Learning Algebraic Functions From a Few Samples

Michael Forbes

MIT

Based on joint works with Amir Shpilka

April 28, 2014

Learning Long Division

Learning Long Division

Learning Long Division

$$f(x, y) := x/y$$

Learning Long Division

$$f(x, y) := x/y$$

Examples:

Learning Long Division

$$f(x, y) := x/y$$

Examples:

- $f(1, 1) = 1$

Learning Long Division

$$f(x, y) := x/y$$

Examples:

- $f(1, 1) = 1 = f(2, 2)$

Learning Long Division

$$f(x, y) := x/y$$

Examples:

- $f(1,1) = 1 = f(2,2)$
- $f(1,2) = 1/2$

Learning Long Division

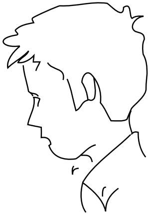
$$f(x, y) := x/y$$

Examples:

- $f(1, 1) = 1 = f(2, 2)$
- $f(1, 2) = 1/2$
- $f(2, 1) = 2$

Learning Long Division

$$f(x, y) := x/y$$



Examples:

- $f(1,1) = 1 = f(2,2)$
- $f(1,2) = 1/2$
- $f(2,1) = 2$

Claim

Now you know long division.

Learning Long Division

$$f(x, y) := x/y$$

Examples:

- $f(1, 1) = 1 = f(2, 2)$
- $f(1, 2) = {}^1\!/_2$
- $f(2, 1) = 2$

Claim

Now you know long division.

Proof.

Learning Long Division

$$f(x, y) := x/y$$

Examples:

- $f(1, 1) = 1 = f(2, 2)$
- $f(1, 2) = {}^1/_2$
- $f(2, 1) = 2$

Claim

Now you know long division.

Proof.

$f(x, y) = x/y$ is the uniquely determined by the above,

$$f(x, y) := x/y$$

Examples:

- $f(1, 1) = 1 = f(2, 2)$
- $f(1, 2) = 1/2$
- $f(2, 1) = 2$

Claim

Now you know long division.

Proof.

$f(x, y) = x/y$ is the uniquely determined by the above, as a function of the form $f(x, y) = (ax + b)/(cy + d)$. □

Theme

Theme

Meta-fact

Theme

Meta-fact

Simple functions are essentially determined by sufficiently many random samples.

Theme

Meta-fact

Simple functions are essentially determined by sufficiently many random samples.

Example

Theme

Meta-fact

Simple functions are essentially determined by sufficiently many random samples.

Example

- PAC (Probably Approximately Correct) learning

Theme

Meta-fact

Simple functions are essentially determined by sufficiently many random samples.

Example

- PAC (Probably Approximately Correct) learning
- Error-correcting Codes

Theme

Meta-fact

Simple functions are essentially determined by sufficiently many random samples.

Example

- PAC (Probably Approximately Correct) learning
- Error-correcting Codes
- (Black-box) Polynomial Identity Testing (PIT)

Theme

Meta-fact

Simple functions are essentially determined by sufficiently many random samples.

Example

- PAC (Probably Approximately Correct) learning
- Error-correcting Codes
- (Black-box) Polynomial Identity Testing (PIT) — given an algebraic circuit C

Theme

Meta-fact

Simple functions are essentially determined by sufficiently many random samples.

Example

- PAC (Probably Approximately Correct) learning
- Error-correcting Codes
- (Black-box) Polynomial Identity Testing (PIT) — given an algebraic circuit C, does the polynomial $C(\vec{x})$ equal zero?

More on the Theme

More on the Theme

Meta-hope

More on the Theme

Meta-hope

Simple functions are essentially determined by random samples.

More on the Theme

Meta-hope

Simple functions are essentially **efficiently** *determined by random samples.*

More on the Theme

Meta-hope

Simple functions are essentially **efficiently** *determined by* **deterministic** *samples.*

More on the Theme

Meta-hope

Simple functions are essentially **efficiently** *determined by* **deterministic** *samples.*

Example (of hope)

More on the Theme

Meta-hope

Simple functions are essentially **efficiently** *determined by* **deterministic** *samples.*

Example (of hope)

- Coding theory:

More on the Theme

Meta-hope

Simple functions are essentially **efficiently** *determined by* **deterministic** *samples.*

Example (of hope)

- Coding theory: need **deterministic** coding schemes for communication

More on the Theme

Meta-hope

Simple functions are essentially **efficiently** *determined by* **deterministic** *samples.*

Example (of hope)

- Coding theory: need **deterministic** coding schemes for communication
- Complexity theory:

More on the Theme

Meta-hope

Simple functions are essentially **efficiently** *determined by* **deterministic** *samples.*

Example (of hope)

- Coding theory: need **deterministic** coding schemes for communication
- Complexity theory: understanding the power of (pseudo)randomness

More on the Theme

Meta-hope

Simple functions are essentially **efficiently** *determined by* **deterministic** *samples.*

Example (of hope)

- Coding theory: need **deterministic** coding schemes for communication
- Complexity theory: understanding the power of (pseudo)randomness
- Complexity theory:

More on the Theme

Meta-hope

Simple functions are essentially **efficiently** *determined by* **deterministic** *samples.*

Example (of hope)

- Coding theory: need **deterministic** coding schemes for communication
- Complexity theory: understanding the power of (pseudo)randomness
- Complexity theory: connections with circuit lower bounds

Bilinear Forms

Bilinear Forms

Let $\mathbb{F}$ be a field.

Bilinear Forms

Bilinear Forms

Let $\mathbb{F}$ be a field.

Bilinear Forms

Let $\mathbb{F}$ be a field. Let M be an $n \times n$ matrix.

Bilinear Forms

Let $\mathbb{F}$ be a field. Let M be an $n \times n$ matrix. Define $f : \mathbb{F}^n \times \mathbb{F}^n \to \mathbb{F}$ by

Bilinear Forms

Let $\mathbb{F}$ be a field. Let M be an $n \times n$ matrix. Define $f : \mathbb{F}^n \times \mathbb{F}^n \to \mathbb{F}$ by

$$f(\vec{x}, \vec{y}) := \vec{x}^t M \vec{y}$$

Bilinear Forms

Let $\mathbb{F}$ be a field. Let M be an $n \times n$ matrix. Define $f : \mathbb{F}^n \times \mathbb{F}^n \to \mathbb{F}$ by

$$f(\vec{x}, \vec{y}) := \vec{x}^t M \vec{y} = \langle M, \vec{x}^t \vec{y} \rangle$$

Bilinear Forms

Let $\mathbb{F}$ be a field. Let M be an $n \times n$ matrix. Define $f : \mathbb{F}^n \times \mathbb{F}^n \to \mathbb{F}$ by

$$f(\vec{x}, \vec{y}) := \vec{x}^t M \vec{y} = \langle M, \vec{x}^t \vec{y} \rangle$$

Question

Bilinear Forms

Let $\mathbb{F}$ be a field. Let M be an $n \times n$ matrix. Define $f : \mathbb{F}^n \times \mathbb{F}^n \to \mathbb{F}$ by

$$f(\vec{x}, \vec{y}) := \vec{x}^t M \vec{y} = \langle M, \vec{x}^t \vec{y} \rangle$$

Question

How many queries to f

Bilinear Forms

Let $\mathbb{F}$ be a field. Let M be an $n \times n$ matrix. Define $f : \mathbb{F}^n \times \mathbb{F}^n \to \mathbb{F}$ by

$$f(\vec{x}, \vec{y}) := \vec{x}^t M \vec{y} = \langle M, \vec{x}^t \vec{y} \rangle$$

Question

How many queries to f do we need to determine if $f \equiv 0$?

Bilinear Forms

Let $\mathbb{F}$ be a field. Let M be an $n \times n$ matrix. Define $f : \mathbb{F}^n \times \mathbb{F}^n \to \mathbb{F}$ by

$$f(\vec{x}, \vec{y}) := \vec{x}^t M \vec{y} = \langle M, \vec{x}^t \vec{y} \rangle$$

Question

How many queries to f do we need to determine if $f \equiv 0$? To learn M?

Bilinear Forms

Let $\mathbb{F}$ be a field. Let M be an $n \times n$ matrix. Define $f : \mathbb{F}^n \times \mathbb{F}^n \to \mathbb{F}$ by

$$f(\vec{x}, \vec{y}) := \vec{x}^t M \vec{y} = \langle M, \vec{x}^t \vec{y} \rangle$$

Question

How many queries to f do we need to determine if $f \equiv 0$? To learn M?

Answer

Bilinear Forms

Let $\mathbb{F}$ be a field. Let M be an $n \times n$ matrix. Define $f : \mathbb{F}^n \times \mathbb{F}^n \to \mathbb{F}$ by

$$f(\vec{x}, \vec{y}) := \vec{x}^t M \vec{y} = \langle M, \vec{x}^t \vec{y} \rangle$$

Question

How many queries to f do we need to determine if $f \equiv 0$? To learn M?

Answer

n^2 *are sufficient*

Bilinear Forms

Let $\mathbb{F}$ be a field. Let M be an $n \times n$ matrix. Define $f : \mathbb{F}^n \times \mathbb{F}^n \to \mathbb{F}$ by

$$f(\vec{x}, \vec{y}) := \vec{x}^t M \vec{y} = \langle M, \vec{x}^t \vec{y} \rangle$$

Question

How many queries to f do we need to determine if $f \equiv 0$? To learn M?

