L}\sqrt{\mathbb{E}\left[\left\|\sum_{i=1}^{L}\epsilon_i X_i\right\|_2^2\right]} \\ &= \frac{8p}{L}\sqrt{\mathbb{E}\left[\left\langle\sum_{i=1}^{L}\epsilon_i X_i, \sum_{j=1}^{L}\epsilon_j X_j\right\rangle\right]} \\ &= \frac{8p}{L}\sqrt{\mathbb{E}\left[\sum_{i\neq j}^{L}\underbrace{\epsilon_i\epsilon_j}_{\epsilon_i\perp\!\!\!\perp\epsilon_j}\langle X_i, X_j\rangle + \sum_{i=1}^{L}\epsilon_i^2\underbrace{\|X_i\|_2^2}_{=1}\right]} \\ &= \frac{8p}{\sqrt{L}}.\end{aligned}$$

□

Bound for V_L

It remains to connect the results proven for $\mathbb{E}[V_L]$ to V_L – which is the quantity that needs to be bounded. To achieve a sufficiently strong bound, a Bernstein inequality for suprema of empirical processes is used:

Lemma 4.11. *[30] Let $\mathcal{F}$ be a countable set of functions $f : \mathbb{R}^n \to \mathbb{R}$ and $X_1, \ldots, X_L$ independent random vectors in $\mathbb{R}^n$ fulfilling*

$$f(X_i) \leq K \text{ almost surely, } \mathbb{E}[f(X_i)] = 0, \text{ and } \mathbb{E}[f(X_i)^2] \leq \sigma_i^2$$

for all $i \in [L]$ and $f \in \mathcal{F}$, and some constants $K, \sigma_1, \ldots, \sigma_L > 0$. Then, for all $t > 0$,

$$\mathbb{P}(V_L \geq \mathbb{E}[V_L] + t) \leq \exp\left(-\frac{t^2/2}{\sigma^2 + 2K\mathbb{E}[V_L] + tK/3}\right)$$

with $V_L = \sup_{f\in\mathcal{F}} \left|\sum_{i=1}^L f(X_i)\right|$ and $\sigma^2 = \sum_{i=1}^L \sigma_i^2$.

$\mathcal{F}$ needs to be slightly adjusted compared to before to fulfill the restrictions of Lemma 4.2 ($\mathbb{E}[f(X_i)] = 0$, the countability of $\mathcal{F}$, and scaling with $\frac{1}{L}$). As S^{n-1} is uncountable, w cannot be varied over the full sphere. Instead, $\mathcal{F} := \{f(x) = \frac{1}{L}\left(\langle w, x\rangle^{2p} - \gamma_{p,n}\right) : w \in S^{n-1} \cap \mathbb{Q}^n\}$ is chosen. As $S^{n-1} \cap \mathbb{Q}^n$ is a dense countable subset of S^{n-1},

$$\sup_{f\in\mathcal{F}} \left|\sum_{i=1}^L f(X_i)\right| = \sup_{w\in S^{n-1}} \left|\frac{1}{L}\sum_{i=1}^L \langle w, X_i\rangle^{2p} - \gamma_{p,n}\right|$$

remains satisfied (e.g., [11]).

Reaching the end of the line of thought and having all necessary lemmata and corollaries available, it remains to correctly determine or bound the required quantities from Lemma 4.11, and combine all results to a theorem:

- $\underline{t > 0}$: $\mathbb{P}(V_L \geq \delta\gamma_{p,n}) \leq \mathbb{P}(V_L \geq \mathbb{E}[V_L] + t)$ is required and by Lemma 4.10, $\mathbb{E}[V_L] \leq \frac{8p}{\sqrt{L}}$.

$$\Rightarrow t := \delta\gamma_{p,n} - \frac{8p}{\sqrt{L}} \qquad \Rightarrow L > \left(\frac{8p}{\delta\gamma_{p,n}}\right)^2$$

- $\underline{K}$: As $\gamma_{p,n} \in (0, 1]$ by definition and w, X_i are unit norm vectors, $f(X_i) = \frac{1}{L}\left(\langle w, X_i\rangle^{2p} - \gamma_{p,n}\right) \leq \frac{1}{L}$

$$\Rightarrow K := \frac{1}{L}$$

- $\underline{\sigma^2}$: $\mathbb{E}[f(X_i)^2] = \frac{1}{L^2}\mathbb{E}[\left(\langle w, X_i\rangle^{2p} - \gamma_{p,n}\right)^2] = \frac{1}{L^2}\left(\gamma_{2p,n} - \gamma_{p,n}^2\right) =: \sigma_i^2$

$$\Rightarrow \sigma^2 = \sum_{i=1}^{L} \sigma_i^2 = \frac{1}{L}\left(\gamma_{2p,n} - \gamma_{p,n}^2\right)$$

Combining all this, the following theorem is proven.

Theorem 4.12. *A set of L i.i.d. random vectors $X_1, \ldots, X_L$ uniformly distributed on S^{n-1} is a δ-approximate spherical t-design for any even $t = 2p$, i.e.,*

$$\sup_{w \in S^{n-1}} \left| \frac{1}{L} \sum_{i=1}^{L} \langle w, X_i\rangle^{2p} - \gamma_{p,n} \right| \leq \delta\gamma_{p,n},$$

with probability at least $1 - \exp\left(-L\frac{\left(\delta\gamma_{p,n} - \frac{8p}{\sqrt{L}}\right)^2/2}{\gamma_{2p,n} - \gamma_{p,n}^2 + \frac{1}{3}\left(\delta\gamma_{p,n} + \frac{40p}{\sqrt{L}}\right)}\right)$ *if*

$$L > \left(\frac{8p}{\delta\gamma_{p,n}}\right)^2.$$

Summarized, an approximate spherical t-design for any dimension n, order t, and deviation $\delta > 0$ can be constructed by sampling a sufficiently large number of vectors uniformly from the sphere at random. It remains to discuss the scaling of L, the size of the approximate design. By definition, $\gamma_{p,n}^{-2} \in \Theta_p(n^{2p})$. While this scaling in the dimension seems promising when compared to the required scaling for known exact spherical designs, combined with the scaling of δ^{-2} – which naturally is desired to be small – concerns for the feasibility of the preprocessing in real-world applications were raised. Based on these insights, the focus shifted from the construction of high-order spherical designs to a successful revision of the initial algorithm which resulted in publication [33] presented in the next section.

5 Sketching with Kerdock's Crayons: Fast Sparsifying Transforms for Arbitrary Linear Maps

This section corresponds to the publication [33] with slight adaptations.

In 2014, Felix Krahmer and Dustin Mixon discussed a potential, efficient algorithm for estimating matrix-vector products based on partially derandomized Johnson-Lindenstrauss projections using spherical designs. Felix Krahmer and I then refined those early ideas realizing that the introduction of sparsity might lead to efficiency gains which potentially not only yield a competitive computational complexity but even real-world runtime advantages over the standard approach for realistic dimensions.

The construction of approximate spherical designs presented in the previous section allowed for a realistic usage of this algorithm. However, only after discussions with David Gross, who suggested the usage of Kerdock sets, and the introduction of median-of-means by Richard Kueng, did the computational complexity of both steps reach the desired level. Refining the discussed concepts into theorems with corresponding proofs was to a large extent my contribution, as well as the required numerical simulations demonstrating the real-world applicability and advantages of the method presented below.

5.1 Introduction

Computing matrix–vector products is a fundamental part of numerical linear algebra. The naive algorithm takes $O(mn)$ operations to multiply an $m \times n$ matrix by a vector. Many structured matrices admit a more efficient implementation of this computation, the most well-known example being the fast Fourier transform, which takes only $O(n \log n)$ operations. In some applications, the desired Fourier transform of the given signal is nearly s-sparse, and as we discuss below, a number of works have proposed methods for such cases that are sublinear in the dimension n.

For the one-dimensional discrete Fourier transform, a randomized algorithm with a runtime scaling quadratically in s up to logarithmic factors in the dimension n has been provided in [35], while a deterministic approach with similar complexity was found in [42, 43]. In later works, this could be reduced to linear scaling in s for both random [36, 38, 39, 42, 43] and deterministic [20, 50] algorithms. For the d-dimensional Fourier transform applied to signals in $n = N^d$ dimensions, the exponential scaling in d presents an instance of the curse of dimensionality. Despite this, for random signals, one may obtain runtimes that are linear in sd up to logarithmic factors in N [16, 17]. For deterministic signals, various deterministic [43, 63] and random [18, 19, 41, 46, 47] sampling strategies have been proposed with a computational complexity which scales polynomially in d, s and N up to logarithmic factors.

Naturally, research on fast transforms is not restricted to Fourier structure. For example, [70] proposes a multidimensional Chebyshev transform with reduced runtime. In [19], a more general approach has been established that yields fast sparse transforms for arbitrary bounded orthonormal product basis with a runtime scaling polynomially in s up to logarithmic factors. These results have been generalized in [18] to signals with only an approximately s-sparse representation while maintaining a computational complexity that is sublinear in the dimension n. While covering a significantly larger class of transforms than just the Fourier transform, all these approaches remain restricted to a specific structure or class of structures of the transformation matrix.

At the same time, data-driven sparsifying transforms, which have been demonstrated to outperform predefined structured representation systems in a variety of contexts [12, 28, 59, 66, 71, 76], typically do not have structural properties that allow for the application of any of the above fast transform methods. This issue was addressed in [72, 73] by imposing structure amenable to fast transforms on the learned representation system A so as to facilitate the computation of Ax. At the same time, this imposed structure significantly limits the space of admissible transforms, and the question remains whether a fast transform can also be constructed for learned representation systems beyond these restrictions.

In this work, we consider cases where the desired product Ax is approximately sparse for a matrix A that does not follow any preset structural constraints, e.g., because it is learned from data. In particular, we assume A is arbitrary. Note that to compute the mapping $(A, x) \mapsto Ax$ for an arbitrary matrix A and vector x, one must first read the input $(A, x) \in \mathbb{R}^{m \times n} \times \mathbb{R}^n$. Since this already requires $\Theta(mn)$ operations, naive matrix–vector multiplication is optimally efficient when computing an individual matrix–vector product.
To obtain a speedup, we instead apply the same transform $x \mapsto Ax$ to a stream of vectors $x \in \mathbb{R}^n$, which models the setting of many applications. An important example of such an application is dictionary learning [2] where a data driven approach is employed to compute a representation system that gives rise to sparse representation for a given type of signals. The idea is that such a system learned with the help of a training set is better adapted to the kind of data at hand than any universal representation system. At the same time, while the outcome of such a procedure will typically not have structural properties amenable to fast multiplication, it is of great relevance for the use of learned dictionaries to be able to apply them efficiently once the training phase is completed.

Our approach will require some upfront preprocessing given A – for dictionary learning, this can be seen as the final part of the training procedure – in exchange for a much faster per-vector computation afterwards. While little work has been done in this vein, there has been quite a bit of work on related problems, which we discuss below.

The first result following this strategy [83] concerns a matrix–vector multiplication algorithm over finite semirings (for general matrices and vectors, hence not assuming any kind of sparsity), which performs $O(n^{2+\epsilon})$ operations of preprocessing on an $n \times n$ matrix before multiplying with an arbitrary vector in $O(n^2/(\epsilon \log n)^2)$ operations. The first and (to our knowledge) only algorithm that achieves a comparable result for real $m \times n$ matrices is the *mailman algorithm* introduced in [56]. Provided the matrix contains only a constant number of distinct values, the algorithm takes $O(mn)$ operations of preprocessing and then takes $O(mn/\log(m+n))$ operations to multiply the preprocessed matrix with an arbitrary vector.

Important steps in the direction of the problem considered in this work were taken in [40] and [44]. These works apply a dimension reduction designed in the context of group testing [22] to compute Ax in the case that it is exactly s-sparse [40] or has entries of exponentially decaying size [44]. After $O(s2^{O(\log^2 \log m)}mn)$ operations of preprocessing, Ax or an approximation, respectively, can be computed in $O(s2^{O(\log^2 \log m)}n)$ operations. Due to the combinatorial nature of the procedure, however, this line of research is restricted to scenarios very close to exact sparsity. The reason for this restriction is that the ideas are based on compressed sensing results and the error of the approximation is bounded by the ℓ_2-norm of the difference between the Ax and $H_s(Ax)$ (i.e., the ℓ_2-norm of the smallest $n-s$ entries of Ax). If Ax contains s-large and $n-s$ identical/similar small entries, this bound would scale with $n-s$ and would, therefore, not be sufficient. Also, we are not aware of numerical implementations feasible for larger problem sizes, which may be due to the fact that there are large constants hidden in the big O notation.

As an alternative, one might batch the stream of vectors into matrices and then perform matrix multiplication. (Granted, such a batched computation is unacceptable for many applications or only very small batch sizes are feasible, e.g., in dictionary learning for image and video processing [60].) Research on matrix multiplication was initiated by the seminal work of Strassen [74], which multiplies two arbitrary $n \times n$ matrices in only $O(n^{2.808})$ operations (i.e., much faster than the naive $O(n^3)$ algorithm). Later algorithms [21, 23, 51, 84] improved this computational complexity to its currently best known scaling of $O(n^{2.373})$. After dividing by the batch size, this gives a per-vector cost of $O(n^{1.373})$ operations. However, we note that such algorithms are infeasible in practice.

A more feasible approach to matrix multiplication was proposed by Drineas, Kanan, and Mahoney [27]. They compute a random approximation of the desired product by multiplying two smaller matrices: one consisting of k randomly selected columns of the first matrix A, and the other consisting of corresponding rows from the second matrix B. With high probability, the Frobenius norm of the estimate error is $O(\|A\|_F\|B\|_F/\sqrt{k})$. Unfortunately, if B represents a batch of column vectors, then this guarantee offers little control of the error in each vector. On the other hand, if B represents a single column vector b, then $\|Ab\|_2$ is typically much smaller than $\|A\|_F\|b\|_2$, so the resulting relative error is quite large even for relatively large values of k.

5.1.1 Our Approach

Given $A \in \mathbb{R}^{m\times n}$, let $\Sigma(A, s, \delta)$ denote the set of all unit vectors $x \in \mathbb{R}^n$ for which

$$\inf\Big\{\|Ax - v\|_\infty : v \in \mathbb{R}^m, |\operatorname{supp}(v)| \leq s\Big\} \leq \delta,$$

i.e., Ax is δ-close to being s-sparse. Given $\epsilon > 0$, let $h_\epsilon \colon \mathbb{R} \to \mathbb{R}$ denote the ϵ-hard thresholding function defined by $h_\epsilon(t) := t \cdot \mathbf{1}\{|t| \geq \epsilon\}$. By abuse of notation, we apply h_ϵ to the entries of a vector v by writing $h_\epsilon(v)$. We seek to solve the following:

Problem 5.1. *Given an arbitrary $A \in \mathbb{R}^{m\times n}$, $s \in \mathbb{N}$, and $\epsilon > \delta \geq 0$, preprocess A so that one may quickly compute $h_\epsilon(Ax)$ for any $x \in \Sigma(A, s, \delta)$.*

Our approach uses a specially designed random vector $z \in \mathbb{R}^n$ such that $\mathbb{E}[zz^\top] = I$:

$$Ax = A(\mathbb{E}[zz^\top])x = \mathbb{E}[Azz^\top x]. \tag{5.1}$$

Denote the random vector $y := Azz^\top x \in \mathbb{R}^m$. The fact that $\mathbb{E}[y] = Ax$ suggests a Monte Carlo approach to estimate Ax. That is, we will approximate the true average Ax with an estimator determined by N independent samples. To obtain a fast algorithm in this vein, we will select a distribution for z and an estimator $\hat{\mu}$ for $\mathbb{E}[y]$ that together satisfy three properties:

(i) the distribution of z is discrete with small support,

(ii) $\hat{\mu}$ can be computed in linear time from independent realizations $\{y_j\}_{j\in[N]}$ of y, and

(iii) for each $x \in \Sigma(A, s, \delta)$, $\|\hat{\mu} - Ax\|_\infty < \frac{\epsilon-\delta}{2}$ with high probability, even for small N.

