Efficient One-bit Compressed Sensing

Sidhant Bansal (A0178042H)
Advisor: Prof. Arnab Bhattacharyya

National University of Singapore

Acknowledgement: This research project was done in collaboration with Prof. Jonathan Scarlett and Anamay Chaturvedi.

Outline

1. Introduction
   - Terminology
   - Real-world Application
2. Related Works
   - $O(k^{1.5}\log\frac{n}{k})$ upper bound for $D = 1$
3. Lower Bound for $\{\pm1\}^{m\times n,0.5}$
   - *ℓ-balanced problem*
   - Proof Idea
4. Conclusions and Future Works

Introduction

- Compressed Sensing (CS) helps us overcome the traditional limits of sampling.
- Instead of sampling at every time-step, we take m measurements of the input signal on the sensor itself.
- Use the fact that input signals in many applications (Ex. radar, spectrum sensing) can be represented using very few non-zero coefficients in a suitable basis.

Introduction

- Compressed Sensing (CS) helps us overcome the traditional limits of sampling.
- Instead of sampling at every time-step, we take m measurements of the input signal on the sensor itself.
- Use the fact that input signals in many applications (Ex. radar, spectrum sensing) can be represented using very few non-zero coefficients in a suitable basis.

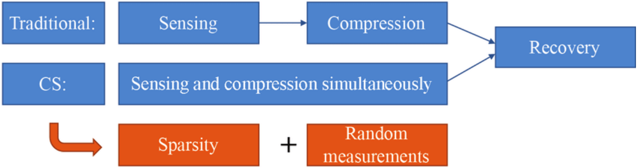

Figure: Differences between traditional sampling and CS. (Credits: In Ma and Yu)

Introduction

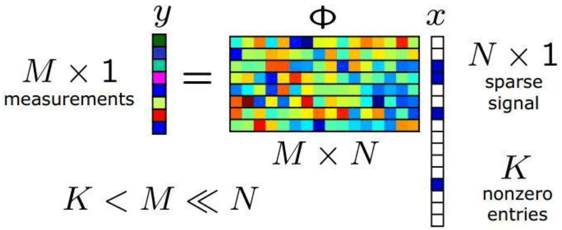

Figure: Compressed Sensing (Credits: Mostafa Morsi and Fakhr)

Questions that arise naturally are:

- How can we compress efficiently?
- How do we recover the original signal back?

Terminology

Definition

A signal $x \in \mathbb{R}^n$ is k-sparse if $\|x\|_0 = \#$non-zero coefficients $\leq k$

Definition

Support vector of a signal is the set of indices which have non-zero entries in the signal.

Why 1-bit CS ?

Definition

Quantization is the process of mapping input values from a large set (often a continuous set) to output values in a smaller set with finite number of elements.

For practical systems, quantization step is essential prior to signal processing, which inevitably introduces quantization error.

Why 1-bit CS ?

Definition

Quantization is the process of mapping input values from a large set (often a continuous set) to output values in a smaller set with finite number of elements.

For practical systems, quantization step is essential prior to signal processing, which inevitably introduces quantization error.
In 1-bit CS we **only preserve the sign information of a measurement** (i.e. just a single bit).

- Allows us to do extreme quantization thus making it robust and tolerant to large quantization error.
- The quantizer which takes the form of a comparator to zero, is cheap and fast hardware device.

CS vs 1-bit CS

-	CS	1-bit CS
Formula	$b = Ax$	$b = sign(Ax)$
Co-domain	$b \in \mathbb{R}^m$	$b \in \{\pm 1\}^m$
Hardware Complexity	Higher	Lower

Remark

$sign : \mathbb{R}^m \to \{\pm 1\}^m$, such that

$$sign(Ax)_i = \begin{cases} 1 & \text{if } (Ax)_i \geq 0 \\ -1 & \text{if } (Ax)_i < 0 \end{cases} \quad \forall i \in [1, m]$$

1-bit CS

Definition

Let $A \in \mathbb{R}^{m \times n}$, then A **is a valid** 1-**bit measurement matrix** iff *there are no collisions in the co-domain for all possible support vectors*

