BEYOND MATRIX COMPLETION

ANKUR MOITRA

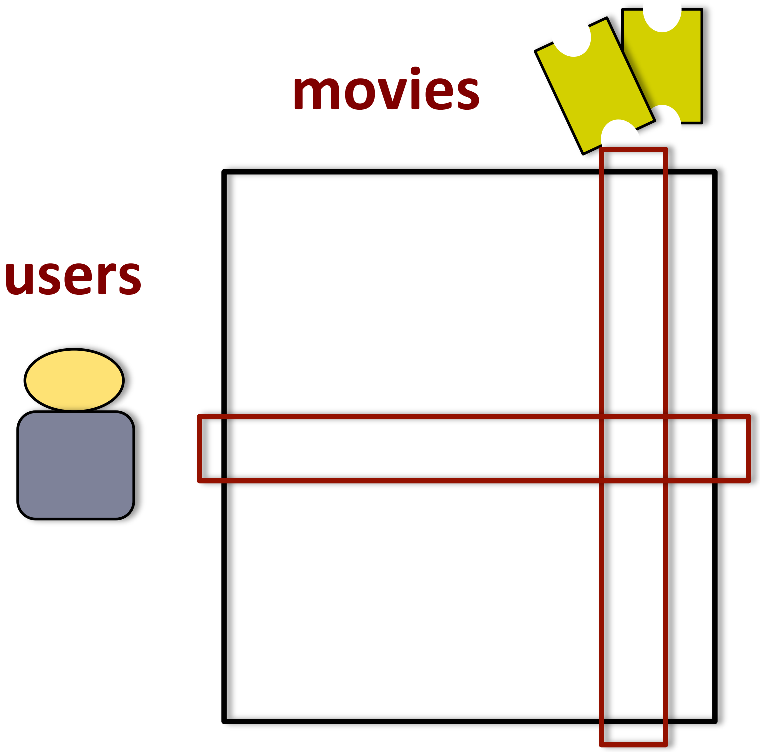
MASSACHUSETTS INSTITUTE OF TECHNOLOGY

Based on joint work with Boaz Barak (Harvard)

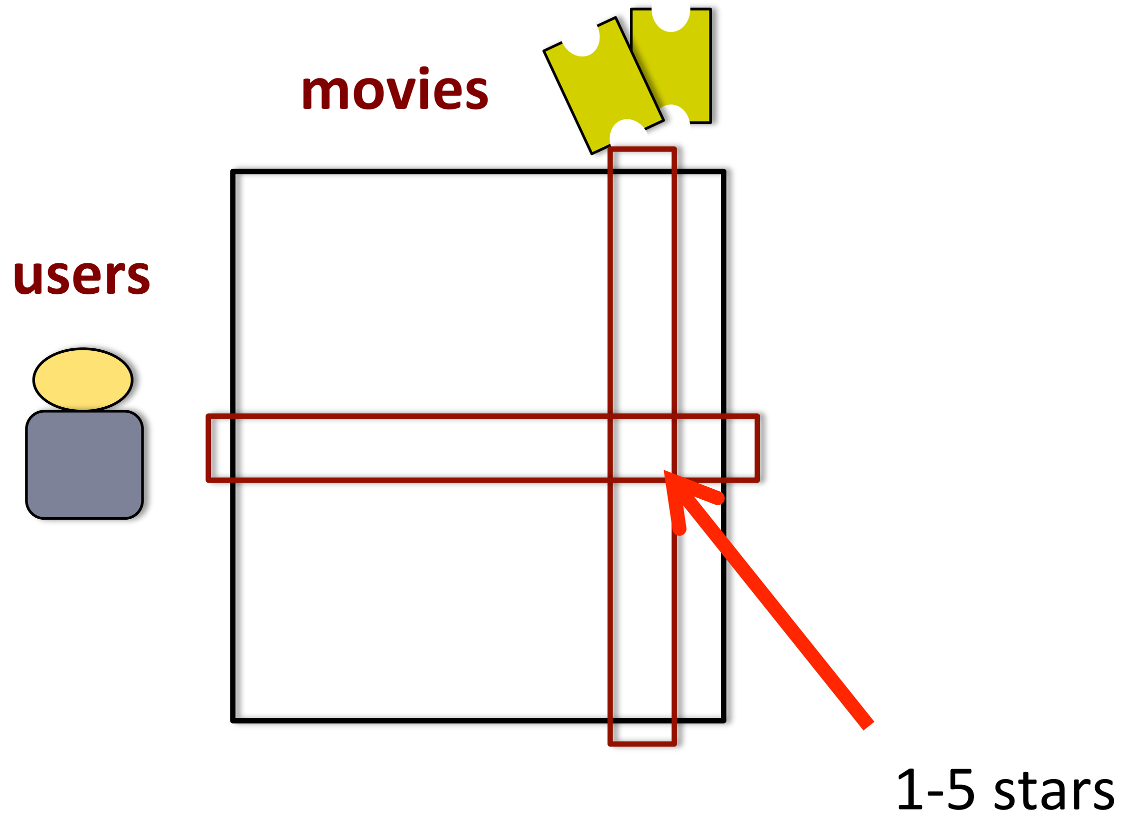
Part I:

Recommendation systems and partially observed matrices

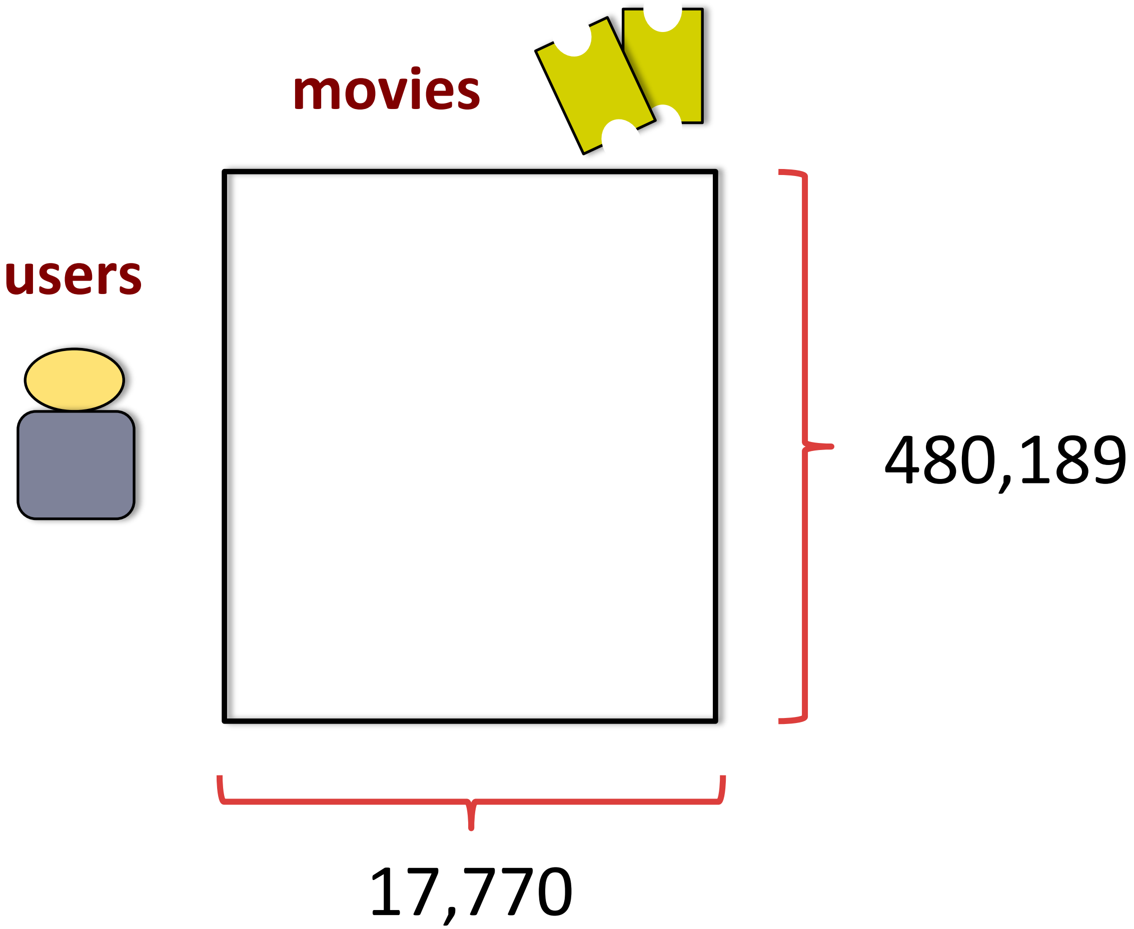
THE NETFLIX PROBLEM



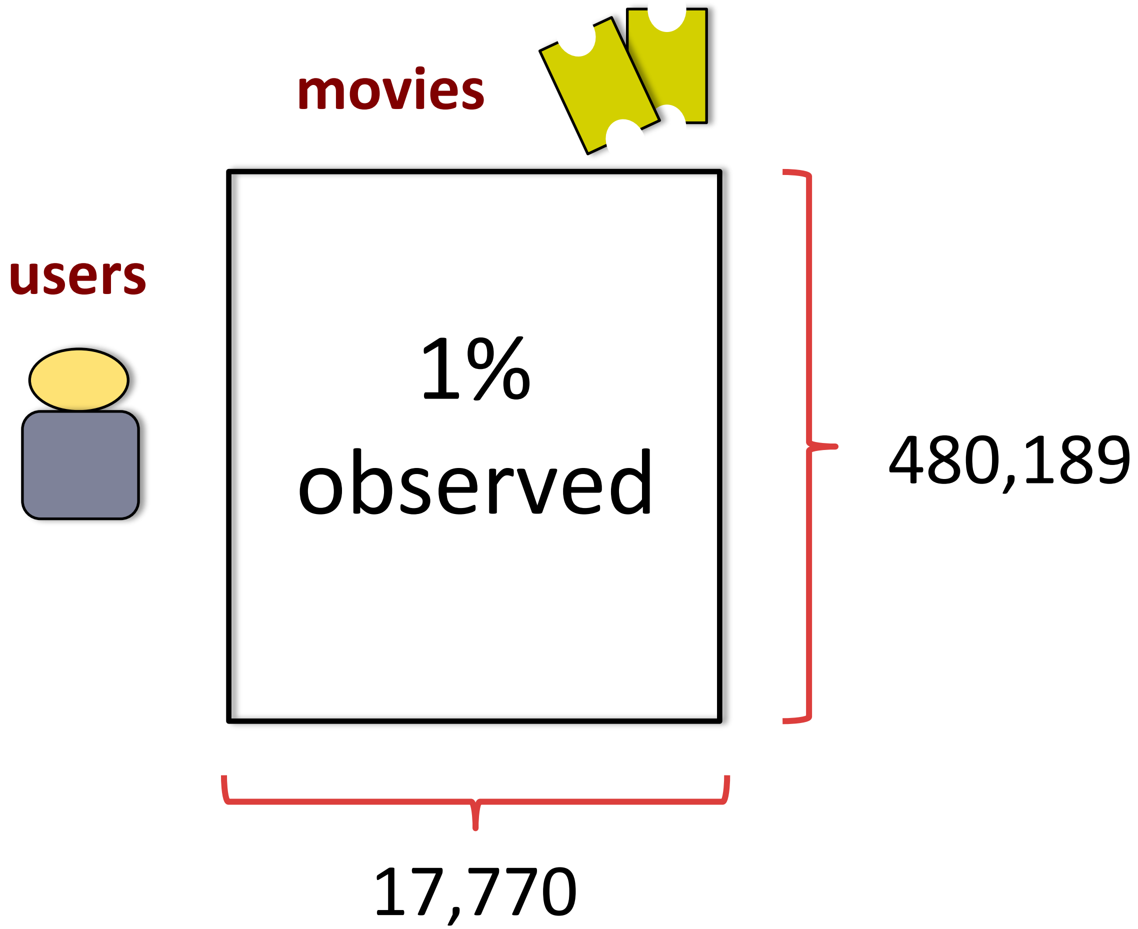
THE NETFLIX PROBLEM



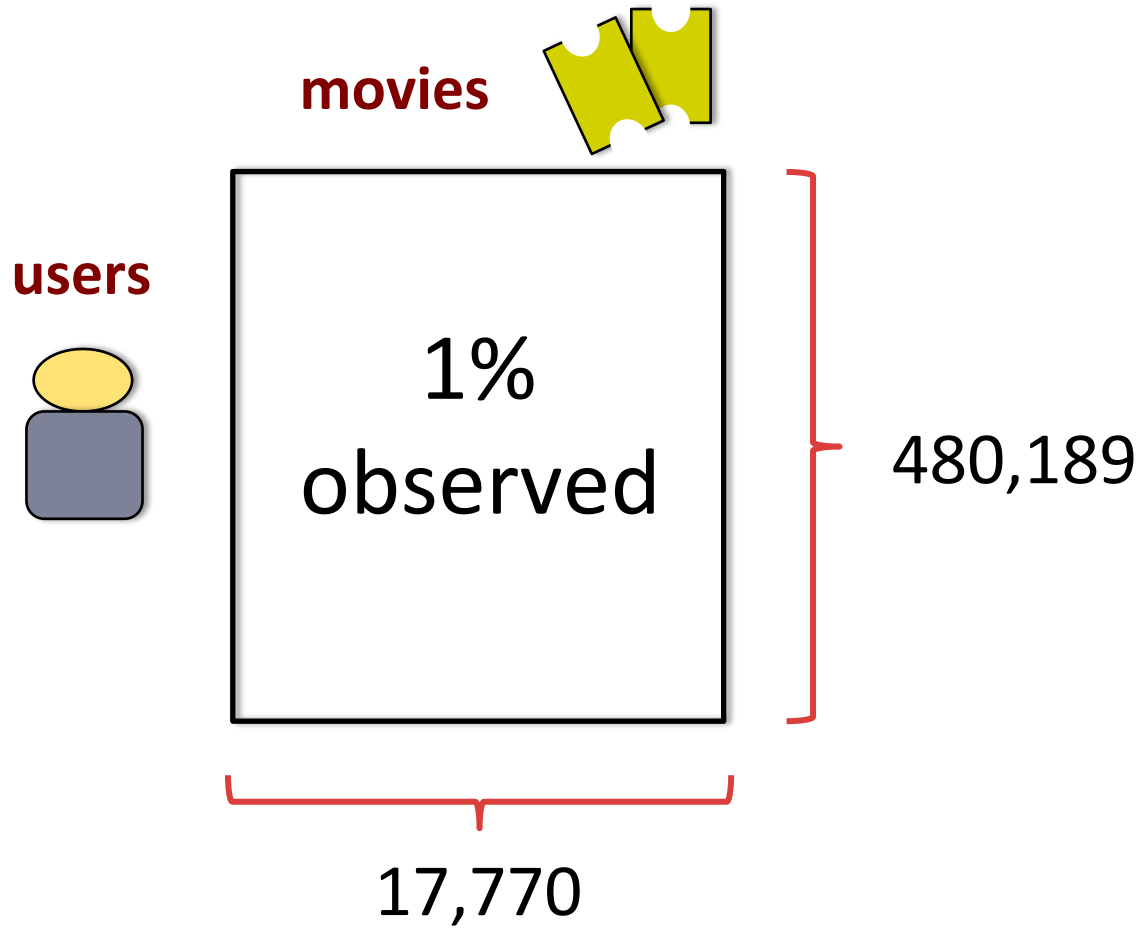
THE NETFLIX PROBLEM



THE NETFLIX PROBLEM



THE NETFLIX PROBLEM



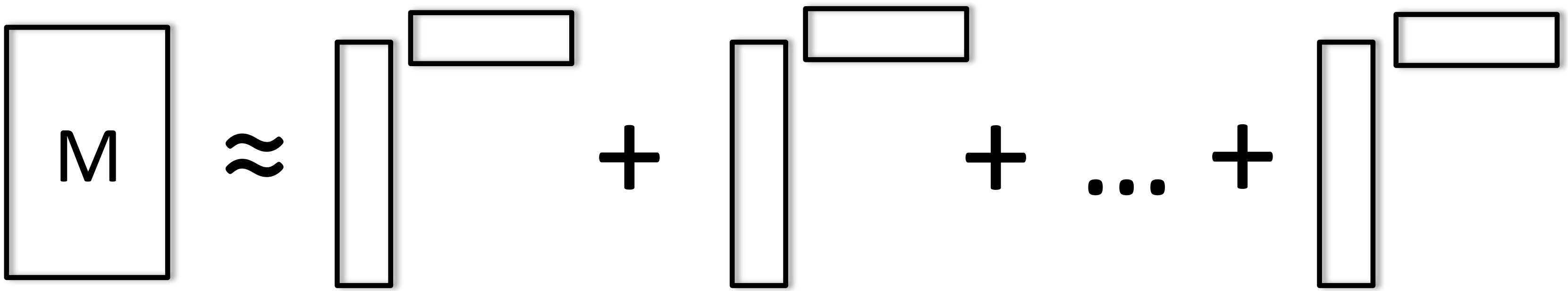
Can we (approximately) fill-in the missing entries?