Indeed, if (i)–(iii) hold, then one may compute $h_\epsilon(Ax)$ using the following (fast) algorithm:

Let $\{s_\ell : \ell \in [L]\} \subseteq \mathbb{R}^n$ denote the support of the distribution of z. Given $A \in \mathbb{R}^{m\times n}$, we run the preprocessing step of computing $\{As_\ell\}_{\ell\in[L]}$ in $O(Lmn)$ operations. Granted, this is more expensive than computing Ax, but we only need to compute $\{As_\ell\}_{\ell\in[L]}$ once, while we expect to compute Ax for a stream of x's. Next, given $x \in \Sigma(A, s, \delta)$, we draw independent realizations $\{z_j\}_{j\in[N]}$ of z. Since we already computed $\{As_\ell\}_{\ell\in[L]}$, we may then compute the corresponding realizations $\{y_j\}_{j\in[N]}$ of y in $O(N(m + n))$ operations. Next, by (ii), we may compute $\hat{\mu}$ from $\{y_j\}_{j\in[N]}$ in $O(mN)$ operations. Finally, let $S \subseteq [m]$ denote the indices of the s entries of $\hat{\mu}$ of largest magnitude. Then by (iii), it holds with high probability that $|\hat{\mu}_i| > \frac{\epsilon+\delta}{2}$ for every $i \in \operatorname{supp}(h_\epsilon(Ax))$ while $|\hat{\mu}_i| < \frac{\epsilon+\delta}{2}$ for every $i \in [m]$ such that $|(Ax)_i| \le \delta$. Since $x \in \Sigma(A, s, \delta)$ by assumption, it follows that $S \supseteq \operatorname{supp}(h_\epsilon(Ax))$, and so $A_S x$ determines $h_\epsilon(Ax)$, which we compute in $O(sn)$ additional operations. (Of course, $A_S x$ might determine other entries in the support of Ax, and we would not discard this information in practice.)

To obtain (i)–(iii), we take z to be uniformly distributed over an appropriately scaled n-dimensional projection of a projective 2-design, and for $\hat{\mu}$, we partition $[N]$ into batches and compute the entrywise median-of-means of $\{y_j\}_{j\in[N]}$ over these batches. The projective 2-design allows us to control the variance of each entry of the random vector y; see Lemma 5.4. Next, the median-of-means estimator improves over the sample mean by being less sensitive to outliers in the small random sample $\{y_j\}_{j\in[N]}$.
Thanks to this behavior, we can get away with drawing only

$$N = O((\epsilon - \delta)^{-2} \|A\|_{2\to\infty}^2 \log(m/\eta))$$

samples, where the induced norm $\|A\|_{2\to\infty}$ equals the largest ℓ_2-norm of the rows of A, and η denotes the failure probability of the randomized algorithm; see Theorem 5.5. As a bonus, the Kerdock set–based projective 2-design we use enjoys a fast matrix–vector multiplication algorithm, yielding a preprocessing step of only $O(mn^2 \log n + n^2 \log^2 n)$ operations despite having $L = \Theta(n^2)$; see Lemma 5.8. See Algorithm 5 for a summary of our approach.

To quickly evaluate the utility of this algorithm, consider the following model: A is an arbitrary $n \times n$ orthogonal matrix, and x is a random vector such that the entries of Ax are drawn independently from the following mixture:

$$e_i^\top Ax \sim \begin{cases} \mathcal{N}(0,1) & \text{with probability } p \\ 0 & \text{with probability } 1-p. \end{cases}$$

Then the expected size of the support of Ax is pn. In this model, our algorithm

Algorithm 5 Fast sparsifying transform for an arbitrary linear map

```
Data: Parameters $\epsilon > \delta \geq 0$, $s, J, K \in \mathbb{N}$, matrix $A \in \mathbb{R}^{m \times n}$, stream of $x \in \Sigma(A, s, \delta)$
Result: Entrywise $\epsilon$-hard threshold of matrix–vector products $h_\epsilon(Ax)$
Preprocessing step

    Let $d$ denote the smallest power of 2 that is at least $n$, and put $L := d(d/2 + 1)$

    Let $\{u_\ell\}_{\ell \in [L]}$ denote a projective 2-design for $\mathbb{R}^d$ arising from a Kerdock set

    Put $s_\ell := \sqrt{d}\Pi u_\ell$, where $\Pi \in \mathbb{R}^{n \times d}$ denotes projection onto the first $n$ coordinates

    Use Lemma 5.8 to compute $\{As_\ell\}_{\ell \in [L]}$
Streaming step

    Draw $N := JK$ indices $\{\ell_j\}_{j \in [N]}$ uniformly from $[L]$

    Compute $y_j := (As_{\ell_j})(s_{\ell_j}^\top x)$ for each $j \in [N]$

    Compute the entrywise median-of-means $\hat{\mu}$ of $\{y_j\}_{j \in [N]}$ over $K$ batches of size $J$

    Let $S \subseteq [m]$ denote the indices of the $s$ entries of $\hat{\mu}$ with largest magnitude

    Compute $A_S x$ and output the indices and values of entries with magnitude at least $\epsilon$
```

provides a speedup over naive matrix–vector multiplication in the regime

$$1 \prec pn \prec \frac{n}{\log n}.$$

To see this, first put $s := 10pn$. Then the multiplicative Chernoff bound implies that Ax is s-sparse with high probability, and so we take $\delta = 0$. Before selecting $\epsilon > 0$, we normalize our vector so that $\hat{x} := x/\|x\|_2 \in \Sigma(A, s, \delta)$ with high probability. Next, standard tail bounds imply that $\|x\|_2^2 \in \Theta(pn)$ with high probability. This suggests the scaling $\epsilon := \alpha/\sqrt{pn}$ with $\alpha \in (0, 1)$. Considering that a fraction $\Theta(\alpha)$ of the support of Ax has magnitude $\Theta(\alpha)$, the hard threshold $h_\epsilon(A\hat{x})$ serves as a decent estimate for the product $A\hat{x}$:

$$\|h_\epsilon(A\hat{x}) - A\hat{x}\|_2^2 = \Theta(\alpha^3).$$

Furthermore, we obtain this quality of estimate with relatively little computation: Since $m = n$, we have $O(n^3 \log n)$ operations of preprocessing, and then for each x, we compute $h_\epsilon(A\hat{x})$ in $O(\alpha^{-2} sn \log n)$ operations since $\|A\|_{2\to\infty} = 1$. By comparison, if an oracle were to reveal the support of $h_\epsilon(A\hat{x})$, then the naive matrix–vector multiplication with the appropriate submatrix of A would cost $O(sn)$ operations.

5.1.2 Outline

In the next section, we review the necessary theory of projective designs, and we show how they can be used in conjunction with a median-of-means estimator to obtain a high-quality random estimate of a sparse matrix–vector product. Next, section 5.3 provides the details of a specific choice of projective design, namely, one that arises from a Kerdock set described by Calderbank, Cameron, Kantor, and Seidel [13]. This particular choice of projective design allows us to leverage the fast Walsh–Hadamard transform to substantially speed up the preprocessing step of our algorithm. We conclude in section 5.4 with some numerical results that demonstrate the plausibility of a real-world implementation of our algorithm.

5.2 Projective Designs and the Median-of-Means

Let σ denote the uniform probability measure on the unit sphere S^{d-1} in $\mathbb{R}^d$, let $\mathrm{Hom}_j(\mathbb{R}^d)$ denote the set of homogeneous polynomials of total degree j in d real variables, and put

$$c_{d,k} := \frac{1 \cdot 3 \cdot 5 \cdots (2k-1)}{d(d+2)\cdots(d+2(k-1))}.$$

A **projective t-design** for $\mathbb{R}^d$ is defined to be any $\{u_\ell\}_{\ell\in[L]}$ in S^{d-1} that satisfies the following equivalent properties:

Proposition 5.2. *Given $\{u_\ell\}_{\ell\in[L]}$ in S^{d-1} and $t \in \mathbb{N}$, the following are equivalent:*

(a) $\frac{1}{L}\sum_{\ell\in[L]} p(u_\ell) = \int_{S^{d-1}} p(u)d\sigma(u)$ for every $p \in \mathrm{Hom}_{2k}(\mathbb{R}^d)$ and every $k \in \{0,\ldots,t\}$.

(b) $\frac{1}{L}\sum_{\ell\in[L]}\langle x, u_\ell\rangle^{2k} = c_{d,k}\|x\|^{2k}$ for every $x \in \mathbb{R}^d$ and every $k \in \{1,\ldots,t\}$.

(c) $\frac{1}{L^2}\sum_{\ell,\ell'\in[L]}\langle u_\ell, u_{\ell'}\rangle^{2k} = c_{d,k}$ for every $k \in \{1,\ldots,t\}$.

The proof of Proposition 5.2 is contained in section 6.4 of [81] and references therein, given the observation that $\{u_\ell\}_{\ell\in[L]}$ satisfies Proposition 5.2(a) precisely when $\{u_\ell\}_{\ell\in[L]} \cup \{-u_\ell\}_{\ell\in[L]}$ forms a so-called *spherical $2t$-design*. We will see that the cubature rule in Proposition 5.2(a) is what makes projective t-designs useful, while Proposition 5.2(c) makes them easy to identify. We note that an analog of Proposition 5.2(c) is used to define projective t-designs in a variety of settings, such as complex projective space and the Cayley plane [65].

Our application of projective 2-designs encourages us to take the size L to be as small as possible. To this end, there is a general lower bound [7] of $L \geq \binom{d+1}{2}$, but to date, equality is only known to be achieved for $d \in \{2, 3, 7, 23\}$; see [37] and references therein. Despite this scarcity, there are *infinite* families of projective 2-designs that take L to be slightly larger, specifically, $L = d(d/2 + 1)$ whenever d is a power of 4; see [13, 14]. For these constructions, $\{u_\ell\}_{\ell\in[L]}$ takes the form of a union of orthonormal bases. Orthonormal bases $\{x_i\}_{i\in[d]}$ and $\{y_i\}_{i\in[d]}$ are said to be **unbiased** if $|\langle x_i, y_j\rangle|^2 = 1/d$ for every $i, j \in [d]$.

Proposition 5.3. *Suppose $\{u_{b,i}\}_{i\in[d]}$ in $\mathbb{R}^d$ is orthonormal for every $b \in [d/2+1]$, and suppose further that $\{u_{b,i}\}_{i\in[d]}$ and $\{u_{b',i}\}_{i\in[d]}$ are unbiased for every $b, b' \in [d/2+1]$ with $b \neq b'$. Then $\{u_{b,i}\}_{b\in[d/2+1],i\in[d]}$ forms a projective 2-design for $\mathbb{R}^d$.*

The proof of Proposition 5.3 follows from Proposition 5.2(c) and the definition of unbiased. In section 5.3, we will provide an explicit construction of this form. In the meantime, we show how projective 2-designs are useful in our application. The following result defines the random vector z in terms of a projective 2-design, and then uses this structure to control the variance of each coordinate of the random vector $y := Azz^\top x$.

Lemma 5.4. *Given $A \in \mathbb{R}^{m\times n}$, fix a projective 2-design $\{u_\ell\}_{\ell\in[L]}$ for $\mathbb{R}^d$ with $d \geq n$, let $\Pi\colon \mathbb{R}^d \to \mathbb{R}^n$ denote the projection onto the first n coordinates, and let z denote a random vector with uniform distribution over $\{\sqrt{d}\Pi u_\ell : \ell \in [L]\}$. Given a unit vector $x \in \mathbb{R}^n$, define the random vector $y := Azz^\top x \in \mathbb{R}^m$. Then for each $i \in [m]$, it holds that*

$$\mathbb{E}[e_i^\top y] = e_i^\top Ax, \qquad \mathbb{V}(e_i^\top y) \leq 2\|A\|_{2\to\infty}^2.$$

Proof. Let Π^* denote the adjoint of Π, namely, the map that embeds $\mathbb{R}^n$ into the first n coordinates of $\mathbb{R}^d$. Fix $i \in [m]$, let $a_i^\top$ denote the ith row of A, and put $\tilde{a}_i := \Pi^* a_i$ and $\tilde{x} := \Pi^* x$. Then

$$e_i^\top y = e_i^\top Azz^\top x = a_i^\top(\sqrt{d}\Pi u_{\ell(z)})(\sqrt{d}\Pi u_{\ell(z)})^\top x = d\tilde{a}_i^\top u_{\ell(z)} u_{\ell(z)}^\top \tilde{x}.$$

To compute $\mathbb{E}[e_i^\top y]$ and $\mathbb{E}[(e_i^\top y)^2]$, we will consider a random vector u that is uniformly distributed on the unit sphere S^{d-1}, as well as some rotation $Q \in \mathrm{O}(d)$ such that

$$Q\tilde{x} = e_1, \qquad Q\tilde{a}_i = \alpha_1 e_1 + \alpha_2 e_2, \qquad \alpha_1, \alpha_2 \in \mathbb{R}.$$

Since $v \mapsto d\tilde{a}_i^\top vv^\top \tilde{x}$ resides in $\mathrm{Hom}_2(\mathbb{R}^d)$, we may apply Proposition 5.2(a) to get

$$\mathbb{E}[e_i^\top y] = \frac{1}{L}\sum_{\ell\in[L]} d\tilde{a}_i^\top u_\ell u_\ell^\top \tilde{x} = \mathbb{E}[d\tilde{a}_i^\top uu^\top \tilde{x}] = \mathbb{E}[d\tilde{a}_i^\top Q^\top Quu^\top Q^\top Q\tilde{x}].$$

A change of variables $Qu \mapsto u$ then gives

$$\mathbb{E}[e_i^\top y] = \mathbb{E}[d(Q\tilde{a}_i)^\top uu^\top (Q\tilde{x})] = \mathbb{E}[d(\alpha_1 e_1 + \alpha_2 e_2)^\top uu^\top e_1] = d\alpha_1 \mathbb{E}[u_1^2].$$

Considering $1 = \mathbb{E}[\|u\|^2] = d\mathbb{E}[u_1^2]$ by symmetry and linearity of expectation, it follows that

$$\mathbb{E}[e_i^\top y] = \alpha_1 = (\alpha_1 e_1 + \alpha_2 e_2)^\top e_1 = \tilde{a}_i^\top \tilde{x} = a_i^\top x = e_i^\top Ax.$$

Indeed, this behavior was the original motivation (5.1) for our approach. We will apply the same technique to compute $\mathbb{E}[(e_i^\top y)^2]$. Since $v \mapsto (d\tilde{a}_i^\top vv^\top \tilde{x})^2$ resides in $\mathrm{Hom}_4(\mathbb{R}^d)$, we may apply Proposition 5.2(a) to get

$$\mathbb{E}[(e_i^\top y)^2] = \frac{1}{L}\sum_{\ell\in[L]} (d\tilde{a}_i^\top u_\ell u_\ell^\top \tilde{x})^2 = \mathbb{E}[(d\tilde{a}_i^\top uu^\top \tilde{x})^2] = \mathbb{E}[(d\tilde{a}_i^\top Q^\top Quu^\top Q^\top Q\tilde{x})^2].$$

A change of variables $Qu \mapsto u$ then gives

$$\mathbb{E}[(e_i^\top y)^2] = \mathbb{E}[(d(\alpha_1 e_1 + \alpha_2 e_2)^\top uu^\top e_1)^2] = d^2(\alpha_1^2 \mathbb{E}[u_1^4] + \alpha_2^2 \mathbb{E}[u_1^2 u_2^2]).$$

Next, the theorem in [29] implies $\mathbb{E}[u_1^4] = \frac{3}{d(d+2)}$ and $\mathbb{E}[u_1^2 u_2^2] = \frac{1}{d(d+2)}$, thereby implying

$$\mathbb{E}[(e_i^\top y)^2] = \alpha_1^2 \cdot \frac{3d}{d+2} + \alpha_2^2 \cdot \frac{d}{d+2}.$$

Finally, we recall $\mathbb{E}[e_i^\top y] = \alpha_1$ to compute the desired variance:

$$\mathbb{V}(e_i^\top y) = \mathbb{E}[(e_i^\top y)^2] - (\mathbb{E}[e_i^\top y])^2 = 2\alpha_1^2 \cdot \frac{d-1}{d+2} + \alpha_2^2 \cdot \frac{d}{d+2} \leq 2\alpha_1^2 + \alpha_2^2 \leq 2\|a_i\|^2.$$

The result then follows from the fact that $\|A\|_{2\to\infty}^2 = \max_{i\in[m]} \|a_i\|^2$. □

Now that we have control of the variance, we can obtain strong deviation bounds on our median-of-means estimator $\hat{\mu}$ of $\mathbb{E}[y] = Ax$.