Formally,

$$\forall u \neq v, u \in Y, v \in Y, sign(Au) \neq sign(Av)$$

where Y denotes the set of k-sparse vectors in $\{0, 1\}^n$

1-bit CS

Examples

$$A = \begin{bmatrix} -1 & 0.7 & 0.5 & 7 \\ 2 & -1.3 & 1.3 & 4 \end{bmatrix}, x = \begin{bmatrix} 3 \\ 0 \\ -2 \\ 0 \end{bmatrix}, Support(x) = \begin{bmatrix} 1 \\ 0 \\ 1 \\ 0 \end{bmatrix}$$

$$b = sign(Ax) = \begin{bmatrix} sign(-4) \\ sign(3.4) \end{bmatrix} = \begin{bmatrix} -1 \\ 1 \end{bmatrix}$$

Types of Recovery

1-bit CS is studied under the following two settings regarding the recovery mechanisms:

Definition

Support Recovery: Here we recover the support of the input k-sparse vector x

Definition

Approximate Vector Recovery: Here we recover $\hat{x}$ which is "close" to the input k-sparse x, i.e. $\left\| \frac{x}{\|x\|} - \frac{\hat{x}}{\|\hat{x}\|} \right\| \leq \epsilon$

Assumptions

- **Universal Scheme**: Using a fixed a measurement matrix for all possible k-sparse input signals.
- Is a practical design choice. In many applications (Ex. Single Pixel Camera) it is not feasible to construct a new matrix A for each different input signal.

Assumptions

- **Universal Scheme**: Using a fixed a measurement matrix for all possible k-sparse input signals.
- Is a practical design choice. In many applications (Ex. Single Pixel Camera) it is not feasible to construct a new matrix A for each different input signal.
- **Bounded Dynamic Range**: All the non-zero entries of the k-sparse input signal are within a bounded range, i.e. $D = \frac{\max_{x_i \neq 0} |x_i|}{\min_{x_i \neq 0} |x_i|}$ is a constant.
- Restricts the set of valid k-sparse input signals thus providing better bounds (upper and lower) on m. In practical conditions when we know that the non-zero values of the input signal are *close-by*.

Real-world Application

Body-area-networks (BAN) are a revolution in the healthcare industry. BAN consists of sensors attached to the patient which transmit real-time bio-sensor data to the healthcare professionals via the smartphone of the patient.

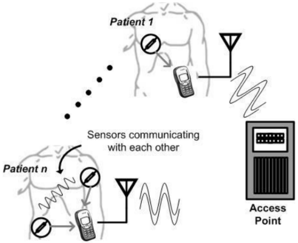

Figure: Body Area Network (BAN) Example (Credits: In Salman et al.)

Real-world Application

The **energy consumed for transmission is proportional to the data sent** so it is ideal to compress the bio-signal as much as possible before its digitisation and transmission.

In Salman et al., universal 1-bit CS under bounded dynamic range achieves

- $16\times$ compression factor
- More energy efficient (compared to CS)

for recording of *Electromyography (EMG) Signals*.

Purpose of EMG

Electromyography (EMG) is a technique used to evaluate and record electrical activity produced by skeletal muscles. It is used to identify neuromuscular diseases.

Properties of EMG

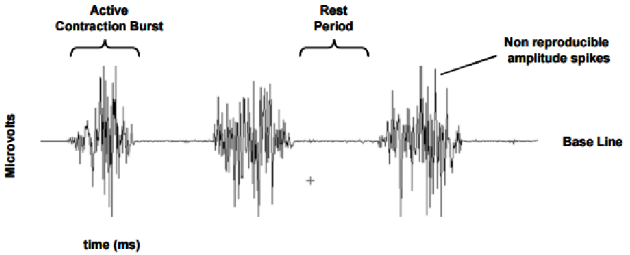

Figure: EMG Signal (Credits: In Arul)

- EMG Signals are sparse in time domain (i.e. satisfy the sparsity condition required by CS)
- The amplitude of the EMG Signals lies in the range from $1 - 10$mV (i.e. the bounded dynamic range $= D \leq 10$).