Answer

n^2 *are sufficient, as* $f(\vec{e}_i, \vec{e}_j) = M_{i,j}$.

Bilinear Forms

Let $\mathbb{F}$ be a field. Let M be an $n \times n$ matrix. Define $f : \mathbb{F}^n \times \mathbb{F}^n \to \mathbb{F}$ by

$$f(\vec{x}, \vec{y}) := \vec{x}^t M \vec{y} = \langle M, \vec{x}^t \vec{y} \rangle$$

Question

How many queries to f do we need to determine if $f \equiv 0$? To learn M?

Answer

n^2 *are sufficient, as* $f(\vec{e}_i, \vec{e}_j) = M_{i,j}$.

Answer

Bilinear Forms

Let $\mathbb{F}$ be a field. Let M be an $n \times n$ matrix. Define $f : \mathbb{F}^n \times \mathbb{F}^n \to \mathbb{F}$ by

$$f(\vec{x}, \vec{y}) := \vec{x}^t M \vec{y} = \langle M, \vec{x}^t \vec{y} \rangle$$

Question

How many queries to f do we need to determine if $f \equiv 0$? To learn M?

Answer

n^2 *are sufficient, as* $f(\vec{e}_i, \vec{e}_j) = M_{i,j}$.

Answer

n^2 *are necessary*

Bilinear Forms

Let $\mathbb{F}$ be a field. Let M be an $n \times n$ matrix. Define $f : \mathbb{F}^n \times \mathbb{F}^n \to \mathbb{F}$ by

$$f(\vec{x}, \vec{y}) := \vec{x}^t M \vec{y} = \langle M, \vec{x}^t \vec{y} \rangle$$

Question

How many queries to f do we need to determine if $f \equiv 0$? To learn M?

Answer

n^2 *are sufficient, as* $f(\vec{e}_i, \vec{e}_j) = M_{i,j}$.

Answer

n^2 *are necessary, by counting.*

Bilinear Forms — Low Rank

Bilinear Forms — Low Rank

Let $\mathbb{F}$ be a field.

Bilinear Forms — Low Rank

Let $\mathbb{F}$ be a field. Let M be an $n \times n$ matrix

Bilinear Forms — Low Rank

Let $\mathbb{F}$ be a field. Let M be an $n \times n$ matrix **of rank** $\leq r$.

Bilinear Forms — Low Rank

Let $\mathbb{F}$ be a field. Let M be an $n \times n$ matrix **of rank** $\leq r$. Define $f : \mathbb{F}^n \times \mathbb{F}^n \to \mathbb{F}$ by $f(\vec{x}, \vec{y}) := \vec{x}^t M \vec{y}$

Bilinear Forms — Low Rank

Let $\mathbb{F}$ be a field. Let M be an $n \times n$ matrix **of rank** $\leq r$. Define $f : \mathbb{F}^n \times \mathbb{F}^n \to \mathbb{F}$ by $f(\vec{x}, \vec{y}) := \vec{x}^t M \vec{y} = \langle M, \vec{x}^t \vec{y} \rangle$

Bilinear Forms — Low Rank

Let $\mathbb{F}$ be a field. Let M be an $n \times n$ matrix **of rank** $\leq r$. Define $f : \mathbb{F}^n \times \mathbb{F}^n \to \mathbb{F}$ by $f(\vec{x}, \vec{y}) := \vec{x}^t M \vec{y} = \langle M, \vec{x}^t \vec{y} \rangle$

Question

How many queries to f do we need to determine if $f \equiv 0$?

Bilinear Forms — Low Rank

Let $\mathbb{F}$ be a field. Let M be an $n \times n$ matrix **of rank** $\leq r$. Define $f : \mathbb{F}^n \times \mathbb{F}^n \to \mathbb{F}$ by $f(\vec{x}, \vec{y}) := \vec{x}^t M \vec{y} = \langle M, \vec{x}^t \vec{y} \rangle$

Question

How many queries to f do we need to determine if $f \equiv 0$? To learn M?

Bilinear Forms — Low Rank

Let $\mathbb{F}$ be a field. Let M be an $n \times n$ matrix **of rank** $\leq r$. Define $f : \mathbb{F}^n \times \mathbb{F}^n \to \mathbb{F}$ by $f(\vec{x}, \vec{y}) := \vec{x}^t M \vec{y} = \langle M, \vec{x}^t \vec{y} \rangle$

Question

How many queries to f do we need to determine if $f \equiv 0$? To learn M?

Answer

Bilinear Forms — Low Rank

Let $\mathbb{F}$ be a field. Let M be an $n \times n$ matrix **of rank** $\leq r$. Define $f : \mathbb{F}^n \times \mathbb{F}^n \to \mathbb{F}$ by $f(\vec{x}, \vec{y}) := \vec{x}^t M \vec{y} = \langle M, \vec{x}^t \vec{y} \rangle$

Question

How many queries to f do we need to determine if $f \equiv 0$? To learn M?

Answer

$\approx 2nr$ *(resp.* $\approx 4nr$*) are sufficient*

Bilinear Forms — Low Rank

Let $\mathbb{F}$ be a field. Let M be an $n \times n$ matrix **of rank** $\leq r$. Define $f : \mathbb{F}^n \times \mathbb{F}^n \to \mathbb{F}$ by $f(\vec{x}, \vec{y}) := \vec{x}^t M \vec{y} = \langle M, \vec{x}^t \vec{y} \rangle$

Question

How many queries to f do we need to determine if $f \equiv 0$? To learn M?

Answer

$\approx 2nr$ *(resp.* $\approx 4nr$*) are sufficient, by the probabilistic method and Schwartz-Zippel*

Bilinear Forms — Low Rank

Let $\mathbb{F}$ be a field. Let M be an $n \times n$ matrix **of rank** $\leq r$. Define $f : \mathbb{F}^n \times \mathbb{F}^n \to \mathbb{F}$ by $f(\vec{x}, \vec{y}) := \vec{x}^t M \vec{y} = \langle M, \vec{x}^t \vec{y} \rangle$

Question

How many queries to f do we need to determine if $f \equiv 0$? To learn M?

Answer

$\approx 2nr$ *(resp.* $\approx 4nr$*) are sufficient, by the probabilistic method and Schwartz-Zippel*

Answer

Bilinear Forms — Low Rank

Let $\mathbb{F}$ be a field. Let M be an $n \times n$ matrix **of rank** $\leq r$. Define $f : \mathbb{F}^n \times \mathbb{F}^n \to \mathbb{F}$ by $f(\vec{x}, \vec{y}) := \vec{x}^t M \vec{y} = \langle M, \vec{x}^t \vec{y} \rangle$

Question

How many queries to f do we need to determine if $f \equiv 0$? To learn M?

Answer

$\approx 2nr$ (resp. $\approx 4nr$) are sufficient, by the probabilistic method and Schwartz-Zippel

Answer

$\Theta(nr)$ are necessary

Bilinear Forms — Low Rank

Let $\mathbb{F}$ be a field. Let M be an $n \times n$ matrix **of rank** $\leq r$. Define $f : \mathbb{F}^n \times \mathbb{F}^n \to \mathbb{F}$ by $f(\vec{x}, \vec{y}) := \vec{x}^t M \vec{y} = \langle M, \vec{x}^t \vec{y} \rangle$

Question

How many queries to f do we need to determine if $f \equiv 0$? To learn M?

Answer

$\approx 2nr$ *(resp.* $\approx 4nr$*) are sufficient, by the probabilistic method and Schwartz-Zippel*

Answer

$\Theta(nr)$ *are necessary, by counting.*

Bilinear Forms — Low Rank

Let $\mathbb{F}$ be a field. Let M be an $n \times n$ matrix **of rank** $\leq r$. Define $f : \mathbb{F}^n \times \mathbb{F}^n \to \mathbb{F}$ by $f(\vec{x}, \vec{y}) := \vec{x}^t M \vec{y} = \langle M, \vec{x}^t \vec{y} \rangle$

Question

How many queries to f do we need to determine if $f \equiv 0$? To learn M?

Answer

$\approx 2nr$ (resp. $\approx 4nr$) are sufficient, by the probabilistic method and Schwartz-Zippel

Answer

$\Theta(nr)$ are necessary, by counting.

Question

Bilinear Forms — Low Rank

Let $\mathbb{F}$ be a field. Let M be an $n \times n$ matrix **of rank** $\leq r$. Define $f : \mathbb{F}^n \times \mathbb{F}^n \to \mathbb{F}$ by $f(\vec{x}, \vec{y}) := \vec{x}^t M \vec{y} = \langle M, \vec{x}^t \vec{y} \rangle$

Question

How many queries to f do we need to determine if $f \equiv 0$? To learn M?

Answer

$\approx 2nr$ (resp. $\approx 4nr$) are sufficient, by the probabilistic method and Schwartz-Zippel

Answer

$\Theta(nr)$ are necessary, by counting.

Question

Can we construct $\approx 2nr$ (resp. $4nr$) **explicit** *samples for identity testing (resp. learning)?*

Results

Results

Definition

Results

Definition

Let $\vec{x} \in \mathbb{F}^n$ be an unknown vector.

Results

Definition

Let $\vec{x} \in \mathbb{F}^n$ be an unknown vector. A **measurement of $\vec{x}$**

Results

Definition

Let $\vec{x} \in \mathbb{F}^n$ be an unknown vector. A **measurement of** $\vec{x}$ is an inner product $\langle \vec{a}, \vec{x} \rangle$ for some known vector $\vec{a} \in \mathbb{F}^n$.