Theorem 5.5. *Given $A \in \mathbb{R}^{m\times n}$, fix a projective 2-design $\{u_\ell\}_{\ell\in[L]}$ for $\mathbb{R}^d$ with $d \geq n$, let $\Pi\colon \mathbb{R}^d \to \mathbb{R}^n$ denote the projection onto the first n coordinates, and let z denote a random vector with uniform distribution over $\{\sqrt{d}\Pi u_\ell : \ell \in [L]\}$. Given a unit vector $x \in \mathbb{R}^n$, define the random vector $y := Azz^\top x \in \mathbb{R}^m$, select*

$$J \geq \frac{4e^2\|A\|_{2\to\infty}^2}{\gamma^2}, \qquad K \geq 2\log\Big(\frac{m}{\eta}\Big),$$

draw independent copies $\{y_{jk}\}_{j\in[J],k\in[K]}$ of y, and compute the entrywise median-of-means:

$$\hat{\mu} := \text{median}\{\overline{y}_k\}_{k\in[K]}, \qquad \overline{y}_k := \frac{1}{J}\sum_{j\in[J]} y_{jk}.$$

Then $\|\hat{\mu} - Ax\|_\infty < \gamma$ with probability at least $1-\eta$.

Proof. Fix $i \in [m]$. For notational convenience, we denote the random variable $Y := e_i^\top y$. Given independent copies $\{Y_j\}_{j\in[J]}$ of Y, let $\overline{Y}$ denote their sample average. Then Chebyshev's inequality and Lemma 5.4 together imply the deviation inequality

$$p := \mathbb{P}\{|\overline{Y} - (Ax)_i| \geq \gamma\} = \mathbb{P}\{|\overline{Y} - \mathbb{E}[\overline{Y}]| \geq \gamma\} \leq \frac{\mathbb{V}(\overline{Y})}{\gamma^2} \leq \frac{2\|A\|_{2\to\infty}^2\|x\|^2}{J\gamma^2}. \quad (5.2)$$

Now take independent copies $\{\overline{Y}_k\}_{k\in[K]}$ of $\overline{Y}$ and put $\hat{\mu}_i := \text{median}\{\overline{Y}_k\}_{k\in[K]}$. Notice that $\hat{\mu}_i \geq (Ax)_i + \gamma$ only if half of the $\overline{Y}_k$'s satisfy $\overline{Y}_k \geq (Ax)_i + \gamma$. Similarly,

$\hat{\mu}_i \leq (Ax)_i - \gamma$ only if half of the $\overline{Y}_k$'s satisfy $\overline{Y}_k \leq (Ax)_i - \gamma$. Thus, defining $I_k := \mathbf{1}\{|\overline{Y}_k - (Ax)_i| \geq \gamma\}$, we have

$$\mathbb{P}\{|\hat{\mu}_i - (Ax)_i| \geq \gamma\} \leq \mathbb{P}\Big\{ \sum_{k\in[K]} I_k \geq \frac{K}{2} \Big\}.$$

Since $\{I_k\}_{k\in[K]}$ are independent Bernoulli random variables with success probability p, we may continue with the help of the multiplicative Chernoff bound:

$$\mathbb{P}\Big\{ \sum_{k\in[K]} I_k \geq (1+\lambda)Kp \Big\} \leq \Big(\frac{e^\lambda}{(1+\lambda)^{1+\lambda}}\Big)^{Kp} = e^{-Kp}\Big(\frac{e}{1+\lambda}\Big)^{(1+\lambda)Kp}, \qquad \lambda > 0.$$

By our choice of J, equation (5.2) implies that $p \leq 1/(2e^2) < 1/2$. As such, there exists $\lambda > 0$ such that $(1+\lambda)Kp = K/2$. Combining the above bounds then gives

$$\mathbb{P}\{|\hat{\mu}_i - (Ax)_i| \geq \gamma\} \leq e^{-Kp}(2ep)^{K/2} \leq (2ep)^{K/2} \leq \Big(\frac{4e\|A\|_{2\to\infty}^2\|x\|^2}{J\gamma^2}\Big)^{K/2} \leq e^{-K/2} \leq \frac{\eta}{m},$$

where the last steps follow from our choices for J and K. Finally, since our choice for $i \in [m]$ was arbitrary, the result follows from a union bound. □

5.3 Fast Preprocessing with Kerdock Sets

Select $k \in 2\mathbb{N}$, and consider the real vector space $\mathbb{R}[\mathbb{F}_2^k]$ of functions $f\colon \mathbb{F}_2^k \to \mathbb{R}$. Calderbank, Cameron, Kantor, and Seidel [13] describe a projective 2-design in this space that takes the form of mutually unbiased orthonormal bases (à la Proposition 5.3). We explicitly construct this projective 2-design with some help from the underlying finite field, and then we leverage its structure to speed up the preprocessing step of our algorithm.

We say $M \in \mathbb{F}_2^{k\times k}$ is **skew-symmetric** if $M_{ii} = 0$ and $M_{ij} = M_{ji}$ for every $i, j \in [k]$. Given a skew-symmetric M, consider the corresponding upper-triangular matrix $\tilde{M} \in \mathbb{F}_2^{k\times k}$ defined by $\tilde{M}_{ij} := M_{ij} \cdot \mathbf{1}\{i < j\}$ and the quadratic form $Q_M : \mathbb{F}_2^k \times \mathbb{F}_2^k \to \mathbb{F}_2$ defined by $Q_M(x) := x^\top \tilde{M} x$. Given a skew-symmetric $M \in \mathbb{F}_2^{k\times k}$ and $w \in \mathbb{F}_2^k$, define $u_{M,w} \in \mathbb{R}[\mathbb{F}_2^k]$ by

$$u_{M,w}(x) := 2^{-k/2}(-1)^{Q_M(x)+w^\top x}. \tag{5.3}$$

The following result uses the vectors in (5.3) to form unbiased orthonormal bases for $\mathbb{R}[\mathbb{F}_2^k]$; this result is contained in [13, 48, 49], but the proof is distributed over dozens of dense pages from multiple papers, so we provide a direct and illustrative proof at the end of this section.

Proposition 5.6. *Consider any skew-symmetric $M, M' \in \mathbb{F}_2^{k\times k}$.*

(a) $\{u_{M,w}\}_{w\in\mathbb{F}_2^k}$ and $\{u_{M',w}\}_{w\in\mathbb{F}_2^k}$ are orthonormal bases.

(b) If $M + M'$ has full rank over $\mathbb{F}_2$, then $\{u_{M,w}\}_{w\in\mathbb{F}_2^k}$ and $\{u_{M',w}\}_{w\in\mathbb{F}_2^k}$ are unbiased.

A **Kerdock set** is a collection $\mathcal{K} \subseteq \mathbb{F}_2^{k\times k}$ of 2^{k-1} skew-symmetric matrices such that $M + M'$ has full rank for every $M, M' \in \mathcal{K}$ with $M \neq M'$. (Note that one cannot hope for a larger set with this property since the first row of each matrix in $\mathcal{K}$ must be distinct, and the first entry of these rows must equal zero.) By Proposition 5.6 and equation (5.3), a Kerdock set $\mathcal{K}$ determines $2^{k-1} + 1$ mutually unbiased bases in $\mathbb{R}[\mathbb{F}_2^k]$, namely, $\{u_{M,w}\}_{w\in\mathbb{F}_2^k}$ for each $M \in \mathcal{K}$ and the identity basis $\{e_w\}_{w\in\mathbb{F}_2^k}$. By Proposition 5.3, these bases combine to form a projective 2-design. Below, we give the "standard" Kerdock set described in Example 9.2 in [13]. (Note that [13] contains a typo and omits the proof of this result; specifically, when they write αx, it should be ax; we will use s instead of a so as to clearly distinguish from α, and for completeness, we supply a proof at the end of this section.) In what follows, $\operatorname{tr}\colon \mathbb{F}_{2^{k-1}} \to \mathbb{F}_2$ denotes the field trace, while for any finite set B, we let $\mathbb{F}_2^{B\times B}$ denote the set of $|B| \times |B|$ matrices with entries in $\mathbb{F}_2$ whose rows and columns are indexed by B.

Proposition 5.7. *Consider the k-dimensional vector space $V := \mathbb{F}_{2^{k-1}} \times \mathbb{F}_2$ over the scalar field $\mathbb{F}_2$, and for each $s \in \mathbb{F}_{2^{k-1}}$, define the linear map $L_s\colon V \to V$ by*

$$L_s(x, \alpha) := (s^2x + s\operatorname{tr}(sx) + \alpha s, \operatorname{tr}(sx)).$$

Next, select a basis B for V, and consider the bilinear form

$$(x, \alpha) \cdot (y, \beta) := \operatorname{tr}(xy) + \alpha\beta.$$

Then $\{M_s \in \mathbb{F}_2^{B\times B} : s \in \mathbb{F}_{2^{k-1}}\}$ defined by $(M_s)_{b,b'} := b \cdot L_s(b')$ is a Kerdock set.

Importantly, Kerdock sets provide speedups for the preprocessing step of our algorithm:

Lemma 5.8. *Given $A \in \mathbb{R}^{m\times n}$, select the smallest $k \in 2\mathbb{N}$ satisfying $d := 2^k \geq n$. Let $\mathcal{K} \subseteq \mathbb{F}_2^{k\times k}$ denote the Kerdock set described in Proposition 5.7, consider $u_{M,w} \in \mathbb{R}[\mathbb{F}_2^k]$ defined by (5.3), let $\{e_w\}_{w\in\mathbb{F}_2^k}$ denote the identity basis in $\mathbb{R}[\mathbb{F}_2^k]$, and let $\Pi\colon \mathbb{R}[\mathbb{F}_2^k] \to \mathbb{R}^n$ denote any coordinate projection. Then*

$$\{A\sqrt{d}\Pi u_{M,w}\}_{M\in\mathcal{K},w\in\mathbb{F}_2^k} \cup \{A\sqrt{d}\Pi e_w\}_{w\in\mathbb{F}_2^k}$$

can be computed in $O(mn^2 \log n + n \log^2 n)$ operations.

Proof. We identify each member of $\mathbb{R}[\mathbb{F}_2^k]$ as a vector in $\mathbb{R}^d$ with entries indexed by $\mathbb{F}_2^k$. By this identification, the vectors $\{u_{M,w}\}_{w\in\mathbb{F}_2^k}$ appear as the columns of the matrix product $D_M H^{\otimes k}$, where $\cdot^{\otimes k}$ denotes the Kronecker power and $D_M, H \in \mathbb{R}^{d\times d}$ are defined by

$$D_M := \operatorname{diag}\{(-1)^{Q_M(x)}\}_{x\in\mathbb{F}_2^k}, \qquad H_{ab} := 2^{-1/2}(-1)^{ab}, \qquad a,b\in\mathbb{F}_2.$$

Let $a_i^\top$ denote the ith row of A. Then the following algorithm runs in the claimed number of operations: For each $M\in\mathcal{K}$, (i) compute $\{(-1)^{Q_M(x)}\}_{x\in\mathbb{F}_2^k}$ in $O(n\log^2 n)$ operations, (ii) for each $i\in[m]$, use the fast Walsh–Hadamard transform to compute $((\Pi^* a_i)^\top D_M)H^{\otimes k}$ in $O(n\log n)$ operations, and (iii) compute the columns of

$$\sqrt{d}\sum_{i\in[m]} e_i(a_i^\top \Pi D_M H^{\otimes k}) = A\sqrt{d}\Pi D_M H^{\otimes k}$$

in $O(mn)$ operations; finally, compute $\{A\sqrt{d}\Pi e_w\}_{w\in\mathbb{F}_2^k}$ in $O(mn)$ operations. □

Proof of Proposition 5.6. Define the *Heisenberg group* $H_{2k+1}(\mathbb{F}_2)$ to be the set $\mathbb{F}_2^k \times \mathbb{F}_2^k \times \mathbb{F}_2$ with multiplication

$$(p,q,\epsilon)\cdot(p',q',\epsilon') := (p+p', q+q', \epsilon+\epsilon'+q^\top p').$$

Fix any skew-symmetric $M\in\mathbb{F}_2^{k\times k}$. For each $w\in\mathbb{F}_2^k$, we define $\psi_{M,w}\colon \mathbb{F}_2^k \to H_{2k+1}(\mathbb{F}_2)$ by

$$\psi_{M,w}(p) := (p, Mp, Q_M(p) + w^\top p).$$

One may apply the polarization identity

$$Q_M(x+y) = Q_M(x) + x^\top \tilde{M} y + y^\top \tilde{M} x + Q_M(y) = Q_M(x) + x^\top M y + Q_M(y). \quad (5.4)$$

to verify that $\psi_{M,w}$ is a group homomorphism.

Next, define the *Schrödinger representation* $\rho\colon H_{2k+1}(\mathbb{F}_2) \to \mathrm{GL}(\mathbb{R}[\mathbb{F}_2^k])$ by

$$\rho(p,q,\epsilon) := (-1)^{\epsilon} X_p Z_q,$$

where X_p and Z_q denote translation and modulation operators, respectively:

$$(X_p f)(x) := f(x+p), \qquad (Z_q f)(x) := (-1)^{q^\top x} f(x). \tag{5.5}$$

Indeed, ρ is a representation of $H_{2k+1}(\mathbb{F}_2)$ as a consequence of the easily verified relation

$$Z_q X_p = (-1)^{q^\top p} X_p Z_q. \tag{5.6}$$

It follows that $\rho \circ \psi_{M,w}$ is a representation of the additive group $\mathbb{F}_2^k$. Explicitly, we have

$$(\rho \circ \psi_{M,w})(p) = (-1)^{Q_M(p)+w^\top p} X_p Z_{Mp}. \tag{5.7}$$

Next, we put

$$P_{M,w} := 2^{-k} \sum_{p \in \mathbb{F}_2^k} (\rho \circ \psi_{M,w})(p). \tag{5.8}$$

Then $P_{M,w}^2 = P_{M,w}$ and $P_{M,w}^* = P_{M,w}$, and so $P_{M,w}$ is an orthogonal projection. Decomposing into irreducible representations reveals that $\operatorname{im} P_{M,w}$ equals the intersection of the eigenspaces of $\{(\rho \circ \psi_{M,w})(p)\}_{p \in \mathbb{F}_2^k}$ with eigenvalue 1.