Related Works

Problem	Upper Bound	Lower Bound	Authors
Support Recovery for $\mathbb{R}^n$	$O(k^3 \log n)$	-	Gopi et al.
	-	$\Omega(k \log\left(\frac{n}{k}\right))$	Folklore
	$O(k^2 \log n)$	$\Omega\left(\frac{k^2 \log n}{\log k}\right)$	Acharya et al.
Support Recovery for dynamic range $D = 1$	$O(k \log\left(\frac{n}{k}\right) + k^{1.5} \log k)$	-	Flodin et al.
Approximate Recovery for $\mathbb{R}^n$	$O(\frac{k}{\epsilon} \log\left(\frac{n}{k}\right))$	-	Jacques et al.
	$O(k^2 \log\left(\frac{n}{k}\right) + \frac{k}{\epsilon})$	$\Omega(k \log\left(\frac{n}{k}\right) + \frac{k}{\epsilon})$	Acharya et al.
	$O(k \log\left(\frac{n}{k}\right) + \frac{k}{\epsilon} \log\left(\frac{k}{\epsilon}\right))$	-	Flodin et al.

Related Works

Problem	Upper Bound	Lower Bound	Authors
Support Recovery for $\mathbb{R}^n$	$O(k^3 \log n)$	-	Gopi et al.
	-	$\Omega(k \log (\frac{n}{k}))$	Folklore
	$O(k^2 \log n)$	$\Omega\left(\frac{k^2 \log n}{\log k}\right)$	Acharya et al.
Support Recovery for dynamic range $D = 1$	$O(k \log (\frac{n}{k}) + k^{1.5} \log k)$	-	Flodin et al.
Approximate Recovery for $\mathbb{R}^n$	$O(\frac{k}{\epsilon} \log (\frac{n}{k}))$	-	Jacques et al.
	$O(k^2 \log (\frac{n}{k}) + \frac{k}{\epsilon})$	$\Omega(k \log (\frac{n}{k}) + \frac{k}{\epsilon})$	Acharya et al.
	$O(k \log (\frac{n}{k}) + \frac{k}{\epsilon} \log (\frac{k}{\epsilon}))$	-	Flodin et al.

- Support recovery for $\mathbb{R}^n$ is almost tight at $\approx k^2 \log n$
- Approximate recovery for $\mathbb{R}^n$ is also tight at $\approx k \log \left(\frac{n}{k}\right) + \frac{k}{\epsilon}$
- **But for dynamic range $D = 1$, there is currently no good lower bound except the trivial lower bound of $\Omega\left(k \log \frac{n}{k}\right)$.**

Related Works

Problem	Upper Bound	Lower Bound	Authors
Support Recovery for $\mathbb{R}^n$	$O(k^3 \log n)$	-	Gopi et al.
	-	$\Omega(k \log\left(\frac{n}{k}\right))$	Folklore
	$O(k^2 \log n)$	$\Omega\left(\frac{k^2 \log n}{\log k}\right)$	Acharya et al.
Support Recovery for dynamic range $D = 1$	$O(k \log\left(\frac{n}{k}\right) + k^{1.5} \log k)$	-	Flodin et al.
Support Recovery for dynamic range $D = 1$ when $A \in \{\pm 1\}^{m \times n, 0.5}$	-	$\Omega(k^{1.5})$	**Our work**
Approximate Recovery for $\mathbb{R}^n$	$O(\frac{k}{\epsilon} \log\left(\frac{n}{k}\right))$	-	Jacques et al.
	$O(k^2 \log\left(\frac{n}{k}\right) + \frac{k}{\epsilon})$	$\Omega(k \log\left(\frac{n}{k}\right) + \frac{k}{\epsilon})$	Acharya et al.
	$O(k \log\left(\frac{n}{k}\right) + \frac{k}{\epsilon} \log\left(\frac{k}{\epsilon}\right))$	-	Flodin et al.