Results

Definition

Let $\vec{x} \in \mathbb{F}^n$ be an unknown vector. A **measurement of** $\vec{x}$ is an inner product $\langle \vec{a}, \vec{x} \rangle$ for some known vector $\vec{a} \in \mathbb{F}^n$. A measurement of an unknown matrix M is an inner product in the space $\mathbb{F}^{n^2}$.

Results

Definition

Let $\vec{x} \in \mathbb{F}^n$ be an unknown vector. A **measurement of** $\vec{x}$ is an inner product $\langle \vec{a}, \vec{x} \rangle$ for some known vector $\vec{a} \in \mathbb{F}^n$. A measurement of an unknown matrix M is an inner product in the space $\mathbb{F}^{n^2}$. A measurement $\langle A, M \rangle$ of the matrix M is **rank 1** if A is rank 1.

Results

Definition

Let $\vec{x} \in \mathbb{F}^n$ be an unknown vector. A **measurement of** $\vec{x}$ is an inner product $\langle \vec{a}, \vec{x} \rangle$ for some known vector $\vec{a} \in \mathbb{F}^n$. A measurement of an unknown matrix M is an inner product in the space $\mathbb{F}^{n^2}$. A measurement $\langle A, M \rangle$ of the matrix M is **rank 1** if A is rank 1.

Thus, evaluations to the bilinear form $f(\vec{x}, \vec{y}) = \vec{x}^t M \vec{y} = \langle \vec{x}^t \vec{y}, M \rangle$ are all rank-1 measurements of M.

Results

Definition

Let $\vec{x} \in \mathbb{F}^n$ be an unknown vector. A **measurement of** $\vec{x}$ is an inner product $\langle \vec{a}, \vec{x} \rangle$ for some known vector $\vec{a} \in \mathbb{F}^n$. A measurement of an unknown matrix M is an inner product in the space $\mathbb{F}^{n^2}$. A measurement $\langle A, M \rangle$ of the matrix M is **rank 1** if A is rank 1.

Thus, evaluations to the bilinear form $f(\vec{x}, \vec{y}) = \vec{x}^t M \vec{y} = \langle \vec{x}^t \vec{y}, M \rangle$ are all rank-1 measurements of M.

Theorem

Results

Definition

Let $\vec{x} \in \mathbb{F}^n$ be an unknown vector. A **measurement of** $\vec{x}$ is an inner product $\langle \vec{a}, \vec{x} \rangle$ for some known vector $\vec{a} \in \mathbb{F}^n$. A measurement of an unknown matrix M is an inner product in the space $\mathbb{F}^{n^2}$. A measurement $\langle A, M \rangle$ of the matrix M is **rank 1** if A is rank 1.

Thus, evaluations to the bilinear form $f(\vec{x}, \vec{y}) = \vec{x}^t M \vec{y} = \langle \vec{x}^t \vec{y}, M \rangle$ are all rank-1 measurements of M.

Theorem

Over large fields, one can efficiently learn an $n \times n$ matrix of rank $\leq r$ with $4nr$ measurements

Results

Definition

Let $\vec{x} \in \mathbb{F}^n$ be an unknown vector. A **measurement of** $\vec{x}$ is an inner product $\langle \vec{a}, \vec{x} \rangle$ for some known vector $\vec{a} \in \mathbb{F}^n$. A measurement of an unknown matrix M is an inner product in the space $\mathbb{F}^{n^2}$. A measurement $\langle A, M \rangle$ of the matrix M is **rank 1** if A is rank 1.

Thus, evaluations to the bilinear form $f(\vec{x}, \vec{y}) = \vec{x}^t M \vec{y} = \langle \vec{x}^t \vec{y}, M \rangle$ are all rank-1 measurements of M.

Theorem

Over large fields, one can efficiently learn an $n \times n$ matrix of rank $\leq r$ with $4nr$ measurements and can even be done with rank-1 measurements.

Results

Definition

Let $\vec{x} \in \mathbb{F}^n$ be an unknown vector. A **measurement of** $\vec{x}$ is an inner product $\langle \vec{a}, \vec{x} \rangle$ for some known vector $\vec{a} \in \mathbb{F}^n$. A measurement of an unknown matrix M is an inner product in the space $\mathbb{F}^{n^2}$. A measurement $\langle A, M \rangle$ of the matrix M is **rank 1** if A is rank 1.

Thus, evaluations to the bilinear form $f(\vec{x}, \vec{y}) = \vec{x}^t M \vec{y} = \langle \vec{x}^t \vec{y}, M \rangle$ are all rank-1 measurements of M.

Theorem

Over large fields, one can efficiently learn an $n \times n$ matrix of rank $\leq r$ with $4nr$ measurements and can even be done with rank-1 measurements.

The number of measurements is optimal over algebraically closed fields.

Results

Definition

Let $\vec{x} \in \mathbb{F}^n$ be an unknown vector. A **measurement of** $\vec{x}$ is an inner product $\langle \vec{a}, \vec{x} \rangle$ for some known vector $\vec{a} \in \mathbb{F}^n$. A measurement of an unknown matrix M is an inner product in the space $\mathbb{F}^{n^2}$. A measurement $\langle A, M \rangle$ of the matrix M is **rank 1** if A is rank 1.

Thus, evaluations to the bilinear form $f(\vec{x}, \vec{y}) = \vec{x}^t M \vec{y} = \langle \vec{x}^t \vec{y}, M \rangle$ are all rank-1 measurements of M.

Theorem

Over large fields, one can efficiently learn an $n \times n$ matrix of rank $\leq r$ with $4nr$ measurements and can even be done with rank-1 measurements.

The number of measurements is optimal over algebraically closed fields.
The use of rank-1 measurements is novel

Results

Definition

Let $\vec{x} \in \mathbb{F}^n$ be an unknown vector. A **measurement of** $\vec{x}$ is an inner product $\langle \vec{a}, \vec{x} \rangle$ for some known vector $\vec{a} \in \mathbb{F}^n$. A measurement of an unknown matrix M is an inner product in the space $\mathbb{F}^{n^2}$. A measurement $\langle A, M \rangle$ of the matrix M is **rank 1** if A is rank 1.

Thus, evaluations to the bilinear form $f(\vec{x}, \vec{y}) = \vec{x}^t M \vec{y} = \langle \vec{x}^t \vec{y}, M \rangle$ are all rank-1 measurements of M.

Theorem

Over large fields, one can efficiently learn an $n \times n$ matrix of rank $\leq r$ with $4nr$ measurements and can even be done with rank-1 measurements.

The number of measurements is optimal over algebraically closed fields. The use of rank-1 measurements is novel, and thus instantiates the meta-hope, by learning a function from deterministic evaluations.

Results (cont'd)

Results (cont'd)

Definition (Tensor)

Results (cont'd)

Definition (Tensor)

- A tensor is a higher dimensional matrix in $[n]^d$.

Results (cont'd)

Definition (Tensor)

- A tensor is a higher dimensional matrix in $[n]^d$.
- Tensors have a **rank**,

Results (cont'd)

Definition (Tensor)

- A tensor is a higher dimensional matrix in $[n]^d$.
- Tensors have a **rank**, dnr parameters for tensors of rank r,

Results (cont'd)

Definition (Tensor)

- A tensor is a higher dimensional matrix in $[n]^d$.
- Tensors have a **rank**, dnr parameters for tensors of rank r, but n^d in general.

Results (cont'd)

Definition (Tensor)

- A tensor is a higher dimensional matrix in $[n]^d$.
- Tensors have a **rank**, dnr parameters for tensors of rank r, but n^d in general.
- Tensors define a d-linear form,

Results (cont'd)

Definition (Tensor)

- A tensor is a higher dimensional matrix in $[n]^d$.
- Tensors have a **rank**, dnr parameters for tensors of rank r, but n^d in general.
- Tensors define a d-linear form, evaluating that form is a rank-1 measurement of the tensor.

Results (cont'd)

Definition (Tensor)

- A tensor is a higher dimensional matrix in $[n]^d$.
- Tensors have a **rank**, dnr parameters for tensors of rank r, but n^d in general.
- Tensors define a d-linear form, evaluating that form is a rank-1 measurement of the tensor.

Theorem

Results (cont'd)

Definition (Tensor)

- A tensor is a higher dimensional matrix in $[n]^d$.
- Tensors have a **rank**, dnr parameters for tensors of rank r, but n^d in general.
- Tensors define a d-linear form, evaluating that form is a rank-1 measurement of the tensor.

Theorem

There is a deterministic $\mathrm{poly}(n, r, d)^{\log d}$*-time algorithm that reconstructs rank r tensors in $[n]^d$ from evaluations of their d-linear forms.*

Results (cont'd)

Definition (Tensor)

- A tensor is a higher dimensional matrix in $[n]^d$.
- Tensors have a **rank**, dnr parameters for tensors of rank r, but n^d in general.
- Tensors define a d-linear form, evaluating that form is a rank-1 measurement of the tensor.

Theorem

There is a deterministic $\mathrm{poly}(n, r, d)^{\log d}$*-time algorithm that reconstructs rank r tensors in $[n]^d$ from evaluations of their d-linear forms.*

This is the *first* deterministic sub-exponential time algorithm for even determining if the d-linear form is non-zero,

Results (cont'd)

Definition (Tensor)

- A tensor is a higher dimensional matrix in $[n]^d$.
- Tensors have a **rank**, dnr parameters for tensors of rank r, but n^d in general.
- Tensors define a d-linear form, evaluating that form is a rank-1 measurement of the tensor.