We claim that $P_{M,w} = u_{M,w} u_{M,w}^*$. To see this, one may first apply the definitions (5.7), (5.5), and (5.3) along with (5.4) and its consequence

$$x^\top M x = Q_M(x+x) + Q_M(x) + Q_M(x) = 0 \tag{5.9}$$

to verify the eigenvector equation

$$[(\rho \circ \psi_{M,w})(p)] u_{M,w} = u_{M,w}$$

for each $p \in \mathbb{F}_2^k$. Next, one may apply the easily verified fact that

$$\operatorname{tr}(X_p Z_q) = \begin{cases} 2^k & \text{if } p = q = 0 \\ 0 & \text{otherwise} \end{cases} \tag{5.10}$$

to the definitions (5.8) and (5.7) to compute $\operatorname{tr} P_{M,w} = 1$. This proves our intermediate claim.

We are now ready to compute $|\langle u_{M,w}, u_{M',w'}\rangle|^2$ in various cases. First, we cycle the trace:

$$|\langle u_{M,w}, u_{M',w'}\rangle|^2 = \operatorname{tr}(u_{M,w}^* u_{M',w'} u_{M',w'}^* u_{M,w})$$
$$= \operatorname{tr}(u_{M,w} u_{M,w}^* u_{M',w'} u_{M',w'}^*) = \operatorname{tr}(P_{M,w} P_{M',w'}).$$

Next, we apply the definitions (5.8) and (5.7) along with identities (5.6), (5.10), and (5.9) to obtain

$$|\langle u_{M,w}, u_{M',w'}\rangle|^2 = \operatorname{tr}(P_{M,w} P_{M',w'}) = 2^{-k} \sum_{p \in \ker(M+M')} (-1)^{Q_M(p) + Q_{M'}(p) + (w+w')^\top p}.$$

From here, we proceed in cases: If $M = M'$, then $\ker(M + M') = \mathbb{F}_2^k$ and $Q_M(p) = Q_{M'}(p)$, and so $|\langle u_{M,w}, u_{M,w'}\rangle|^2 = \delta_{w,w'}$. This gives (a). For (b), if $M + M'$ has full rank, then $\ker(M + M') = \{0\}$, and so $|\langle u_{M,w}, u_{M,w'}\rangle|^2 = 2^{-k}$, as claimed. □

Proof of Proposition 5.7. In what follows, we demonstrate three things:

(i) For every $s, x \in \mathbb{F}_{2^{k-1}}$ and $\alpha \in \mathbb{F}_2$, it holds that $(x, \alpha) \cdot L_s(x, \alpha) = 0$.

(ii) For every $s, x, y \in \mathbb{F}_{2^{k-1}}$ and $\alpha, \beta \in \mathbb{F}_2$, it holds that $(x, \alpha) \cdot L_s(y, \beta) = L_s(x, \alpha) \cdot (y, \beta)$.

(iii) For every $r, s \in \mathbb{F}_{2^{k-1}}$ with $r \neq s$, $L_r(x, \alpha) = L_s(x, \alpha)$ implies $(x, \alpha) = (0, 0)$.

Thanks to the non-degeneracy of the bilinear form, (i)–(iii) together imply the result.

First, (i) is easily verified by applying three properties of the trace: tr is $\mathbb{F}_2$-linear, $\operatorname{tr}(z^2) = \operatorname{tr}(z)$, and $\operatorname{tr}(z)^2 = \operatorname{tr}(z)$ since $\operatorname{tr}(z) \in \mathbb{F}_2$. Also, (ii) quickly follows from the linearity of the trace. For (iii), take $r, s \in \mathbb{F}_{2^{k-1}}$ with $r \neq s$ and suppose $L_r(x, \alpha) = L_s(x, \alpha)$. The second argument of this identity is $\operatorname{tr}(rx) = \operatorname{tr}(sx)$. Rearrange the first argument to get

$$0 = (r^2 + s^2)x + r\operatorname{tr}(rx) + s\operatorname{tr}(sx) + \alpha(r + s) = (r + s)\Big((r + s)x + \operatorname{tr}(rx) + \alpha\Big),$$

where the last step applies the fact that $\operatorname{tr}(rx) = \operatorname{tr}(sx)$. Since $r + s \neq 0$ by assumption, it follows that

$$(r + s)x + \operatorname{tr}(rx) = \alpha \in \mathbb{F}_2. \tag{5.11}$$

Since $\alpha^2 = \alpha$, the left-hand side of (5.11) satisfies the same quadratic, which in turn implies $x \in \{0, (r + s)^{-1}\}$. Since x also satisfies $\operatorname{tr}((r + s)x) = 0$, it follows that $x = 0$. Plugging into (5.11) then gives $\alpha = 0$, as desired. □

5.4 Numerical Results

In this section, we report the real-world performance of our fast sparsifying transform. In our experiments, we take $A \in \mathbb{R}^{n\times n}$ to be a random orthogonal matrix, and then we select a unit vector x such that Ax has exactly s nonzero entries of the same size in random positions. (This is straightforward to implement since $A^{-1} = A^{\top}$.) Due to limitations in computing power and storage capacity, we restrict our experiments to dimension $n = 4096$, and we select sparsity level $s = 20$.

What follows are some details about our implementation. Since n is a power of 2, we have $d = n$, and so our projective 2-design has size $L = n(n/2+1) = 8392704$. For such a large value of L, it turns out that the runtime of selecting N random members of the precomputed sketch $\{As_\ell\}_{\ell\in[L]}$ is sensitive to the design of the underlying data structure. In order to provide a useful runtime comparison, we therefore assume that this random selection is performed by an oracle before we start the runtime clock in our algorithm. To compute medians, we apply the quickselect algorithm [58]. Finally, to boost performance, we take $S \subseteq [n]$ to index the $10s$ largest-magnitude entries of $\hat{\mu}$ instead of the top s entries.

The results of our experiments are summarized in Figure 6. Figure 6(top-left) illustrates that the median-of-means estimator $\hat{\mu}$ performs better in practice than predicted by Theorem 5.5. In particular, we can take J and K to be smaller than suggested by the bounds in our guarantee, which is good for runtime considerations. (In fact, taking $\eta = 1$ in Theorem 5.5 delivers a lower bound on K that is greater than 16, meaning our theoretical guarantees are off the scale in this plot.) Figure 6(top-right) illustrates the performance of our entire algorithm for different choices of J and K. Notably, we perfectly computed Ax in all of our 1000 random trials when $K = 2$ and $J = 375$. Figure 6(bottom) illustrates the runtime of our algorithm relative to naive matrix–vector multiplication. For this plot, we divide the runtime of our algorithm by the runtime of the naive algorithm. Throughout, gray denotes choices

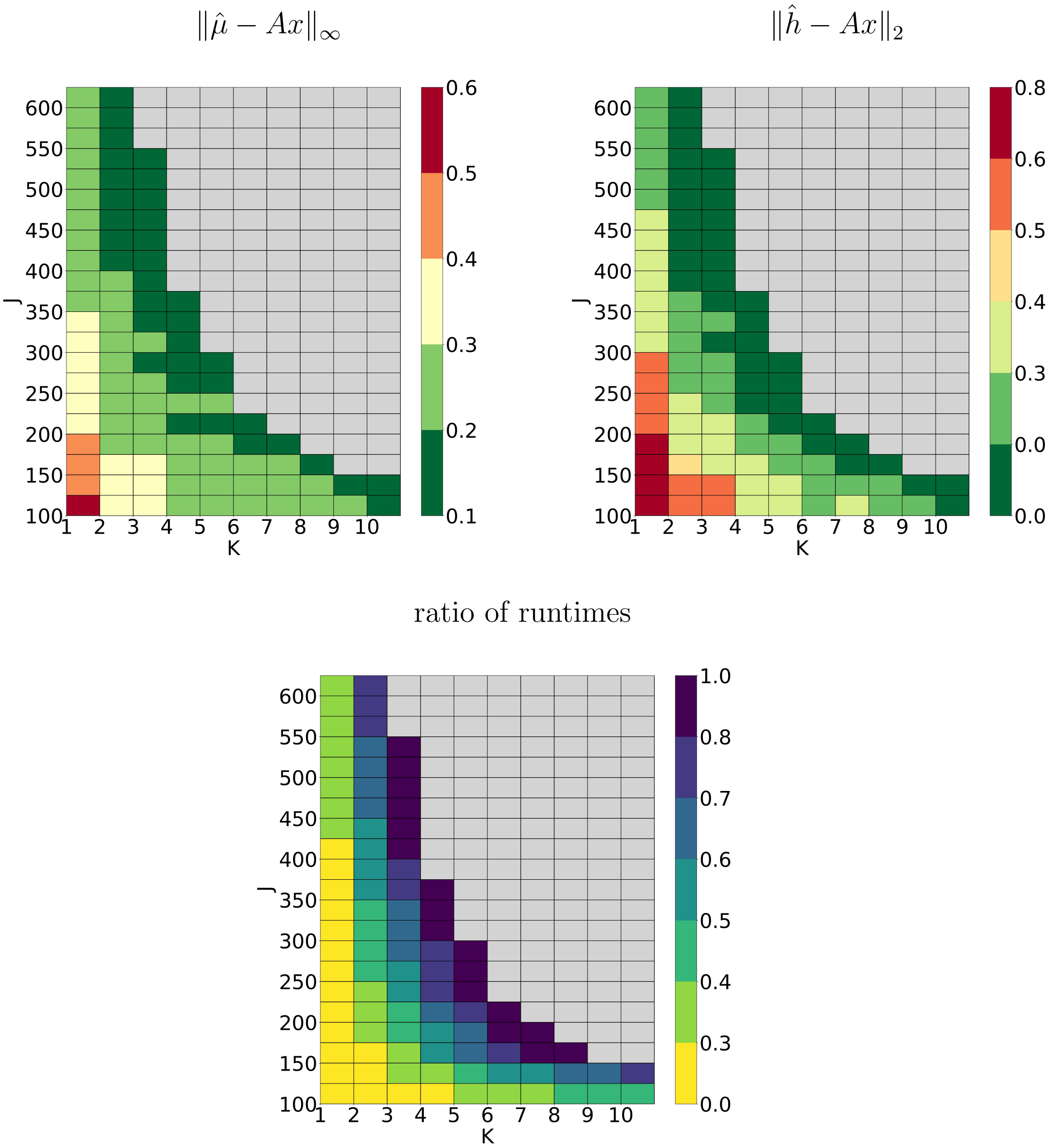

Figure 6: Performance of fast sparsifying transform for a random orthogonal matrix $A \in \mathbb{R}^{n \times n}$ with $n = 4096$ and vectors $x \in \Sigma(A, s, \delta)$ with $s = 20$ and $\delta = 0$. **(top-left)** We compute the median-of-means estimator $\hat{\mu} \in \mathbb{R}^n$ and plot the worst-case behavior over 1000 trials. **(top-right)** We compute $\hat{h} \in \mathbb{R}^n$ by multiplying x by the rows of A corresponding to the $10s$ largest-magnitude entries of $\hat{\mu}$, and we plot the worst-case behavior over 1000 trials. **(bottom)** We plot the quotient of our algorithm's runtime with the runtime of naive matrix–vector multiplication. (Here, we ignore the runtime of randomly sampling N vectors from the precomputed sketch $\{As_\ell\}_{\ell \in [L]}$ since optimizing data structures is beyond the scope of this work.) Throughout, gray denotes choices of parameters for which our algorithm is no faster than the naive algorithm.

of (K, L) for which our algorithm provides no speedup over the naive algorithm. In particular, when $K = 2$ and $J = 375$, our method is about twice as fast as the naive approach. Interestingly, the median-of-means estimator reduces to the empirical mean when $K = 2$; this might suggest an opportunity to improve our theory, and perhaps even speed up our algorithm.

6 Greedy-Type Sparse Recovery from Heavy-Tailed Measurements

Subsections 6.1-6.5 correspond to the publication [34] with slight adaptations. Subsection 6.6 demonstrates an adaptation of the CoSaMP-algorithm using the idea of an iterative median-of-mean algorithm outlined in this publication. Further, a numerical analysis of this algorithm is shown and a proof sketch is discussed.

After extensive discussions about the advantages (but also limitations) of median-of-means in the context of sparsifying transforms during the project presented in the last section, Felix Krahmer and I explored further applications where substituting a sample mean by a median-of-means could be beneficial. As the sample mean is at the core of basically every established recovery algorithm in compressed sensing, this was the main target of my last research project before finalizing this dissertation. Refining early discussions with Richard Kueng, we published the most fundamental approach outlining the main idea of using median-of-means for compressed sensing and continued exploring more complex recovery algorithms and recovery guarantees afterwards. Further numerical results and proof ideas are discussed in subsection 6.6. After completing my research at Felix Krahmer's research group, Anna Veselovska and Felix Krahmer have taken over the continuation of this project.

6.1 Introduction

Motivated by various applications in signal processing and the publications of Candès, Romberg, Tao, and Donoho [15, 26], a variety of research in the field of compressed sensing, targeting the recovery of a sparse signal from a small number of measurements, has been established.

In the following, it is assumed that the measurements are of the form

$$y_j = a^{(j)*}x \qquad \longleftrightarrow \qquad y = Ax,$$

where $x \in \mathbb{C}^n$ denotes the s-sparse signal, $y \in \mathbb{C}^m$ the measurement vector with $m \ll n$, and $A \in \mathbb{C}^{m \times n}$ a random measurement matrix with $\mathbb{E}[A^*Ax] = x$.

Besides algorithms for solving the initially proposed Basis Pursuit ($\min \|\hat{x}\|_1$ s.t. $A\hat{x} = y$), more efficient greedy algorithms as the Orthogonal Matching Pursuit (OMP)[77], Compressive Sampling Matching Pursuit (CoSaMP) [67] or Iterative Hard Thresholding [9] have been established. However, those methods are based on a strong concentration of A^*Ax around x, namely, the Restricted Isometry Property (RIP).

Requiring a strong concentration of A^*Ax is equivalent to requiring a strong concentration of the sample mean of m i.i.d. random variables $X^{(j)} := ma^{(j)}a^{(j)*}x$ around their mean $\mathbb{E}[X^{(j)}] = x$:

$$A^*Ax = \sum_{j=1}^{m} a^{(j)}a^{(j)*}x =: \frac{1}{m}\sum_{j=1}^{m} X^{(j)} =: \bar{X}$$

For Gaussian measurement matrices (and other well-concentrated distributions), comparably sharp tail bounds for $|\bar{X}_i - x_i|$ exist. However, this is a major challenge for heavy-tailed distributions or in scenarios with only limited knowledge about the underlying distribution.

In [25], the authors pointed out the difficulties of an RIP-based analysis for matrices with weak concentration and instead established a new, ℓ_1-specific technique to obtain recovery guarantees for the Basis Pursuit covering heavy-tailed matrices. However, their theory is not applicable to greedy algorithms. To the best of our knowledge, there are no successful recovery guarantees for greedy algorithms for heavy-tailed matrices.

In section 6.2, the median-of-means is introduced as a viable alternative to the mean, and a subroutine based on this estimator is presented. In section 6.3, this subroutine is then expanded to an iterative algorithm – the main contribution of this work. The performance of this algorithm is then presented in section 6.4 and possible improvements are discussed.

6.2 Median-of-Means

By definition, heavy-tailed distributions have a significantly higher probability for outliers, which negatively affects the sample mean and, as a consequence, prevents successful recovery guarantees for greedy algorithms. For that reason, we suggest replacing the inadequately concentrating sample mean by a more robust median-of-means estimator $\hat{\mu}$.

