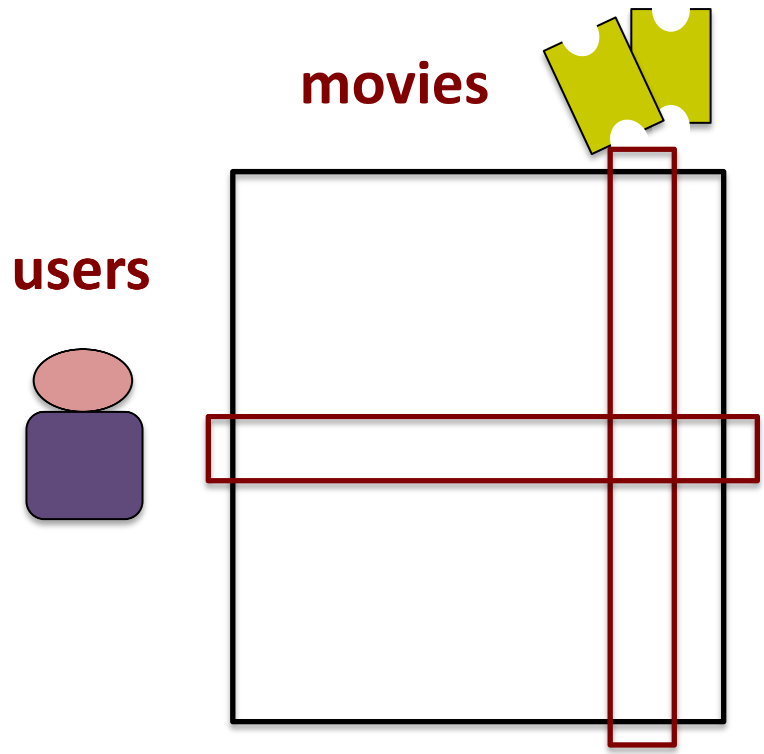
Tensor Completion and Refuting Random CSPs

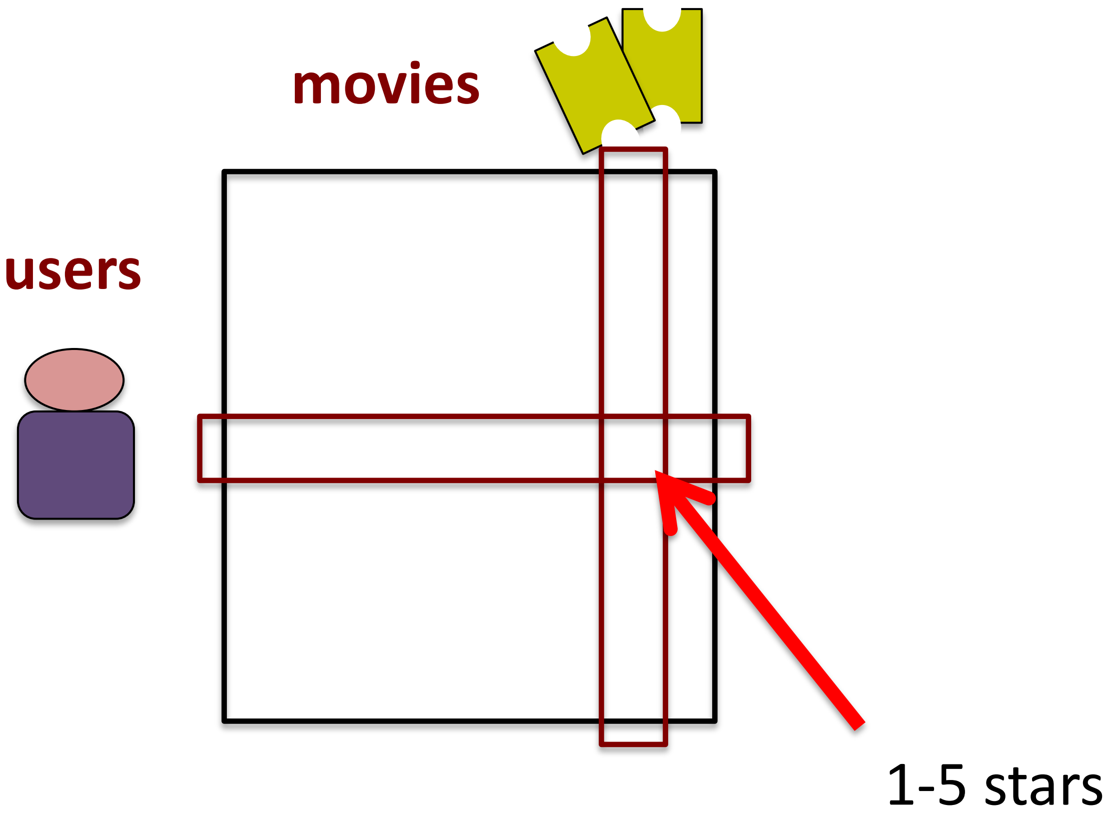
Ankur Moitra (MIT)

IPAM Tutorial, Part 2

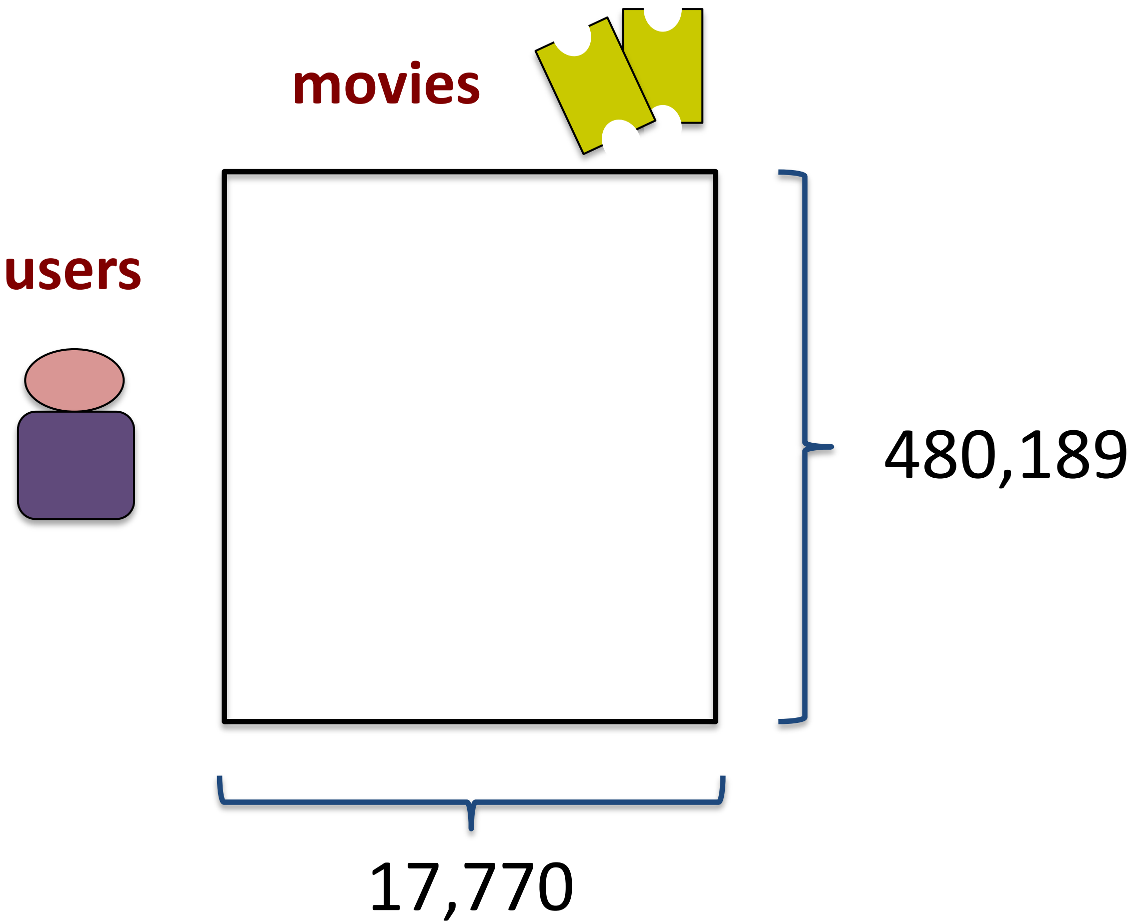
THE NETFLIX PROBLEM



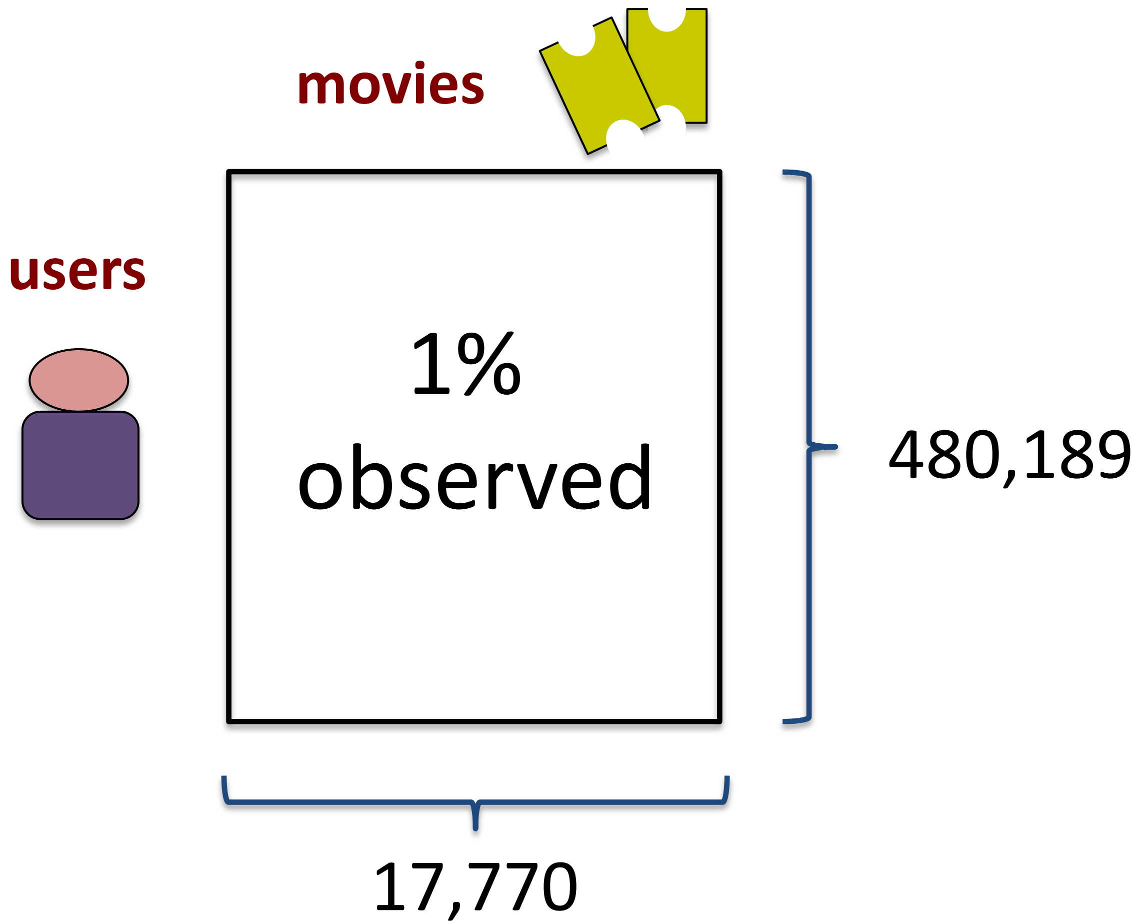
THE NETFLIX PROBLEM



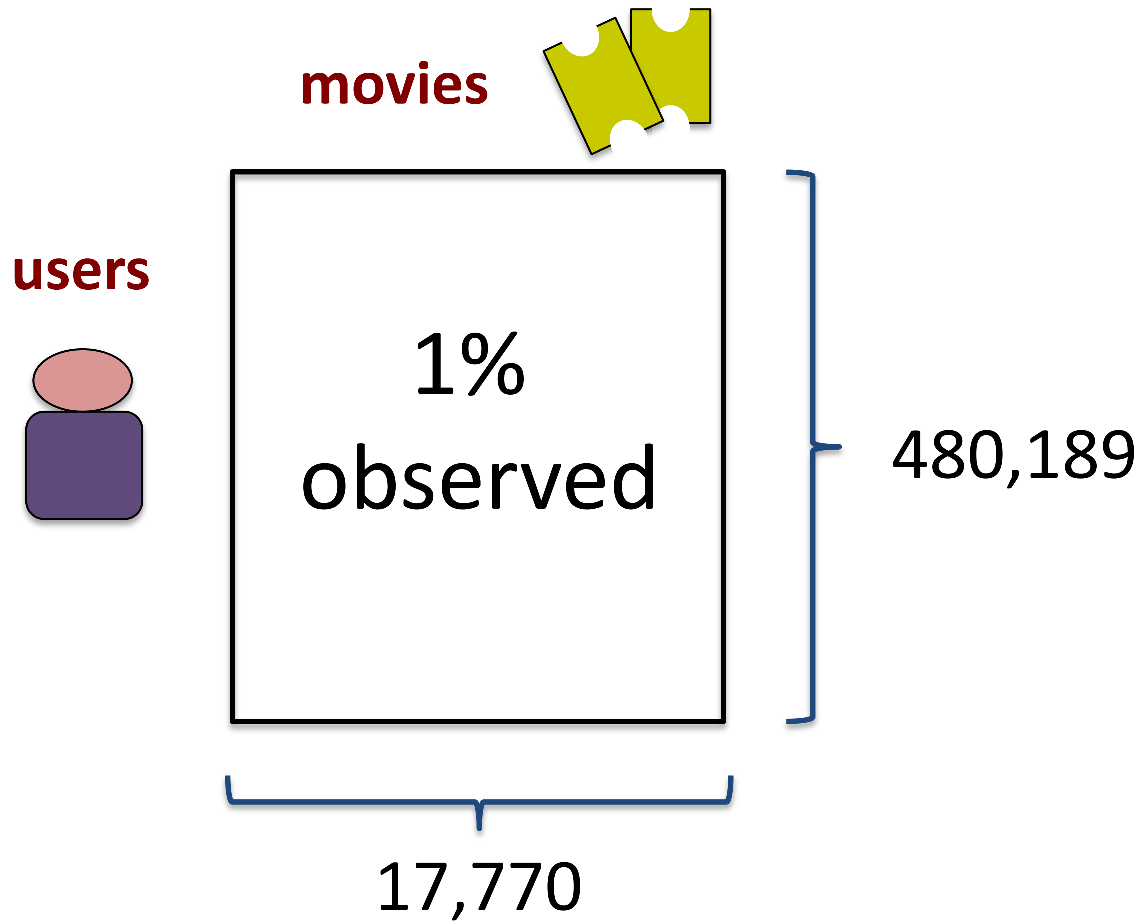
THE NETFLIX PROBLEM



THE NETFLIX PROBLEM



THE NETFLIX PROBLEM



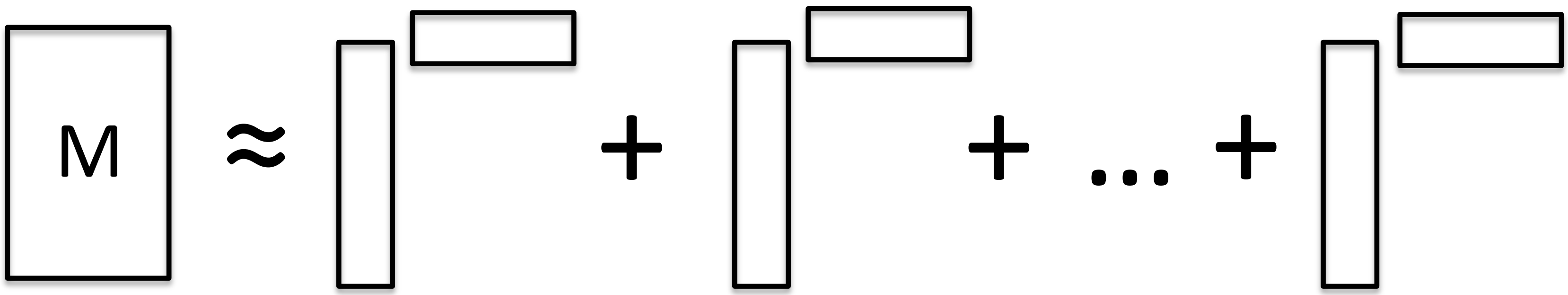
Can we (approximately) fill-in the missing entries?