Theorem

There is a deterministic $\mathrm{poly}(n, r, d)^{\log d}$*-time algorithm that reconstructs rank r tensors in $[n]^d$ from evaluations of their d-linear forms.*

This is the *first* deterministic sub-exponential time algorithm for even determining if the d-linear form is non-zero, by only using its evaluations.

Results — Low-Rank Recovery

Results — Low-Rank Recovery

Theorem

Results — Low-Rank Recovery

Theorem

Any r-sparse-recovery oracle with measurements $\mathcal{V}$

Results — Low-Rank Recovery

Theorem

Any r-sparse-recovery oracle with measurements $\mathcal{V}$ can be turned into a rank $\leq r$ low-rank recovery algorithm, with measurements $\mathcal{H}$

Results — Low-Rank Recovery

Theorem

Any r-sparse-recovery oracle with measurements $\mathcal{V}$ can be turned into a rank $\leq r$ low-rank recovery algorithm, with measurements $\mathcal{H}$, and $|\mathcal{H}| = 2n|\mathcal{V}|$.

Results — Low-Rank Recovery

Theorem

Any r-sparse-recovery oracle with measurements $\mathcal{V}$ can be turned into a rank $\leq r$ low-rank recovery algorithm, with measurements $\mathcal{H}$, and $|\mathcal{H}| = 2n|\mathcal{V}|$.

An r-sparse-recovery oracle is simply an error-correcting-code with that can correct r errors (thus, distance $2r$)

Results — Low-Rank Recovery

Theorem

Any r-sparse-recovery oracle with measurements $\mathcal{V}$ can be turned into a rank $\leq r$ low-rank recovery algorithm, with measurements $\mathcal{H}$, and $|\mathcal{H}| = 2n|\mathcal{V}|$.

An r-sparse-recovery oracle is simply an error-correcting-code with that can correct r errors (thus, distance $2r$)

$\implies$ take the Reed-Solomon code with $2r$ measurements (for large fields).

Results — Low-Rank Recovery

Theorem

Any r-sparse-recovery oracle with measurements $\mathcal{V}$ can be turned into a rank $\leq r$ low-rank recovery algorithm, with measurements $\mathcal{H}$, and $|\mathcal{H}| = 2n|\mathcal{V}|$.

An r-sparse-recovery oracle is simply an error-correcting-code with that can correct r errors (thus, distance $2r$)

$\implies$ take the Reed-Solomon code with $2r$ measurements (for large fields).

$\implies$ get $2n \cdot 2r = 4nr$ measurements

Results — Low-Rank Recovery

Theorem

Any r-sparse-recovery oracle with measurements $\mathcal{V}$ can be turned into a rank $\leq r$ low-rank recovery algorithm, with measurements $\mathcal{H}$, and $|\mathcal{H}| = 2n|\mathcal{V}|$.

An r-sparse-recovery oracle is simply an error-correcting-code with that can correct r errors (thus, distance $2r$)

$\Longrightarrow$ take the Reed-Solomon code with $2r$ measurements (for large fields).

$\Longrightarrow$ get $2n \cdot 2r = 4nr$ measurements

rank 1 measurements:

Results — Low-Rank Recovery

Theorem

Any r-sparse-recovery oracle with measurements $\mathcal{V}$ can be turned into a rank $\leq r$ low-rank recovery algorithm, with measurements $\mathcal{H}$, and $|\mathcal{H}| = 2n|\mathcal{V}|$.

An r-sparse-recovery oracle is simply an error-correcting-code with that can correct r errors (thus, distance $2r$)

$\implies$ take the Reed-Solomon code with $2r$ measurements (for large fields).

$\implies$ get $2n \cdot 2r = 4nr$ measurements

rank 1 measurements: do a clever change of basis from $\mathcal{H}$ to $\mathcal{H}'$

Sparsity and Rank (I)

Sparsity and Rank (I)

want: M has rank $\leq r$

Sparsity and Rank (I)

want: M has rank $\leq r \implies$ related r sparse vector.

Sparsity and Rank (I)

want: M has rank $\leq r \implies$ related r sparse vector.
hope:

Sparsity and Rank (I)

want: M has rank $\leq r \implies$ related r sparse vector.
hope: some row or column of M is sparse.

Sparsity and Rank (I)

want: M has rank $\leq r \implies$ related r sparse vector.
hope: some row or column of M is sparse. (false)

Sparsity and Rank (I)

want: M has rank $\leq r \implies$ related r sparse vector.
hope: some row or column of M is sparse. (false)
a new hope:

Sparsity and Rank (I)

want: M has rank $\leq r \implies$ related r sparse vector.
hope: some row or column of M is sparse. (false)
a new hope: the some *diagonal* of M is sparse.

Sparsity and Rank (I)

want: M has rank $\leq r \implies$ related r sparse vector.
hope: some row or column of M is sparse. (false)
a new hope: the some *diagonal* of M is sparse.

Definition

Sparsity and Rank (I)

want: M has rank $\leq r \implies$ related r sparse vector.
hope: some row or column of M is sparse. (false)
a new hope: the some *diagonal* of M is sparse.

Definition

Let M be $n \times n$.

Sparsity and Rank (I)

want: M has rank $\leq r \implies$ related r sparse vector.
hope: some row or column of M is sparse. (false)
a new hope: the some *diagonal* of M is sparse.

Definition

Let M be $n \times n$. The k-**diagonal** of M are the entries $\{M_{i,j}\}_{i+j=k}$.

Sparsity and Rank (I)

want: M has rank $\leq r \implies$ related r sparse vector.
hope: some row or column of M is sparse. (false)
a new hope: the some *diagonal* of M is sparse.

Definition

Let M be $n \times n$. The k-**diagonal** of M are the entries $\{M_{i,j}\}_{i+j=k}$.

Lemma

Sparsity and Rank (I)

want: M has rank $\leq r \implies$ related r sparse vector.
hope: some row or column of M is sparse. (false)
a new hope: the some *diagonal* of M is sparse.

Definition

Let M be $n \times n$. The k-**diagonal** of M are the entries $\{M_{i,j}\}_{i+j=k}$.

Lemma

Let M be $n \times n$, of rank $\leq r$.

Sparsity and Rank (I)

want: M has rank $\leq r \implies$ related r sparse vector.
hope: some row or column of M is sparse. (false)
a new hope: the some *diagonal* of M is sparse.

Definition

Let M be $n \times n$. The k-**diagonal** of M are the entries $\{M_{i,j}\}_{i+j=k}$.

Lemma

Let M be $n \times n$, of rank $\leq r$. The first non-zero diagonal is r-sparse.

Sparsity and Rank (II)

Sparsity and Rank (II)

Lemma

Sparsity and Rank (II)

Lemma

Let M be $n \times n$, of rank $\leq r$. The first non-zero diagonal is r-sparse.

Sparsity and Rank (II)

Lemma

Let M be $n \times n$, of rank $\leq r$. The first non-zero diagonal is r-sparse.

Proof by example.

Sparsity and Rank (II)

Lemma

Let M be $n \times n$, of rank $\leq r$. The first non-zero diagonal is r-sparse.

Proof by example.

$$\begin{bmatrix} 0 & 0 & 0 & 0 & a & \star & \star & \star \\ 0 & 0 & 0 & b & \star & \star & \star & \star \\ 0 & 0 & c & \star & \star & \star & \star & \star \\ 0 & d & \star & \star & \star & \star & \star & \star \\ e & \star & \star & \star & \star & \star & \star & \star \\ \star & \star & \star & \star & \star & \star & \star & \star \\ \star & \star & \star & \star & \star & \star & \star & \star \\ \star & \star & \star & \star & \star & \star & \star & \star \end{bmatrix}$$

Sparsity and Rank (II)

Lemma

Let M be $n \times n$, of rank $\leq r$. The first non-zero diagonal is r-sparse.

Proof by example.

$$\begin{bmatrix} 0 & 0 & 0 & 0 & a & \star & \star & \star \\ 0 & 0 & 0 & b & \star & \star & \star & \star \\ 0 & 0 & c & \star & \star & \star & \star & \star \\ 0 & d & \star & \star & \star & \star & \star & \star \\ e & \star & \star & \star & \star & \star & \star & \star \\ \star & \star & \star & \star & \star & \star & \star & \star \\ \star & \star & \star & \star & \star & \star & \star & \star \\ \star & \star & \star & \star & \star & \star & \star & \star \end{bmatrix}$$

The rows with non-zeros amongst the entries a, b, c, d, e are linearly independent

Sparsity and Rank (II)

Lemma

Let M be $n \times n$, of rank $\leq r$. The first non-zero diagonal is r-sparse.

Proof by example.

$$\begin{bmatrix} 0 & 0 & 0 & 0 & a & \star & \star & \star \\ 0 & 0 & 0 & b & \star & \star & \star & \star \\ 0 & 0 & c & \star & \star & \star & \star & \star \\ 0 & d & \star & \star & \star & \star & \star & \star \\ e & \star & \star & \star & \star & \star & \star & \star \\ \star & \star & \star & \star & \star & \star & \star & \star \\ \star & \star & \star & \star & \star & \star & \star & \star \\ \star & \star & \star & \star & \star & \star & \star & \star \end{bmatrix}$$

The rows with non-zeros amongst the entries a, b, c, d, e are linearly independent, as this is a triangular system

Sparsity and Rank (II)

Lemma

Let M be $n \times n$, of rank $\leq r$. The first non-zero diagonal is r-sparse.