For computing $\hat{\mu}$, the m measurements have to be split into K subsets of size J. In the next step, the sample mean $\bar{X}^{(k)}$ of every subset has to be computed.

$$\underbrace{\begin{bmatrix} X^{(1,1)} \\ \vdots \\ X^{(J,1)} \end{bmatrix}}_{\Rightarrow \bar{X}^{(1)}} \quad \underbrace{\begin{bmatrix} X^{(1,2)} \\ \vdots \\ X^{(J,2)} \end{bmatrix}}_{\Rightarrow \bar{X}^{(2)}} \quad \cdots \quad \underbrace{\begin{bmatrix} X^{(1,K)} \\ \vdots \\ X^{(J,K)} \end{bmatrix}}_{\Rightarrow \bar{X}^{(K)}}$$

By taking the entrywise median (in the complex case separately for the real and imaginary part) over all sample means $\bar{X}^{(1)}, \dots, \bar{X}^{(K)}$, the median-of-means estimator $\hat{\mu}$ is obtained. As the median is very robust against outliers, $\hat{\mu}$ even exhibits an exponential concentration in K:

Lemma 6.1. *Assume a random variable X_i has mean $\mathbb{E}[X_i] = x_i$ and variance $\mathbb{V}[X_i] \leq \sigma^2 \|x\|_2^2 < \infty$. Then, the median-of-means estimator $\hat{\mu}_i$, defined as*

$$\hat{\mu}_i = \mathit{median}\{\bar{X}_i^{(1)}, \dots, \bar{X}_i^{(K)}\} \quad \mathit{with}\ \bar{X}_i^{(k)} = \frac{1}{J} \sum_{j=1}^{J} X_i^{(j,k)},$$

fulfills

$$\mathbb{P}(|\hat{\mu}_i - x_i| \geq \gamma) \leq 2e^{-K/2}$$

if $J \geq \frac{2e^2\sigma^2\|x\|_2^2}{\gamma^2}$.

Proof. The proof follows the proof idea of [33, Theorem 5] with appropriate adaptations.

– **Case I:** x, y, A **are real**:

By assumption,

$$\mathbb{V}[\bar{X}_i] = \frac{1}{J^2} \sum_{j=1}^{J} \mathbb{V}[X_i^{(j)}] \leq \frac{\sigma^2 \|x\|_2^2}{J}.$$

By applying Chebyshev's inequality, the following tail bound is obtained

$$p_J := \mathbb{P}(|\bar{X}_i - x_i| \geq \gamma) \leq \frac{\mathbb{V}[\bar{X}_i]}{\gamma^2} \leq \frac{\sigma^2 \|x\|_2^2}{J\gamma^2}.$$

For every $k \in [K]$, one can define the Bernoulli random variable $I^{(k)} := \mathbf{1}\{|\bar{X}_i^{(k)} - x| \geq \gamma\}$ with parameter p_J. By the definition of the median, $|\hat{\mu}_i - x_i| \geq \gamma$ can only be fulfilled if either at least half of the $\bar{X}_i^{(k)}$ are larger than $x_i + \gamma$ or at least half of them are smaller than $x_i - \gamma$. Therefore,

$$\mathbb{P}\left(|\hat{\mu}_i - x_i| \geq \gamma\right) \leq \mathbb{P}(\sum_{k=1}^{K} I^{(k)} \geq \frac{K}{2}).$$

Applying the multiplicative Chernoff bound, yields

$$\begin{aligned} \mathbb{P}\left(\sum_{k=1}^{K} I^{(k)} \geq (1+\lambda)Kp\right) &\leq \left(\frac{e^{\lambda}}{(1+\lambda)^{1+\lambda}}\right)^{Kp} \\ &= e^{-Kp}\left(\frac{e}{1+\lambda}\right)^{(1+\lambda)Kp}, \qquad \lambda > 0. \end{aligned}$$

By assumption, $J \geq \frac{2e^2\sigma^2\|x\|_2^2}{\gamma^2}$ and therefore $p_J \leq \frac{1}{2e^2}$. Choosing $(1+\lambda)Kp = K/2$

concludes the proof for the real case:

$$\begin{aligned}\mathbb{P}(|\hat{\mu}_i - x_i| \geq \gamma) \leq e^{-Kp_J}(2ep_J)^{K/2} &\leq (2ep_J)^{K/2} \\ &\leq \left(\frac{2e\sigma^2\|x\|_2^2}{J\gamma^2}\right)^{K/2} \leq e^{-K/2}.\end{aligned}$$

– **Case II:** x**,** y**,** A **are complex**:

Denote by $\Re(x_i)$ the real part and by $\Im(x_i)$ the imaginary part of x_i. By our definition, the median over a complex set has to be taken separately for the real part and imaginary part of its elements. Therefore, $\hat{\mu}_i =: \Re(\hat{\mu}_i) + i\Im(\hat{\mu}_i)$, where $\Re(\hat{\mu}_i) = \text{median}\{\Re(\bar{X}_i^{(1)}), \ldots, \Re(\bar{X}_i^{(K)})\}$ (resp. for $\Im(\hat{\mu}_i)$).

By triangle inequality and union bound,

$$\begin{aligned}&\mathbb{P}(|\hat{\mu}_i - x_i| \geq \gamma) \\ &\leq \mathbb{P}(|\Re(\hat{\mu}_i) - \Re(x_i)| + |\Im(\hat{\mu}_i) - \Im(x_i)| \geq \gamma) \\ &\leq \mathbb{P}(|\Re(\hat{\mu}_i) - \Re(x_i)| \geq \gamma) + \mathbb{P}(|\Im(\hat{\mu}_i) - \Im(x_i)| \geq \gamma) \\ &\leq 2e^{-K/2}.\end{aligned}$$

As $\mathbb{P}(|\Re(\bar{X}_i - x_i)| \geq \gamma) \leq \mathbb{P}(|\bar{X}_i - x_i| \geq \gamma)$ (resp. for $\Im$), the bound for the real case above holds for both summands in the second line separately, which concludes the proof. □

Corollary 6.2. *Assume* $X_1, \ldots, X_n$ *are random variables with mean* $\mathbb{E}[X_i] = x_i$ *and variance* $\mathbb{V}[X_i] \leq \sigma^2 < \infty$ *for all* $i \in [n]$*. Then,*

$$\mathbb{P}(\|\hat{\mu} - x\|_\infty \geq \gamma) \leq \eta$$

for

$$J \geq \frac{2e^2\sigma^2\|x\|_2^2}{\gamma^2} \qquad K \geq 2\log\left(\frac{2n}{\eta}\right).$$

Proof. The theorem follows directly from Lemma 6.1 by choosing K such $2e^{-K/2} \leq \frac{\eta}{n}$ and applying a union bound over all $i \in [n]$. □

Algorithm 6 Approximation from random measurements via median-of-means

Require: Measurement matrix $A \in \mathbb{C}^{m\times n}$ and vector of measurements $y \in \mathbb{C}^m$ with $m = JK$;

Ensure: Approximation $\hat{\mu}$ of the s-sparse signal $x \in \mathbb{C}^n$ fulfilling $\|\hat{\mu} - x\|_\infty < \gamma$ with high probability.

```
function MoM($y; A; J; K$)
    Split $A$ in matrices $A^{(k)} \in \mathbb{C}^{J\times n}$ and $y$ in corresponding vectors $y^{(k)} \in \mathbb{C}^J$ $\forall k \in [K]$.
    for $k = 1$ to $K$ do
        Compute $\bar{X}^{(k)} = \frac{m}{J} A^{(k)*} y^{(k)}$
    end for
    return median$\{\bar{X}^{(1)}, \dots, \bar{X}^{(K)}\}$
end function
```

Theorem 6.3. *Let $x \in \mathbb{C}^n$ be a signal, $A \in \mathbb{C}^{m\times n}$ a random measurement matrix with centered i.i.d. entries with moments*

$$\mathbb{E}[|a_i^{(j)}|^2] = \frac{1}{m} \quad \textit{and} \quad \mathbb{E}[|a_i^{(j)}|^4] \leq \frac{\sigma^2}{m^2} < \infty,$$

and $y := (Ax) \in \mathbb{C}^m$ the corresponding measurement vector.

Then, the output $\hat{\mu}$ of Algorithm 6 with J and K as in Corollary 6.2 and $m \geq JK$ fulfills

$$\|\hat{\mu} - x\|_\infty < \gamma$$

with probability $1 - \eta$.

Proof. This follows directly from Corollary 6.2 as

$$\begin{aligned}\mathbb{E}[X_i^{(j)}] &= m\mathbb{E}[e_i^* a^{(j)} a^{(j)*} x] \\ &= \underbrace{m\mathbb{E}[|a_i^{(j)}|^2]x_i}_{=x_i} + m \underbrace{\mathbb{E}[a_i^{(j)}]}_{=0} \mathbb{E}[\textstyle\sum_{l\neq i} \bar{a}_l^{(j)} x_l] = x_i\end{aligned}$$

and

$$\begin{aligned}\mathbb{V}[X_i^{(j)}] &= \mathbb{E}[|X_i^{(j)}|^2] - |\mathbb{E}[X_i^{(j)}]|^2 \\ &= \underbrace{m^2\mathbb{E}[|a_i^{(j)}|^4]|x_i|^2}_{\leq \sigma^2 |x_i|^2} + \underbrace{m\mathbb{E}[|a_i^{(j)}|^2]}_{=1} \underbrace{m\mathbb{E}[|\textstyle\sum_{l\neq i} \bar{a}_l^{(j)} x_l|^2]}_{=m\sum_{l\neq i}\mathbb{E}[|\bar{a}_l^{(j)}|^2]|x_l|^2} - |x_i|^2 \\ &\leq (\sigma^2 - 1)|x_i|^2 + \textstyle\sum_{l\neq i} |x_l|^2 \leq \sigma^2 \|x\|_2^2.\end{aligned}$$

The last inequality holds as $\mathbb{E}[|a_i^{(j)}|^4] \geq \mathbb{E}[|a_i^{(j)}|^2]^2$ (Jensen's inequality) $\Rightarrow \sigma^2 \geq 1$. □

Remark 6.4. *Even if the original measurement matrix does not fulfill* $\mathbb{E}[|a_i^{(j)}|^2] = \frac{1}{m}$, *matrix and measurements can be scaled to fulfill the corresponding requirement of Theorem 6.3 as long as the fourth moment is bounded and the second moment is known.*

6.3 Iterative Median-of-Means Algorithm

As proven in Theorem 6.3, the approximation $\hat{\mu}(x)$ obtained by Algorithm 6 fulfills $\|\hat{\mu}(x) - x\|_\infty < \gamma$ with high probability. While the ℓ_∞-bound can be used to identify large entries of x, the naive ℓ_2-bound exhibits an undesirable scaling in n:

$$\begin{aligned} \|\hat{\mu}(x) - x\|_2 &= \sqrt{\sum_{i=1}^{n} |\hat{\mu}_i(x) - x_i|^2} \\ &\leq \sqrt{n}\|\hat{\mu}(x) - x\|_\infty \leq \sqrt{n}\gamma \end{aligned}$$

The scaling in n can be reduced to a scaling in s by applying an entrywise hard-thresholding operator

$$h_\gamma(\hat{\mu})_i := h_\gamma(\hat{\mu}_i) := \begin{cases} \hat{\mu}_i & \text{for } |\hat{\mu}_i| \geq \gamma \\ 0 & \text{for } |\hat{\mu}_i| < \gamma. \end{cases}$$

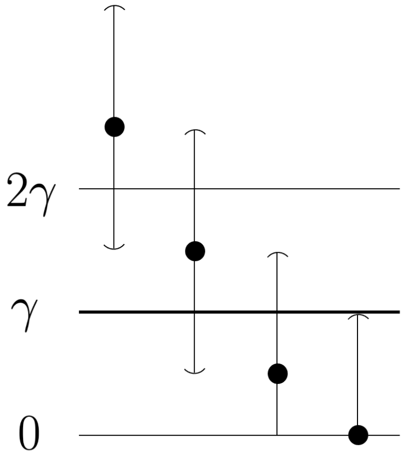

Figure 7: Visualization of the possible deviation of $|\hat{\mu}|$ from $|x|$ for a precision of γ. The black dots symbolize the entries of x, while the entries of $\hat{\mu}_i(x)$ lie within the open intervals.

In Figure 7, the possible intervals of the deviations of $\hat{\mu}(x)$ from x are visualized.

As $\|\hat{\mu}(x) - x\|_\infty < \gamma$, $x_i = 0$ implies $h_\gamma(\hat{\mu}_i(x)) = 0$, and, further,

$$\operatorname{supp}(h_\gamma(\hat{\mu}(x))) \subseteq \operatorname{supp}(x). \tag{6.1}$$

No effect can be seen for $|x_i| > 2\gamma$ which implies $|\hat{\mu}_i(x)| > \gamma$, and therefore, $h_\gamma(\hat{\mu}_i(x)) = \hat{\mu}_i(x)$.

While the last two properties are beneficial, applying the thresholding operator can increase the ℓ_∞-error for $|x_i| \in [\gamma, 2\gamma)$ with $|\hat{\mu}_i(x)| < \gamma$, and therefore, $h_\gamma(\hat{\mu}_i(x)) = 0$ which doubles the ℓ_∞ bound $|h_\gamma(\hat{\mu}_i(x)) - x_i| < 2\gamma$.

Combined, this leads to the ℓ_2-bound

$$\begin{aligned} \|h_\gamma(\hat{\mu}(x)) - x\|_2 &= \sqrt{\sum_{i \in \operatorname{supp}(h_\gamma(\hat{\mu}(x)))} |h_\gamma(\hat{\mu}_i(x)) - x_i|^2} \\ &\le \sqrt{s}\|h_\gamma(\hat{\mu}(x)) - x\|_\infty \le \sqrt{s}2\gamma. \end{aligned} \tag{6.2}$$

Due to the strong scaling of $m \in \mathcal{O}(\frac{1}{\gamma^2})$ in γ, a small ℓ_2-norm can only be achieved by a large increase of the number of measurements.

Instead, an iterative procedure will be defined which allows for an increasing precision while keeping J constant. For simplicity, assume that x has unit norm (i.e., $\|x\|_2 = 1$). Setting $x^{(1)} = h_\gamma(\hat{\mu}(x))$, in the second iteration not x but $x - x^{(1)}$ has to be recovered. By Eq. 6.1, the sparsity is still bounded by s, while, by Eq. 6.2, the ℓ_2-norm is bounded by $\sqrt{s}2\gamma$.

So, in order to obtain an approximation $\hat{\mu}(x - x^{(1)})$ with a precision of $\alpha\gamma$ (for an $\alpha \in (0,1)$) while keeping J constant, the following inequality has to be fulfilled:

$$J \ge \frac{\overbrace{\|x\|_2^2}^{=1}}{\gamma^2} \overset{!}{\ge} \frac{\overbrace{\|x - x^{(1)}\|_2^2}^{s(2\gamma)^2 \ge}}{(\alpha\gamma)^2} \qquad \Rightarrow \gamma \le \alpha\frac{1}{2\sqrt{s}}.$$

Therefore, set $\gamma := \alpha\frac{1}{2\sqrt{s}}$ (the choice of α will be discussed in Remark 6.6).