- Support recovery for $\mathbb{R}^n$ is almost tight at $\approx k^2 \log n$
- Approximate recovery for $\mathbb{R}^n$ is also tight at $\approx k \log\left(\frac{n}{k}\right) + \frac{k}{\epsilon}$
- **But for dynamic range $D = 1$, there is currently no good lower bound except the trivial lower bound of $\Omega\left(k \log \frac{n}{k}\right)$.**

$O(k^{1.5} \log \frac{n}{k})$ upper bound for $D = 1$

An $m = O(k^{1.5} \log \left(\frac{n}{k}\right))$ universal 1-bit measurement matrix for bounded dynamic range of $D = 1$ is described in Jacques et al.

Construction Method

Let the distribution $\mathbb{D} = \{\pm 1\}^{m\times n, 0.5}$ be a probability distribution on matrices of dimension $m \times n$ where each entry of the matrix is an independent Rademacher random variable (RV), i.e. it is -1 with probability 0.5 and $+1$ with probability 0.5

$$A = \begin{bmatrix} +1 & -1 & \dots \\ \vdots & \ddots & \\ -1 & & +1 \end{bmatrix}$$

Lower Bound for $\{\pm 1\}^{m \times n, 0.5}$

Assumptions

- Under universal scheme
- $D = 1$

Lower Bound for $\{\pm 1\}^{m\times n,0.5}$

Assumptions

- Under universal scheme
- $D = 1$

Main Theorem

Given we pick a matrix $A \in \mathbb{D} = \{\pm 1\}^{m\times n,0.5}$ and $m = O(k^{1.5})$ then with $\Omega(1)$ probability the 1-bit measurement matrix will NOT be valid.

ℓ-balanced problem

Definition

A set $V \subseteq \{\pm 1\}^n$ is *ℓ-balanced* if for any $S \subseteq [n]$ of size ℓ, there exists $v \in V$ satisfying $|\sum_{i \in S} v_i| \leq 1$.

Example

$$V = \begin{cases} (+1, -1, +1, -1, +1, -1, +1) \\ (-1, +1, +1, +1, -1, +1, -1) \\ (-1, -1, +1, -1, +1, -1, -1) \\ (+1, +1, -1, -1, +1, +1, -1) \\ (+1, -1, -1, +1, -1, +1, -1) \end{cases}$$

$V \subseteq \{\pm 1\}^7, |V| = 5,\ V$ is 4-balanced.

ℓ-balanced problem

Definition

A set $V \subseteq \{\pm 1\}^n$ is *ℓ-balanced* if for any $S \subseteq [n]$ of size ℓ, there exists $v \in V$ satisfying $|\sum_{i\in S} v_i| \leq 1$.

Example

$$V = \begin{cases} (+1, -1, +1, -1, +1, -1, +1) & \text{Sum} = 0\ ✓ \\ (-1, +1, +1, +1, -1, +1, -1) & \text{Sum} = 2\ ✗ \\ (-1, -1, +1, -1, +1, -1, -1) & \text{Sum} = 0\ ✓ \\ (+1, +1, -1, -1, +1, +1, -1) & \text{Sum} = 2\ ✗ \\ (+1, -1, -1, +1, -1, +1, -1) & \text{Sum} = \text{-}2\ ✗ \end{cases} \quad s = \begin{bmatrix} 0 \\ 1 \\ 1 \\ 0 \\ 1 \\ 1 \\ 0 \end{bmatrix} \in S$$

$V \subseteq \{\pm 1\}^7, |V| = 5,\ V$ is 4-balanced.

ℓ-balanced problem $\leq$ 1-bit CS

Let A be a valid 1-bit CS measurement matrix (for k-sparse inputs) with the rows labelled as $r_1, r_2, \ldots, r_m$.

ℓ-balanced problem $\leq$ 1-bit CS

Let A be a valid 1-bit CS measurement matrix (for k-sparse inputs) with the rows labelled as $r_1, r_2, \ldots, r_m$. Then $V = \{r_1, r_2, \ldots, r_m\}$ forms a valid set for the $(k-1)$-balanced problem.

ℓ-balanced problem $\leq$ 1-bit CS

Let A be a valid 1-bit CS measurement matrix (for k-sparse inputs) with the rows labelled as $r_1, r_2, \ldots, r_m$. Then $V = \{r_1, r_2, \ldots, r_m\}$ forms a valid set for the $(k-1)$-balanced problem.