Proof by example.

$$\begin{bmatrix} 0 & 0 & 0 & 0 & a & \star & \star & \star \\ 0 & 0 & 0 & b & \star & \star & \star & \star \\ 0 & 0 & c & \star & \star & \star & \star & \star \\ 0 & d & \star & \star & \star & \star & \star & \star \\ e & \star & \star & \star & \star & \star & \star & \star \\ \star & \star & \star & \star & \star & \star & \star & \star \\ \star & \star & \star & \star & \star & \star & \star & \star \\ \star & \star & \star & \star & \star & \star & \star & \star \end{bmatrix}$$

The rows with non-zeros amongst the entries a, b, c, d, e are linearly independent, as this is a triangular system, so this follows from standard linear algebra.

Sparsity and Rank (II)

Lemma

Let M be $n \times n$, of rank $\leq r$. The first non-zero diagonal is r-sparse.

Proof by example.

$$\begin{bmatrix} 0 & 0 & 0 & 0 & a & \star & \star & \star \\ 0 & 0 & 0 & b & \star & \star & \star & \star \\ 0 & 0 & c & \star & \star & \star & \star & \star \\ 0 & d & \star & \star & \star & \star & \star & \star \\ e & \star & \star & \star & \star & \star & \star & \star \\ \star & \star & \star & \star & \star & \star & \star & \star \\ \star & \star & \star & \star & \star & \star & \star & \star \\ \star & \star & \star & \star & \star & \star & \star & \star \end{bmatrix}$$

The rows with non-zeros amongst the entries a, b, c, d, e are linearly independent, as this is a triangular system, so this follows from standard linear algebra. So if the rank is at most 3,

Sparsity and Rank (II)

Lemma

Let M be $n \times n$, of rank $\leq r$. The first non-zero diagonal is r-sparse.

Proof by example.

$$\begin{bmatrix} 0 & 0 & 0 & 0 & a & \star & \star & \star \\ 0 & 0 & 0 & b & \star & \star & \star & \star \\ 0 & 0 & c & \star & \star & \star & \star & \star \\ 0 & d & \star & \star & \star & \star & \star & \star \\ e & \star & \star & \star & \star & \star & \star & \star \\ \star & \star & \star & \star & \star & \star & \star & \star \\ \star & \star & \star & \star & \star & \star & \star & \star \\ \star & \star & \star & \star & \star & \star & \star & \star \end{bmatrix}$$

The rows with non-zeros amongst the entries a, b, c, d, e are linearly independent, as this is a triangular system, so this follows from standard linear algebra. So if the rank is at most 3, then this diagonal is 3-sparse.

□

Proof of Low-Rank Recovery

Proof of Low-Rank Recovery

Learning an 7×7, rank ≤ 3 matrix, essentially using 3-sparse recovery:

Proof of Low-Rank Recovery

Learning an 7×7, rank ≤ 3 matrix, essentially using 3-sparse recovery:

$$\begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 1 & 1 & 1 \\ 0 & 0 & 1 & 1 & 0 & 0 & 1 \\ 0 & 1 & 1 & 1 & 0 & 1 & 1 \\ 0 & 1 & 2 & 3 & 1 & 2 & 3 \\ 0 & 1 & 2 & 3 & 1 & 2 & 3 \\ 0 & 1 & 2 & 3 & 1 & 2 & 3 \end{bmatrix}$$

Proof of Low-Rank Recovery

Learning an 7×7, rank ≤ 3 matrix, essentially using 3-sparse recovery:

$$\begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 1 & 1 & 1 \\ 0 & 0 & 1 & 1 & 0 & 0 & 1 \\ 0 & 1 & 1 & 1 & 0 & 1 & 1 \\ 0 & 1 & 2 & 3 & 1 & 2 & 3 \\ 0 & 1 & 2 & 3 & 1 & 2 & 3 \\ 0 & 1 & 2 & 3 & 1 & 2 & 3 \end{bmatrix}$$

$$\begin{bmatrix} \\ \\ \\ \\ \\ \\ \\ \end{bmatrix}$$

Proof of Low-Rank Recovery

Learning an 7×7, rank ≤ 3 matrix, essentially using 3-sparse recovery:

$$\begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 1 & 1 & 1 \\ 0 & 0 & 1 & 1 & 0 & 0 & 1 \\ 0 & 1 & 1 & 1 & 0 & 1 & 1 \\ 0 & 1 & 2 & 3 & 1 & 2 & 3 \\ 0 & 1 & 2 & 3 & 1 & 2 & 3 \\ 0 & 1 & 2 & 3 & 1 & 2 & 3 \end{bmatrix} \qquad \begin{bmatrix} ? & & & & & & \\ & & & & & & \\ & & & & & & \\ & & & & & & \\ & & & & & & \\ & & & & & & \\ & & & & & & \end{bmatrix}$$

Proof of Low-Rank Recovery

Learning an 7×7, rank ≤ 3 matrix, essentially using 3-sparse recovery:

$$\begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 1 & 1 & 1 \\ 0 & 0 & 1 & 1 & 0 & 0 & 1 \\ 0 & 1 & 1 & 1 & 0 & 1 & 1 \\ 0 & 1 & 2 & 3 & 1 & 2 & 3 \\ 0 & 1 & 2 & 3 & 1 & 2 & 3 \\ 0 & 1 & 2 & 3 & 1 & 2 & 3 \end{bmatrix} \qquad \begin{bmatrix} 0 & ? & & & & & \\ ? & & & & & & \\ & & & & & & \\ & & & & & & \\ & & & & & & \\ & & & & & & \\ & & & & & & \end{bmatrix}$$

Proof of Low-Rank Recovery

Learning an 7×7, rank ≤ 3 matrix, essentially using 3-sparse recovery:

$$\begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 1 & 1 & 1 \\ 0 & 0 & 1 & 1 & 0 & 0 & 1 \\ 0 & 1 & 1 & 1 & 0 & 1 & 1 \\ 0 & 1 & 2 & 3 & 1 & 2 & 3 \\ 0 & 1 & 2 & 3 & 1 & 2 & 3 \\ 0 & 1 & 2 & 3 & 1 & 2 & 3 \end{bmatrix} \qquad \begin{bmatrix} 0 & 0 & \boxed{?} & & & & \\ 0 & \boxed{?} & & & & & \\ \boxed{?} & & & & & & \\ & & & & & & \\ & & & & & & \\ & & & & & & \\ & & & & & & \end{bmatrix}$$

Proof of Low-Rank Recovery

Learning an 7×7, rank ≤ 3 matrix, essentially using 3-sparse recovery:

$$\begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 1 & 1 & 1 \\ 0 & 0 & 1 & 1 & 0 & 0 & 1 \\ 0 & 1 & 1 & 1 & 0 & 1 & 1 \\ 0 & 1 & 2 & 3 & 1 & 2 & 3 \\ 0 & 1 & 2 & 3 & 1 & 2 & 3 \\ 0 & 1 & 2 & 3 & 1 & 2 & 3 \end{bmatrix} \qquad \begin{bmatrix} 0 & 0 & 0 & ? & & & \\ 0 & 0 & ? & & & & \\ 0 & ? & & & & & \\ ? & & & & & & \\ & & & & & & \\ & & & & & & \\ & & & & & & \end{bmatrix}$$

Proof of Low-Rank Recovery

Learning an 7×7, rank ≤ 3 matrix, essentially using 3-sparse recovery:

$$\begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 1 & 1 & 1 \\ 0 & 0 & 1 & 1 & 0 & 0 & 1 \\ 0 & 1 & 1 & 1 & 0 & 1 & 1 \\ 0 & 1 & 2 & 3 & 1 & 2 & 3 \\ 0 & 1 & 2 & 3 & 1 & 2 & 3 \\ 0 & 1 & 2 & 3 & 1 & 2 & 3 \end{bmatrix} \qquad \begin{bmatrix} 0 & 0 & 0 & 0 & ? & & \\ 0 & 0 & 0 & ? & & & \\ 0 & 0 & ? & & & & \\ 0 & ? & & & & & \\ ? & & & & & & \\ & & & & & & \\ & & & & & & \end{bmatrix}$$

Proof of Low-Rank Recovery

Learning an 7×7, rank ≤ 3 matrix, essentially using 3-sparse recovery:

$$\begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 1 & 1 & 1 \\ 0 & 0 & 1 & 1 & 0 & 0 & 1 \\ 0 & 1 & 1 & 1 & 0 & 1 & 1 \\ 0 & 1 & 2 & 3 & 1 & 2 & 3 \\ 0 & 1 & 2 & 3 & 1 & 2 & 3 \\ 0 & 1 & 2 & 3 & 1 & 2 & 3 \end{bmatrix} \qquad \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & \boxed{?} & \\ 0 & 0 & 0 & 1 & \boxed{?} & & \\ 0 & 0 & 1 & \boxed{?} & & & \\ 0 & 1 & \boxed{?} & & & & \\ 0 & \boxed{?} & & & & & \\ \boxed{?} & & & & & & \\ & & & & & & \end{bmatrix}$$