A last issue has to be addressed: Theorem 6.3 assumes a fixed x which is independent of A. $\hat{\mu}(x)$, and consequently, $x - x^{(1)}$ does not fulfill the independence on A. Therefore, A and y have to be partitioned into L blocks, where L denotes the number of iterations. Due to this, the current approximation and the next block will always be independent. Defining $x^{(l)} = x^{(l-1)} + h_{\alpha^{l-1}\gamma}(\hat{\mu}(x - x^{(l-1)}))$ recursively, leads to the iterative Algorithm 7 and the main result, Theorem 6.5.

Algorithm 7 Approximation from random measurements via iterative median-of-means

Require: Measurement matrix $A \in \mathbb{C}^{m \times n}$ and vector of measurements $y \in \mathbb{C}^m$ with $m = JKL$; $\alpha \in (0, 1)$.

Ensure: Approximation $\hat{x}$ of the s-sparse signal $x \in \mathbb{C}^n$ fulfilling $\|x - \hat{x}\|_2 \leq \alpha^L \|x\|_2$.

```
function ITERATIVE-MOM($y; A; N; K; L; \alpha$)
    Split $A$ in matrices $A^{(k,l)} \in \mathbb{C}^{J \times n}$ and $y$ in corresponding vectors $y^{(k,l)} \in \mathbb{C}^J$ $\forall k \in [K], l \in [L]$.
    Set $x^{(0)} = 0$
    for $l = 1$ to $L$ do
        for $k = 1$ to $K$ do
            Compute $\bar{X}^{(k)} = \frac{m}{J} A^{(k,l)*}(y^{(k,l)} - A^{(k,l)} x^{(l-1)})$
        end for
        $\hat{\mu} = \text{median}\{\bar{X}^{(1)}, \ldots, \bar{X}^{(K)}\}$
        $x^{(l)} = x^{(l-1)} + h_{\alpha^l \frac{\|x\|_2}{2\sqrt{s}}}(\hat{\mu})$
    end for
    return $x^{(L)}$
end function
```

Theorem 6.5. *Let $x \in \mathbb{C}^n$ be an s-sparse signal with unit norm, $A \in \mathbb{C}^{m \times n}$ a random measurement matrix with centered i.i.d. entries with moments*

$$\mathbb{E}[|a_i^{(j)}|^2] = \frac{1}{m} \quad \textit{and} \quad \mathbb{E}[|a_i^{(j)}|^4] \leq \frac{\sigma^2}{m^2} < \infty,$$

and $y := (Ax) \in \mathbb{C}^m$ the corresponding measurement vector.

Then, the output $\hat{x}$ of Algorithm 7 fulfills

$$\|\hat{x} - x\|_2 < \epsilon$$

with probability $1-\eta$ *if* $m \geq JKL$ *and*

$$J \geq \mathbf{s}\frac{8e^2\sigma^2}{\alpha^2} \quad K \geq 2\log(\mathbf{n}\frac{2L}{\eta}) \quad L \geq \frac{\log(\epsilon)}{\log(\alpha)} \quad \alpha \in (0,1).$$

Proof. By induction,

$$\operatorname{supp}(x - x^{(l)}) \subseteq \cdots \subseteq \operatorname{supp}(x - x^{(1)}) \subseteq \operatorname{supp}(x)$$
$$\|x - x^{(l)}\|_2 \leq \sqrt{s}2(\alpha^{l-1}\gamma) = \alpha^l.$$

The choice of L guarantees $\alpha^L \leq \epsilon$, while the slight adaptation of K is the result of a union bound over all L iterations. The proof follows now directly from Theorem 6.3 as $x - x^{(l-1)}$ is independent of $A^{(l)}$ and $y^{(l)}$ for all $l \in [L]$, due to the partitioning of A and y. □

Remark 6.6. *As mentioned before, a decrease of* γ *strongly increases* J*, and consequently, the number of measurements. As* $\gamma := \alpha\frac{1}{2\sqrt{s}}$*,* α *cannot be chosen too small. On the other hand, an* α *close to 1 increases* L*, the number of iterations. Minimizing over the product* JL*,* $\alpha = \frac{1}{\sqrt{e}}$ *is obtained.*

6.4 Numerical Analysis

In the following, the numerical performance of Algorithm 7 will be analyzed.

The entries of A are chosen to be i.i.d. Student's t distributed with 5 degrees of freedom, and are then scaled to fulfill the requirements of Theorem 7, which leads to $\sigma^2 = 9$.

For a dimension of $n = 2000$ and sparsity $s = 10$, the required number of measurements in Theorem 7 appeared to be too large. Instead, we chose $J = 160$ and $K = 7$. The parameter α is set to $\frac{1}{\sqrt{e}}$ as suggested by Remark 6.6. The sparse vector x is chosen to have unit norm with s linearly increasing entries (from approx. 0.05 to 0.5) on random positions.

Then, Algorithm 7 has been performed 10 times for different matrices A and the worst result for every step has been plotted in Figure 8. Despite the significantly lower values for J and K, the ℓ_2-error $\|x^{(l)} - x\|_2$ of the iterates of our algorithm (blue) stayed consistently below the theoretical bound α^l (red).

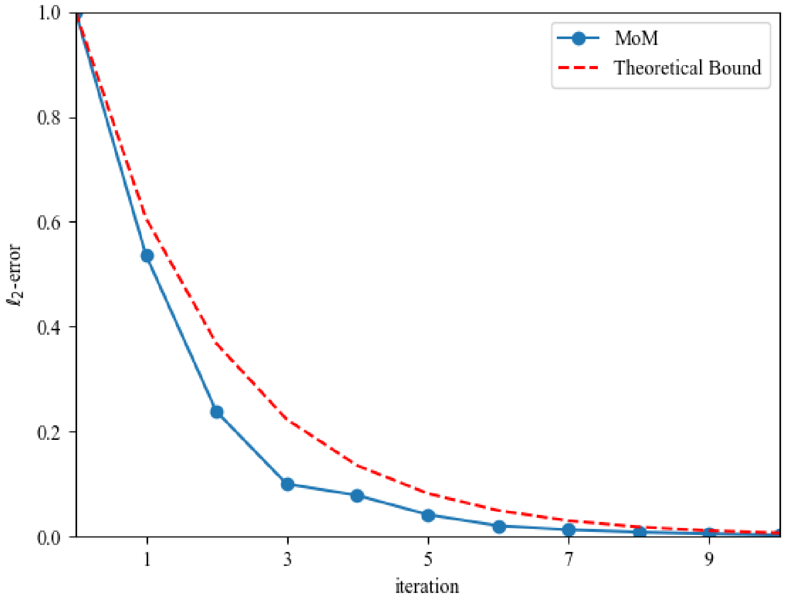

Figure 8: Comparison between $\|x^{(l)} - x\|_2$ and its theoretical bound α^l for $x^{(l)}$ obtained by Algorithm 7.
(max. ℓ_2-error of every iteration for 10 different matrices A)

In Figure 9, the performance of two modifications of Algorithm 7 can be seen. For a good comparison, each of the three methods is applied to the same A and y. As explained in the last section, A and y have to be partitioned to guarantee the necessary independence between underlying signal and measurement matrix. If only one of those samples is used for every iteration, the algorithm appears way more unstable and often fails (orange), which indicates that the required independence is not only a proof artifact. Nevertheless, without partitioning, a larger number of measurements could be used for every iteration - a trade-off which remains subject of further research.

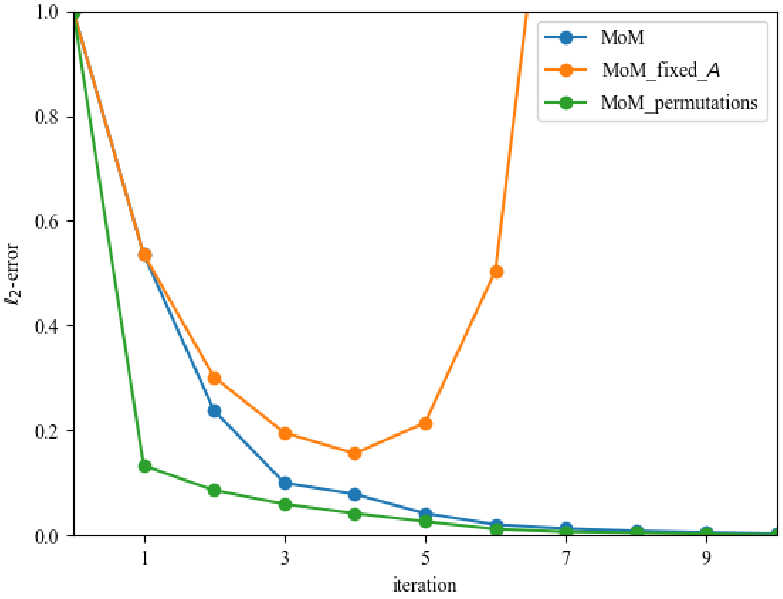

Figure 9: Comparison between the ℓ_2-error of the iterates of Algorithm 7 and two modifications of the algorithm. For the orange results, $A^{(l)}$ is fixed for all iterations. For the green line, in every iteration, the median over 20 median-of-means estimates for different permutations of the measurements is taken.
(max. ℓ_2-error of every iteration for 10 different matrices A)

The second modification targets an underlying weakness of the median-of-means estimator. Different to mean or median, the median-of-means of a data set potentially changes when changing the order of the samples. If all outliers end up in only few of the K subsets, the median-of-means will most likely not be affected by them. If they are distributed over all K subsets and affect all means, the median-of-means should be affected as well. To compensate for this, the median-of-means is not only computed for one single ordering of the measurements. Instead, the median-of-means has to be computed again for multiple different permutations of the measurements. The 'improved' estimate is then obtained by taking the median over those median-of-means. While even further improvements can be expected for a larger number of permutations, we restricted ourselves to only 20 random permutations for performance reasons.

As indicated by the graph, this leads to a significant increase in performance in the first iteration. The performance of further steps might be restricted by the slow decrease of the threshold (i.e., x might already be recovered with high precision, but smaller values of the support of x are still set to 0 by the high threshold). During our work on this modification, there appeared two preprints of Stanislav Minsker [61, 62] showing a significantly improved constant in the tail bound compared to the standard median-of-means. While this does not affect the scaling of the required number of measurements with the dimension n or sparsity s, those results can significantly improve the applicability and runtime of our algorithm for real world scenarios.

Using this modification to expand our theory to uniform guarantees is the topic of ongoing research.

6.5 Conclusion

The greedy algorithm presented in this work reliably approximates an s-sparse vector from random measurements while requiring a comparably small number of measurements. The big advantage of the presented method – besides the efficient implementation – is the lack of strong concentration requirements on the measurement matrix. As long as the fourth moment can be bounded, our algorithm will provably work for any centered measurement matrix A.

Furthermore, an additional performance increase of a modified median-of-means estimator has been demonstrated empirically in the last section.

As listed in the introduction, there is a variety of greedy algorithms for recovering sparse signals which are based on the concentration of A^*Ax. We are convinced that the algorithm presented here is only one example where replacing the sample mean by the median-of-means is beneficial and suggests further research for different, more involved recovery algorithms.

6.6 Iterative Median-of-Means for CoSaMP

The possible advantage of estimating the expectation of a weakly concentrating random variable via median-of-means instead of sample mean has been explained extensively in section 2.4. The main contribution of [34] was applying this theory for sparse approximation with a heavy-tailed measurement matrix A.

The construction of Algorithm 7 was centered around the idea of defining a random variable based on measurements and measurement matrix which has the desired mean, x, and, then, estimating it via median-of-means. The final iterative algorithm reminds of a projected gradient descent with the added twist of using median-of-means.

Yet, there remains one considerable weakness. By applying a proof technique similar to the one presented in section 5, the probability of a large deviation of any entry of x has been bounded. This has been achieved by establishing a tail bound for the deviation of a single component and – as the entries of $\mu(x)$ are not independent of each other – applying a union bound to obtain a tail bound for the largest deviation of any entry.

While this allows for a tail bound of $\|\hat{\mu} - x\|_\infty$ – depending on the algorithm – such a strong result might not be required. As outlined above, there already exists a successful method for the ℓ_1-minimization for heavy-tailed measurements [25] which also does not require such strong results. Unfortunately, there seems to be no hope of expanding their theory to greedy algorithms.

Nevertheless, when recalling the definition of the well-established CoSaMP algorithm, it is clear that it does not require every single entry to be either large (if in the support of x) or close to zero (if not in the support). Instead, it considers the $2s$-largest entries of $A^*(y - Ax^{(i-1)})$ and adds them to the potential support set. Therefore, there is no need to control every single entry of the current approximation $\hat{\mu}$ via an ℓ_∞-bound, but it would be sufficient to bound the probability of the following two events

- E_1: More than s of the $n - s$ entries of $\hat{\mu}$ which do not correspond to the support of x are larger than $\gamma\|y\|_2$.
- E_2: Entries of $\hat{\mu}$ whose corresponding entries in x are responsible for a significant fraction of $\|x\|_2$ are smaller than $\gamma\|y\|_2$.

Compared to the ℓ_∞-bound, there is hope that those bounds can be established with a significantly lower number of measurements.

After adapting the algorithm using the median-of-means, a numerical analysis of the recovery success of this new method is discussed. Concluding this subsection, a proof idea for bounding the probability of the first event is presented.

Algorithm

Following the theme of this thesis, CoSaMP is centered around a sample mean which is used for selecting a potential support set for the approximation of x. Our adapted version replaces this sample mean by a median-of-means estimator. This change is marked in blue.

Algorithm 2 Compressive Sampling Matching Pursuit (CoSaMP)

```
**Data:** Matrix $A \in \mathbb{C}^{m \times n}$, measurement $y \in \mathbb{C}^m$, sparsity level $s$
**Result:** $s$-sparse approximation of vector $x \in \mathbb{C}^n$
$S^{(0)} = \varnothing$
$x^{(0)} = 0$
**for** $l$ in $1, \ldots, L$ **do**
    $S^{(l)} = \text{supp}(x^{(i-1)}) \cup L_{2s}(A^*(y - Ax^{(i-1)}))$
    $\tilde{x}^{(l)} = \text{argmin}_{x \in \mathbb{R}^n}\{(\|Ax - y\|)_2, \text{supp}(x) \subseteq S^{(l)}\}$
    $x^{(l)} = H_s(\tilde{x}^{(l)})$
**end for**
```

Algorithm 8 Compressive Sampling Matching Pursuit with MoM (CoSaMP-MoM)

```
Data: Matrix $A \in \mathbb{C}^{m \times n}$, measurement $y \in \mathbb{C}^m$, sparsity level $s$
Result: $s$-sparse approximation of vector $x \in \mathbb{C}^n$
$S^{(0)} = \varnothing$
$x^{(0)} = 0$
for $l$ in $1, \dots, L$ do
    Split $A$ in matrices $A^{(k)} \in \mathbb{C}^{J \times n}$ and $y$ in corresponding vectors $y^{(k)} \in \mathbb{C}^J$ $\forall k \in [K]$.
    for $k = 1$ to $K$ do
        Compute $\bar{X}^{(k)} = \frac{m}{J} A^{(k)*} y^{(k)}$
    end for
    $S^{(l)} = \mathrm{supp}(x^{(i-1)}) \cup L_{2s}(\mathrm{median}\{\bar{X}^{(1)}, \dots, \bar{X}^{(K)}\})$
    $\tilde{x}^{(l)} = \mathrm{argmin}_{x \in \mathbb{R}^n} \{(\|Ax - y\|)_2, \mathrm{supp}(x) \subseteq S^{(l)}\}$
    $x^{(l)} = H_s(\tilde{x}^{(l)})$
end for
```

Numerical Analysis

As in the numerical analysis before, $n = 2000$ and $s = 10$ are chosen and A is sampled from a Student's t distribution ($df = 5$). As it can be seen in Figure 9, fixing the matrix A for all iterations, the algorithm already failed the recovery for $J = 160$ and $K = 7$, demonstrating that requiring new measurements for each iteration was not a proof artifact. For CoSaMP, this does not seem to be an issue as can be seen in Figure 10.