MATRIX COMPLETION

Let M be an unknown, approximately low-rank matrix

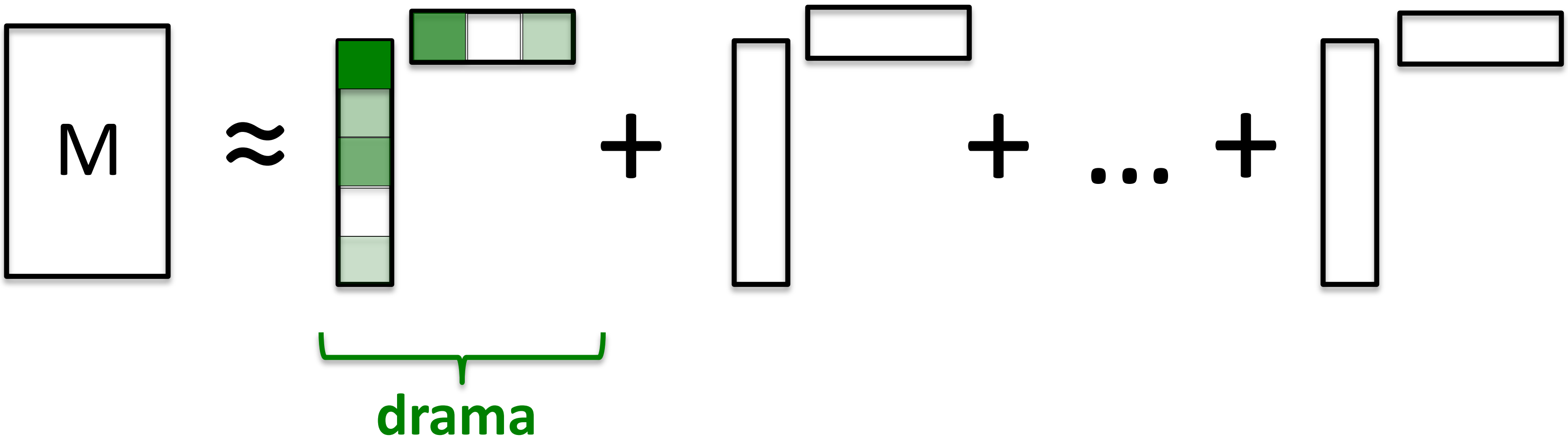
MATRIX COMPLETION

Let M be an unknown, approximately low-rank matrix



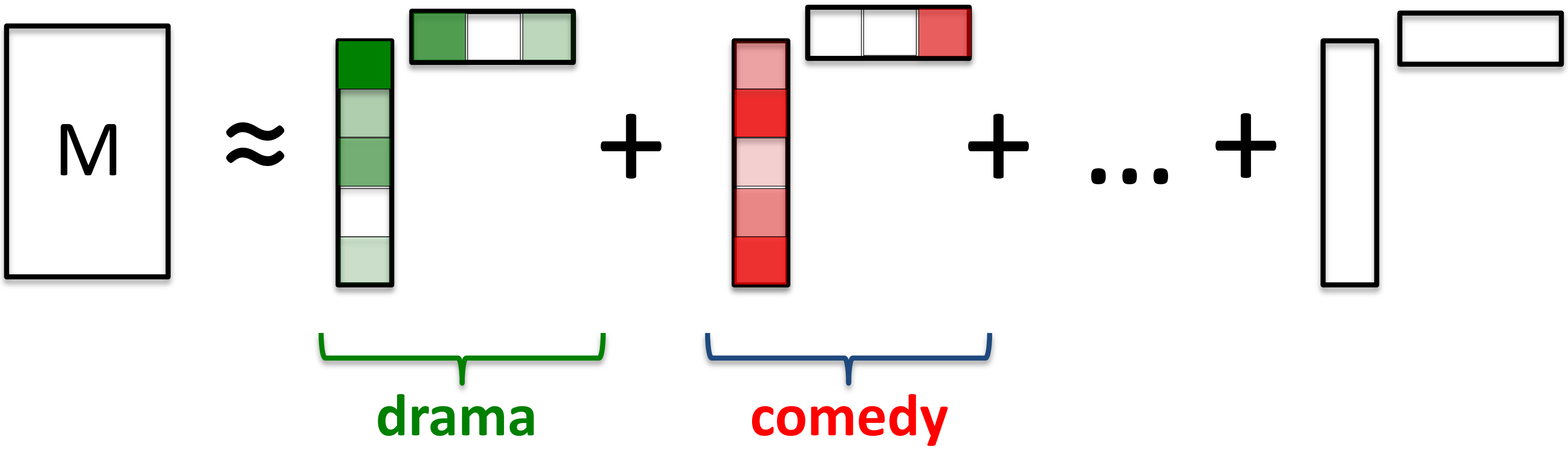
MATRIX COMPLETION

Let M be an unknown, approximately low-rank matrix



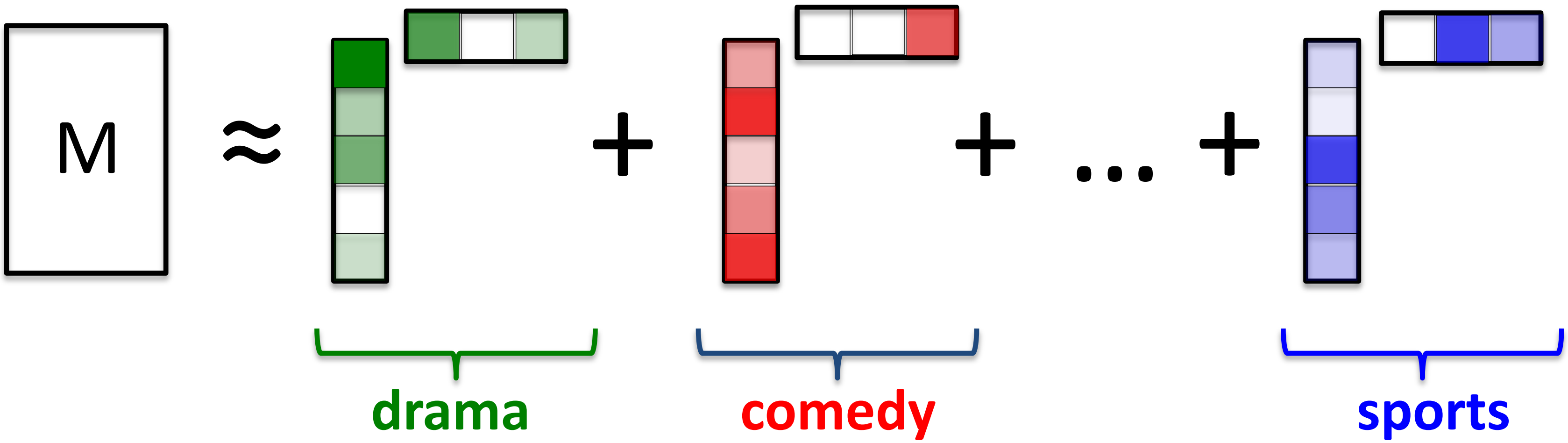
MATRIX COMPLETION

Let M be an unknown, approximately low-rank matrix



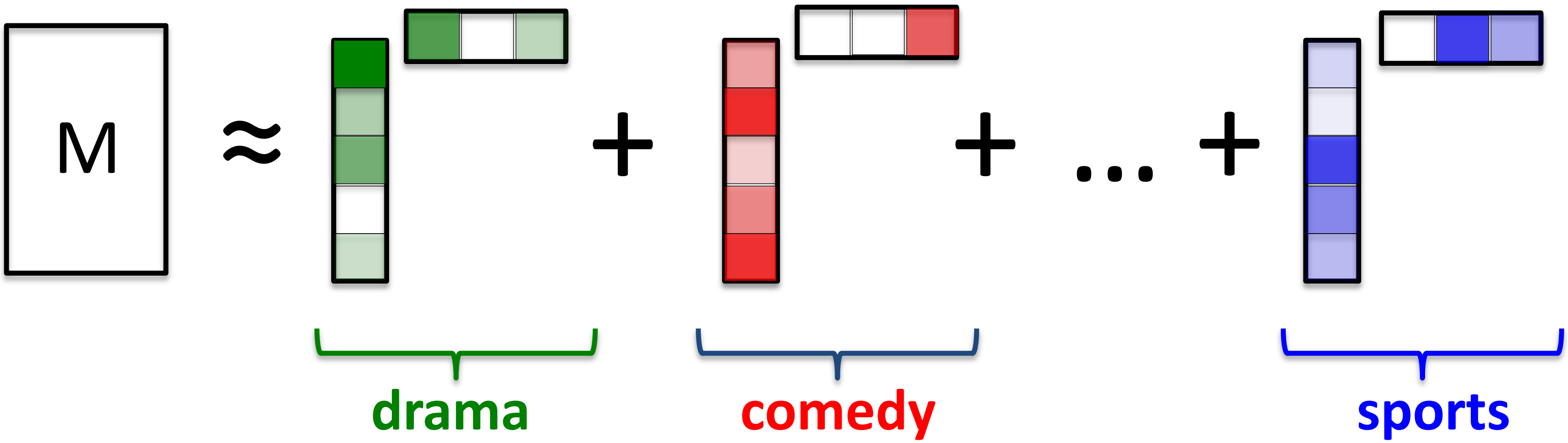
MATRIX COMPLETION

Let M be an unknown, approximately low-rank matrix



MATRIX COMPLETION

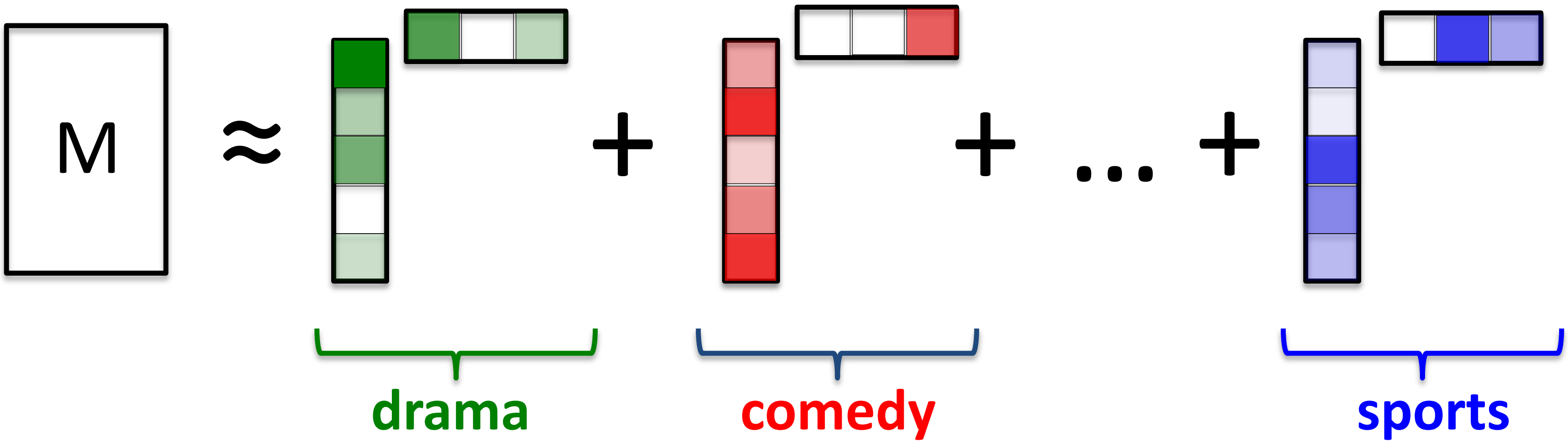
Let M be an unknown, approximately low-rank matrix



Model: we are given random observations $M_{i,j}$ for all $i,j \in \Omega$

MATRIX COMPLETION

Let M be an unknown, approximately low-rank matrix



Model: we are given random observations $M_{i,j}$ for all $i,j \in \Omega$

Is there an efficient algorithm to recover M?