Proof of Low-Rank Recovery

Learning an 7×7, rank ≤ 3 matrix, essentially using 3-sparse recovery:

$$\begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 1 & 1 & 1 \\ 0 & 0 & 1 & 1 & 0 & 0 & 1 \\ 0 & 1 & 1 & 1 & 0 & 1 & 1 \\ 0 & 1 & 2 & 3 & 1 & 2 & 3 \\ 0 & 1 & 2 & 3 & 1 & 2 & 3 \\ 0 & 1 & 2 & 3 & 1 & 2 & 3 \end{bmatrix}$$

$$\begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 1 & 1 & 1 \\ 0 & 0 & 1 & 0 & -1 & -1 & 0 \\ 0 & 1 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

$$\begin{bmatrix} 0 & 0 & 0 & 0 & 0 & \boxed{?} & \\ 0 & 0 & 0 & 1 & \boxed{?} & & \\ 0 & 0 & 1 & \boxed{?} & & & \\ 0 & 1 & \boxed{?} & & & & \\ 0 & \boxed{?} & & & & & \\ \boxed{?} & & & & & & \\ & & & & & & \end{bmatrix}$$

Proof of Low-Rank Recovery

Learning an 7×7, rank ≤ 3 matrix, essentially using 3-sparse recovery:

$$\begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 1 & 1 & 1 \\ 0 & 0 & 1 & 1 & 0 & 0 & 1 \\ 0 & 1 & 1 & 1 & 0 & 1 & 1 \\ 0 & 1 & 2 & 3 & 1 & 2 & 3 \\ 0 & 1 & 2 & 3 & 1 & 2 & 3 \\ 0 & 1 & 2 & 3 & 1 & 2 & 3 \end{bmatrix}$$

$$\begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 1 & 1 & 1 \\ 0 & 0 & 1 & 0 & -1 & -1 & 0 \\ 0 & 1 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

$$\begin{bmatrix} 0 & 0 & 0 & 0 & 0 & ? & \\ 0 & 0 & 0 & 1 & ? & & \\ 0 & 0 & 1 & ? & & & \\ 0 & 1 & ? & & & & \\ 0 & ? & & & & & \\ ? & & & & & & \\ & & & & & & \end{bmatrix}$$

Given a row-reduced upper part of M, the next diagonal is essentially 3-sparse.

Learning an 7×7, rank ≤ 3 matrix, essentially using 3-sparse recovery:

$$\begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 1 & 1 & 1 \\ 0 & 0 & 1 & 1 & 0 & 0 & 1 \\ 0 & 1 & 1 & 1 & 0 & 1 & 1 \\ 0 & 1 & 2 & 3 & 1 & 2 & 3 \\ 0 & 1 & 2 & 3 & 1 & 2 & 3 \\ 0 & 1 & 2 & 3 & 1 & 2 & 3 \end{bmatrix}$$

$$\begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 1 & 1 & 1 \\ 0 & 0 & 1 & 0 & -1 & -1 & 0 \\ 0 & 1 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

$$\begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & \boxed{?} \\ 0 & 0 & 0 & 1 & 1 & \boxed{?} & \\ 0 & 0 & 1 & 1 & \boxed{?} & & \\ 0 & 1 & 1 & \boxed{?} & & & \\ 0 & 1 & \boxed{?} & & & & \\ 0 & \boxed{?} & & & & & \\ \boxed{?} & & & & & & \end{bmatrix}$$

Given a row-reduced upper part of M, the next diagonal is essentially 3-sparse.

Proof of Low-Rank Recovery

Learning an 7×7, rank ≤ 3 matrix, essentially using 3-sparse recovery:

$$\begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 1 & 1 & 1 \\ 0 & 0 & 1 & 1 & 0 & 0 & 1 \\ 0 & 1 & 1 & 1 & 0 & 1 & 1 \\ 0 & 1 & 2 & 3 & 1 & 2 & 3 \\ 0 & 1 & 2 & 3 & 1 & 2 & 3 \\ 0 & 1 & 2 & 3 & 1 & 2 & 3 \end{bmatrix}$$

$$\begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & \boxed{?} \\ 0 & 0 & 0 & 1 & 1 & \boxed{?} & \\ 0 & 0 & 1 & 1 & \boxed{?} & & \\ 0 & 1 & 1 & \boxed{?} & & & \\ 0 & 1 & \boxed{?} & & & & \\ 0 & \boxed{?} & & & & & \\ \boxed{?} & & & & & & \end{bmatrix}$$

$$\begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 1 & 1 & 1 \\ 0 & 0 & 1 & 0 & -1 & -1 & 0 \\ 0 & 1 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

Given a row-reduced upper part of M, the next diagonal is essentially 3-sparse. The effect of (downward) row-reduction on the measurements can be offset.

Proof of Low-Rank Recovery

Learning an 7×7, rank ≤ 3 matrix, essentially using 3-sparse recovery:

$$\begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 1 & 1 & 1 \\ 0 & 0 & 1 & 1 & 0 & 0 & 1 \\ 0 & 1 & 1 & 1 & 0 & 1 & 1 \\ 0 & 1 & 2 & 3 & 1 & 2 & 3 \\ 0 & 1 & 2 & 3 & 1 & 2 & 3 \\ 0 & 1 & 2 & 3 & 1 & 2 & 3 \end{bmatrix}$$

$$\begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 1 & 1 & 1 \\ 0 & 0 & 1 & 0 & -1 & -1 & 0 \\ 0 & 1 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

$$\begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & ? \\ 0 & 0 & 0 & 1 & 1 & ? & \\ 0 & 0 & 1 & 0 & ? & & \\ 0 & 1 & 0 & ? & & & \\ 0 & 0 & ? & & & & \\ 0 & ? & & & & & \\ ? & & & & & & \end{bmatrix}$$

Given a row-reduced upper part of M, the next diagonal is essentially 3-sparse. The effect of (downward) row-reduction on the measurements can be offset.

Proof of Low-Rank Recovery

Learning an 7×7, rank ≤ 3 matrix, essentially using 3-sparse recovery:

$$\begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 1 & 1 & 1 \\ 0 & 0 & 1 & 1 & 0 & 0 & 1 \\ 0 & 1 & 1 & 1 & 0 & 1 & 1 \\ 0 & 1 & 2 & 3 & 1 & 2 & 3 \\ 0 & 1 & 2 & 3 & 1 & 2 & 3 \\ 0 & 1 & 2 & 3 & 1 & 2 & 3 \end{bmatrix}$$

$$\begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 1 & 1 & 1 \\ 0 & 0 & 1 & 0 & -1 & -1 & 0 \\ 0 & 1 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

$$\begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 1 & 1 & \boxed{?} \\ 0 & 0 & 1 & 0 & -1 & \boxed{?} & \\ 0 & 1 & 0 & 0 & \boxed{?} & & \\ 0 & 0 & 1 & \boxed{?} & & & \\ 0 & 1 & \boxed{?} & & & & \\ 0 & \boxed{?} & & & & & \end{bmatrix}$$

Given a row-reduced upper part of M, the next diagonal is essentially 3-sparse. The effect of (downward) row-reduction on the measurements can be offset.

Proof of Low-Rank Recovery

Learning an 7×7, rank ≤ 3 matrix, essentially using 3-sparse recovery:

$$\begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 1 & 1 & 1 \\ 0 & 0 & 1 & 1 & 0 & 0 & 1 \\ 0 & 1 & 1 & 1 & 0 & 1 & 1 \\ 0 & 1 & 2 & 3 & 1 & 2 & 3 \\ 0 & 1 & 2 & 3 & 1 & 2 & 3 \\ 0 & 1 & 2 & 3 & 1 & 2 & 3 \end{bmatrix}$$

$$\begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 1 & 1 & 1 \\ 0 & 0 & 1 & 0 & -1 & -1 & 0 \\ 0 & 1 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

$$\begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 1 & 1 & ? \\ 0 & 0 & 1 & 0 & -1 & ? & \\ 0 & 1 & 0 & 0 & ? & & \\ 0 & 0 & 0 & ? & & & \\ 0 & 0 & ? & & & & \\ 0 & ? & & & & & \end{bmatrix}$$

Given a row-reduced upper part of M, the next diagonal is essentially 3-sparse. The effect of (downward) row-reduction on the measurements can be offset.

Proof of Low-Rank Recovery

Learning an 7×7, rank ≤ 3 matrix, essentially using 3-sparse recovery:

$$\begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 1 & 1 & 1 \\ 0 & 0 & 1 & 1 & 0 & 0 & 1 \\ 0 & 1 & 1 & 1 & 0 & 1 & 1 \\ 0 & 1 & 2 & 3 & 1 & 2 & 3 \\ 0 & 1 & 2 & 3 & 1 & 2 & 3 \\ 0 & 1 & 2 & 3 & 1 & 2 & 3 \end{bmatrix}$$

$$\begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 1 & 1 & 1 \\ 0 & 0 & 1 & 0 & -1 & -1 & 0 \\ 0 & 1 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

$$\begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 1 & 1 & 1 \\ 0 & 0 & 1 & 0 & \text{-1} & \text{-1} & ? \\ 0 & 1 & 0 & 0 & 0 & ? & \\ 0 & 0 & 0 & 2 & ? & & \\ 0 & 0 & 2 & ? & & & \\ 0 & 1 & ? & & & & \end{bmatrix}$$

Given a row-reduced upper part of M, the next diagonal is essentially 3-sparse. The effect of (downward) row-reduction on the measurements can be offset.