MATRIX COMPLETION

Let M be an unknown, approximately low-rank matrix

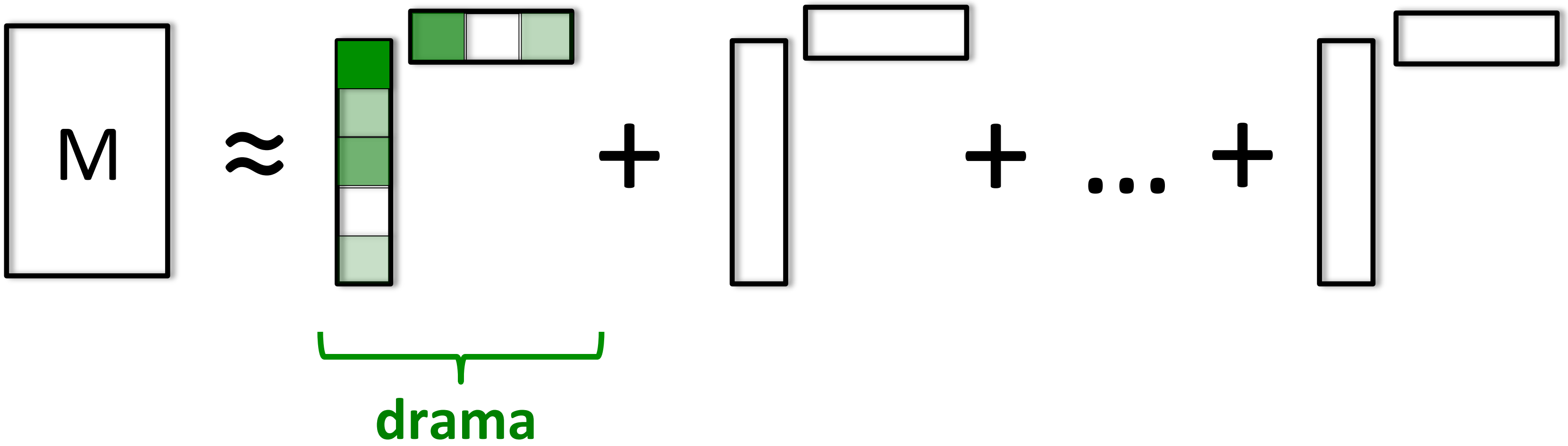
MATRIX COMPLETION

Let M be an unknown, approximately low-rank matrix



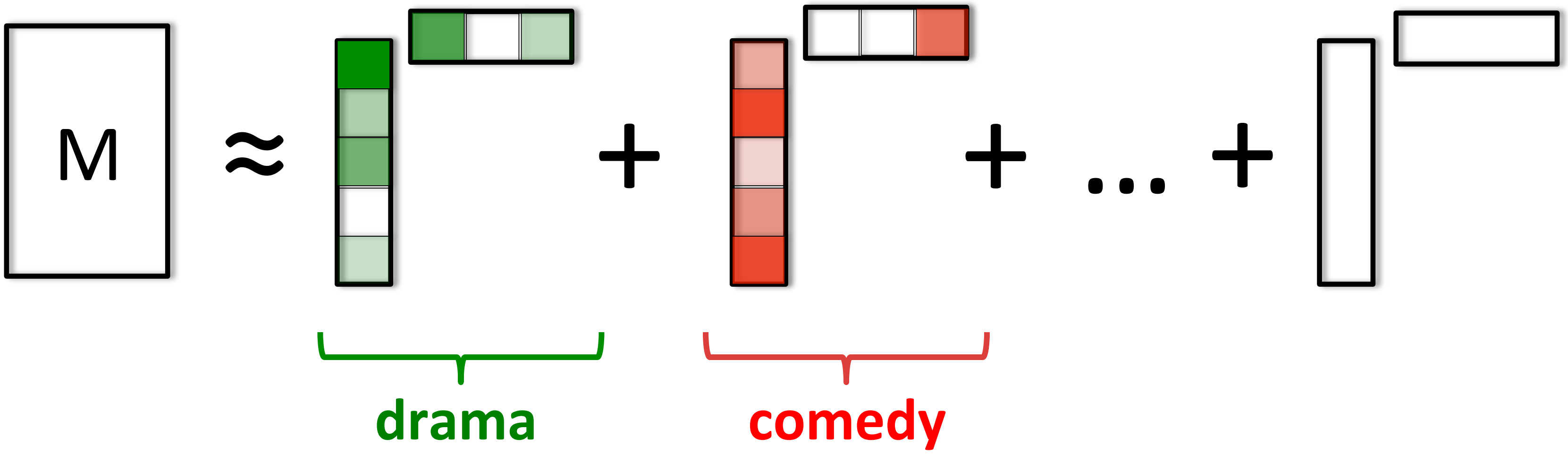
MATRIX COMPLETION

Let M be an unknown, approximately low-rank matrix



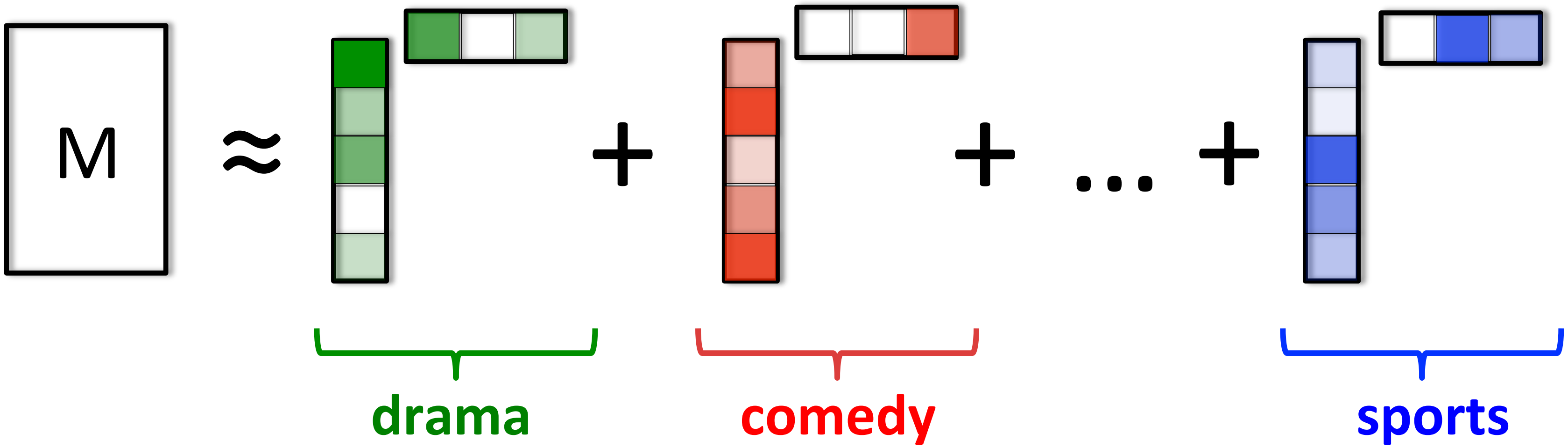
MATRIX COMPLETION

Let M be an unknown, approximately low-rank matrix



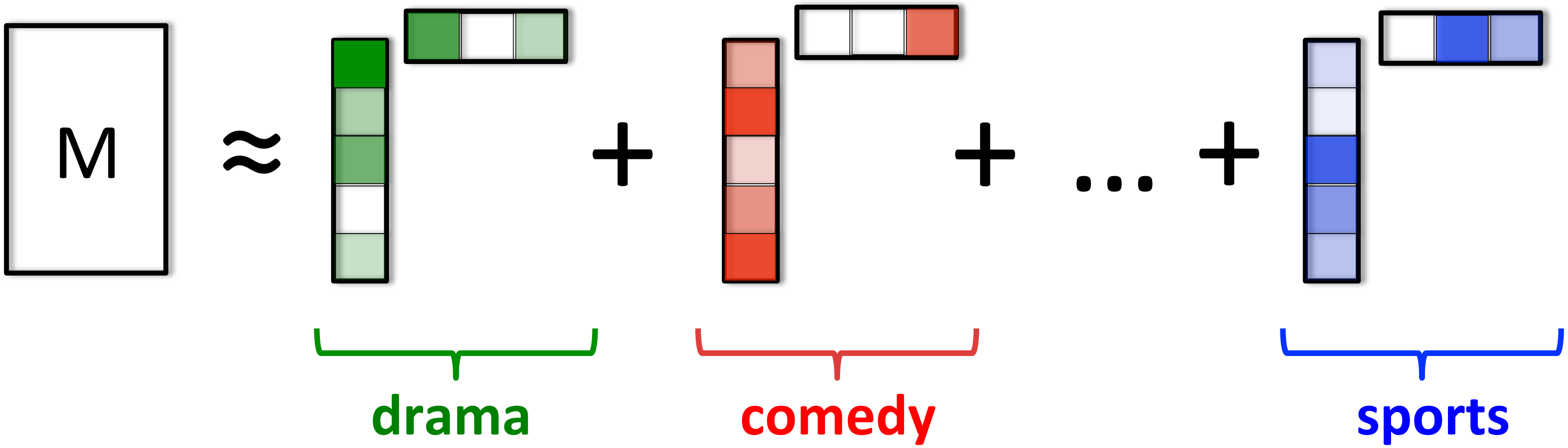