MATRIX COMPLETION

The natural formulation is **non-convex**, and **NP-hard**

$$\min \operatorname{rank}(X) \quad \text{s.t.} \quad \frac{1}{|\Omega|} \sum_{(i,j)\in\Omega} |X_{i,j} - M_{i,j}| \le \eta$$

MATRIX COMPLETION

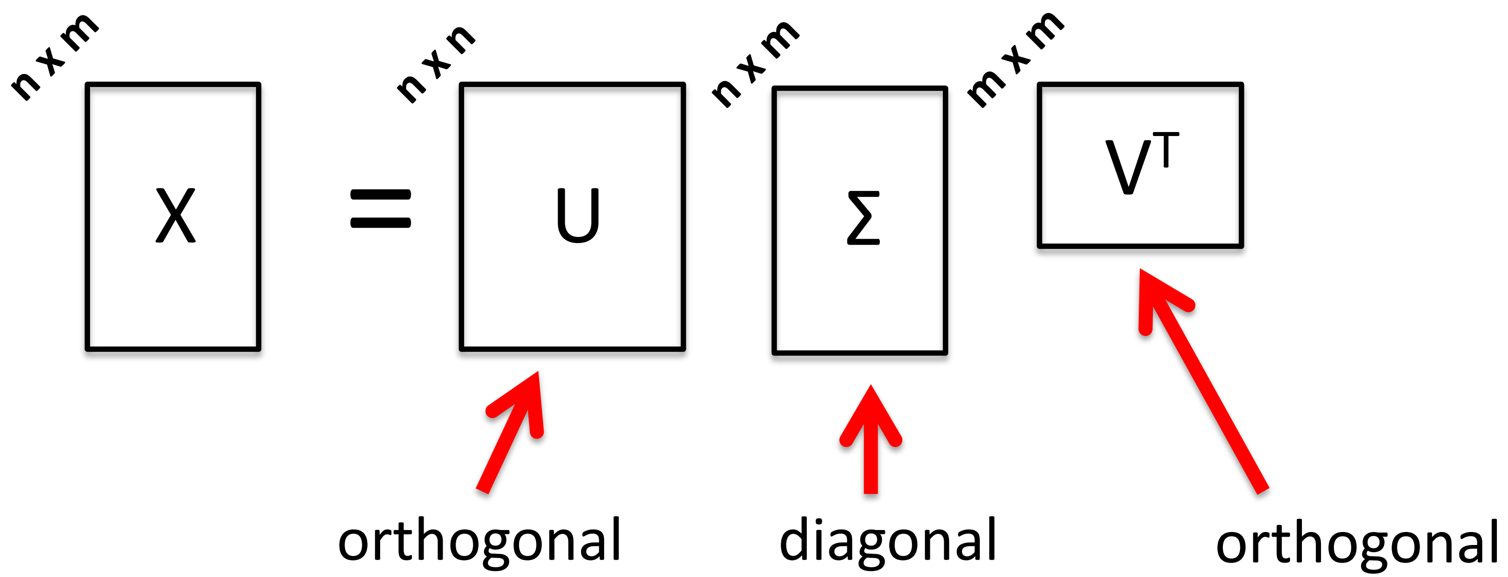
The natural formulation is **non-convex**, and **NP-hard**

$$\min \operatorname{rank}(X) \ \text{ s.t. } \frac{1}{|\Omega|} \sum_{(i,j)\in\Omega} |X_{i,j} - M_{i,j}| \le \eta$$

There is a powerful, convex relaxation…

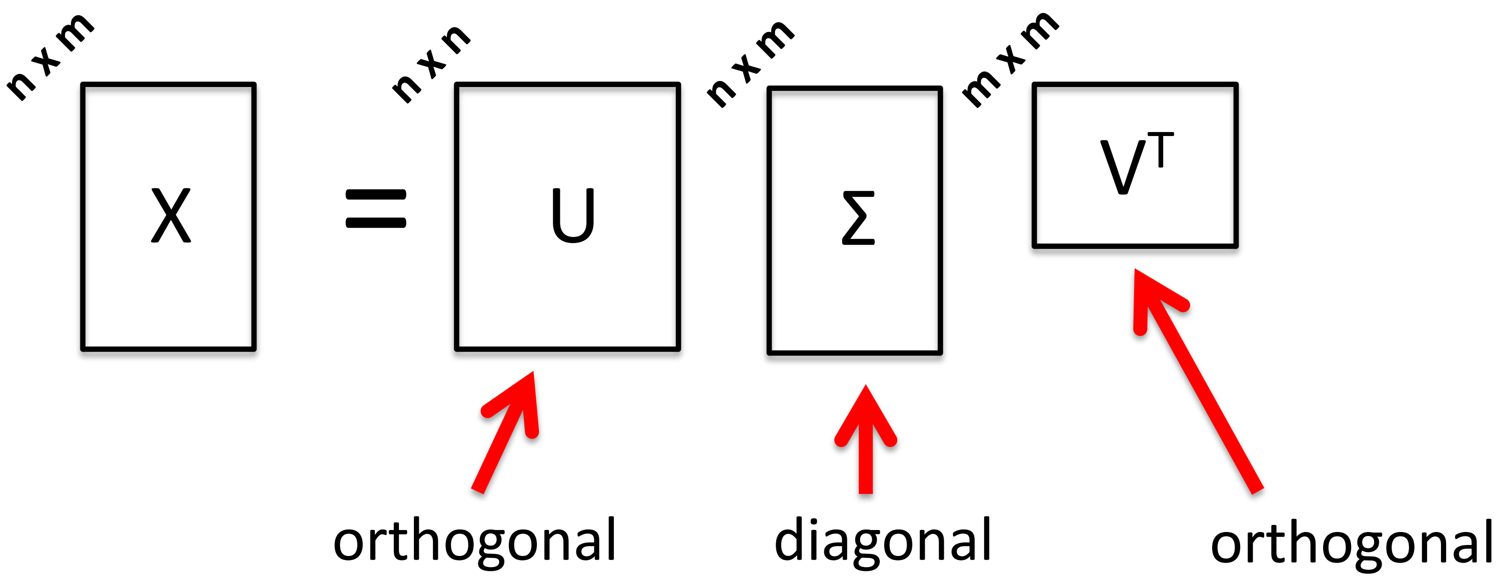
THE NUCLEAR NORM

Consider the **singular value decomposition** of X:



THE NUCLEAR NORM

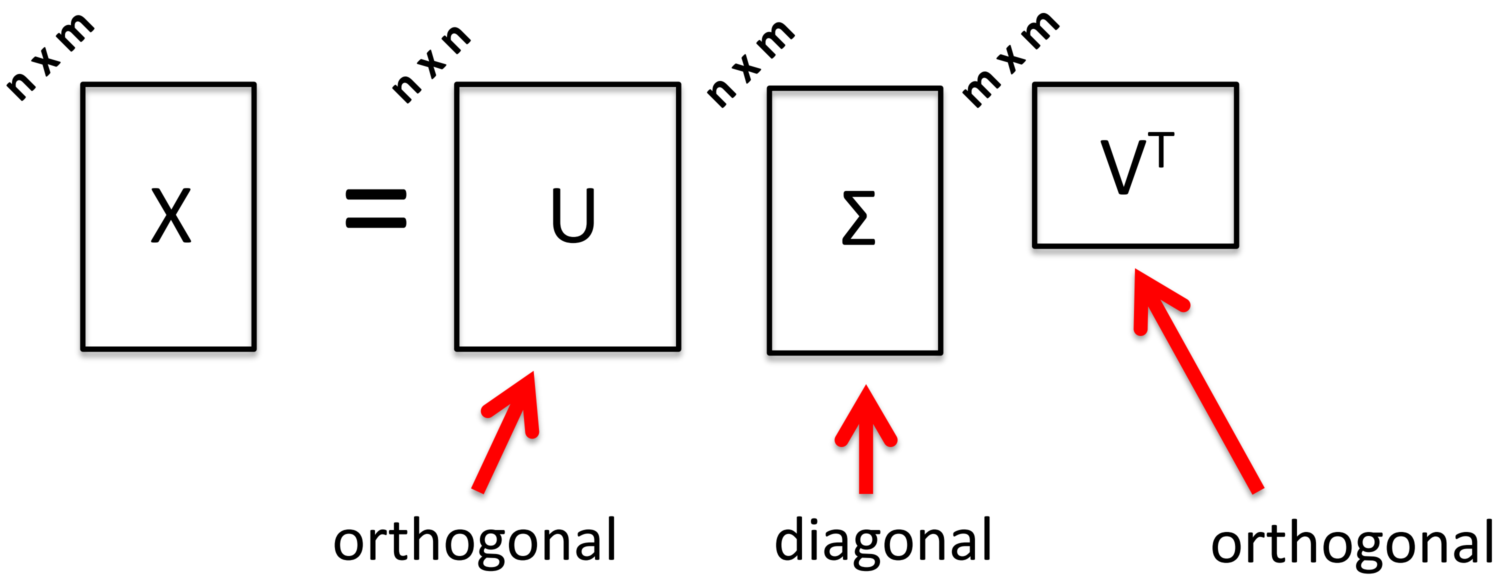
Consider the **singular value decomposition** of X:



Let $\sigma_1 \geq \sigma_2 \geq \ldots \sigma_r > \sigma_{r+1} = \ldots \sigma_m = 0$ be the singular values

THE NUCLEAR NORM

Consider the **singular value decomposition** of X:



Let $\sigma_1 \geq \sigma_2 \geq \ldots \sigma_r > \sigma_{r+1} = \ldots \sigma_m = 0$ be the singular values

Then $\mathrm{rank}(X) = r$, and $\|X\|_* = \sigma_1 + \sigma_2 + \ldots + \sigma_r$ (**nuclear norm**)

This yields a convex relaxation, that can be solved efficiently:

$$\min \|X\|_* \text{ s.t. } \frac{1}{|\Omega|} \sum_{(i,j)\in\Omega} |X_{i,j} - M_{i,j}| \leq \eta \qquad \textbf{(P)}$$

[Fazel], [Srebro, Shraibman], [Recht, Fazel, Parrilo], [Candes, Recht], [Candes, Tao], [Candes, Plan], [Recht],

This yields a convex relaxation, that can be solved efficiently:

$$\min \|X\|_* \text{ s.t. } \frac{1}{|\Omega|} \sum_{(i,j)\in\Omega} |X_{i,j} - M_{i,j}| \leq \eta \quad \textbf{(P)}$$

[Fazel], [Srebro, Shraibman], [Recht, Fazel, Parrilo], [Candes, Recht], [Candes, Tao], [Candes, Plan], [Recht],

Theorem: If M is n x n and has rank r, and is C-incoherent then **(P)** recovers M exactly from $C^6 n r \log^2 n$ observations

This yields a convex relaxation, that can be solved efficiently:

$$\min \|X\|_* \text{ s.t. } \frac{1}{|\Omega|} \sum_{(i,j)\in\Omega} |X_{i,j} - M_{i,j}| \leq \eta \quad \textbf{(P)}$$

[Fazel], [Srebro, Shraibman], [Recht, Fazel, Parrilo], [Candes, Recht], [Candes, Tao], [Candes, Plan], [Recht],

Theorem: If M is n x n and has rank r, and is C-incoherent then **(P)** recovers M exactly from $C^6 n r \log^2 n$ observations

This is nearly optimal, since there are O(nr) parameters

OUTLINE

Part I: Introduction

- The Netflix Problem and Matrix Completion
- Tensor Completion

Part II: To Random CSPs and Back

- A Distinguishing Problem
- Random CSPs and Strong Refutation
- Rademacher Complexity

Part III: Resolution

OUTLINE

Part I: Introduction

- The Netflix Problem and Matrix Completion
- **Tensor Completion**

Part II: To Random CSPs and Back

- A Distinguishing Problem
- Random CSPs and Strong Refutation
- Rademacher Complexity

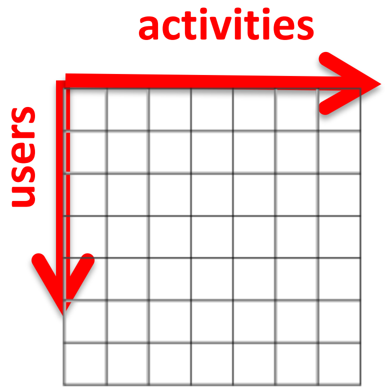
Part III: Resolution

TENSOR COMPLETION

Can using **more than two** attributes can lead to better recommendations?

TENSOR COMPLETION

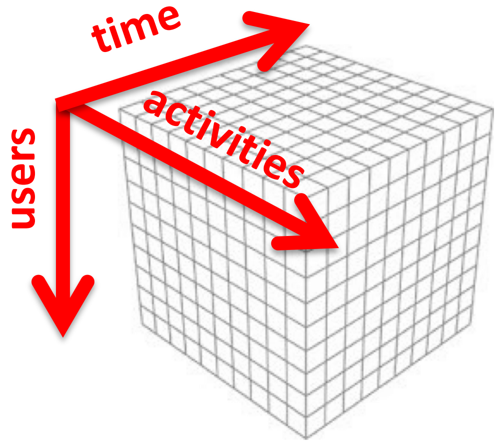
Can using **more than two** attributes can lead to better recommendations?



e.g. Groupon

TENSOR COMPLETION

Can using **more than two** attributes can lead to better recommendations?

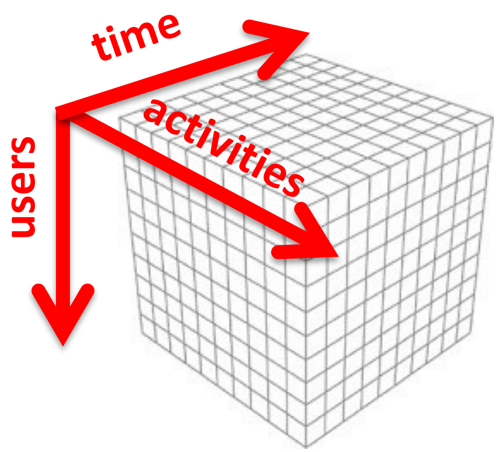


e.g. Groupon

time: season, time of day, weekday/weekend, etc

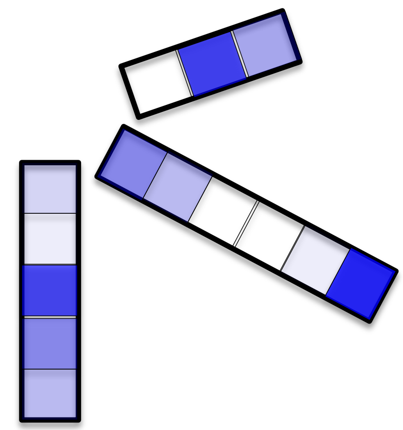
TENSOR COMPLETION

Can using **more than two** attributes can lead to better recommendations?



e.g. Groupon

$$T \approx \sum_{i=1}^{r}$$

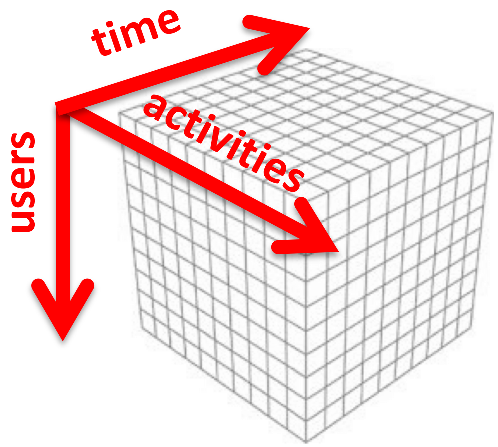


snow sports

time: season, time of day, weekday/weekend, etc

TENSOR COMPLETION

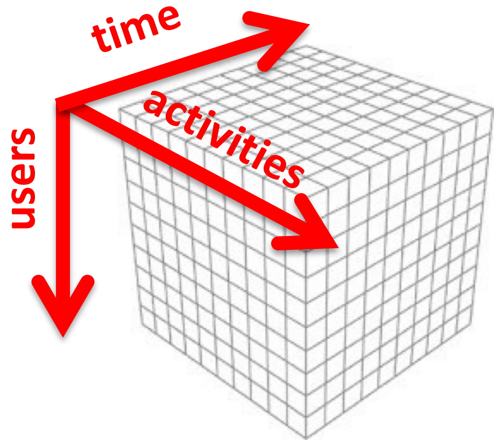
Can using **more than two** attributes can lead to better recommendations?



$$T \approx \sum_{i=1}^{r} a_i \otimes b_i \otimes c_i$$

TENSOR COMPLETION

Can using **more than two** attributes can lead to better recommendations?



$$T \approx \sum_{i=1}^{r} a_i \otimes b_i \otimes c_i$$

Can we (approximately) fill-in the missing entries?