To heuristically identify the sufficient number of measurements for a successful recovery, Figure 11 shows the combinations of J and K for which a sparse x has been recovered successfully 10 times within at most 10 iterations – while keeping A fixed for all iterations.

Considering that all 10 recoveries were successful using only 90 measurements ($J = 20, K = 3$), the recovery success of CoSaMP-MoM looks very promising. In the following, the proof idea is outlined and missing parts and issues are addressed.

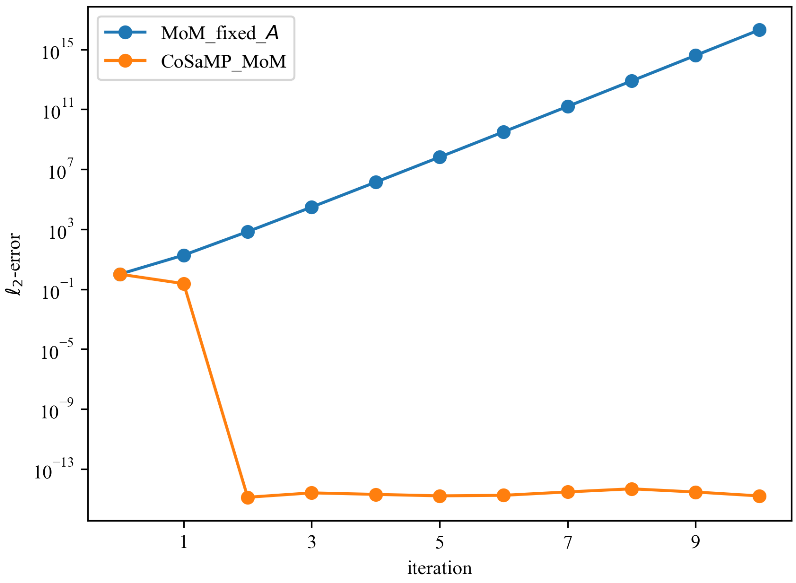

Figure 10: Comparison between the ℓ_2-error of the iterates of Algorithm 7 with a fixed $A^{(l)}$ for all iterations and the modified CoSaMP-MoM. *(max. ℓ_2-error of every iteration for 10 different matrices A)*

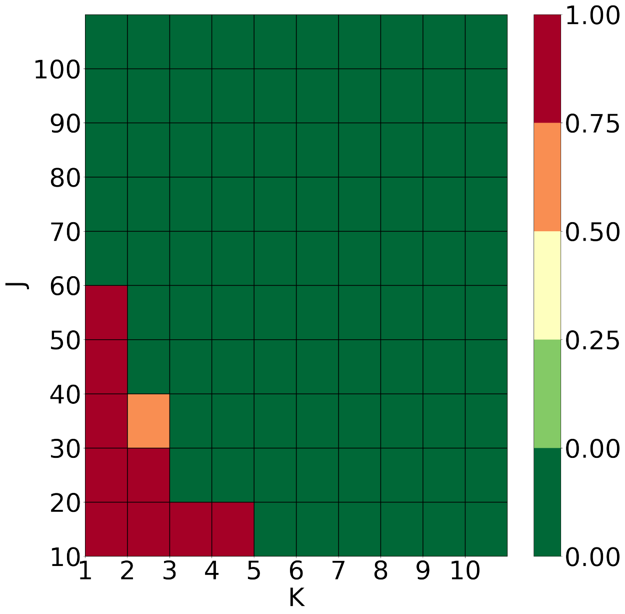

Figure 11: ℓ_2-error after 10 iterations of Algorithm 8 for various combinations of J and K. *(max. ℓ_2-error for 10 different matrices A)*

Proof Idea

The proof idea follows again the technique outlined in section 2.4 with an additional twist to avoid a union bound. By conditioning on the measurements y, an independence between the entries of the approximation corresponding to entries that are not in the support of x can be obtained. Denote $S := \operatorname{supp}(x)$, fix $i \in [n] \setminus S$, and recall

$$e_i^* A^{(1)*} A^{(1)} x = \sum_{j=1}^{J} e_i^* a^{(j)} a^{(j)*} x = \sum_{j=1}^{J} a_i^{(j)} a^{(j)*} x = \sum_{j=1}^{J} a_i^{(j)} a_S^{(j)*} x_S.$$

As all $a_i^{(j)}$ are i.i.d., $a_i^{(j)}$ and $a_S^{(j)*} x_S$ are independent for $i \in [n] \setminus S$. Recalling the split of A and y into K blocks for the median-of-means, we obtain $y_j^{(1)} = a_S^{(j)*} x_S$ and $y^{(1)} = (y_1^{(1)}, \dots, y_J^{(1)})^T = A^{(1)} x$. By conditioning on $y^{(1)}$, all $\sum_{j=1}^{J} a_i^{(j)} y^{(1)} \Big| y^{(1)}$ are independent for all $i \in [n] \setminus S$ and, therefore, allow for expanding a tail bound for one entry to all $i \in [n] \setminus S$ without relying on the union bound.

Using Chebyshev's inequality and the properties of $a_i^{(j)}$, the following conditional probability can be bounded

$$\begin{aligned}
&\mathbb{P}\left(|\frac{m}{J} e_i^* A^{(1)*} y^{(1)}| \geq \gamma \; \Big| y^{(1)} \right) \\
&\leq \frac{E\left[(\frac{m}{J} e_i^* A^{(1)*} y^{(1)})^2 \; \Big| y^{(1)} \right]}{\gamma^2} \\
&= \frac{m^2}{J^2 \gamma^2} E\left[(\sum_{j=1}^{J} a_i^{(j)} y_j^{(1)})^2 \; \Bigg| y^{(1)} \right] \\
&= \frac{m^2}{J^2 \gamma^2} (\underbrace{\sum_{j=1}^{J} \underbrace{E\left[(a_i^{(j)})^2 \right]}_{=\frac{1}{m}} (y_j^{(1)})^2}_{=\frac{1}{m} \|y^{(1)}\|_2^2} + 2 \sum_{1 \leq k < l \leq J} \underbrace{E\left[a_i^{(k)} a_i^{(l)} \right]}_{=0} y_k^{(1)} y_l^{(1)}) \\
&\leq \frac{m \|y^{(1)}\|_2^2}{J^2 \gamma^2} = \frac{K \|y^{(1)}\|_2^2}{J \gamma^2} =: p_1.
\end{aligned}$$

As previously explained, the idea behind median-of-means is based on splitting the measurements into K independent blocks. A fixed entry i of the median-of-means

$\hat{\mu}(x)$ only deviates 'too much' if such a deviation occurs for more than half of the blocks. Therefore, defining

$$I_k := \mathbf{1}\left\{ \left| \frac{m}{J} e_i^* A^{(k)*} y^{(k)} \right| \geq \gamma \; \middle| y^{(k)} \right\},$$

the probability of the deviation of the median-of-means estimator can be bounded again via the multiplicative Chernoff bound. It should be noted that, in contrast to before, the probability of I_k being equal to 1 is bounded by different p_k, which requires slight adjustments and the more general version of the multiplicative Chernoff bound:

$$\begin{aligned}\mathbb{P}\left(\sum_{k=1}^K I_k \geq (1+\lambda) \sum_{k=1}^K p_k \; \middle| y \right) &\leq \left(\frac{e^\lambda}{(1+\lambda)^{1+\lambda}} \right)^{\sum_{k=1}^K p_k} \\ &= e^{-\sum_{k=1}^K p_k} \left(\frac{e}{1+\lambda} \right)^{(1+\lambda)\sum_{k=1}^K p_k}, \qquad \lambda > 0.\end{aligned}$$

Assume $J \geq \frac{2e^2 \|y\|_2^2}{\gamma^2}$ and therefore $\frac{1}{K}\sum_{k=1}^K p_k \leq \frac{1}{2e^2}$. Choosing $(1+\lambda)\sum_{k=1}^K p_k = K/2$, one obtains the bound for a single entry,

$$\begin{aligned}\mathbb{P}(|\hat{\mu}_i| \geq \gamma \; |y) &\leq e^{-\sum_{k=1}^K p_k} \left(2e \frac{1}{K} \sum_{k=1}^K p_k \right)^{K/2} \leq \left(2e \frac{1}{K} \sum_{k=1}^K p_k \right)^{K/2} \\ &= \left(2e \frac{1}{K} \sum_{k=1}^K \frac{K \|y^{(k)}\|_2^2}{J\gamma^2} \right)^{K/2} = \left(\frac{2e\|y\|_2^2}{J\gamma^2} \right)^{K/2} \\ &\leq e^{-K/2}.\end{aligned}$$

The key difference to the proofs in [33] and [34] is that they always required a union bound as the to-be-bounded entries were not independent. Conditioning on y, the entries of $\hat{\mu}_{S^C}$ are indeed independent allowing for a stronger bound of the probability of the corresponding event. There are $\binom{n-s}{s+1}$ combinations of picking $s+1$ entries from the $n-s$ entries not corresponding to the support of x. For each of those variants, the probability of each entry to exceed γ is bounded by $e^{-K/2}$ as shown above. Combined, the probability of the event can now be bounded by

$$\mathbb{P}(E_1 \,|y) \leq \binom{n-s}{s+1}(e^{-K/2})^{s+1} \leq \left(\frac{(n-s)e}{s+1}\right)^{s+1}(e^{-K/2})^{s+1} \leq \eta^{s+1}\left(\frac{ne}{s}\right)^{-(s+1)}.$$

The last inequality can be obtained by picking $K = 2\log\left(\frac{ne}{s}\frac{(n-s)e}{(s+1)\eta}\right) \in \mathcal{O}\left(\log(\frac{n}{s\sqrt{\eta}})\right)$. As the dimension of y is equal to m, keeping such an indirect dependency in $m = JK$ should be avoided. Nevertheless, recalling that $y = Ax = A_S x_S$ and $E[|a_i^{(j)}|^2] = \frac{1}{m}$, a scaling of $\|y\|_2$ depending on s but not on m or n can be expected and, therefore, is not of concern. Hence, summarizing those results and choosing $\gamma = \tilde{\gamma}\|y\|_2$, the following lemma has been proven:

Lemma 6.7. *Let $x \in \mathbb{C}^n$ be a signal, $A \in \mathbb{C}^{m\times n}$ a random measurement matrix with centered i.i.d. entries and $E[|a_i^{(j)}|^2] = \frac{1}{m}$, and $y = Ax \in \mathbb{C}^m$ the corresponding measurement vector. Define*

$$B = \{i \in [n] \setminus S : |\hat{\mu}_i| \geq \gamma\|y\|_2\}.$$

Then, the output $\hat{\mu}$ of Algorithm 8 with $J \geq \frac{2e^2}{\gamma^2}$, $K = 2\log\left(\frac{ne}{s}\frac{(n-s)e}{(s+1)\eta}\right)$, and $m \geq JK$ satisfies

$$\mathbb{P}(|B| > s \,|y) \leq \eta^{s+1}\left(\frac{ne}{s}\right)^{-(s+1)}.$$

Lemma 6.7 exhibits an attractive scaling of the number of measurements with $m \in \mathcal{O}\left(\gamma^{-2}\log(\frac{n}{s\sqrt{\eta}})\right)$. For RIP-based proofs, a bound scaling with $\left(\frac{ne}{s}\right)^{-s}$ usually allows a union bound over all s-sparse vectors. If this is possible also in this setting, the result of Lemma 6.7 could be expanded to hold not only for a fixed vector x but all s-sparse vectors, still with high probability. Continuing this line of thought to the full proof of the CoSaMP-MoM algorithm, the successful recovery, even for a low number of measurements and a fixed matrix A for all iterations (as seen in Figure 11), could be explained. It remains to finalize the union bound argument over all s-sparse vectors and resolve the conditioning on y. Further, the probability of event E_2 needs to be bounded.

The ideas presented in this subsection are based on discussions during my time in Felix Krahmer's research group. Unfortunately, the proof could not be completed before my departure and finalizing this thesis, but Anna Veselovska and Felix Krahmer have taken over the continuation of this project.

7 Conclusion and Outlook

This thesis has addressed significant challenges in the fields of sparse data representation and signal recovery, presenting novel solutions that enhance both computational efficiency and robustness. The research has focused on two primary contributions:

Fast Sparsifying Transforms

A randomized algorithm was developed to compute Ax for any $x \in \mathbb{R}^n$ such that Ax is s-sparse, even for unstructured matrices A. Leveraging spherical designs derived from Kerdock sets and the robustness of the median-of-means estimator, this approach efficiently computes the representation of A during preprocessing. Subsequently, the fast transform computes the entrywise ϵ-hard threshold of Ax with high probability, significantly reducing computational complexity compared to traditional methods. Performance guarantees and numerical results underscore the practical feasibility of this algorithm.

Sparse Recovery from Heavy-Tailed Measurements

Traditional compressed sensing algorithms often struggle with heavy-tailed measurement matrices due to their weak concentration properties. To address this challenge, an adapted greedy algorithm based on the median-of-means estimator was introduced. This method ensures robust recovery of any s-sparse unit vector $x \in \mathbb{C}^n$ with high probability, achieving small ℓ_2-error while imposing minimal assumptions on the measurement matrix A. The successful adaptation of the CoSaMP algorithm and its numerical validation provide practical insights and lay a foundation for further advancements.

In summary, this thesis represents significant advancements in fast sparse transforms and robust compressed sensing. These methodologies not only enhance computational efficiency but also guarantee reliable signal recovery in challenging environments. Future research directions will focus on refining the theoretical underpinnings and extending numerical evaluations to solidify these contributions' impact. Continued development promises to expand the capabilities and applications of sparse models in solving complex real-world problems.

Bibliography

[1] A. Adiga, D. Dubhashi, B. Lewis, M. Marathe, S. Venkatramanan, and A. Vullikanti. “Mathematical models for covid-19 pandemic: a comparative analysis”. In: *Journal of the Indian Institute of Science* 100.4 (2020), pp. 793–807.

[2] M. Aharon, M. Elad, and A. Bruckstein. “K-SVD: An algorithm for designing overcomplete dictionaries for sparse representation”. In: *IEEE Transactions on signal processing* 54.11 (2006), pp. 4311–4322.

[3] N. Alon, Y. Matias, and M. Szegedy. “The space complexity of approximating the frequency moments”. In: *Proceedings of the twenty-eighth annual ACM symposium on Theory of computing*. 1996, pp. 20–29.

[4] A. Ambainis and J. Emerson. “Quantum t-designs: t-wise independence in the quantum world”. In: *Twenty-Second Annual IEEE Conference on Computational Complexity (CCC’07)*. IEEE. 2007, pp. 129–140.

[5] S. Bag, L. C. Wood, L. Xu, P. Dhamija, and Y. Kayikci. “Big data analytics as an operational excellence approach to enhance sustainable supply chain performance”. In: *Resources, conservation and recycling* 153 (2020), p. 104559.

[6] E. Bannai and E. Bannai. “A survey on spherical designs and algebraic combinatorics on spheres”. In: *European Journal of Combinatorics* 30.6 (2009), pp. 1392–1425.

[7] E. Bannai and S. G. Hoggar. “On tight t-designs in compact symmetric spaces of rank one”. In: *Proceedings of the Japan Academy, Series A, Mathematical Sciences* 61.3 (1985), pp. 78–82.