MATRIX COMPLETION

Let M be an unknown, approximately low-rank matrix



MATRIX COMPLETION

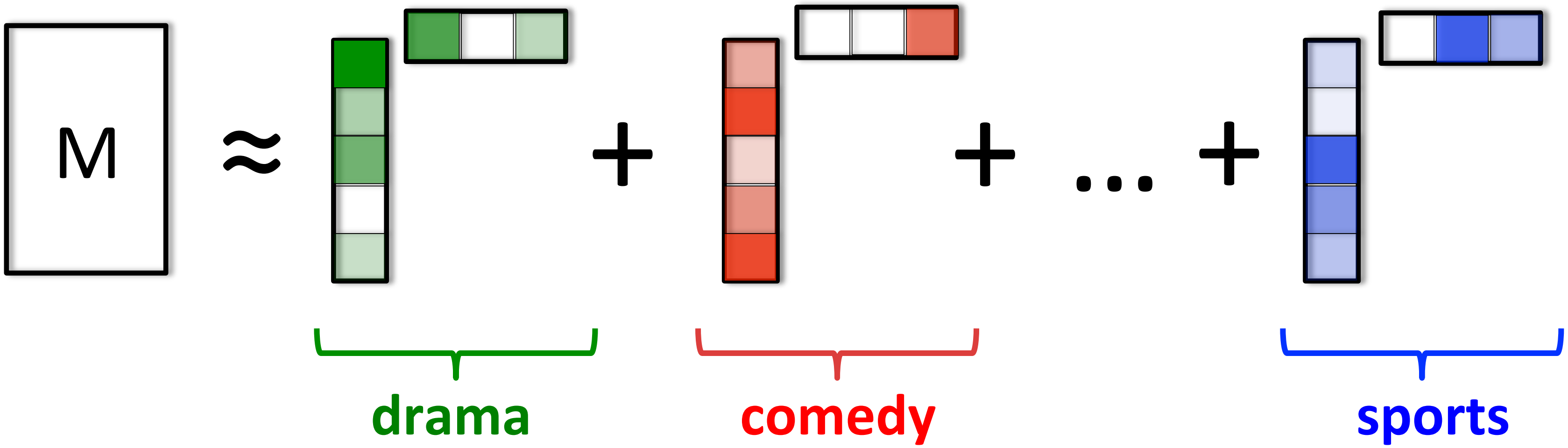
Let M be an unknown, approximately low-rank matrix



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**

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

MATRIX COMPLETION

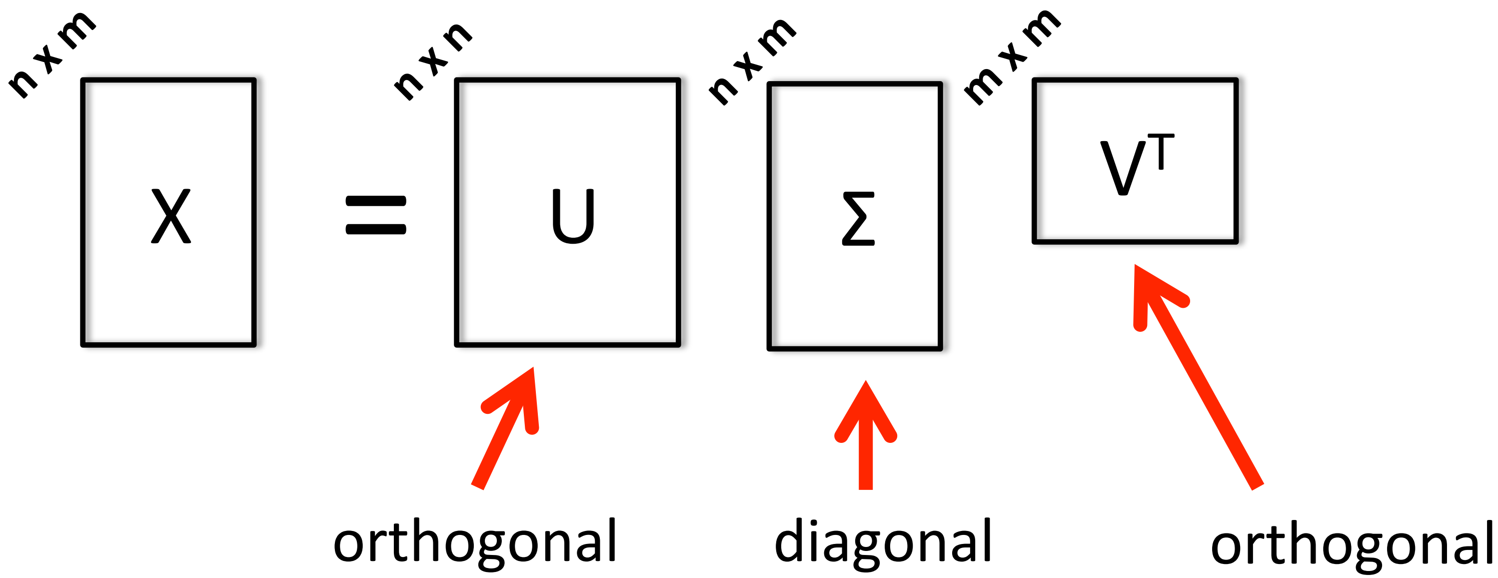
The natural formulation is **non-convex**, and **NP-hard**