THE TROUBLE WITH TENSORS

Natural approach (suggested by many authors):

$$\min \|X\|_* \text{ s.t. } \frac{1}{|\Omega|} \sum_{(i,j,k)\in\Omega} |X_{i,j,k} - T_{i,j,k}| \leq \eta \quad \textbf{(P)}$$

tensor nuclear norm

THE TROUBLE WITH TENSORS

Natural approach (suggested by many authors):

$$\min \|X\|_* \text{ s.t. } \frac{1}{|\Omega|}\sum_{(i,j,k)\in\Omega} |X_{i,j,k} - T_{i,j,k}| \leq \eta \quad \textbf{(P)}$$

tensor nuclear norm

The tensor nuclear norm is **NP-hard** to compute!

[Gurvits], [Liu], [Harrow, Montanaro]

In fact, most of the linear algebra toolkit is **ill-posed**, or **computationally hard** for tensors…

In fact, most of the linear algebra toolkit is **ill-posed**, or **computationally hard** for tensors…

e.g. [Hillar, Lim] "Most Tensor Problems are NP-Hard"

In fact, most of the linear algebra toolkit is **ill-posed**, or **computationally hard** for tensors…

e.g. [Hillar, Lim] "Most Tensor Problems are NP-Hard"

Table I. Tractability of Tensor Problems

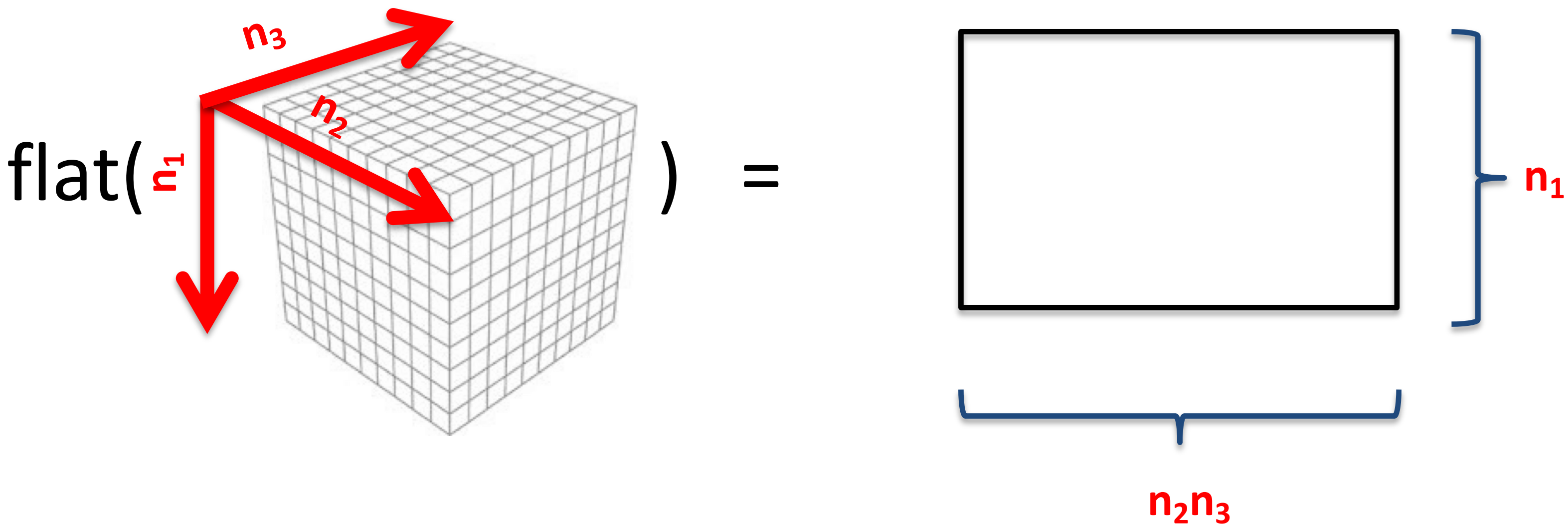
Problem	**Complexity**
Bivariate Matrix Functions over $\mathbb{R}$, $\mathbb{C}$	Undecidable (Proposition 12.2)
Bilinear System over $\mathbb{R}$, $\mathbb{C}$	NP-hard (Theorems 2.6, 3.7, 3.8)
Eigenvalue over $\mathbb{R}$	NP-hard (Theorem 1.3)
Approximating Eigenvector over $\mathbb{R}$	NP-hard (Theorem 1.5)
Symmetric Eigenvalue over $\mathbb{R}$	NP-hard (Theorem 9.3)
Approximating Symmetric Eigenvalue over $\mathbb{R}$	NP-hard (Theorem 9.6)
Singular Value over $\mathbb{R}$, $\mathbb{C}$	NP-hard (Theorem 1.7)
Symmetric Singular Value over $\mathbb{R}$	NP-hard (Theorem 10.2)
Approximating Singular Vector over $\mathbb{R}$, $\mathbb{C}$	NP-hard (Theorem 6.3)
Spectral Norm over $\mathbb{R}$	NP-hard (Theorem 1.10)
Symmetric Spectral Norm over $\mathbb{R}$	NP-hard (Theorem 10.2)
Approximating Spectral Norm over $\mathbb{R}$	NP-hard (Theorem 1.11)
Nonnegative Definiteness	NP-hard (Theorem 11.2)
Best Rank-1 Approximation	NP-hard (Theorem 1.13)
Best Symmetric Rank-1 Approximation	NP-hard (Theorem 10.2)
Rank over $\mathbb{R}$ or $\mathbb{C}$	NP-hard (Theorem 8.2)
Enumerating Eigenvectors over $\mathbb{R}$	#P-hard (Corollary 1.16)
Combinatorial Hyperdeterminant	NP-, #P-, VNP-hard (Theorems 4.1 , 4.2, Corollary 4.3)
Geometric Hyperdeterminant	Conjectures 1.9, 13.1
Symmetric Rank	Conjecture 13.2
Bilinear Programming	Conjecture 13.4
Bilinear Least Squares	Conjecture 13.5

FLATTENING A TENSOR

Many tensor methods rely on **flattening**:

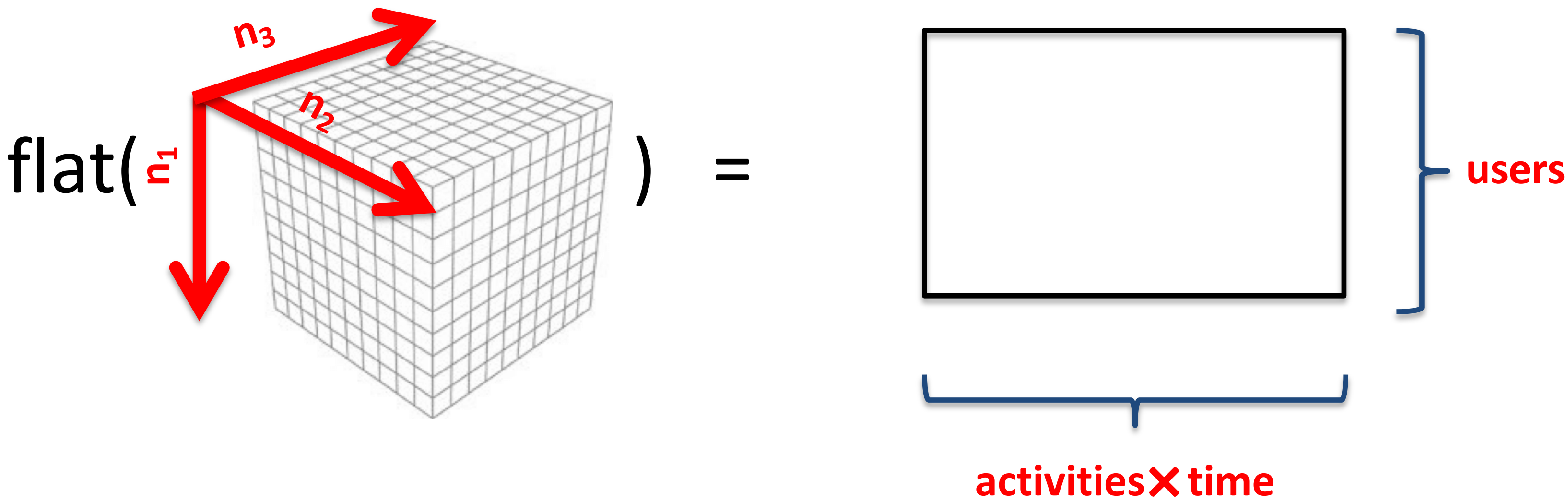
FLATTENING A TENSOR

Many tensor methods rely on **flattening**:



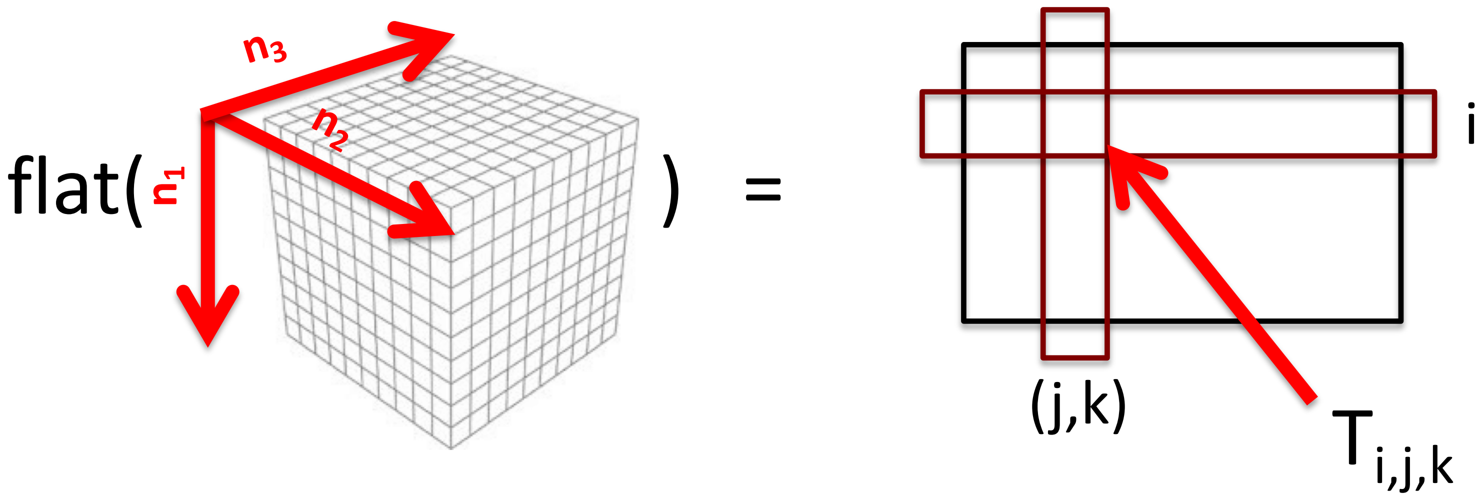
FLATTENING A TENSOR

Many tensor methods rely on **flattening**:



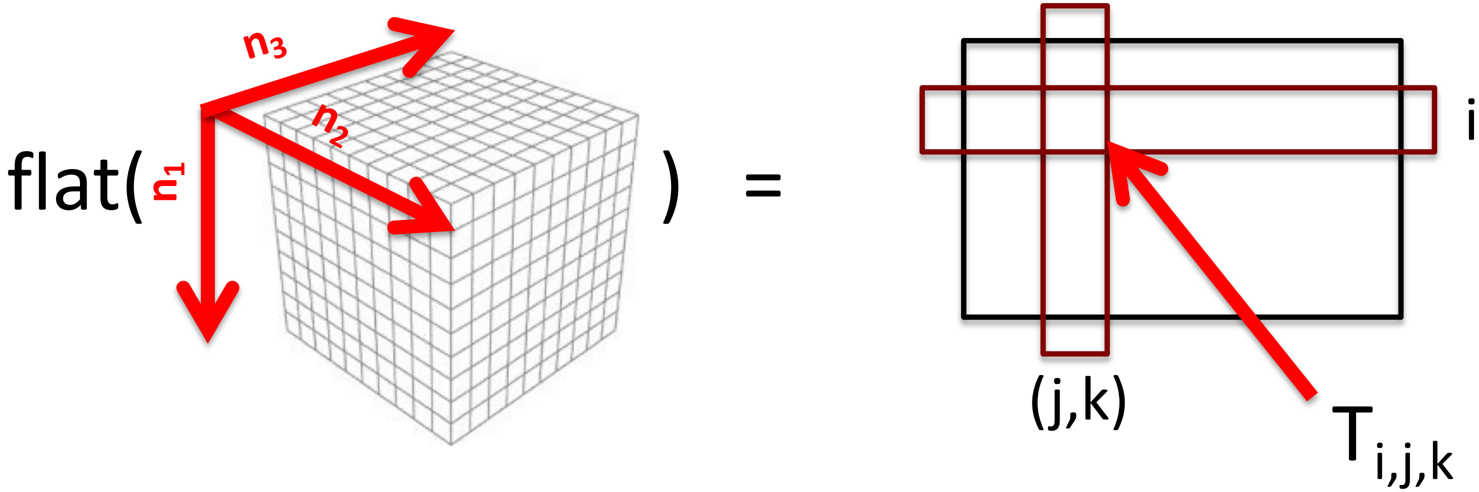
FLATTENING A TENSOR

Many tensor methods rely on **flattening**:



FLATTENING A TENSOR

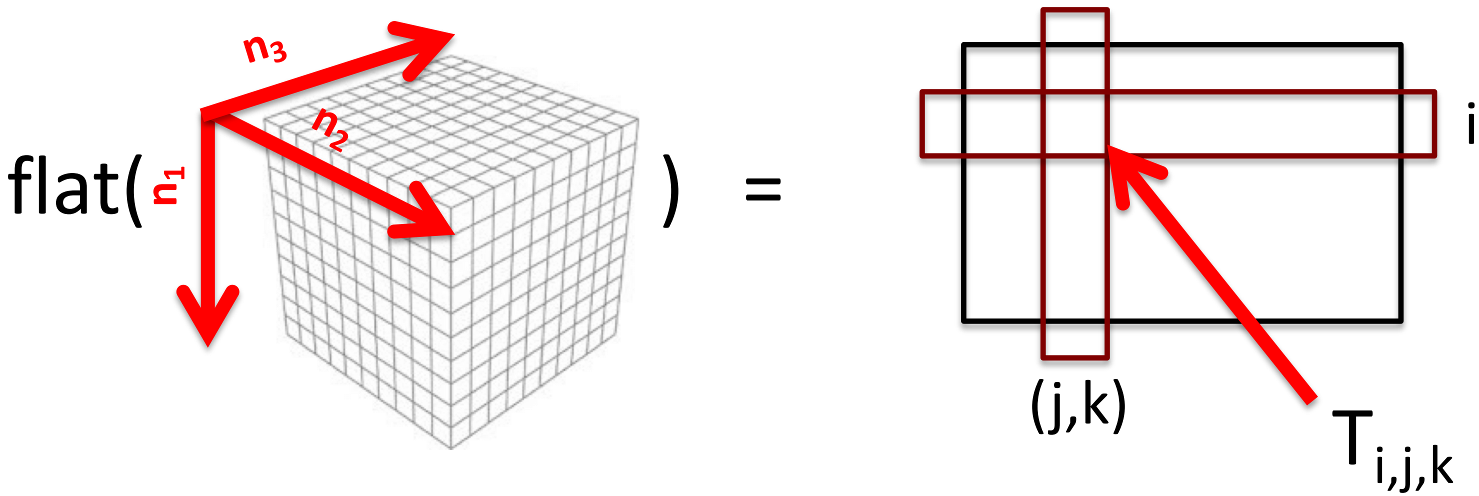
Many tensor methods rely on **flattening**:



This is a **rearrangement** of the entries, into a matrix, that does not increase its **rank**

FLATTENING A TENSOR

Many tensor methods rely on **flattening**:



$$\mathrm{flat}\left(\sum_{i=1}^{r} a_i \otimes b_i \otimes c_i\right) = \sum_{i=1}^{r} a_i \otimes \underbrace{\mathrm{vec}(b_i c_i^T)}_{n_2 n_3\text{-dimensional vector}}$$

Let $n_1 = n_2 = n_3 = n$

We would need $\tilde{O}(n^2 r)$ observations to fill-in flat(T)

Let $n_1 = n_2 = n_3 = n$

We would need $\widetilde{O}(n^2 r)$ observations to fill-in flat(T)

There are many other variants of **flattening**, but with comparable guarantees

[Liu, Musialski, Wonka, Ye], [Gandy, Recht, Yamada], [Signoretto, De Lathauwer, Suykens], [Tomioko, Hayashi, Kashima], [Mu, Huang, Wright, Goldfarb], …

Let $n_1 = n_2 = n_3 = n$

We would need $\widetilde{O}(n^2 r)$ observations to fill-in flat(T)

There are many other variants of **flattening**, but with comparable guarantees

[Liu, Musialski, Wonka, Ye], [Gandy, Recht, Yamada], [Signoretto, De Lathauwer, Suykens], [Tomioko, Hayashi, Kashima], [Mu, Huang, Wright, Goldfarb], …

Can we beat flattening?

Let $n_1 = n_2 = n_3 = n$

We would need $\widetilde{O}(n^2 r)$ observations to fill-in flat(T)

There are many other variants of **flattening**, but with comparable guarantees

[Liu, Musialski, Wonka, Ye], [Gandy, Recht, Yamada],
[Signoretto, De Lathauwer, Suykens], [Tomioko, Hayashi, Kashima],
[Mu, Huang, Wright, Goldfarb], …

Can we beat flattening?

Can we make better predictions than we do by treating each **activity x time** as unrelated?

BEATING FLATTENING

Suppose we are given $|\Omega| = m$ noisy observations from T:

$$T = \sum_{i=1}^{r} \sigma_i \, a_i \otimes b_i \otimes c_i + \textbf{noise}$$

with $|\sigma_i|$, $|a_i|_\infty$, $|b_i|_\infty$, $|c_i|_\infty \leq C$

bdd by η

BEATING FLATTENING

Suppose we are given $|\Omega| = m$ noisy observations from T:

$$T = \sum_{i=1}^{r} \sigma_i \, a_i \otimes b_i \otimes c_i + \textbf{noise}$$

with $|\sigma_i|$, $|a_i|_\infty$, $|b_i|_\infty$, $|c_i|_\infty \leq C$

bdd by η

Theorem [Barak, Moitra]: There is an efficient algorithm that with prob 1-δ, outputs X with

$$\frac{1}{n^3} \sum_{i,j,k} |X_{i,j,k} - T_{i,j,k}| \leq C^3 r \sqrt{\frac{n^{3/2} \log^4 n}{m}} + 2C^3 r \sqrt{\frac{\ln(2/\delta)}{m}} + 2\eta$$

LOWER BOUNDS

Not only is the **tensor nuclear norm** hard to compute, but…

LOWER BOUNDS

Not only is the **tensor nuclear norm** hard to compute, but…

Noisy tensor completion with m observations $\rightarrow$ Refute random 3-SAT with m clauses

LOWER BOUNDS

Not only is the **tensor nuclear norm** hard to compute, but…

Noisy tensor completion with m observations → Refute random 3-SAT with m clauses

Believed to be hard,
If $m = n^{3/2-\delta}$

OUTLINE

Part I: Introduction

- The Netflix Problem and Matrix Completion
- Tensor Completion

Part II: To Random CSPs and Back

- A Distinguishing Problem
- Random CSPs and Strong Refutation
- Rademacher Complexity

Part III: Resolution

OUTLINE

Part I: Introduction

- The Netflix Problem and Matrix Completion
- Tensor Completion

Part II: To Random CSPs and Back

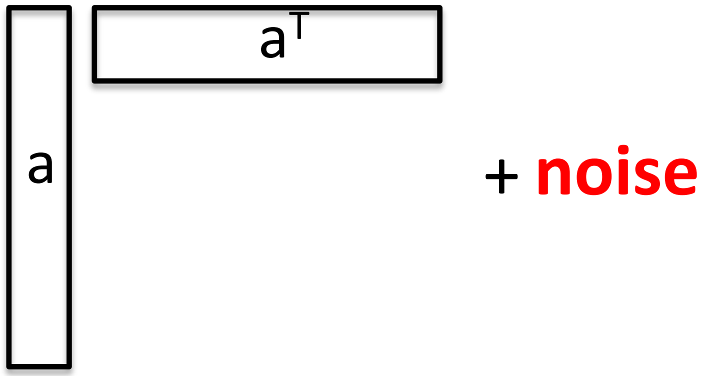
- **A Distinguishing Problem**
- Random CSPs and Strong Refutation
- Rademacher Complexity

Part III: Resolution

Can we distinguish between low-rank and random?

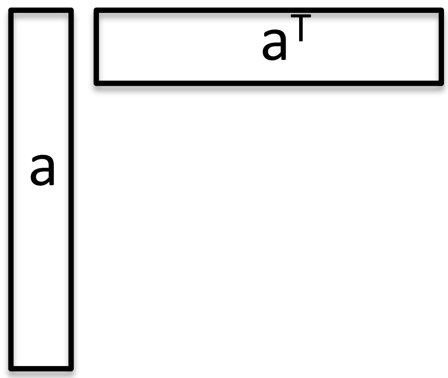
Can we distinguish between low-rank and random?

Case #1: Approximately low-rank



Can we distinguish between low-rank and random?

Case #1: Approximately low-rank



+ **noise**

For each $(i,j) \in \Omega$

$$M_{i,j} = \begin{cases} a_i a_j & \text{w/ probability } 3/4 \\ \text{random } \pm 1 & \text{w/ probability } 1/4 \end{cases}$$

where each $a_i = \pm 1$

Can we distinguish between low-rank and random?

Can we distinguish between low-rank and random?

Case #2: Random

For each $(i,j) \in \Omega$, $M_{i,j}$ = random ± 1

Can we distinguish between low-rank and random?

Case #2: Random

random

For each $(i,j) \in \Omega$, $M_{i,j}$ = random ± 1

In **Case #1** the entries are (somewhat) predictable, but in **Case #2** they are completely **unpredictable**

OUTLINE

Part I: Introduction

- The Netflix Problem and Matrix Completion
- Tensor Completion

Part II: To Random CSPs and Back

- A Distinguishing Problem
- Random CSPs and Strong Refutation
- Rademacher Complexity

Part III: Resolution

OUTLINE

Part I: Introduction

- The Netflix Problem and Matrix Completion
- Tensor Completion

Part II: To Random CSPs and Back

- A Distinguishing Problem
- **Random CSPs and Strong Refutation**
- Rademacher Complexity

Part III: Resolution

There are two very different communities that (essentially) attacked this same distinguishing problem:

There are two very different communities that (essentially) attacked this same distinguishing problem:

The community working on **matrix completion**

There are two very different communities that (essentially) attacked this same distinguishing problem:

The community working on **matrix completion**

The community working on **refuting random CSPs**

AN INTERPRETATION

We can interpret:

$$(i_1, j_1; \boldsymbol{\sigma_1}), (i_2, j_2; \boldsymbol{\sigma_2}), \ldots, (i_m, j_m; \boldsymbol{\sigma_m})$$

±1 r.v.

as a random 2-XOR formula ψ

AN INTERPRETATION

We can interpret:

$$(i_1, j_1; \boldsymbol{\sigma_1}), (i_2, j_2; \boldsymbol{\sigma_2}), \ldots, (i_m, j_m; \boldsymbol{\sigma_m})$$

±1 r.v.

as a random 2-XOR formula ψ

In particular each observation/fctn value maps to a clause:

$$(i, j, \boldsymbol{\sigma}) \longrightarrow \underbrace{v_i \bullet v_j}_{\text{variables}} = \underbrace{\sigma}_{\text{constraint}}$$

AN INTERPRETATION

We can interpret:

$$(i_1, j_1; \boldsymbol{\sigma_1}), (i_2, j_2; \boldsymbol{\sigma_2}), \ldots, (i_m, j_m; \boldsymbol{\sigma_m})$$

±1 r.v.

as a random 2-XOR formula ψ (and vice-versa)

In particular each observation/fctn value maps to a clause:

$$(i, j, \boldsymbol{\sigma}) \longrightarrow \underbrace{v_i \bullet v_j}_{\text{variables}} = \underbrace{\sigma}_{\text{constraint}}$$

STRONG REFUTATION

We will say that an algorithm **strongly refutes*** random 2-XOR with m clauses if:

STRONG REFUTATION

We will say that an algorithm **strongly refutes*** random 2-XOR with m clauses if:

(1) On any 2-XOR formula ψ, it outputs **val** where:

$$\mathrm{OPT}(\psi) \leq \mathrm{val}(\psi)$$

STRONG REFUTATION

We will say that an algorithm **strongly refutes*** random 2-XOR with m clauses if:

(1) On any 2-XOR formula ψ, it outputs **val** where:

$$\underbrace{\mathrm{OPT}(\psi)}_{\text{largest fraction of clauses of } \psi \text{ that can be satisfied}} \leq \mathrm{val}(\psi)$$

STRONG REFUTATION

We will say that an algorithm **strongly refutes*** random 2-XOR with m clauses if:

(1) On any 2-XOR formula ψ, it outputs **val** where:

$$\mathrm{OPT}(\psi) \leq \mathrm{val}(\psi)$$

STRONG REFUTATION

We will say that an algorithm **strongly refutes*** random 2-XOR with m clauses if:

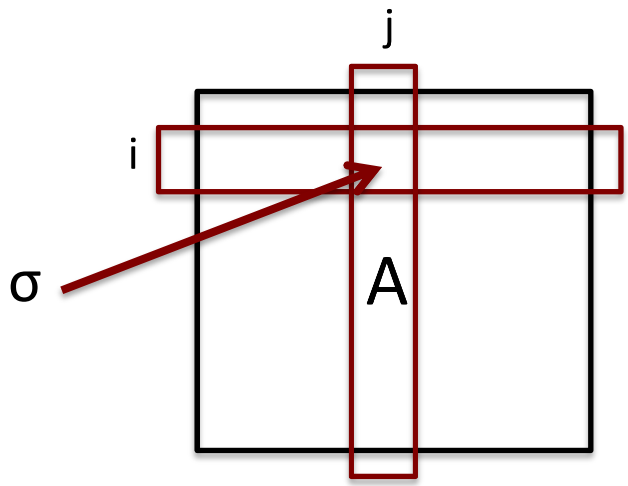
(1) On any 2-XOR formula ψ, it outputs **val** where:

$$\mathrm{OPT}(\psi) \leq \mathrm{val}(\psi)$$

(2) With high probability (for random ψ with m clauses):

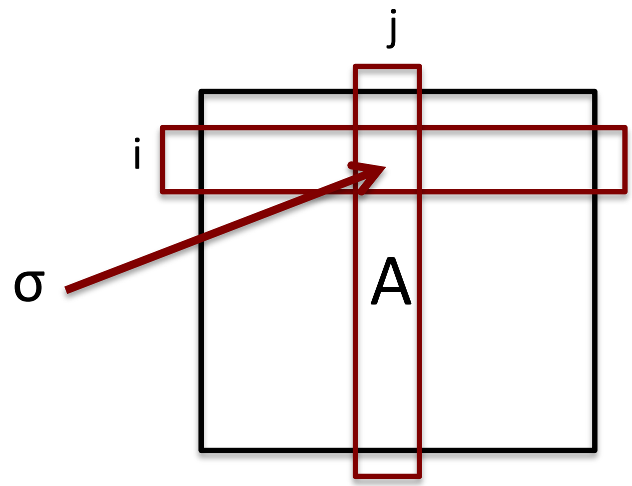
$$\mathrm{val}(\psi) = \frac{1}{2} + o(1)$$