Proof Idea

Let $k = 5$ and we know A is valid.

$$A = \begin{bmatrix} +1 & -1 & +1 & -1 & +1 & -1 & +1 \\ -1 & +1 & +1 & +1 & -1 & +1 & -1 \\ -1 & -1 & +1 & -1 & +1, & -1 & -1 \\ +1 & +1 & -1 & -1 & +1 & +1 & -1 \\ +1 & -1 & -1 & +1 & -1 & +1 & -1 \end{bmatrix}$$

ℓ-balanced problem $\leq$ 1-bit CS

Let A be a valid 1-bit CS measurement matrix (for k-sparse inputs) with the rows labelled as $r_1, r_2, \ldots, r_m$. Then $V = \{r_1, r_2, \ldots, r_m\}$ forms a valid set for the $(k-1)$-balanced problem.

Proof Idea

Let $k = 5$ and we know A is valid.

$$A = \begin{bmatrix} +1 & -1 & +1 & -1 & +1 & -1 & +1 \\ -1 & +1 & +1 & +1 & -1 & +1 & -1 \\ -1 & -1 & +1 & -1 & +1, & -1 & -1 \\ +1 & +1 & -1 & -1 & +1 & +1 & -1 \\ +1 & -1 & -1 & +1 & -1 & +1 & -1 \end{bmatrix} s = \begin{bmatrix} 0 \\ 1 \\ 1 \\ 0 \\ 1 \\ 1 \\ 0 \end{bmatrix}$$

ℓ-balanced problem $\leq$ 1-bit CS

Let A be a valid 1-bit CS measurement matrix (for k-sparse inputs) with the rows labelled as $r_1, r_2, \ldots, r_m$. Then $V = \{r_1, r_2, \ldots, r_m\}$ forms a valid set for the $(k-1)$-balanced problem.

Proof Idea

Let $k = 5$ and we know A is valid.

$$A = \begin{bmatrix} +1 & -1 & +1 & -1 & +1 & -1 & +1 \\ -1 & +1 & +1 & +1 & -1 & +1 & -1 \\ -1 & -1 & +1 & -1 & +1, & -1 & -1 \\ +1 & +1 & -1 & -1 & +1 & +1 & -1 \\ +1 & -1 & -1 & +1 & -1 & +1 & -1 \end{bmatrix} s = x_1 \cap x_2 = \begin{bmatrix} 0 \\ 1 \\ 1 \\ 0 \\ 1 \\ 1 \\ 1 \end{bmatrix} \cap \begin{bmatrix} 0 \\ 1 \\ 1 \\ 0 \\ 1 \\ 1 \\ 0 \end{bmatrix}$$

ℓ-balanced problem $\leq$ 1-bit CS

Let A be a valid 1-bit CS measurement matrix (for k-sparse inputs) with the rows labelled as $r_1, r_2, \ldots, r_m$. Then $V = \{r_1, r_2, \ldots, r_m\}$ forms a valid set for the $(k-1)$-balanced problem.

Proof Idea

Let $k = 5$ and we know A is valid.

$$A = \begin{bmatrix} +1 & -1 & +1 & -1 & +1 & -1 & +1 \\ -1 & +1 & +1 & +1 & -1 & +1 & -1 \\ -1 & -1 & +1 & -1 & +1, & -1 & \textbf{-1} \\ +1 & +1 & -1 & -1 & +1 & +1 & -1 \\ +1 & -1 & -1 & +1 & -1 & +1 & -1 \end{bmatrix} s = x_1 \cap x_2 = \begin{bmatrix} 0 \\ 1 \\ 1 \\ 0 \\ 1 \\ 1 \\ 1 \end{bmatrix} \cap \begin{bmatrix} 0 \\ 1 \\ 1 \\ 0 \\ 1 \\ 1 \\ 0 \end{bmatrix}$$

$sign(r_3 \cdot x_1) \neq sign(r_3 \cdot x_2)$

ℓ-balanced problem $\leq$ 1-bit CS

Let A be a valid 1-bit CS measurement matrix (for k-sparse inputs) with the rows labelled as $r_1, r_2, \ldots, r_m$. Then $V = \{r_1, r_2, \ldots, r_m\}$ forms a valid set for the $(k-1)$-balanced problem.