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

There is a powerful, convex relaxation…

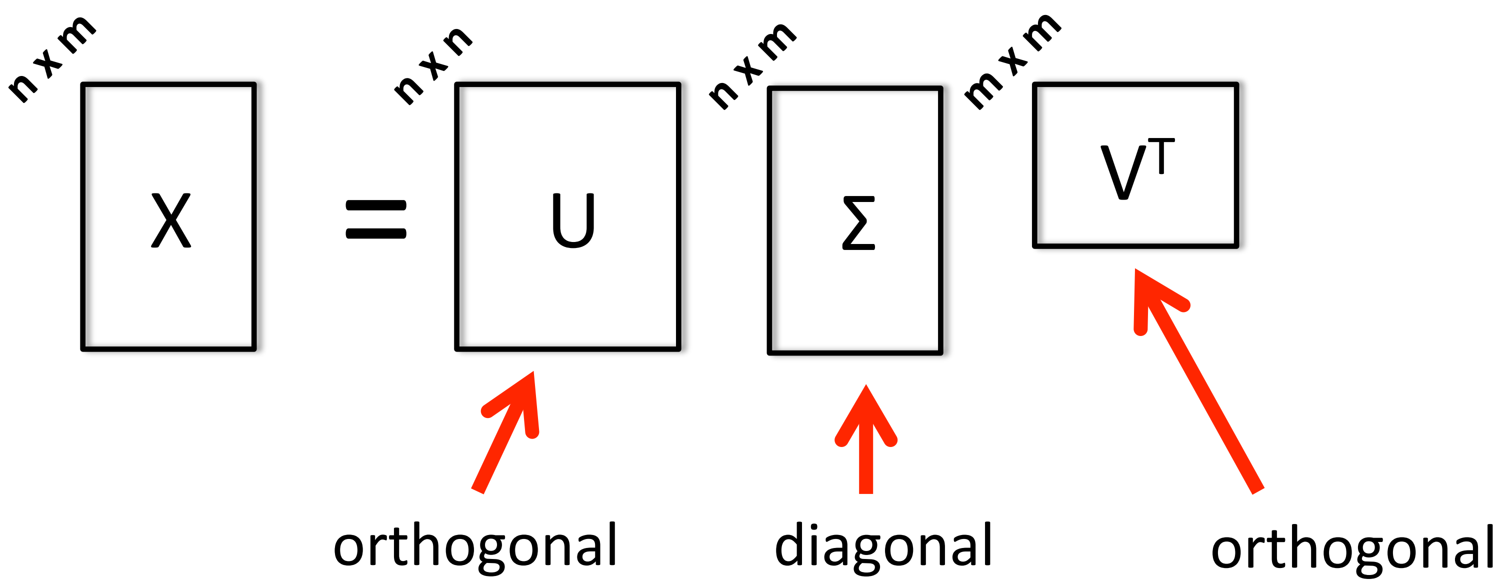
THE NUCLEAR NORM

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



THE NUCLEAR NORM

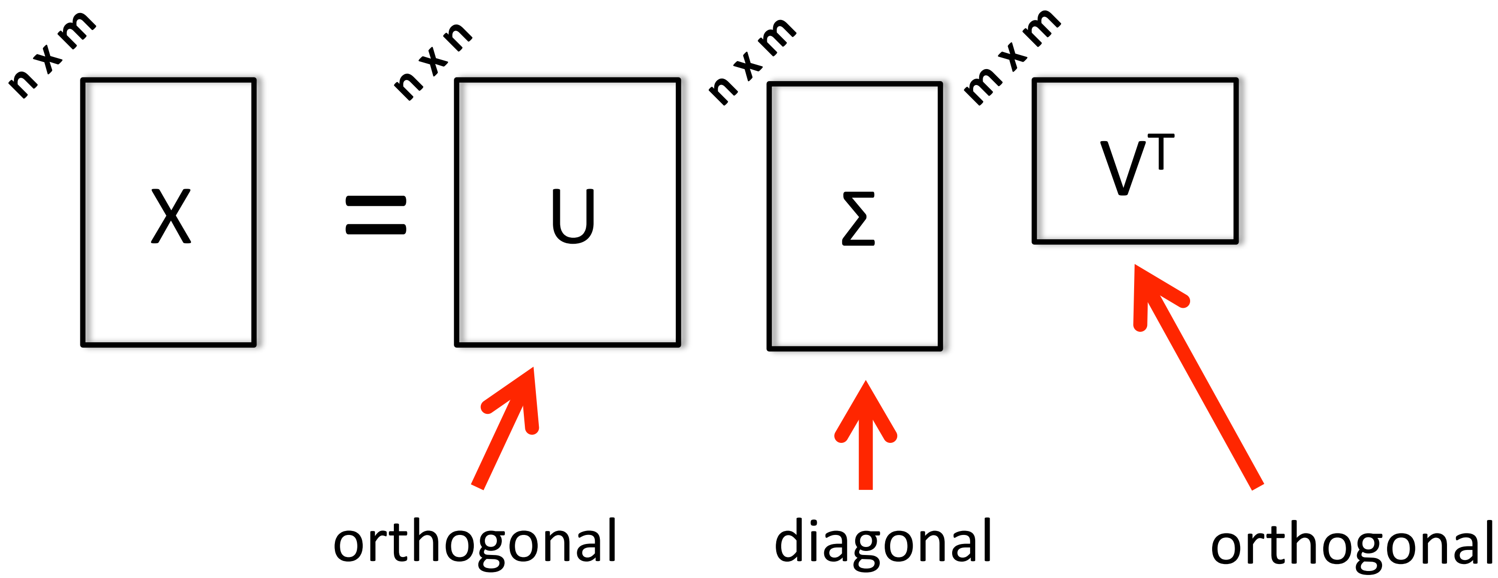
Consider the **singular value decomposition** of X:



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

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

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

[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 $C^6 nr\log^2 n$ 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 $C^6 nr\log^2 n$ observations

This is nearly optimal, since there are 2nr parameters

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 $C^6 nr\log^2 n$ observations

This is nearly optimal, since there are 2nr parameters

Many other approaches, e.g. **alternating minimization**:

[Keshavan, Montanari, Oh], [Jain, Netrapalli, Sanghavi], [Hardt], …

Part II:

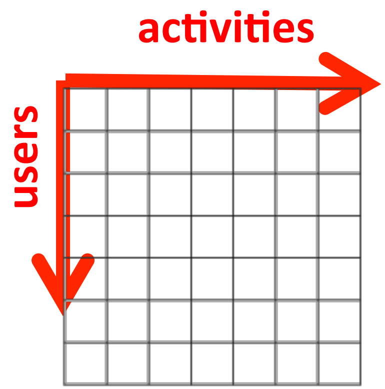
Higher order structure?

TENSOR PREDICTION

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

TENSOR PREDICTION

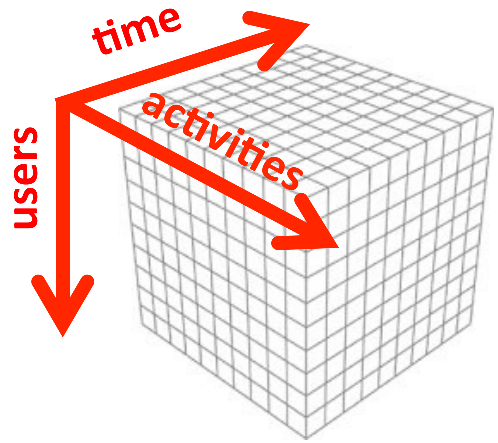
Can using **more than two** attributes can lead to better recommendations?



e.g. Groupon

TENSOR PREDICTION

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

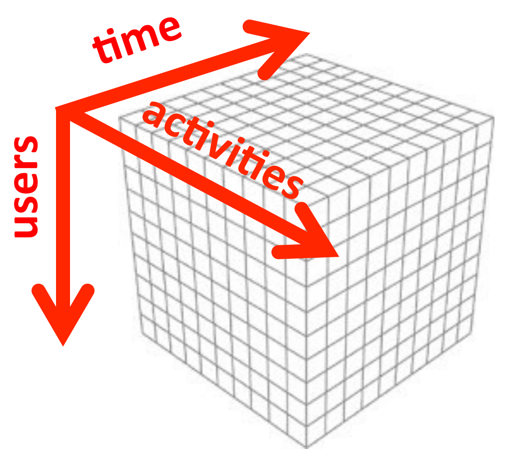


e.g. Groupon

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

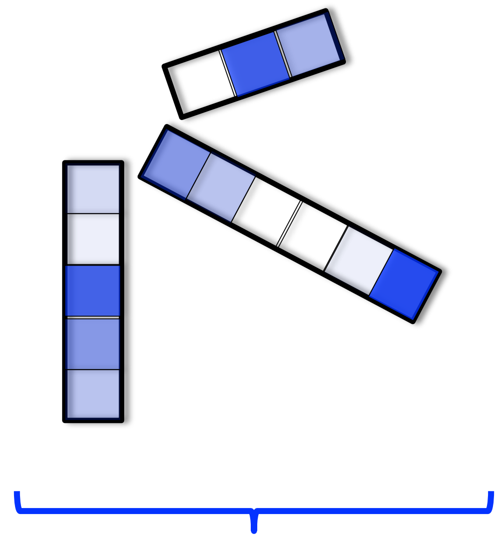
TENSOR PREDICTION

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



e.g. Groupon

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

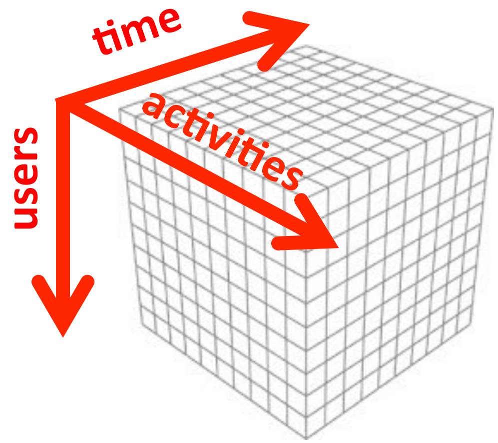


snow sports

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

TENSOR PREDICTION

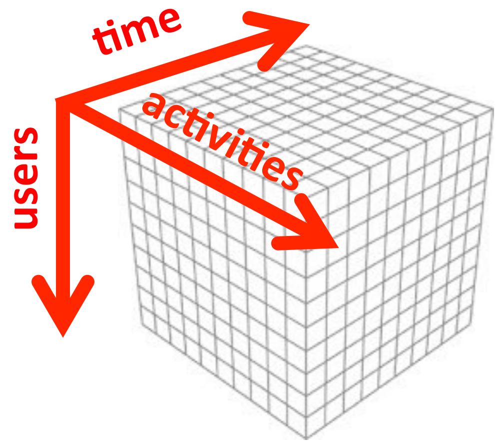
Can using **more than two** attributes can lead to better recommendations?