Lemma: If $(i_1, j_1; \boldsymbol{\sigma_1}), \ldots, (i_m, j_m; \boldsymbol{\sigma_m}) \leftrightarrow \psi$ then



$$\frac{2\,\mathrm{OPT}(\psi) - 1}{n} \leq \frac{1}{m} \|A\|_2$$

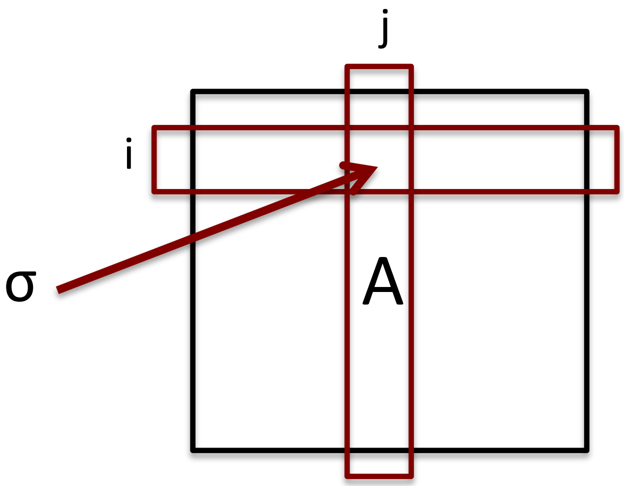
Lemma: If $(i_1, j_1; \boldsymbol{\sigma_1}), \ldots, (i_m, j_m; \boldsymbol{\sigma_m}) \leftrightarrow \psi$ then



$$\frac{2\,\mathrm{OPT}(\psi) - 1}{n} \le \frac{1}{m} \|A\|_2$$

Proof: Map the assignment to a unit vector so that $x_i = \pm 1/\sqrt{n}$ and take the quadratic form on A ■

Lemma: If $(i_1, j_1; \boldsymbol{\sigma_1}), \ldots, (i_m, j_m; \boldsymbol{\sigma_m}) \leftrightarrow \psi$ then

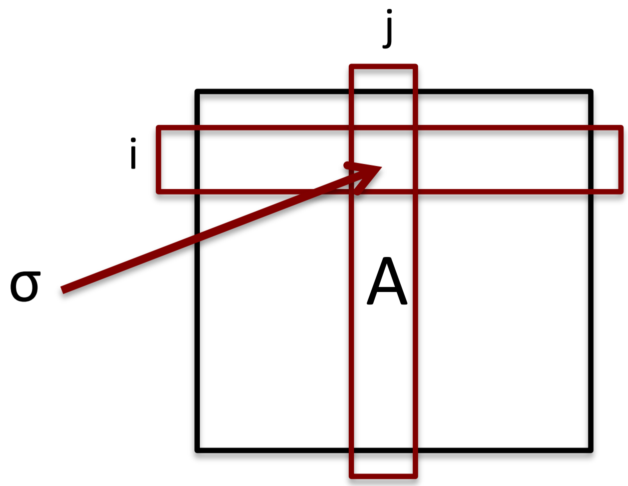


$$\frac{2\,\mathrm{OPT}(\psi) - 1}{n} \leq \frac{1}{m} \|A\|_2$$

Proof: Map the assignment to a unit vector so that $x_i = \pm 1/\sqrt{n}$ and take the quadratic form on A ∎

$$\|A\| \sim \sqrt{\frac{m}{n}}$$

Lemma: If $(i_1, j_1; \boldsymbol{\sigma_1}), \ldots, (i_m, j_m; \boldsymbol{\sigma_m}) \leftrightarrow \psi$ then

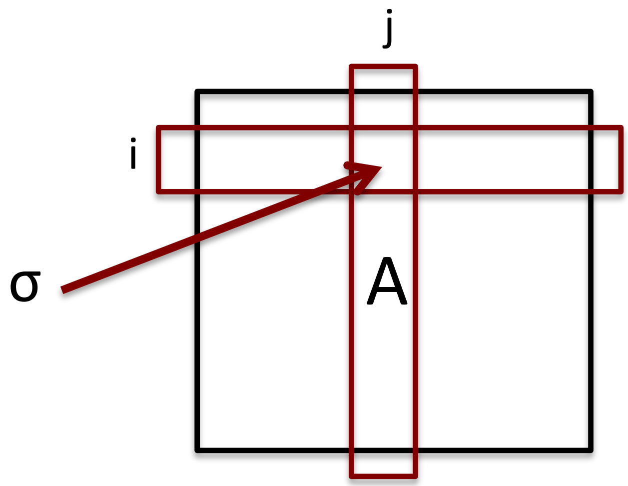


$$\frac{2\,\mathrm{OPT}(\psi) - 1}{n} \leq \frac{1}{m} \|A\|_2$$

Proof: Map the assignment to a unit vector so that $x_i = \pm 1/\sqrt{n}$ and take the quadratic form on A ■

$$\|A\| \sim \sqrt{\frac{m}{n}} \xrightarrow{m = \omega(n)} \mathrm{OPT}(\psi) \leq \frac{1}{2} + o(1)$$

Lemma: If $(i_1, j_1; \sigma_1), \ldots, (i_m, j_m; \sigma_m) \leftrightarrow \psi$ then



$$\frac{2\,\mathrm{OPT}(\psi) - 1}{n} \leq \frac{1}{m} \|A\|_2$$

Proof: Map the assignment to a unit vector so that $x_i = \pm 1/\sqrt{n}$ and take the quadratic form on A ■

$$\|A\| \sim \sqrt{\frac{m}{n}} \xrightarrow{m = \omega(n)} \mathrm{OPT}(\psi) \leq \frac{1}{2} + o(1)$$

This solves the strong refutation problem…

OUTLINE

Part I: Introduction

- The Netflix Problem and Matrix Completion
- Tensor Completion

Part II: To Random CSPs and Back

- A Distinguishing Problem
- Random CSPs and Strong Refutation
- Rademacher Complexity

Part III: Resolution

OUTLINE

Part I: Introduction

- The Netflix Problem and Matrix Completion
- Tensor Completion

Part II: To Random CSPs and Back

- A Distinguishing Problem
- Random CSPs and Strong Refutation
- **Rademacher Complexity**

Part III: Resolution

There are two very different communities that (essentially) attacked this same distinguishing problem:

The community working on **matrix completion**

The community working on **refuting random CSPs**

There are two very different communities that (essentially) attacked this same distinguishing problem:

The community working on **matrix completion**

The community working on **refuting random CSPs**

The **same** spectral bound implies:

(1) An algorithm for strongly refuting random 2-XOR

There are two very different communities that (essentially) attacked this same distinguishing problem:

The community working on **matrix completion**

The community working on **refuting random CSPs**

The **same** spectral bound implies:

(1) An algorithm for strongly refuting random 2-XOR

(2) An algorithm for the distinguishing problem

There are two very different communities that (essentially) attacked this same distinguishing problem:

The community working on **matrix completion**

The community working on **refuting random CSPs**

The **same** spectral bound implies:

(1) An algorithm for strongly refuting random 2-XOR

(2) An algorithm for the distinguishing problem

(3) Generalization bounds for the nuclear norm

It also yields bounds on how well the solution to the convex program generalizes **[Srebro, Shraibman]** …

$$\min \|X\|_* \text{ s.t. } \frac{1}{|\Omega|} \sum_{(i,j)\in\Omega} |X_{i,j} - M_{i,j}| \leq \eta$$

It also yields bounds on how well the solution to the convex program generalizes **[Srebro, Shraibman]** …

$$\min \|X\|_* \text{ s.t. } \frac{1}{|\Omega|} \sum_{(i,j)\in\Omega} |X_{i,j} - M_{i,j}| \leq \eta$$

An approach through **statistical learning theory**:

It also yields bounds on how well the solution to the convex program generalizes **[Srebro, Shraibman]** …

$$\min \|X\|_* \text{ s.t. } \frac{1}{|\Omega|} \sum_{(i,j)\in\Omega} |X_{i,j} - M_{i,j}| \leq \eta$$

An approach through **statistical learning theory**:

empirical error: $$\frac{1}{|\Omega|} \sum_{(i,j)\in\Omega} |X_{i,j} - M_{i,j}|$$

It also yields bounds on how well the solution to the convex program generalizes **[Srebro, Shraibman]** …

$$\min \|X\|_* \text{ s.t. } \frac{1}{|\Omega|} \sum_{(i,j)\in\Omega} |X_{i,j} - M_{i,j}| \leq \eta$$

An approach through **statistical learning theory**:

empirical error: $$\frac{1}{|\Omega|} \sum_{(i,j)\in\Omega} |X_{i,j} - M_{i,j}| \qquad (\leq \eta)$$

It also yields bounds on how well the solution to the convex program generalizes **[Srebro, Shraibman]** …

$$\min \|X\|_* \text{ s.t. } \frac{1}{|\Omega|} \sum_{(i,j)\in\Omega} |X_{i,j} - M_{i,j}| \leq \eta$$

An approach through **statistical learning theory**:

empirical error: $$\frac{1}{|\Omega|} \sum_{(i,j)\in\Omega} |X_{i,j} - M_{i,j}|$$ **(≤ η)**

prediction error: $$\frac{1}{n^2} \sum |X_{i,j} - M_{i,j}|$$

It also yields bounds on how well the solution to the convex program generalizes **[Srebro, Shraibman]** …

Then if we let

$$\mathcal{K} = \{ X \text{ s.t. } \|X\|_* \leq 1 \} = \text{conv}\{ ab^T \text{ s.t. } \|a\|, \|b\| \leq 1 \}$$

It also yields bounds on how well the solution to the convex program generalizes [Srebro, Shraibman] ...

Then if we let

$$\mathcal{K} = \{ X \text{ s.t. } ||X||_* \leq 1 \} = \text{conv}\{ ab^T \text{ s.t. } ||a||, ||b|| \leq 1 \}$$

generalization error:

$$\sup_{X \in \mathcal{K}} \left| \underset{\text{(on } \Omega)}{\textbf{empirical error}} (X) - \textbf{prediction error} (X) \right|$$

It also yields bounds on how well the solution to the convex program generalizes [Srebro, Shraibman] …

Then if we let

$$\mathcal{K} = \{ X \text{ s.t. } ||X||_* \leq 1 \} = \text{conv}\{ ab^T \text{ s.t. } ||a||, ||b|| \leq 1 \}$$

generalization error:

$$\sup_{X \in \mathcal{K}} \left| \underset{(\text{on } \Omega)}{\textbf{empirical error}} (X) - \textbf{prediction error} (X) \right|$$

Theorem:

"generalization error $\leq$ best agreement with random function"
(on Ω)

It also yields bounds on how well the solution to the convex program generalizes [Srebro, Shraibman] ...

Then if we let

$$\mathcal{K} = \{ X \text{ s.t. } \|X\|_* \leq 1 \} = \text{conv}\{ ab^T \text{ s.t. } \|a\|, \|b\| \leq 1 \}$$

generalization error:

$$\sup_{X \in \mathcal{K}} \left| \underset{(\text{on } \Omega)}{\textbf{empirical error}} (X) - \textbf{prediction error} (X) \right|$$

Theorem:

$$\text{“generalization error} \leq \underbrace{\underset{(\text{on } \Omega)}{\text{best agreement with random function}}}_{\textbf{Rademacher complexity}}\text{”}$$

It also yields bounds on how well the solution to the convex program generalizes **[Srebro, Shraibman]** …

More precisely:

$$\underbrace{\sup_{X \in \mathcal{K}} \frac{1}{m} \left| \sum_{a=1}^{m} \sigma_a X_{i_a, j_a} \right|}_{}$$

Rademacher complexity ($R^m(\mathcal{K})$)

It also yields bounds on how well the solution to the convex program generalizes **[Srebro, Shraibman]** ...

More precisely:

$$\underbrace{\sup_{X \in \mathcal{K}} \frac{1}{m} \left| \sum_{a=1} \sigma_a X_{i_a, j_a} \right|}_{\textbf{Rademacher complexity } (\mathbf{R}^m(\mathcal{K}))} = \frac{1}{m} || A ||$$

It also yields bounds on how well the solution to the convex program generalizes [Srebro, Shraibman] ...

More precisely:

$$\underbrace{\sup_{X \in \mathcal{K}} \frac{1}{m} \left| \sum_{a=1} \sigma_a X_{i_a, j_a} \right|}_{\textbf{Rademacher complexity } (R^m(\mathcal{K}))} = \frac{1}{m} \| A \|$$

$$\frac{1}{m} \| A \| \sim \sqrt{\frac{1}{mn}}$$

It also yields bounds on how well the solution to the convex program generalizes **[Srebro, Shraibman]** …

More precisely:

$$\underbrace{\sup_{X \in \mathcal{K}} \frac{1}{m} \left| \sum_{a=1} \sigma_a X_{i_a, j_a} \right|}_{\textbf{Rademacher complexity } (\mathbf{R^m(\mathcal{K})})} = \frac{1}{m} \| A \|$$

$$\frac{1}{m} \| A \| \sim \sqrt{\frac{1}{mn}} \xrightarrow{m = \omega(n)} R^m(\mathcal{K}) = o\left(\frac{1}{n}\right)$$

There are two very different communities that (essentially) attacked this same distinguishing problem:

The community working on **matrix completion**

The community working on **refuting random CSPs**

Noisy matrix completion with m observations $\xrightarrow[\text{Rademacher Complexity}]{}$ **Strongly refute*** random 2-XOR/2-SAT with m clauses

*Want an algorithm that certifies a formula is far from satisfiable

There are two very different communities that (essentially) attacked this same distinguishing problem:

The community working on **matrix completion**

The community working on **refuting random CSPs**

Noisy **tensor** completion with m observations $\xrightarrow{\text{Rademacher Complexity}}$ **Strongly refute*** random **3-XOR/3-SAT** with m clauses

*Want an algorithm that certifies a formula is far from satisfiable

There are two very different communities that (essentially) attacked this same distinguishing problem:

The community working on **matrix completion**

The community working on **refuting random CSPs**

Noisy **tensor** completion with m observations $\xrightarrow{\text{Rademacher Complexity}}$ **Strongly refute*** random **3-XOR/3-SAT** with m clauses

[Coja-Oghlan, Goerdt, Lanka]

*Want an algorithm that certifies a formula is far from satisfiable

OUTLINE

Part I: Introduction

- The Netflix Problem and Matrix Completion
- Tensor Completion

Part II: To Random CSPs and Back

- A Distinguishing Problem
- Random CSPs and Strong Refutation
- Rademacher Complexity