Proof Idea

Let $k = 5$ and we know A is valid.

$$A = \begin{bmatrix} +1 & -1 & +1 & -1 & +1 & -1 & +1 \\ -1 & +1 & +1 & +1 & -1 & +1 & -1 \\ -1 & \mathbf{-1} & \mathbf{+1} & -1 & \mathbf{+1}, & \mathbf{-1} & \mathbf{-1} \\ +1 & +1 & -1 & -1 & +1 & +1 & -1 \\ +1 & -1 & -1 & +1 & -1 & +1 & -1 \end{bmatrix} s = x_1 \cap x_2 = \begin{bmatrix} 0 \\ 1 \\ 1 \\ 0 \\ 1 \\ 1 \\ \mathbf{1} \end{bmatrix} \cap \begin{bmatrix} 0 \\ 1 \\ 1 \\ 0 \\ 1 \\ 1 \\ \mathbf{0} \end{bmatrix}$$

$r_3 \cdot x_1 < 0, r_3 \cdot x_2 \geq 0$

ℓ-balanced problem $\leq$ 1-bit CS

Let A be a valid 1-bit CS measurement matrix (for k-sparse inputs) with the rows labelled as $r_1, r_2, \dots, r_m$. Then $V = \{r_1, r_2, \dots, r_m\}$ forms a valid set for the $(k-1)$-balanced problem.

Proof Idea

Let $k = 5$ and we know A is valid.

$$A = \begin{bmatrix} +1 & -1 & +1 & -1 & +1 & -1 & +1 \\ -1 & +1 & +1 & +1 & -1 & +1 & -1 \\ -1 & -1 & +1 & -1 & +1, & -1 & \textbf{-1} \\ +1 & +1 & -1 & -1 & +1 & +1 & -1 \\ +1 & -1 & -1 & +1 & -1 & +1 & -1 \end{bmatrix} s = x_1 \cap x_2 = \begin{bmatrix} 0 \\ 1 \\ 1 \\ 0 \\ 1 \\ 1 \\ 1 \end{bmatrix} \cap \begin{bmatrix} 0 \\ 1 \\ 1 \\ 0 \\ 1 \\ 1 \\ 0 \end{bmatrix}$$

$$\implies |r_3 \cdot (x_1 \cap x_2)| \leq 1 \implies |r_3 \cdot s| \leq 1$$

ℓ-balanced problem $\leq$ 1-bit CS

Let A be a valid 1-bit CS measurement matrix (for k-sparse inputs) with the rows labelled as $r_1, r_2, \ldots, r_m$. Then $V = \{r_1, r_2, \ldots, r_m\}$ forms a valid set for the $(k-1)$-balanced problem.

Proof Idea

Let $k = 5$ and we know A is valid.

$$A = \begin{bmatrix} +1 & -1 & +1 & -1 & +1 & -1 & +1 \\ -1 & +1 & +1 & +1 & -1 & +1 & -1 \\ -1 & -1 & +1 & -1 & +1, & -1 & \mathbf{-1} \\ +1 & +1 & -1 & -1 & +1 & +1 & -1 \\ +1 & -1 & -1 & +1 & -1 & +1 & -1 \end{bmatrix} s = x_1 \cap x_2 = \begin{bmatrix} 0 \\ 1 \\ 1 \\ 0 \\ 1 \\ 1 \\ 1 \end{bmatrix} \cap \begin{bmatrix} 0 \\ 1 \\ 1 \\ 0 \\ 1 \\ 1 \\ 0 \end{bmatrix}$$

$\implies |\sum_{i \in s} r_{3,i}| \leq 1 \implies s$ is balanced [Do this $\forall s \in S$]

$(k-1)$-balanced problem $\leq$ 1-bit CS

So $(k-1)$-balanced problem is *easier* than 1-bit CS.

Remark

Therefore a lower bound of
$|V| > \Omega(k^{1.5})$ for (k - 1)-balanced problem $\implies$ $m > \Omega(k^{1.5})$ for 1-bit CS.