[8] R. G. Baraniuk, T. Goldstein, A. C. Sankaranarayanan, C. Studer, A. Veeraraghavan, and M. B. Wakin. “Compressive video sensing: Algorithms, architectures, and applications”. In: *IEEE Signal Processing Magazine* 34.1 (2017), pp. 52–66.

[9] T. Blumensath and M. E. Davies. “Iterative hard thresholding for compressed sensing”. In: *Applied and computational harmonic analysis* 27.3 (2009), pp. 265–274.

[10] A. Bondarenko, D. Radchenko, and M. Viazovska. “Optimal asymptotic bounds for spherical designs”. In: *Annals of mathematics* (2013), pp. 443–452.

[11] H. Brezis and H. Brézis. *Functional analysis, Sobolev spaces and partial differential equations.* Vol. 2. 3. Springer, 2011.

[12] A. M. Bruckstein, D. L. Donoho, and M. Elad. “From sparse solutions of systems of equations to sparse modeling of signals and images”. In: *SIAM review* 51.1 (2009), pp. 34–81.

[13] A. R. Calderbank, P. J. Cameron, W. M. Kantor, and J. J. Seidel. “$\mathbb{Z}_4$-Kerdock codes, orthogonal spreads, and extremal euclidean line-sets”. In: *Proceedings of the London Mathematical Society* 75.2 (1997), pp. 436–480.

[14] P. J. Cameron and J. J. Seidel. “Quadratic forms over GF (2)”. In: *Geometry and Combinatorics.* Elsevier, 1991, pp. 290–297.

[15] E. J. Candès, J. K. Romberg, and T. Tao. “Stable signal recovery from incomplete and inaccurate measurements”. In: *Communications on Pure and Applied Mathematics: A Journal Issued by the Courant Institute of Mathematical Sciences* 59.8 (2006), pp. 1207–1223.

[16] B. Choi, A. Christlieb, and Y. Wang. “High-dimensional sparse Fourier algorithms”. In: *Numerical Algorithms* 87.1 (2021), pp. 161–186.

[17] B. Choi, A. Christlieb, and Y. Wang. “Multiscale high-dimensional sparse fourier algorithms for noisy data”. In: *arXiv preprint arXiv:1907.03692* (2019).

[18] B. Choi, M. Iwen, and T. Volkmer. “Sparse harmonic transforms ii: Best s-term approximation guarantees for bounded orthonormal product bases in sublinear-time”. In: *Numerische Mathematik* (2021), pp. 1–70.

[19] B. Choi, M. A. Iwen, and F. Krahmer. "Sparse harmonic transforms: A new class of sublinear-time algorithms for learning functions of many variables". In: *Foundations of Computational Mathematics* 21.2 (2021), pp. 275–329.

[20] A. Christlieb, D. Lawlor, and Y. Wang. "A multiscale sub-linear time Fourier algorithm for noisy data". In: *Applied and Computational Harmonic Analysis* 40.3 (2016), pp. 553–574.

[21] D. Coppersmith and S. Winograd. "Matrix multiplication via arithmetic progressions". In: *Proceedings of the nineteenth annual ACM symposium on Theory of computing*. 1987, pp. 1–6.

[22] G. Cormode and S. Muthukrishnan. *Combinatorial algorithms for compressed sensing*. Springer, 2006.

[23] A. M. Davie and A. J. Stothers. "Improved bound for complexity of matrix multiplication". In: *Proceedings of the Royal Society of Edinburgh Section A: Mathematics* 143.2 (2013), pp. 351–369.

[24] P. Delsarte, J.-M. Goethals, and J. J. Seidel. "Spherical codes and designs". In: *Geometry and Combinatorics*. Elsevier, 1991, pp. 68–93.

[25] S. Dirksen, G. Lecué, and H. Rauhut. "On the gap between restricted isometry properties and sparse recovery conditions". In: *IEEE Transactions on Information Theory* 64.8 (2016), pp. 5478–5487.

[26] D. Donoho. "Compressed sensing". In: *IEEE Transactions on Information Theory* 52.4 (2006), pp. 1289–1306.

[27] P. Drineas, R. Kannan, and M. W. Mahoney. "Fast Monte Carlo algorithms for matrices I: Approximating matrix multiplication". In: *SIAM Journal on Computing* 36.1 (2006), pp. 132–157.

[28] M. Elad and M. Aharon. “Image denoising via sparse and redundant representations over learned dictionaries”. In: *IEEE Transactions on Image processing* 15.12 (2006), pp. 3736–3745.

[29] G. B. Folland. “How to integrate a polynomial over a sphere”. In: *The American Mathematical Monthly* 108.5 (2001), pp. 446–448.

[30] S. Foucart and H. Rauhut. *A Mathematical Introduction to Compressive Sensing.* Applied and Numerical Harmonic Analysis. Springer New York, 2013.

[31] S. Foucart. “Hard thresholding pursuit: an algorithm for compressive sensing”. In: *SIAM Journal on numerical analysis* 49.6 (2011), pp. 2543–2563.

[32] T. Fuchs, D. Gross, P. Jung, F. Krahmer, R. Kueng, and D. Stöger. “Proof methods for robust low-rank matrix recovery”. In: *Compressed Sensing in Information Processing.* Springer, 2022, pp. 37–75.

[33] T. Fuchs, D. Gross, F. Krahmer, R. Kueng, and D. Mixon. “Sketching with Kerdock’s Crayons: Fast Sparsifying Transforms for Arbitrary Linear Maps”. In: *SIAM Journal on Matrix Analysis and Applications* 43.2 (2022), 939–952, © 2022 SIAM.

[34] T. Fuchs, F. Krahmer, and R. Kueng. “Greedy-type sparse recovery from heavy-tailed measurements”. In: *2023 International Conference on Sampling Theory and Applications (SampTA).* 2023, 1–5, © 2023 IEEE.

[35] A. C. Gilbert, S. Guha, P. Indyk, S. Muthukrishnan, and M. Strauss. “Near-optimal sparse Fourier representations via sampling”. In: *Proceedings of the thiry-fourth annual ACM symposium on Theory of computing.* 2002, pp. 152–161.

[36] A. C. Gilbert, S. Muthukrishnan, and M. Strauss. “Improved time bounds for near-optimal sparse Fourier representations”. In: *Wavelets XI.* Vol. 5914. International Society for Optics and Photonics. 2005, 59141A.

[37] N. I. Gillespie. “Equiangular lines, incoherent sets and quasi-symmetric designs”. In: *arXiv preprint arXiv:1809.05739* (2018).

[38] H. Hassanieh, P. Indyk, D. Katabi, and E. Price. “Nearly optimal sparse Fourier transform”. In: *Proceedings of the forty-fourth annual ACM symposium on Theory of computing*. 2012, pp. 563–578.

[39] H. Hassanieh, P. Indyk, D. Katabi, and E. Price. “Simple and practical algorithm for sparse Fourier transform”. In: *Proceedings of the twenty-third annual ACM-SIAM symposium on Discrete Algorithms*. SIAM. 2012, pp. 1183–1194.

[40] P. Indyk. “Explicit constructions for compressed sensing of sparse signals.” In: 8 (2008), pp. 30–33.

[41] P. Indyk and M. Kapralov. “Sample-optimal Fourier sampling in any constant dimension”. In: *2014 IEEE 55th Annual Symposium on Foundations of Computer Science*. IEEE. 2014, pp. 514–523.

[42] M. A. Iwen. “Combinatorial sublinear-time Fourier algorithms”. In: *Foundations of Computational Mathematics* 10.3 (2010), pp. 303–338.

[43] M. A. Iwen. “Improved approximation guarantees for sublinear-time Fourier algorithms”. In: *Applied And Computational Harmonic Analysis* 34.1 (2013), pp. 57–82.

[44] M. A. Iwen and C. V. Spencer. “A note on compressed sensing and the complexity of matrix multiplication”. In: *Information Processing Letters* 109.10 (2009), pp. 468–471.

[45] M. R. Jerrum, L. G. Valiant, and V. V. Vazirani. “Random generation of combinatorial structures from a uniform distribution”. In: *Theoretical computer science* 43 (1986), pp. 169–188.

[46] L. Kämmerer, F. Krahmer, and T. Volkmer. “A sample efficient sparse FFT for arbitrary frequency candidate sets in high dimensions”. In: *Numerical Algorithms* (2022), pp. 1–42.

[47] L. Kämmerer, D. Potts, and T. Volkmer. “High-dimensional sparse FFT based on sampling along multiple rank-1 lattices”. In: *Applied and Computational Harmonic Analysis* 51 (2021), pp. 225–257.

[48] W. M. Kantor. “Spreads, translation planes and Kerdock sets. I”. In: *SIAM Journal on Algebraic Discrete Methods* 3.2 (1982), pp. 151–165.

[49] W. Kantor. “Spreads, translation planes and Kerdock sets. II”. In: *SIAM Journal on Algebraic Discrete Methods* 3.3 (1982), pp. 308–318.

[50] D. Lawlor, Y. Wang, and A. Christlieb. “Adaptive sub-linear time Fourier algorithms”. In: *Advances in Adaptive Data Analysis* 5.01 (2013), p. 1350003.

[51] F. Le Gall. “Powers of tensors and fast matrix multiplication”. In: *Proceedings of the 39th international symposium on symbolic and algebraic computation.* 2014, pp. 296–303.

[52] G. Lecué and S. Mendelson. “Sparse recovery under weak moment assumptions”. In: *Journal of the European Mathematical Society* 19.3 (2017), pp. 881–904.

[53] M. Ledoux and M. Talagrand. *Probability in Banach Spaces: isoperimetry and processes.* Springer, 1991.

[54] R. Leipus, J. Šiaulys, and D. Konstantinides. “Heavy-Tailed and Related Classes of Distributions”. In: *Closure Properties for Heavy-Tailed and Related Distributions: An Overview.* Cham: Springer Nature Switzerland, 2023, pp. 7–30.

[55] S. Li, L. Da Xu, and X. Wang. “Compressed sensing signal and data acquisition in wireless sensor networks and internet of things”. In: *IEEE transactions on industrial informatics* 9.4 (2012), pp. 2177–2186.

[56] E. Liberty and S. W. Zucker. "The mailman algorithm: A note on matrix–vector multiplication". In: *Information Processing Letters* 109.3 (2009), pp. 179–182.

[57] M. Lustig, D. Donoho, and J. M. Pauly. "Sparse MRI: The application of compressed sensing for rapid MR imaging". In: *Magnetic Resonance in Medicine: An Official Journal of the International Society for Magnetic Resonance in Medicine* 58.6 (2007), pp. 1182–1195.

[58] H. M. Mahmoud, R. Modarres, and R. T. Smythe. "Analysis of quickselect: An algorithm for order statistics". In: *RAIRO-Theoretical Informatics and Applications-Informatique Théorique et Applications* 29.4 (1995), pp. 255–276.

[59] J. Mairal, F. Bach, and J. Ponce. "Task-driven dictionary learning". In: *IEEE transactions on pattern analysis and machine intelligence* 34.4 (2011), pp. 791–804.

[60] J. Mairal, F. Bach, J. Ponce, and G. Sapiro. "Online learning for matrix factorization and sparse coding." In: *Journal of Machine Learning Research* 11.1 (2010).

[61] S. Minsker. "Efficient median of means estimator". In: *The Thirty Sixth Annual Conference on Learning Theory.* PMLR. 2023, pp. 5925–5933.

[62] S. Minsker. "U-statistics of growing order and sub-gaussian mean estimators with sharp constants". In: *Mathematical Statistics and Learning* 7.1 (2023), pp. 1–39.

[63] L. Morotti. "Explicit universal sampling sets in finite vector spaces". In: *Applied and Computational Harmonic Analysis* 43.2 (2017), pp. 354–369.

[64] R. Motwani and P. Raghavan. *Randomized algorithms.* Cambridge university press, 1995.

[65] A. Munemasa. "Spherical designs". In: *Handbook of Combinatorial Designs.* Chapman and Hall/CRC, 2006, pp. 643–647.

[66] S. Nam, M. E. Davies, M. Elad, and R. Gribonval. “The cosparse analysis model and algorithms”. In: *Applied and Computational Harmonic Analysis* 34.1 (2013), pp. 30–56.

[67] D. Needell and J. Tropp. “CoSaMP: Iterative signal recovery from incomplete and inaccurate samples”. In: *Applied and Computational Harmonic Analysis* 26.3 (2009), pp. 301–321. ISSN: 1063-5203.

[68] A. S. Nemirovskij and D. B. Yudin. *Problem complexity and method efficiency in optimization.* Wiley-Interscience, 1983.

[69] W. K. Newey and D. McFadden. “Large sample estimation and hypothesis testing”. In: *Handbook of econometrics* 4 (1994), pp. 2111–2245.

[70] D. Potts and T. Volkmer. “Fast, exact and stable reconstruction of multivariate algebraic polynomials in Chebyshev form”. In: (2015).

[71] S. Ravishankar and Y. Bresler. “Learning sparsifying transforms”. In: *IEEE Transactions on Signal Processing* 61.5 (2012), pp. 1072–1086.

[72] C. Rusu, N. Gonzalez-Prelcic, and R. W. Heath. “Fast orthonormal sparsifying transforms based on householder reflectors”. In: *IEEE Transactions on Signal Processing* 64.24 (2016), pp. 6589–6599.

[73] C. Rusu and J. Thompson. “Learning fast sparsifying transforms”. In: *IEEE Transactions on Signal Processing* 65.16 (2017), pp. 4367–4378.

[74] V. Strassen. “Gaussian elimination is not optimal”. In: *Numerische mathematik* 13.4 (1969), pp. 354–356.

[75] P. Taylor. *Volume of data/information created, captured, copied, and consumed worldwide from 2010 to 2020, with forecasts from 2021 to 2025.* 2021. URL: `https://www.statista.com/statistics/871513/worldwide-data-created/#statisticContainer` (visited on 05/19/2024).

[76] I. Tošić and P. Frossard. “Dictionary learning”. In: *IEEE Signal Processing Magazine* 28.2 (2011), pp. 27–38.

[77] J. A. Tropp and A. C. Gilbert. “Signal Recovery From Random Measurements Via Orthogonal Matching Pursuit”. In: *IEEE Transactions on Information Theory* 53.12 (2007), pp. 4655–4666.

[78] B. Venkov. “Réseaux et designs sphériques”. In: *Réseaux euclidiens, designs sphériques et formes modulaires* 37 (2001), pp. 10–86.

[79] C. M. Verdun*, T. Fuchs*, et al. “Group testing for SARS-CoV-2 allows for up to 10-fold efficiency increase across realistic scenarios and testing strategies”. In: *Frontiers in Public Health* 9 (2021), p. 583377.

[80] M. J. Wainwright. *High-dimensional statistics: A non-asymptotic viewpoint.* Vol. 48. Cambridge university press, 2019.

[81] S. F. Waldron. *An introduction to finite tight frames.* Springer, 2018.

[82] L. Wasserman. *All of statistics: a concise course in statistical inference.* Vol. 26. Springer, 2004.

[83] R. Williams. “Matrix-vector multiplication in sub-quadratic time:(some preprocessing required)”. In: *SODA*. Vol. 7. 2007, pp. 995–1001.

[84] V. V. Williams. “Multiplying matrices faster than Coppersmith-Winograd”. In: *Proceedings of the forty-fourth annual ACM symposium on Theory of computing.* 2012, pp. 887–898.

[85] R. S. Womersley. “Efficient Spherical Designs with Good Geometric Properties”. In: *Contemporary Computational Mathematics - A Celebration of the 80th Birthday of Ian Sloan.* Ed. by J. Dick, F. Y. Kuo, and H. Woźniakowski. Cham: Springer International Publishing, 2018, pp. 1243–1285.