Part III: Resolution

OUTLINE

Part I: Introduction

- The Netflix Problem and Matrix Completion
- Tensor Completion

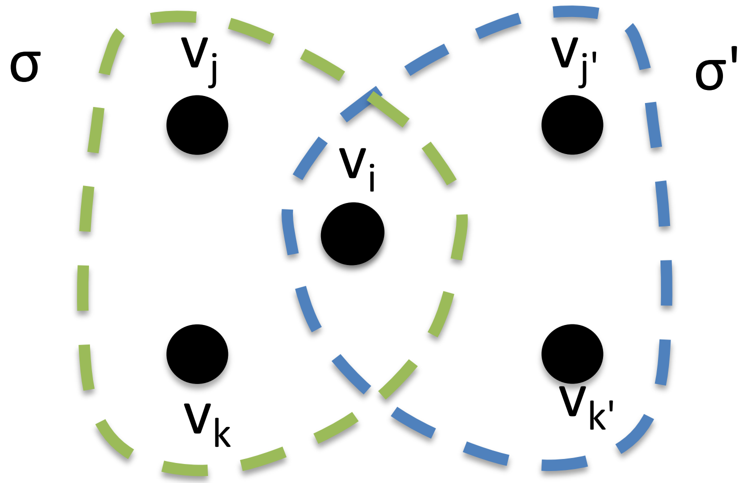
Part II: To Random CSPs and Back

- A Distinguishing Problem
- Random CSPs and Strong Refutation
- Rademacher Complexity

Part III: Resolution

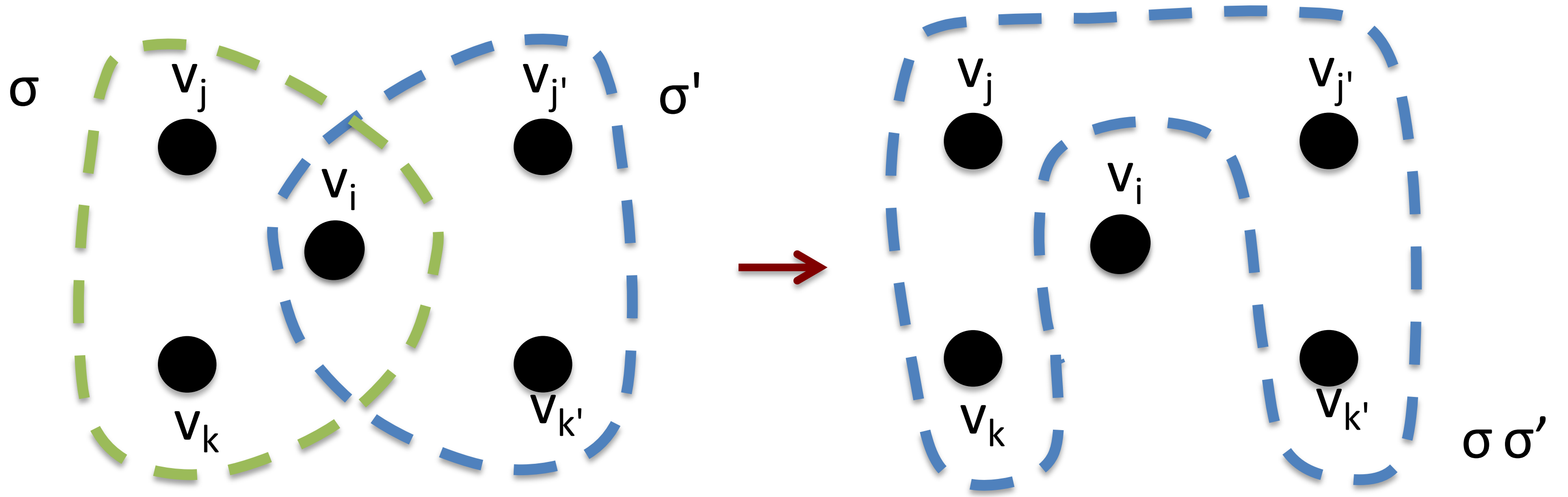
[Coja-Ohglan, Goerdt, Lanka]: Reduce 3-XOR to 4-XOR

$$\left(v_i \bullet v_j \bullet v_k = \sigma\right) \bullet \left(v_i \bullet v_{j'} \bullet v_{k'} = \sigma'\right)$$



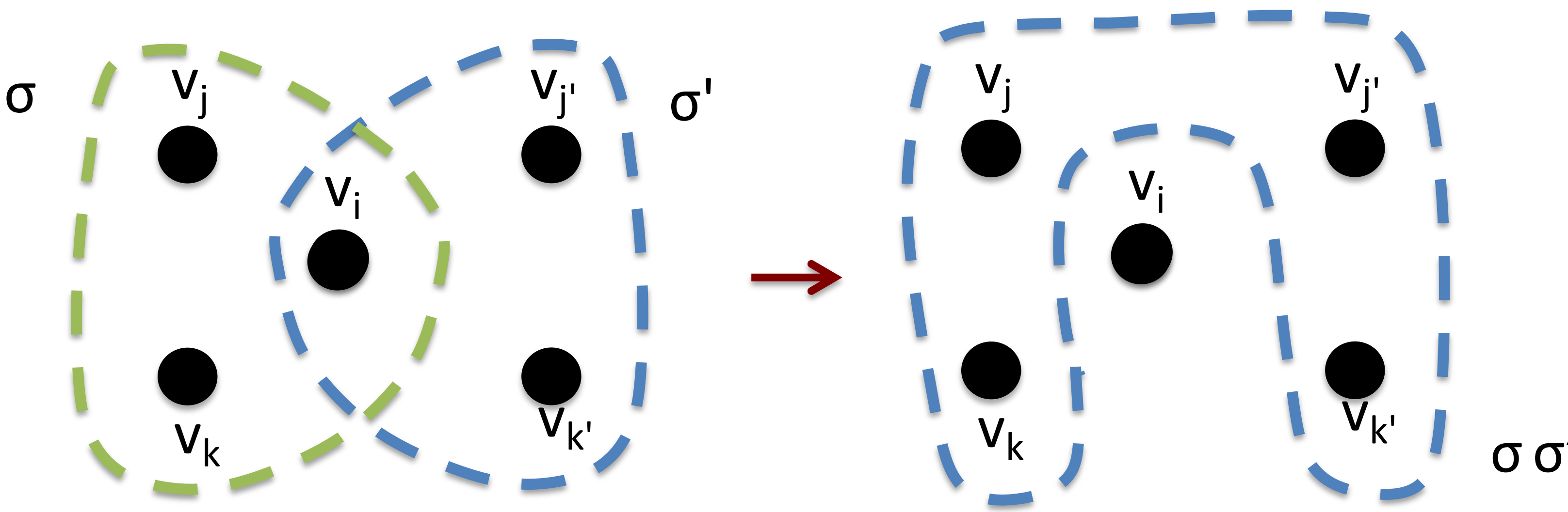
[Coja-Ohglan, Goerdt, Lanka]: Reduce 3-XOR to 4-XOR

$$\left(v_i \bullet v_j \bullet v_k = \sigma\right) \bullet \left(v_i \bullet v_{j'} \bullet v_{k'} = \sigma'\right) \longrightarrow \left(v_j \bullet v_k \bullet v_{j'} \bullet v_{k'} = \sigma\,\sigma'\right)$$



[Coja-Ohglan, Goerdt, Lanka]: Reduce 3-XOR to 4-XOR

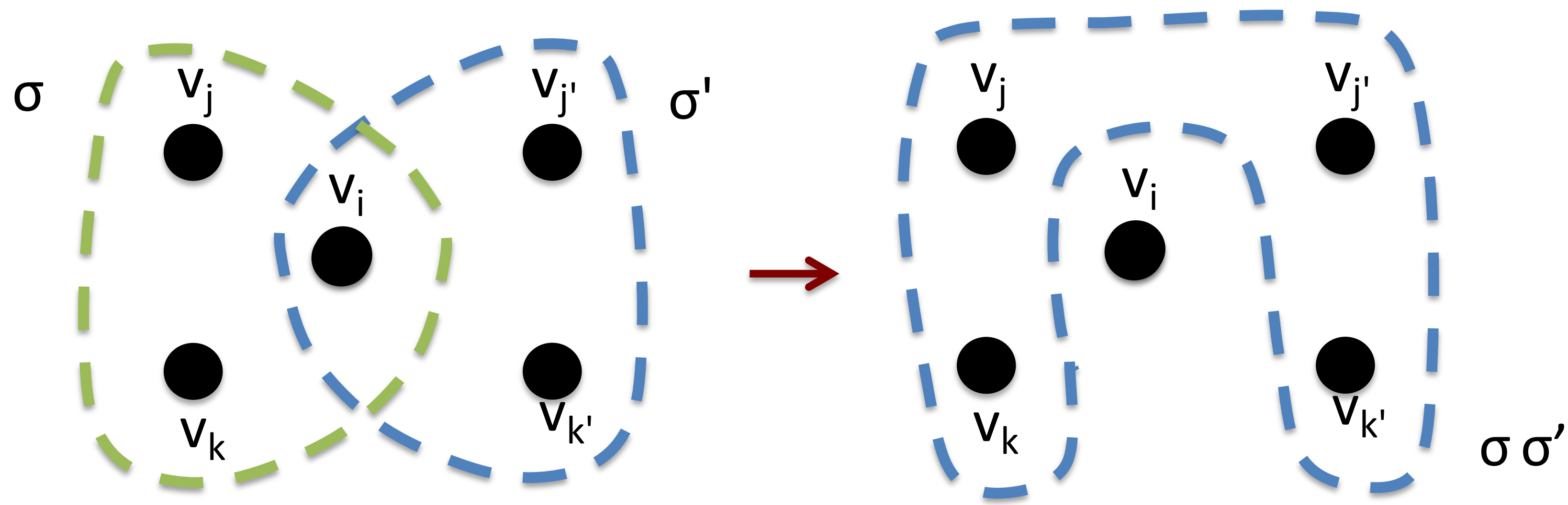
$$(v_i \cdot v_j \cdot v_k = \sigma) \cdot (v_i \cdot v_{j'} \cdot v_{k'} = \sigma') \longrightarrow (v_j \cdot v_k \cdot v_{j'} \cdot v_{k'} = \sigma\,\sigma')$$



This yields $n^5p^2 = n^2 \log^{O(1)} n$ clauses

[Coja-Ohglan, Goerdt, Lanka]: Reduce 3-XOR to 4-XOR

$$(v_i \cdot v_j \cdot v_k = \sigma) \cdot (v_i \cdot v_{j'} \cdot v_{k'} = \sigma') \longrightarrow (v_j \cdot v_k \cdot v_{j'} \cdot v_{k'} = \sigma\,\sigma')$$

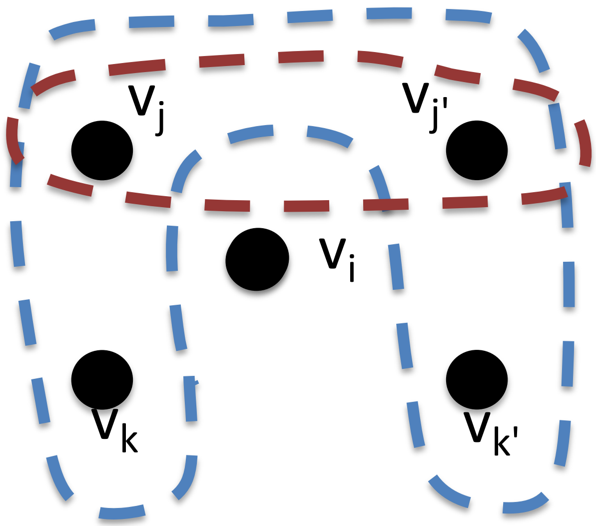


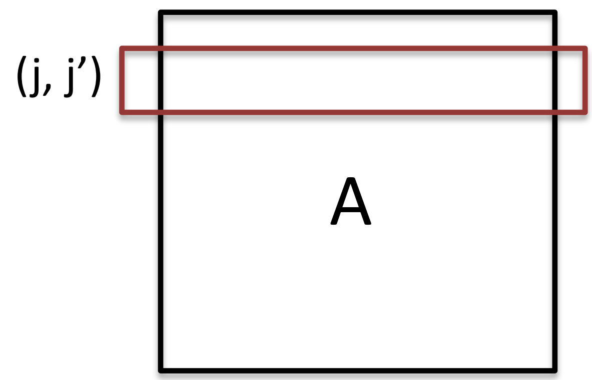
This yields $n^5p^2 = n^2 \log^{O(1)} n$ clauses

Warning: The 4-XOR clauses are not independent!