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

TENSOR PREDICTION

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

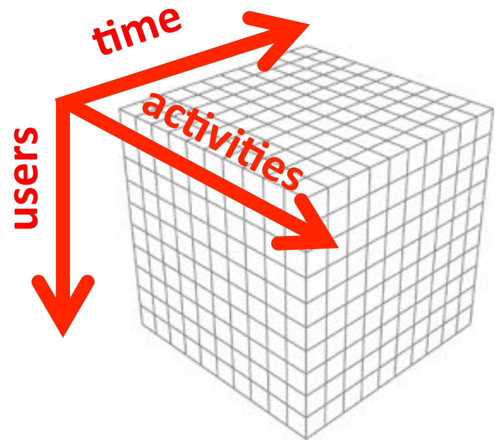


$$T = \sum_{i=1}^{r} a_i \otimes b_i \otimes c_i$$

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

TENSOR PREDICTION

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



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

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

More attributes lead to better recommendations, but more complex objects…

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

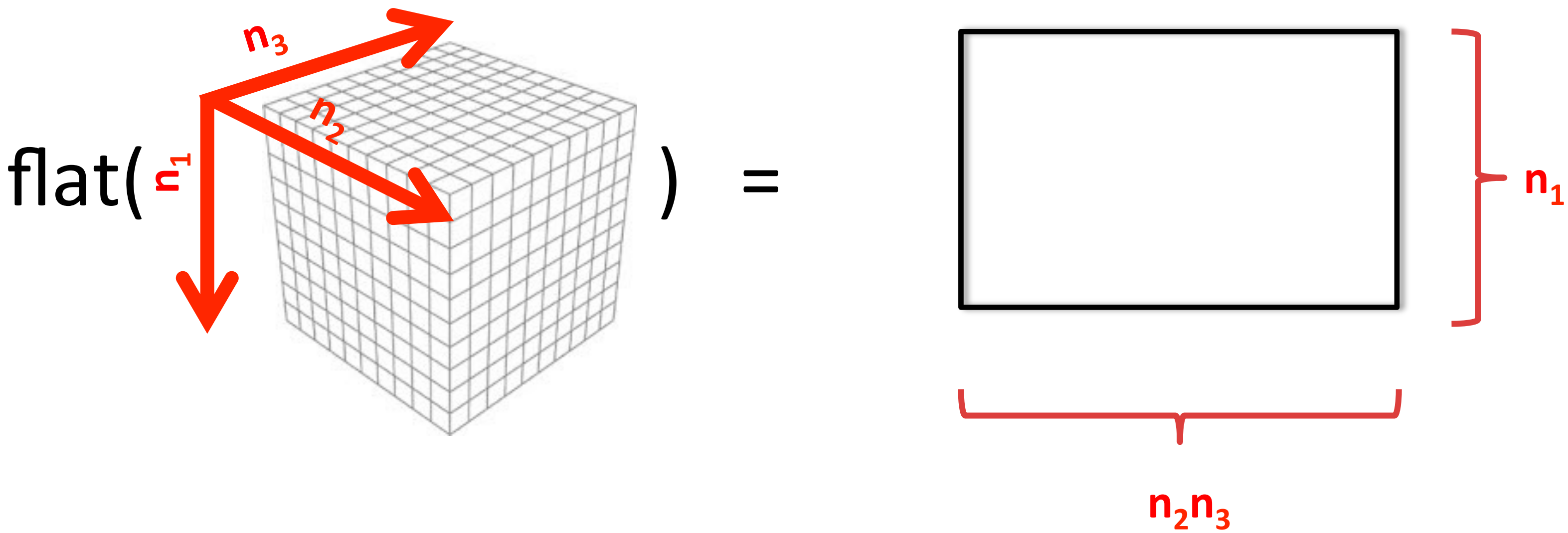
Problem	**Complexity**
Bivariate Matrix Functions over $\mathbb{R}$, $\mathbb{C}$	Undecidable (Proposition 12.2)
Bilinear System over $\mathbb{R}$, $\mathbb{C}$	NP-hard (Theorems 2.6, 3.7, 3.8)
Eigenvalue over $\mathbb{R}$	NP-hard (Theorem 1.3)
Approximating Eigenvector over $\mathbb{R}$	NP-hard (Theorem 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**:

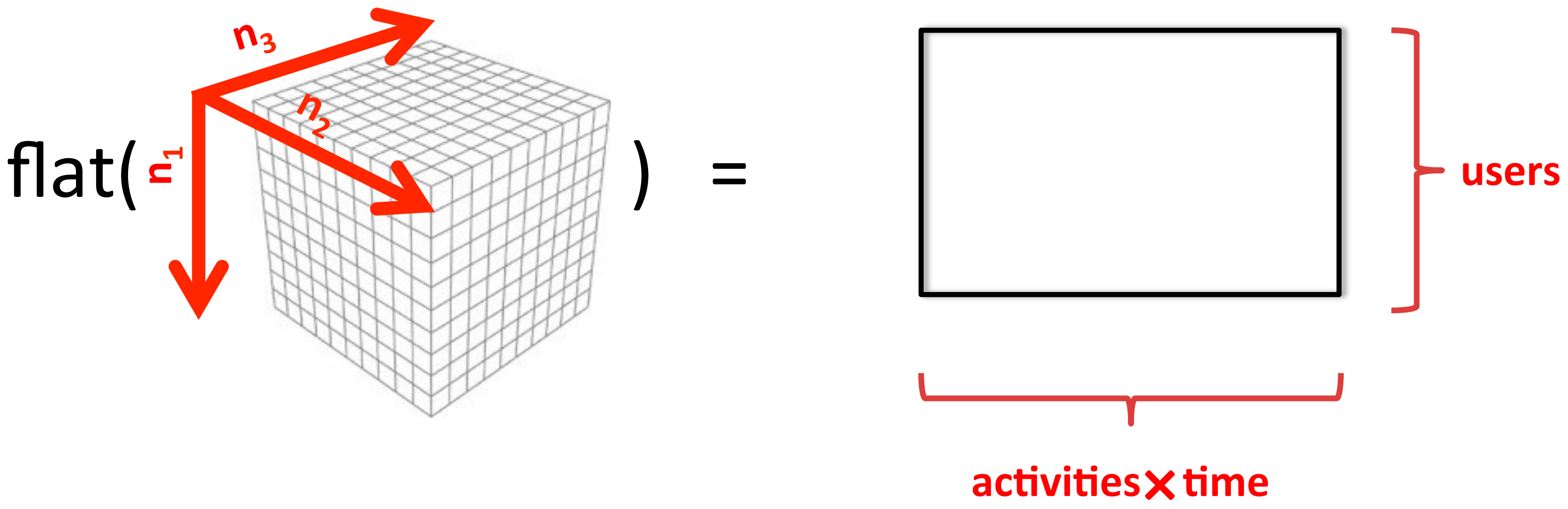
FLATTENING A TENSOR

Many tensor methods rely on **flattening**:



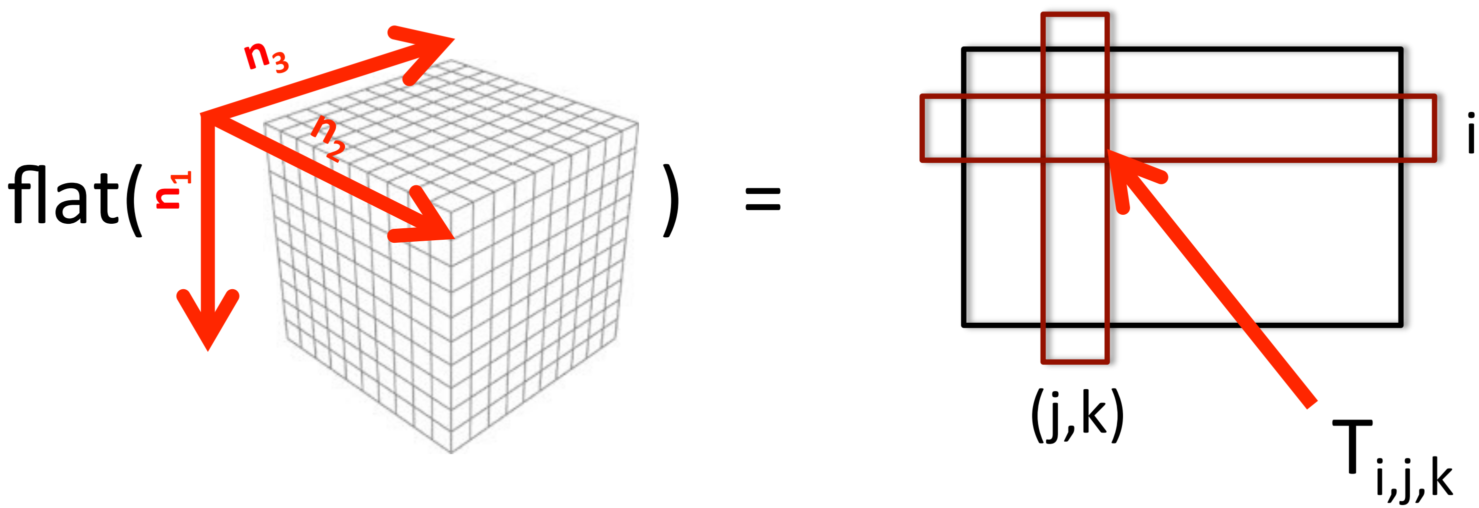
FLATTENING A TENSOR

Many tensor methods rely on **flattening**:



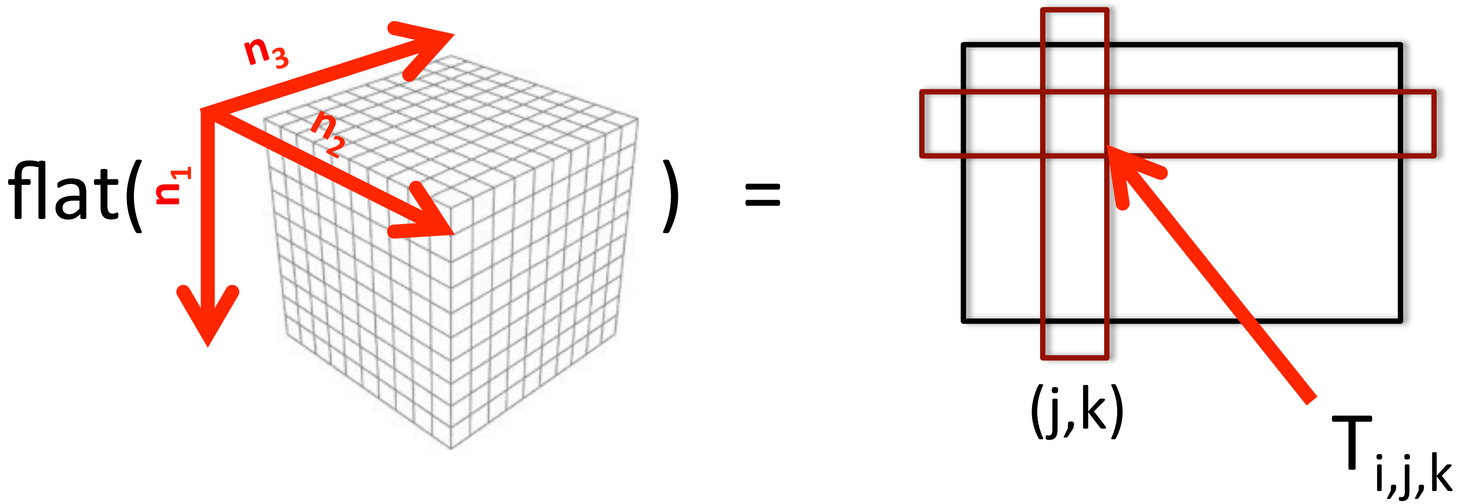
FLATTENING A TENSOR

Many tensor methods rely on **flattening**:



FLATTENING A TENSOR

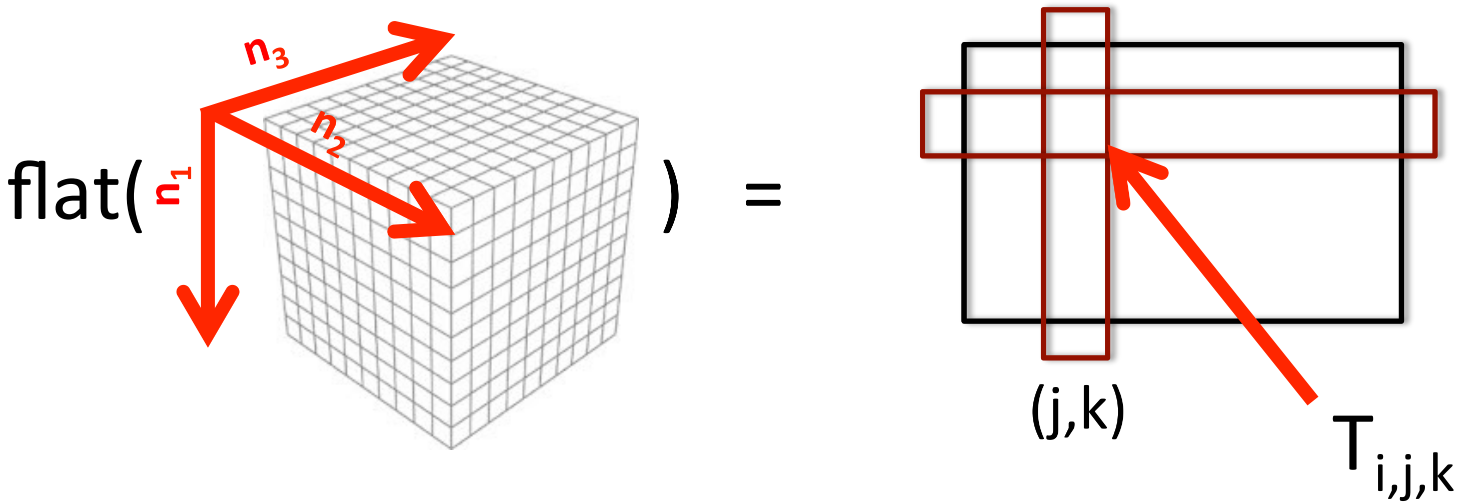
Many tensor methods rely on **flattening**:



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**:



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

Let $n_1 = n_2 = n_3 = n$

We would need $\widetilde{O}(n^2 r)$ 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?

Part III:

Nearly optimal algorithms for tensor prediction

OUR RESULTS

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

standard Gaussian r.v.

OUR RESULTS

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

standard Gaussian r.v.

Theorem: Suppose $var(T_{i,j,k}) \geq r$. Then there is an efficient algorithm that outputs X which satisfies:

$$X_{i,j,k} = (1 \pm o(1)) T_{i,j,k}$$

for a 1-o(1) fraction of entries, provided $m = \widetilde{\Omega}(n^{3/2} r)$

OUR RESULTS

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

standard Gaussian r.v.

Theorem: Suppose $\mathrm{var}(T_{i,j,k}) \geq r$. Then there is an efficient algorithm that outputs X which satisfies:

$$X_{i,j,k} = (1 \pm o(1)) T_{i,j,k}$$

for a 1-o(1) fraction of entries, provided $m = \tilde{\Omega}(n^{3/2} r)$

This variance bound holds for **random** tensors, but also tensors where the factors (a_i's, b_i's, c_i's) have **large inner-product**

OUR RESULTS

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

standard Gaussian r.v.

Theorem: Suppose $var(T_{i,j,k}) \geq r$. Then there is an efficient algorithm that outputs X which satisfies:

$$X_{i,j,k} = (1 \pm o(1)) T_{i,j,k}$$

for a 1-o(1) fraction of entries, provided $m = \tilde{\Omega}(n^{3/2} r)$

Even for $r = n^{3/2-\delta}$ (**highly overcomplete**), we only need to observe an o(1) fraction of the entries to predict almost everything

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…

Tensor prediction with m observations → Refute random 3-SAT with m clauses

LOWER BOUNDS

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

Tensor prediction with m observations → Refute random 3-SAT with m clauses

The best known algorithms require $m = n^{3/2}$, and even **powerful** SDP hierarchies fail with fewer clauses

[Shor], [Nesterov], [Parrilo], [Lasserre], [Grigoriev], [Schoenebeck]

LOWER BOUNDS

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

Tensor prediction with m observations → Refute random 3-SAT with m clauses

The best known algorithms require $m = n^{3/2}$, and even **powerful** SDP hierarchies fail with fewer clauses

[Shor], [Nesterov], [Parrilo], [Lasserre], [Grigoriev], [Schoenebeck]

Corollary [informal]: Any algorithm for solving tensor prediction, in the sum-of-squares hierarchy that uses $m=n^{3/2-\delta}r$ observations must run in exponential time

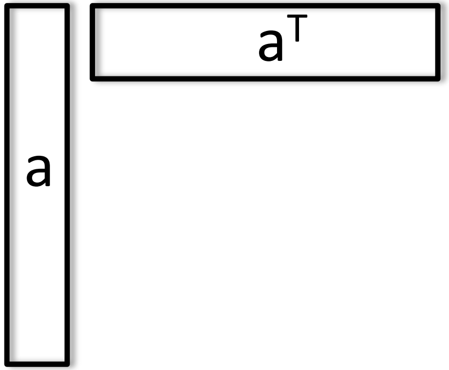
Part IV:

Our techniques: connections to random CSPs

Can we distinguish between low-rank and random?

Can we distinguish between low-rank and random?

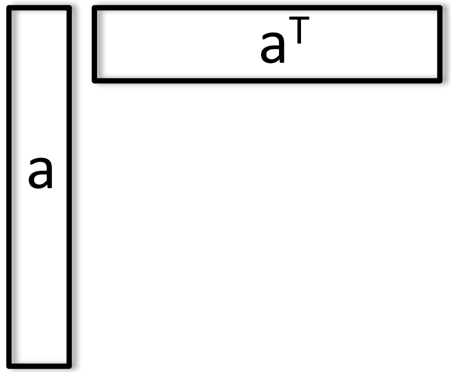
Case #1: Approximately low-rank



+ **noise**

Can we distinguish between low-rank and random?