Proof Idea

We want to now show the following:

Main Theorem

A set $V \subseteq \{\pm 1\}^n$, where $|V| = m$ and each $V_{i,j}$ is an independent Rademacher RV is NOT k-balanced with probability $\Omega(1)$ if $m < O(k^{1.5})$

Proof Idea

Pr(Failure) = Pr(one of the subsets of size k is not balanced by V)

Proof Idea

Pr(Failure) = Pr(one of the subsets of size k is not balanced by V)

Definition

Let $X = \{x | x \subseteq [n], |x| = k\}$.
Then $|X| = \binom{n}{k}$ and each element in X is denoted by $x_i, \forall i \in [1, \binom{n}{k}]$

Proof Idea

Pr(Failure) = Pr(one of the subsets of size k is not balanced by V)

Definition

Let $X = \{x | x \subseteq [n], |x| = k\}$.
Then $|X| = \binom{n}{k}$ and each element in X is denoted by $x_i, \forall i \in [1, \binom{n}{k}]$

Definition

Let F_i be the failure event, i.e. the event that $|v \cdot x_i| > 1, \forall v \in V$ for the k-balanced problem, where each $V_{i,j}$ is an independent Rademacher RV.

Proof Idea

Pr(Failure) = Pr(one of the subsets of size k is not balanced by V)

Definition

Let $X = \{x | x \subseteq [n], |x| = k\}$.
Then $|X| = \binom{n}{k}$ and each element in X is denoted by $x_i, \forall i \in [1, \binom{n}{k}]$

Definition

Let F_i be the failure event, i.e. the event that $|v \cdot x_i| > 1, \forall v \in V$ for the k-balanced problem, where each $V_{i,j}$ is an independent Rademacher RV.

$$\Pr(\text{one of the subsets of size k is not balanced by } V) = \Pr\left(\bigcup_{i=1}^{|X|} F_i\right)$$

Proof Idea

$\Pr(\text{Failure}) = \Pr(\bigcup_{i=1}^{|X|} F_i)$

Definition

De Caen's Inequality:

$$\Pr(\bigcup_{i\in I} F_i) \geq \sum_{i\in I} \frac{\Pr(F_i)^2}{\sum_{j\in I} \Pr(F_i \cap F_j)}$$

In De Caen, this inequality was proven for a finite set of events.

Proof Idea

$\Pr(\text{Failure}) = \Pr(\bigcup_{i=1}^{|X|} F_i)$

Definition

De Caen's Inequality:

$$\Pr(\bigcup_{i\in I} F_i) \geq \sum_{i\in I} \frac{\Pr(F_i)^2}{\sum_{j\in I} \Pr(F_i \cap F_j)}$$

In De Caen, this inequality was proven for a finite set of events.

So we now know, $\Pr(\text{Failure}) \geq \sum_{i=1}^{|X|} \frac{\Pr(F_i)^2}{\sum_{j=1}^{|X|} \Pr(F_i \cap F_j)} \geq \frac{\binom{n}{k} \Pr(F_1)^2}{\sum_{j=1}^{\binom{n}{k}} \Pr(F_1 \cap F_j)}$

Proof Idea

$$\Pr(\text{Failure}) \geq \frac{\binom{n}{k} \Pr(F_1)^2}{\sum_{j=1}^{\binom{n}{k}} \Pr(F_1 \cap F_j)}$$

We want to show that the denominator here is at-most constant times bigger than the numerator.

Proof Idea

$$\Pr(\text{Failure}) \geq \frac{\binom{n}{k} \Pr(F_1)^2}{\sum_{j=1}^{\binom{n}{k}} \Pr(F_1 \cap F_j)}$$

We want to show that the denominator here is at-most constant times bigger than the numerator.

- Case 1 $|x_1 \cap x_j| < \sqrt{k}$
- Case 2 $\sqrt{k} \leq |x_1 \cap x_j| < 0.9k$
- Case 3 $0.9k \leq |x_1 \cap x_j| < k$

Proof Idea

$$\Pr(\text{Failure}) \geq \frac{\binom{n}{k}\Pr(F_1)^2}{\sum_{j=1}^{\binom{n}{k}} \Pr(F_1 \cap F_j)}$$

We want to show that the denominator here is at-most constant times bigger than the numerator.