[Coja-Ohglan, Goerdt, Lanka]: Reduce 3-XOR to 4-XOR

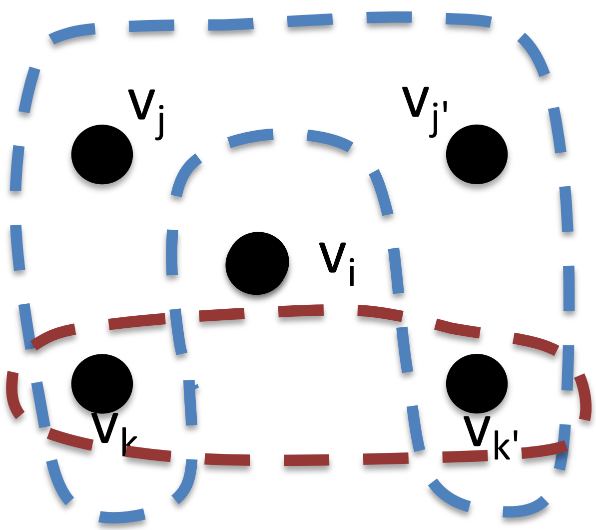
$$(v_i \cdot v_j \cdot v_k = \sigma) \cdot (v_i \cdot v_{j'} \cdot v_{k'} = \sigma') \longrightarrow (v_j \cdot v_k \cdot v_{j'} \cdot v_{k'} = \sigma \sigma')$$

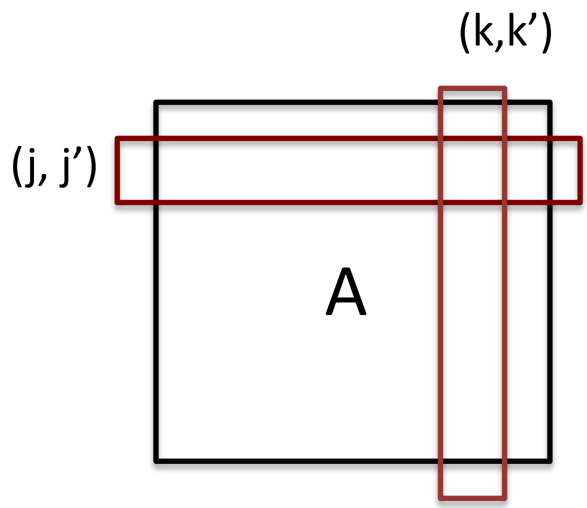




[Coja-Ohglan, Goerdt, Lanka]: Reduce 3-XOR to 4-XOR

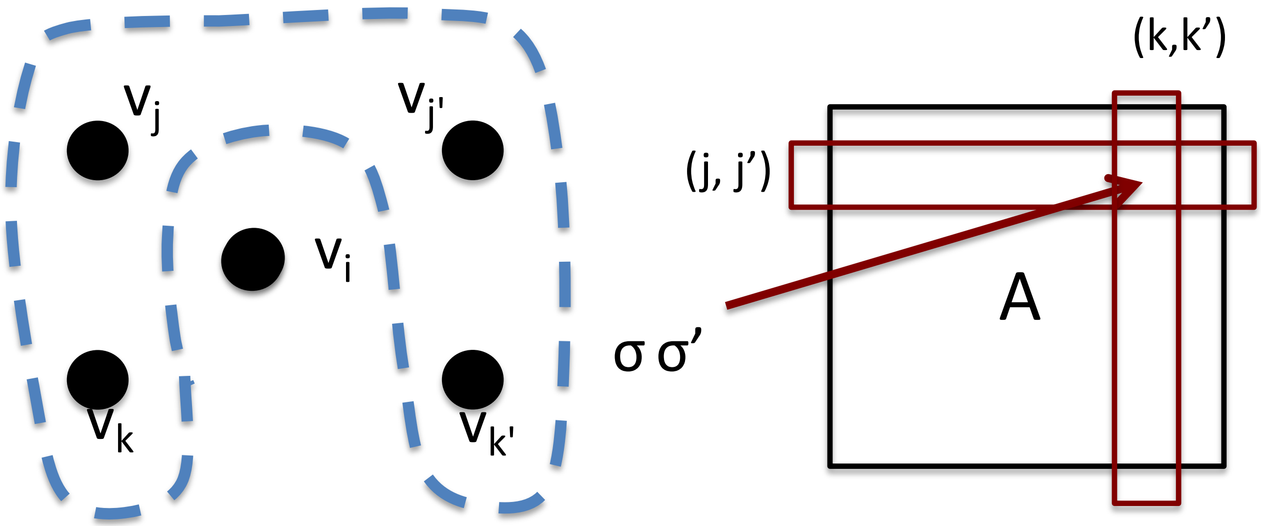
$$(v_i \bullet v_j \bullet v_k = \sigma) \bullet (v_i \bullet v_{j'} \bullet v_{k'} = \sigma') \longrightarrow (v_j \bullet v_k \bullet v_{j'} \bullet v_{k'} = \sigma\sigma')$$





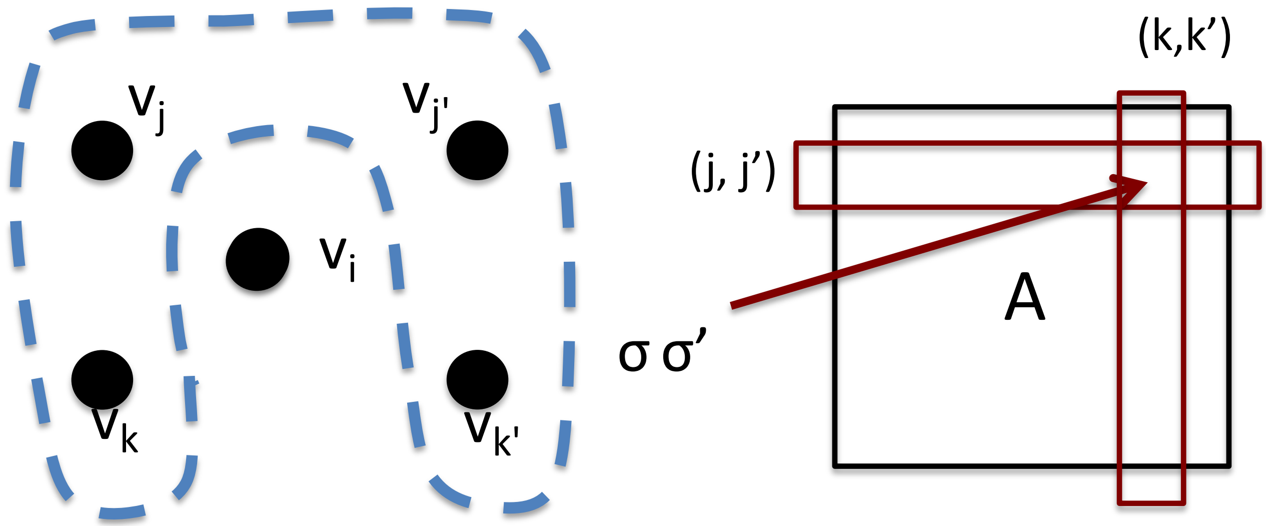
[Coja-Ohglan, Goerdt, Lanka]: Reduce 3-XOR to 4-XOR

$$(v_i \bullet v_j \bullet v_k = \sigma) \bullet (v_i \bullet v_{j'} \bullet v_{k'} = \sigma') \longrightarrow (v_j \bullet v_k \bullet v_{j'} \bullet v_{k'} = \sigma \sigma')$$



[Coja-Ohglan, Goerdt, Lanka]: Reduce 3-XOR to 4-XOR

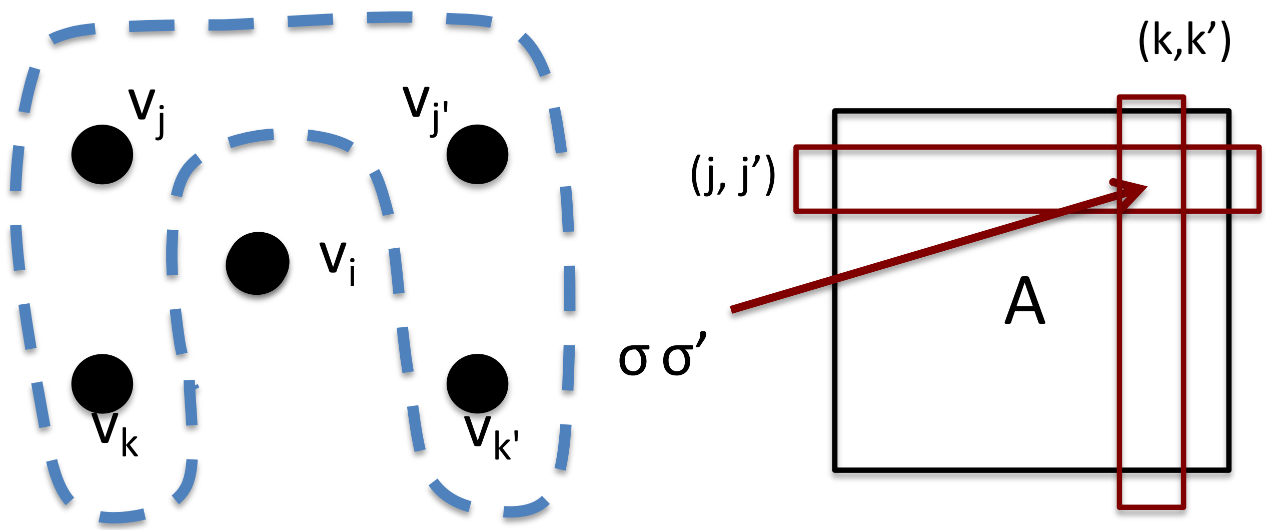
$$(v_i \bullet v_j \bullet v_k = \sigma) \bullet (v_i \bullet v_{j'} \bullet v_{k'} = \sigma') \longrightarrow (v_j \bullet v_k \bullet v_{j'} \bullet v_{k'} = \sigma\sigma')$$



Hence the paired variables for the rows (and colns) come from different clauses!

[Coja-Ohglan, Goerdt, Lanka]: Reduce 3-XOR to 4-XOR

$$(v_i \bullet v_j \bullet v_k = \sigma) \bullet (v_i \bullet v_{j'} \bullet v_{k'} = \sigma') \longrightarrow (v_j \bullet v_k \bullet v_{j'} \bullet v_{k'} = \sigma\sigma')$$

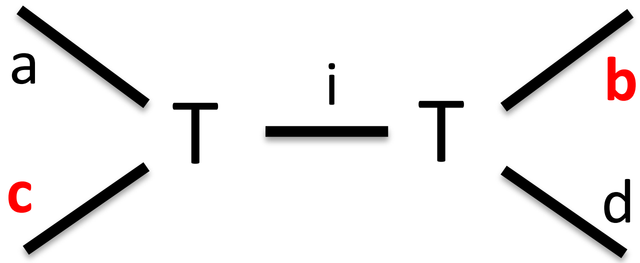


+ trace method

Hence the paired variables for the rows (and colns) come from different clauses!

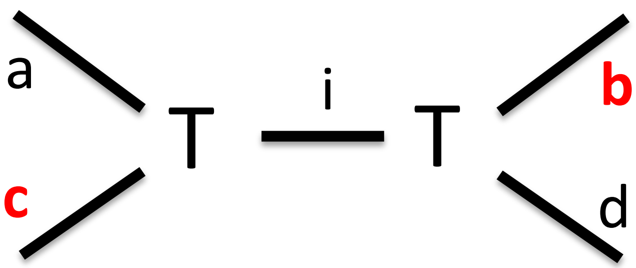
This reduction works because of **tensor networks**

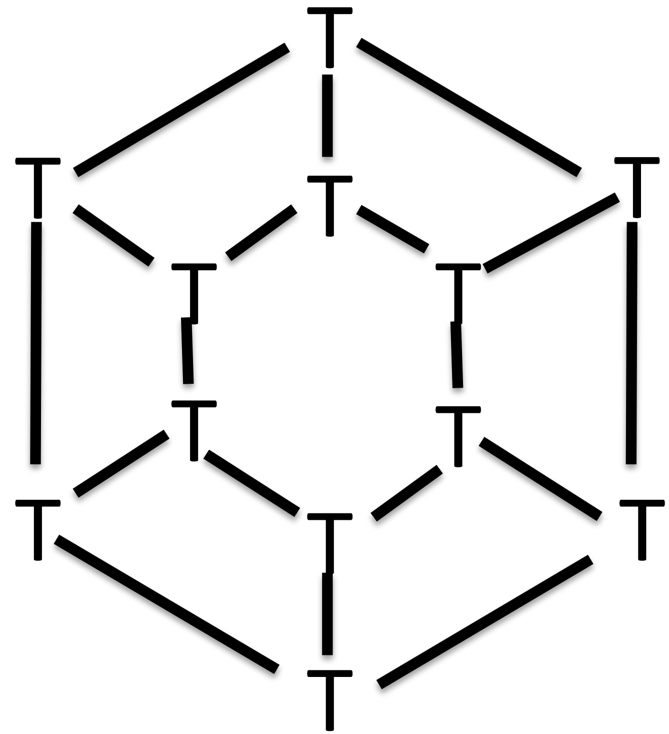
This reduction works because of **tensor networks**



Applying the trace method
to the (ab,cd)-flattening

This reduction works because of **tensor networks**





Applying the trace method to the (ab,cd)-flattening

Counts certainly labelings over this graph

There are two very different communities that (essentially) attacked this same distinguishing problem:

The community working on **matrix completion**

The community working on **refuting random CSPs**

Noisy **tensor** completion with m observations → **Strongly refute*** random **3-XOR/3-SAT** with m clauses

Rademacher Complexity

[Coja-Oghlan, Goerdt, Lanka]

*Want an algorithm that certifies a formula is far from satisfiable

There are two very different communities that (essentially) attacked this same distinguishing problem:

The community working on **matrix completion**

The community working on **refuting random CSPs**

Noisy **tensor** completion with m observations ← **Embedding in SOS** — **Strongly refute*** random **3-XOR/3-SAT** with m clauses

[Coja-Oghlan, Goerdt, Lanka]

We then embed this algorithm into the **sixth** level of the sum-of-squares hierarchy, to get a relaxation for tensor prediction

*Want an algorithm that certifies a formula is far from satisfiable

MORE ALGORITHMS

Is there an algorithm for exact completion?

MORE ALGORITHMS

Is there an algorithm for exact completion?

[Potechin, Steurer]: Yes, assuming the factors are **orthogonal**

MORE ALGORITHMS

Is there an algorithm for exact completion?

[Potechin, Steurer]: Yes, assuming the factors are **orthogonal**

[Kivva, Potechin]: Yes, for random overcomplete tensors

MORE ALGORITHMS

Is there a practical algorithm?

MORE ALGORITHMS

Is there a practical algorithm?

Many fast algorithms that avoid SOS altogether

MORE ALGORITHMS

Is there a practical algorithm?

Many fast algorithms that avoid SOS altogether

[Montanari, Sun]: Yes, using spectral methods that work up to $r \leq n^{3/4}$ but only approximate completion

MORE ALGORITHMS

Is there a practical algorithm?

Many fast algorithms that avoid SOS altogether

[Montanari, Sun]: Yes, using spectral methods that work up to $r \leq n^{3/4}$ but only approximate completion

[Cai et al.]: Yes, assuming the factors are nearly **orthogonal**

MORE ALGORITHMS

Is there a practical algorithm?

Many fast algorithms that avoid SOS altogether

[Montanari, Sun]: Yes, using spectral methods that work up to $r \leq n^{3/4}$ but only approximate completion

[Cai et al.]: Yes, assuming the factors are **nearly orthogonal**

[Liu, Moitra]: Yes, assuming linear independence, but needs $n^{3/2}\text{poly}(r)$ observations

MORE ALGORITHMS

Is there a practical algorithm?

Many fast algorithms that avoid SOS altogether

[Montanari, Sun]: Yes, using spectral methods that work up to $r \leq n^{3/4}$ but only approximate completion

[Cai et al.]: Yes, assuming the factors are nearly **orthogonal**

[Liu, Moitra]: Yes, assuming linear independence, but needs $n^{3/2}$poly(r) observations

Scales to thousand-dimensional tensors!

Summary:

- Connections between tensors and random CSPs
- New algorithms for completing third-order tensors that beat flattening
- **Is practical tensor completion within reach?**

Thanks! Any Questions?