Proof of Low-Rank Recovery

Learning an 7×7, rank ≤ 3 matrix, essentially using 3-sparse recovery:

$$\begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 1 & 1 & 1 \\ 0 & 0 & 1 & 1 & 0 & 0 & 1 \\ 0 & 1 & 1 & 1 & 0 & 1 & 1 \\ 0 & 1 & 2 & 3 & 1 & 2 & 3 \\ 0 & 1 & 2 & 3 & 1 & 2 & 3 \\ 0 & 1 & 2 & 3 & 1 & 2 & 3 \end{bmatrix}$$

$$\begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 1 & 1 & 1 \\ 0 & 0 & 1 & 0 & -1 & -1 & 0 \\ 0 & 1 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

$$\begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 1 & 1 & 1 \\ 0 & 0 & 1 & 0 & -1 & -1 & ? \\ 0 & 1 & 0 & 0 & 0 & ? & \\ 0 & 0 & 0 & 0 & ? & & \\ 0 & 0 & 0 & ? & & & \\ 0 & 0 & ? & & & & \end{bmatrix}$$

Given a row-reduced upper part of M, the next diagonal is essentially 3-sparse. The effect of (downward) row-reduction on the measurements can be offset.

Proof of Low-Rank Recovery

Learning an 7×7, rank ≤ 3 matrix, essentially using 3-sparse recovery:

$$\begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 1 & 1 & 1 \\ 0 & 0 & 1 & 1 & 0 & 0 & 1 \\ 0 & 1 & 1 & 1 & 0 & 1 & 1 \\ 0 & 1 & 2 & 3 & 1 & 2 & 3 \\ 0 & 1 & 2 & 3 & 1 & 2 & 3 \\ 0 & 1 & 2 & 3 & 1 & 2 & 3 \end{bmatrix}$$

$$\begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 1 & 1 & 1 \\ 0 & 0 & 1 & 0 & -1 & -1 & 0 \\ 0 & 1 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

$$\begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 1 & 1 & 1 \\ 0 & 0 & 1 & 0 & -1 & -1 & 0 \\ 0 & 1 & 0 & 0 & 0 & 1 & ? \\ 0 & 0 & 0 & 0 & 0 & ? & \\ 0 & 0 & 0 & 3 & ? & & \\ 0 & 0 & 2 & ? & & & \end{bmatrix}$$

Given a row-reduced upper part of M, the next diagonal is essentially 3-sparse. The effect of (downward) row-reduction on the measurements can be offset.

Proof of Low-Rank Recovery

Learning an 7×7, rank ≤ 3 matrix, essentially using 3-sparse recovery:

$$\begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 1 & 1 & 1 \\ 0 & 0 & 1 & 1 & 0 & 0 & 1 \\ 0 & 1 & 1 & 1 & 0 & 1 & 1 \\ 0 & 1 & 2 & 3 & 1 & 2 & 3 \\ 0 & 1 & 2 & 3 & 1 & 2 & 3 \\ 0 & 1 & 2 & 3 & 1 & 2 & 3 \end{bmatrix}$$

$$\begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 1 & 1 & 1 \\ 0 & 0 & 1 & 0 & -1 & -1 & 0 \\ 0 & 1 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

$$\begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 1 & 1 & 1 \\ 0 & 0 & 1 & 0 & -1 & -1 & 0 \\ 0 & 1 & 0 & 0 & 0 & 1 & \boxed{?} \\ 0 & 0 & 0 & 0 & 0 & \boxed{?} & \\ 0 & 0 & 0 & 0 & \boxed{?} & & \\ 0 & 0 & 0 & \boxed{?} & & & \end{bmatrix}$$

Given a row-reduced upper part of M, the next diagonal is essentially 3-sparse. The effect of (downward) row-reduction on the measurements can be offset.

Proof of Low-Rank Recovery

Learning an 7×7, rank ≤ 3 matrix, essentially using 3-sparse recovery:

$$\begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 1 & 1 & 1 \\ 0 & 0 & 1 & 1 & 0 & 0 & 1 \\ 0 & 1 & 1 & 1 & 0 & 1 & 1 \\ 0 & 1 & 2 & 3 & 1 & 2 & 3 \\ 0 & 1 & 2 & 3 & 1 & 2 & 3 \\ 0 & 1 & 2 & 3 & 1 & 2 & 3 \end{bmatrix}$$

$$\begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 1 & 1 & 1 \\ 0 & 0 & 1 & 0 & -1 & -1 & 0 \\ 0 & 1 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

$$\begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 1 & 1 & 1 \\ 0 & 0 & 1 & 0 & -1 & -1 & 0 \\ 0 & 1 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & \boxed{?} \\ 0 & 0 & 0 & 0 & 0 & \boxed{?} & \\ 0 & 0 & 0 & 3 & \boxed{?} & & \end{bmatrix}$$

Given a row-reduced upper part of M, the next diagonal is essentially 3-sparse. The effect of (downward) row-reduction on the measurements can be offset.

Proof of Low-Rank Recovery

Learning an 7×7, rank ≤ 3 matrix, essentially using 3-sparse recovery:

$$\begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 1 & 1 & 1 \\ 0 & 0 & 1 & 1 & 0 & 0 & 1 \\ 0 & 1 & 1 & 1 & 0 & 1 & 1 \\ 0 & 1 & 2 & 3 & 1 & 2 & 3 \\ 0 & 1 & 2 & 3 & 1 & 2 & 3 \\ 0 & 1 & 2 & 3 & 1 & 2 & 3 \end{bmatrix}$$

$$\begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 1 & 1 & 1 \\ 0 & 0 & 1 & 0 & -1 & -1 & 0 \\ 0 & 1 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

$$\begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 1 & 1 & 1 \\ 0 & 0 & 1 & 0 & -1 & -1 & 0 \\ 0 & 1 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & ? \\ 0 & 0 & 0 & 0 & 0 & ? & \\ 0 & 0 & 0 & 0 & ? & & \end{bmatrix}$$

Given a row-reduced upper part of M, the next diagonal is essentially 3-sparse. The effect of (downward) row-reduction on the measurements can be offset.

Proof of Low-Rank Recovery

Learning an 7×7, rank ≤ 3 matrix, essentially using 3-sparse recovery:

$$\begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 1 & 1 & 1 \\ 0 & 0 & 1 & 1 & 0 & 0 & 1 \\ 0 & 1 & 1 & 1 & 0 & 1 & 1 \\ 0 & 1 & 2 & 3 & 1 & 2 & 3 \\ 0 & 1 & 2 & 3 & 1 & 2 & 3 \\ 0 & 1 & 2 & 3 & 1 & 2 & 3 \end{bmatrix}$$

$$\begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 1 & 1 & 1 \\ 0 & 0 & 1 & 0 & -1 & -1 & 0 \\ 0 & 1 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

$$\begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 1 & 1 & 1 \\ 0 & 0 & 1 & 0 & -1 & -1 & 0 \\ 0 & 1 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & ? \\ 0 & 0 & 0 & 0 & 0 & ? & \end{bmatrix}$$

Given a row-reduced upper part of M, the next diagonal is essentially 3-sparse. The effect of (downward) row-reduction on the measurements can be offset.

Proof of Low-Rank Recovery

Learning an 7×7, rank ≤ 3 matrix, essentially using 3-sparse recovery:

$$\begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 1 & 1 & 1 \\ 0 & 0 & 1 & 1 & 0 & 0 & 1 \\ 0 & 1 & 1 & 1 & 0 & 1 & 1 \\ 0 & 1 & 2 & 3 & 1 & 2 & 3 \\ 0 & 1 & 2 & 3 & 1 & 2 & 3 \\ 0 & 1 & 2 & 3 & 1 & 2 & 3 \end{bmatrix}$$

$$\begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 1 & 1 & 1 \\ 0 & 0 & 1 & 0 & -1 & -1 & 0 \\ 0 & 1 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

$$\begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 1 & 1 & 1 \\ 0 & 0 & 1 & 0 & -1 & -1 & 0 \\ 0 & 1 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & \boxed{?} \end{bmatrix}$$

Given a row-reduced upper part of M, the next diagonal is essentially 3-sparse. The effect of (downward) row-reduction on the measurements can be offset.

Proof of Low-Rank Recovery

Learning an 7×7, rank ≤ 3 matrix, essentially using 3-sparse recovery:

$$\begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 1 & 1 & 1 \\ 0 & 0 & 1 & 1 & 0 & 0 & 1 \\ 0 & 1 & 1 & 1 & 0 & 1 & 1 \\ 0 & 1 & 2 & 3 & 1 & 2 & 3 \\ 0 & 1 & 2 & 3 & 1 & 2 & 3 \\ 0 & 1 & 2 & 3 & 1 & 2 & 3 \end{bmatrix}$$

$$\begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 1 & 1 & 1 \\ 0 & 0 & 1 & 0 & -1 & -1 & 0 \\ 0 & 1 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

$$\begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 1 & 1 & 1 \\ 0 & 0 & 1 & 0 & -1 & -1 & 0 \\ 0 & 1 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

Given a row-reduced upper part of M, the next diagonal is essentially 3-sparse. The effect of (downward) row-reduction on the measurements can be offset.