Case 1 $|x_1 \cap x_j| < \sqrt{k}$

Since the intersection is *small* so F_i and F_j are *almost-independent* $\implies \Pr(F_i \cap F_j) \approx O(\Pr(F_i)^2)$

$$\therefore \sum_{|x_1 \cap x_j| < \sqrt{k}} \Pr(F_1 \cap F_j) \leq \binom{n}{k} O(\Pr(F_1)^2) = O\left(\binom{n}{k}\Pr(F_1)^2\right)$$

Case 2 $\sqrt{k} \leq |x_1 \cap x_j| < 0.9k$

Case 3 $0.9k \leq |x_1 \cap x_j| < k$

Proof Idea

$$\Pr(\text{Failure}) \geq \frac{\binom{n}{k}\Pr(F_1)^2}{\sum_{j=1}^{\binom{n}{k}} \Pr(F_1 \cap F_j)}$$

We want to show that the denominator here is at-most constant times bigger than the numerator.

- **Case 1** $|x_1 \cap x_j| < \sqrt{k}$
- **Case 2** $\sqrt{k} \leq |x_1 \cap x_j| < 0.9k$

  The number of pairs of $\{F_1, F_i\}$ such that the intersection size is at-least $\sqrt{k}$, i.e. $\binom{k}{|x_i \cap x_j|}\binom{n-k}{k-|x_1\cap x_j|} \ll \binom{n}{k}$

  $$\therefore \sum_{\sqrt{k} \leq |x_1 \cap x_j| < 0.9k} \Pr(F_1 \cap F_j) = O\left(\binom{n}{k}\Pr(F_1)^2\right)$$

- **Case 3** $0.9k \leq |x_1 \cap x_j| < k$

Proof Idea

$$\Pr(\text{Failure}) \geq \frac{\binom{n}{k} \Pr(F_1)^2}{\sum_{j=1}^{\binom{n}{k}} \Pr(F_1 \cap F_j)}$$

We want to show that the denominator here is at-most constant times bigger than the numerator.

- Case 1 $|x_1 \cap x_j| < \sqrt{k}$
- Case 2 $\sqrt{k} \leq |x_1 \cap x_j| < 0.9k$
- Case 3 $0.9k \leq |x_1 \cap x_j| < k$
  We again bound the number of pairs to be $< \binom{n}{k}$

$$\therefore \sum_{0.9k \leq |x_1 \cap x_j|} \Pr(F_1 \cap F_j) \leq \sum_{0.9k \leq |x_1 \cap x_j|} 1 \leq O\left(\binom{n}{k} \Pr(F_1)^2\right)$$

Proof Idea

$$\Pr(\text{Failure}) \geq \frac{\binom{n}{k}\Pr(F_1)^2}{\sum_{j=1}^{\binom{n}{k}}\Pr(F_1 \cap F_j)} \geq \Omega(1) \text{ when } m = O(k^{1.5})$$

We want to show that the denominator here is at-most constant times bigger than the numerator.

- Case 1 $|x_1 \cap x_j| < \sqrt{k}$
- Case 2 $\sqrt{k} \leq |x_1 \cap x_j| < 0.9k$
- Case 3 $0.9k \leq |x_1 \cap x_j| < k$

Remark

All the algebraic details are presented in the report.

Our contributions

- Established the probabilistic lower bound of $m = \Omega(k^{1.5})$ for distribution $\{\pm 1\}^{m\times n,0.5}$ under the universal scheme setting with $D = 1$.
- Designed and analysed an efficient recovery algorithm for $D = 1$.

Future Work

- Extend the current probabilistic lower bound from the distribution $\{\pm 1\}^{m\times n,0.5}$ to $N(0,1)^{m\times n}$ for bounded dynamic range, i.e. $D \geq 1$.
- We suspect the lower bound of $m = \Omega(k^{1.5})$ to even hold for all possible matrices of the form $\mathbb{R}^{m\times n}$ for bounded dynamic range. Research is possible in this area as well.

Thank You!