Proof of Low-Rank Recovery

Learning an 7×7, rank ≤ 3 matrix, essentially using 3-sparse recovery:

$$\begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 1 & 1 & 1 \\ 0 & 0 & 1 & 1 & 0 & 0 & 1 \\ 0 & 1 & 1 & 1 & 0 & 1 & 1 \\ 0 & 1 & 2 & 3 & 1 & 2 & 3 \\ 0 & 1 & 2 & 3 & 1 & 2 & 3 \\ 0 & 1 & 2 & 3 & 1 & 2 & 3 \end{bmatrix}$$

$$\begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 1 & 1 & 1 \\ 0 & 0 & 1 & 1 & 0 & 0 & 1 \\ 0 & 1 & 1 & 1 & 0 & 1 & 1 \\ 0 & 1 & 2 & 3 & 1 & 2 & 3 \\ 0 & 1 & 2 & 3 & 1 & 2 & 3 \\ 0 & 1 & 2 & 3 & 1 & 2 & 3 \end{bmatrix}$$

$$\begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 1 & 1 & 1 \\ 0 & 0 & 1 & 0 & -1 & -1 & 0 \\ 0 & 1 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

Given a row-reduced upper part of M, the next diagonal is essentially 3-sparse. The effect of (downward) row-reduction on the measurements can be offset.

Proof of Low-Rank Recovery (cont'd)

Proof of Low-Rank Recovery (cont'd)

Theorem

Proof of Low-Rank Recovery (cont'd)

Theorem

Any r-sparse-recovery oracle with measurements $\mathcal{V}$ can be turned into a rank $\leq r$ low-rank recovery algorithm, with measurements $\mathcal{H}$, and $|\mathcal{H}| = 2n|\mathcal{V}|$.

Proof of Low-Rank Recovery (cont'd)

Theorem

Any r-sparse-recovery oracle with measurements $\mathcal{V}$ can be turned into a rank $\leq r$ low-rank recovery algorithm, with measurements $\mathcal{H}$, and $|\mathcal{H}| = 2n|\mathcal{V}|$.

Proof.

Proof of Low-Rank Recovery (cont'd)

Theorem

Any r-sparse-recovery oracle with measurements $\mathcal{V}$ can be turned into a rank $\leq r$ low-rank recovery algorithm, with measurements $\mathcal{H}$, and $|\mathcal{H}| = 2n|\mathcal{V}|$.

Proof.

- M rank $\leq r$

Proof of Low-Rank Recovery (cont'd)

Theorem

Any r-sparse-recovery oracle with measurements $\mathcal{V}$ can be turned into a rank $\leq r$ low-rank recovery algorithm, with measurements $\mathcal{H}$, and $|\mathcal{H}| = 2n|\mathcal{V}|$.

Proof.

- M rank $\leq r \implies$ the first non-zero diagonal is r-sparse

Proof of Low-Rank Recovery (cont'd)

Theorem

Any r-sparse-recovery oracle with measurements $\mathcal{V}$ can be turned into a rank $\leq r$ low-rank recovery algorithm, with measurements $\mathcal{H}$, and $|\mathcal{H}| = 2n|\mathcal{V}|$.

Proof.

- M rank $\leq r \implies$ the first non-zero diagonal is r-sparse
- M rank $\leq r$,

Proof of Low-Rank Recovery (cont'd)

Theorem

Any r-sparse-recovery oracle with measurements $\mathcal{V}$ can be turned into a rank $\leq r$ low-rank recovery algorithm, with measurements $\mathcal{H}$, and $|\mathcal{H}| = 2n|\mathcal{V}|$.

Proof.

- M rank $\leq r \implies$ the first non-zero diagonal is r-sparse
- M rank $\leq r$, reduced row-echelon form

Proof of Low-Rank Recovery (cont'd)

Theorem

Any r-sparse-recovery oracle with measurements $\mathcal{V}$ can be turned into a rank $\leq r$ low-rank recovery algorithm, with measurements $\mathcal{H}$, and $|\mathcal{H}| = 2n|\mathcal{V}|$.

Proof.

- M rank $\leq r \implies$ the first non-zero diagonal is r-sparse
- M rank $\leq r$, reduced row-echelon form $\implies$ every diagonal is r-sparse

Proof of Low-Rank Recovery (cont'd)

Theorem

Any r-sparse-recovery oracle with measurements $\mathcal{V}$ can be turned into a rank $\leq r$ low-rank recovery algorithm, with measurements $\mathcal{H}$, and $|\mathcal{H}| = 2n|\mathcal{V}|$.

Proof.

- M rank $\leq r \implies$ the first non-zero diagonal is r-sparse
- M rank $\leq r$, reduced row-echelon form $\implies$ every diagonal is r-sparse
- M rank $\leq r$,

Proof of Low-Rank Recovery (cont'd)

Theorem

Any r-sparse-recovery oracle with measurements $\mathcal{V}$ can be turned into a rank $\leq r$ low-rank recovery algorithm, with measurements $\mathcal{H}$, and $|\mathcal{H}| = 2n|\mathcal{V}|$.

Proof.

- M rank $\leq r \implies$ the first non-zero diagonal is r-sparse
- M rank $\leq r$, reduced row-echelon form $\implies$ every diagonal is r-sparse
- M rank $\leq r$, first k diagonals in reduced row-echelon form

Proof of Low-Rank Recovery (cont'd)

Theorem

Any r-sparse-recovery oracle with measurements $\mathcal{V}$ can be turned into a rank $\leq r$ low-rank recovery algorithm, with measurements $\mathcal{H}$, and $|\mathcal{H}| = 2n|\mathcal{V}|$.

Proof.

- M rank $\leq r \implies$ the first non-zero diagonal is r-sparse
- M rank $\leq r$, reduced row-echelon form $\implies$ every diagonal is r-sparse
- M rank $\leq r$, first k diagonals in reduced row-echelon form $\implies$ $(k+1)$-diagonal of M is essentially r-sparse

Proof of Low-Rank Recovery (cont'd)

Theorem

Any r-sparse-recovery oracle with measurements $\mathcal{V}$ can be turned into a rank $\leq r$ low-rank recovery algorithm, with measurements $\mathcal{H}$, and $|\mathcal{H}| = 2n|\mathcal{V}|$.

Proof.

- M rank $\leq r \implies$ the first non-zero diagonal is r-sparse
- M rank $\leq r$, reduced row-echelon form $\implies$ every diagonal is r-sparse
- M rank $\leq r$, first k diagonals in reduced row-echelon form $\implies$ $(k+1)$-diagonal of M is essentially r-sparse (and actually $2r$-sparse)

Proof of Low-Rank Recovery (cont'd)

Theorem

Any r-sparse-recovery oracle with measurements $\mathcal{V}$ can be turned into a rank $\leq r$ low-rank recovery algorithm, with measurements $\mathcal{H}$, and $|\mathcal{H}| = 2n|\mathcal{V}|$.

Proof.

- M rank $\leq r \implies$ the first non-zero diagonal is r-sparse
- M rank $\leq r$, reduced row-echelon form $\implies$ every diagonal is r-sparse
- M rank $\leq r$, first k diagonals in reduced row-echelon form $\implies$ $(k+1)$-diagonal of M is essentially r-sparse (and actually $2r$-sparse)
- learning M via iteratively learning diagonals and row reducing (downward)

Proof of Low-Rank Recovery (cont'd)

Theorem

Any r-sparse-recovery oracle with measurements $\mathcal{V}$ can be turned into a rank $\leq r$ low-rank recovery algorithm, with measurements $\mathcal{H}$, and $|\mathcal{H}| = 2n|\mathcal{V}|$.

Proof.

- M rank $\leq r \implies$ the first non-zero diagonal is r-sparse
- M rank $\leq r$, reduced row-echelon form $\implies$ every diagonal is r-sparse
- M rank $\leq r$, first k diagonals in reduced row-echelon form $\implies$ $(k+1)$-diagonal of M is essentially r-sparse (and actually $2r$-sparse)
- learning M via iteratively learning diagonals and row reducing (downward)
- undo row-reduction at the end

Proof of Low-Rank Recovery (cont'd)

Theorem

Any r-sparse-recovery oracle with measurements $\mathcal{V}$ can be turned into a rank $\leq r$ low-rank recovery algorithm, with measurements $\mathcal{H}$, and $|\mathcal{H}| = 2n|\mathcal{V}|$.

Proof.

- M rank $\leq r \implies$ the first non-zero diagonal is r-sparse
- M rank $\leq r$, reduced row-echelon form $\implies$ every diagonal is r-sparse
- M rank $\leq r$, first k diagonals in reduced row-echelon form $\implies$ $(k+1)$-diagonal of M is essentially r-sparse (and actually $2r$-sparse)
- learning M via iteratively learning diagonals and row reducing (downward)
- undo row-reduction at the end
- called sparse-recovery $2n$ times — once per diagonal $\square$

Summary

Summary

Results:

Summary

Results:

- Low-rank recovery of matrices is reducible to sparse recovery of vectors.

Summary

Results:

- Low-rank recovery of matrices is reducible to sparse recovery of vectors.
- There is a deterministic quasi-polynomial-time algorithm for learning low-rank tensors.

Summary

Results:

- Low-rank recovery of matrices is reducible to sparse recovery of vectors.
- There is a deterministic quasi-polynomial-time algorithm for learning low-rank tensors.

Open Questions:

Summary

Results:

- Low-rank recovery of matrices is reducible to sparse recovery of vectors.
- There is a deterministic quasi-polynomial-time algorithm for learning low-rank tensors.

Open Questions:

- Can our reduction from low-rank recovery to sparse recovery be made stable?

Summary

Results:

- Low-rank recovery of matrices is reducible to sparse recovery of vectors.
- There is a deterministic quasi-polynomial-time algorithm for learning low-rank tensors.

Open Questions:

- Can our reduction from low-rank recovery to sparse recovery be made stable?
- Deterministic *polynomial*-time algorithm for learning tensors?

TOC