Case #1: Approximately low-rank



+ **noise**

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

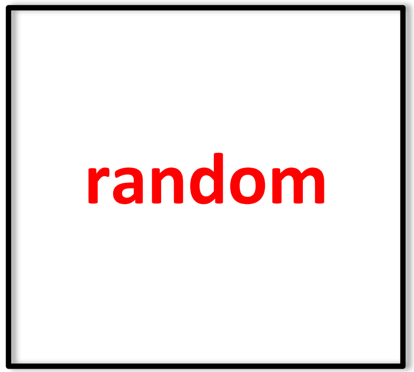
$$M_{i,j} = \begin{cases} a_i a_j & \text{w/ probability } 3/4 \\ \text{random } \pm 1 & \text{w/ probability } 1/4 \end{cases}$$

where each $a_i = \pm 1$

Can we distinguish between low-rank and random?

Can we distinguish between low-rank and random?

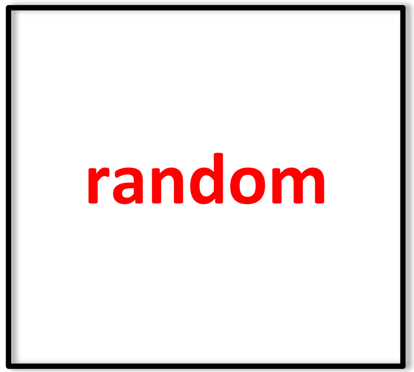
Case #2: Random



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

Can we distinguish between low-rank and random?

Case #2: Random



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**

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:

$$OPT(\psi) \leq val(\psi)$$

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:

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

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:

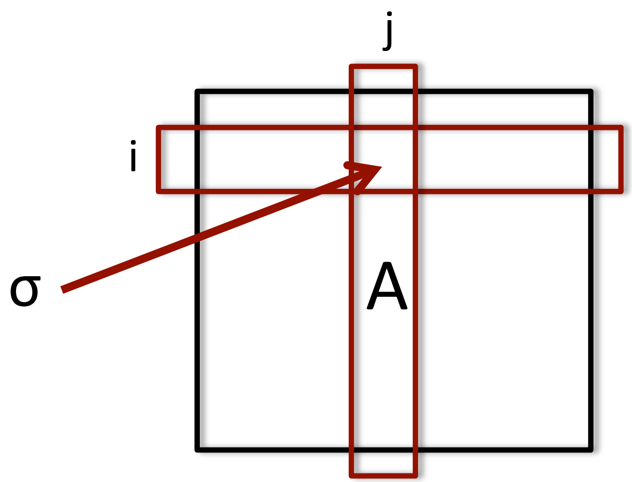
(1) On any 2-XOR formula ψ, it outputs **val** where:

$$OPT(\psi) \leq val(\psi)$$

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

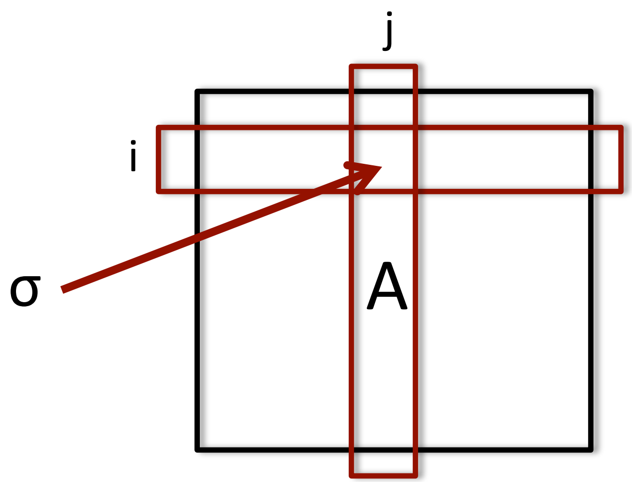
$$val(\psi) = \frac{1}{2} + o(1)$$

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



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

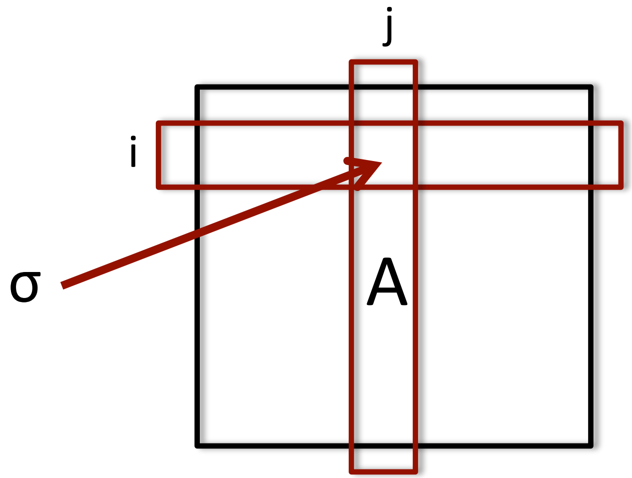
Lemma: If $(i_1, j_1; \boldsymbol{\sigma_1}), \ldots, (i_m, j_m; \boldsymbol{\sigma_m}) \Leftrightarrow \psi$ then



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

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

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

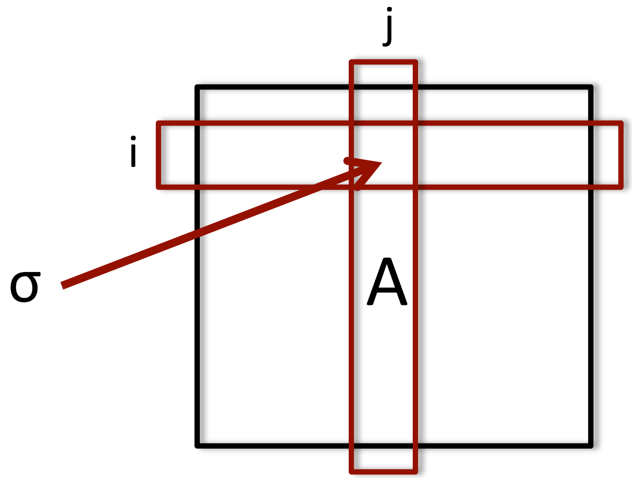


$$\frac{2\,\mathrm{OPT}(\psi) - 1}{n} \leq \frac{1}{m} \|A\|_2$$

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

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

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

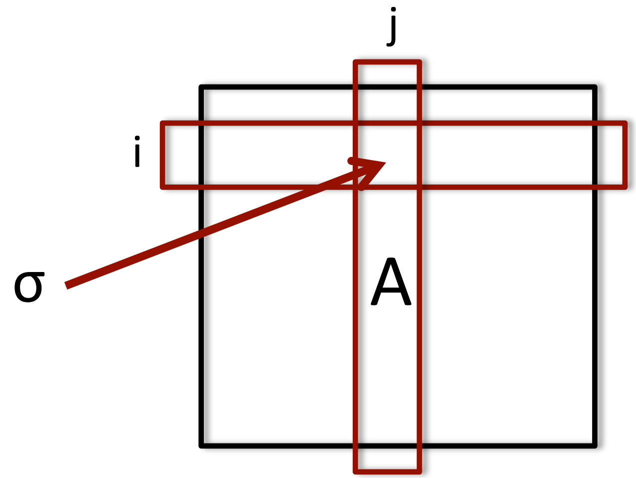


$$\frac{2\,\mathrm{OPT}(\psi) - 1}{n} \leq \frac{1}{m} \|A\|_2$$

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

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

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



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

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

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

This solves the strong refutation 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**

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 → **Strongly refute*** random 2-XOR/2-SAT with m clauses

Rademacher Complexity

*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 → **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 → (**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

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

Embedding in SOS

[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

GENERALIZATION BOUNDS

Suppose we are given $|\Omega| = m$ noisy observations $T_{i,j,k} \pm \eta$, and the factors of T are C-incoherent:

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

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

GENERALIZATION BOUNDS

Suppose we are given $|\Omega| = m$ noisy observations $T_{i,j,k} \pm \eta$, and the factors of T are C-incoherent:

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

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

This comes from giving an efficiently computable norm $\|\cdot\|_K$ whose Rademacher complexity is asymptotically smaller than the trivial bound whenever $m=\Omega(n^{3/2}\log^4 n)$

SUMMARY

New Algorithm:

We gave an algorithm for 3rd-order tensor prediction that uses $m = n^{3/2} r \log^4 n$ observations

SUMMARY

New Algorithm:

We gave an algorithm for 3rd-order tensor prediction that uses $m = n^{3/2} r \log^4 n$ observations

---

An Inefficient Algorithm: (via **tensor nuclear norm**)

There is an inefficient algorithm that use $m = nr\log n$ observations

SUMMARY

New Algorithm:

We gave an algorithm for 3rd-order tensor prediction that uses $m = n^{3/2} r \log^4 n$ observations

---

An Inefficient Algorithm: (via **tensor nuclear norm**)

There is an inefficient algorithm that use $m = nr \log n$ observations

---

A Phase Transition:

Even for n^{δ} rounds of the powerful sum-of-squares hierarchy, no norm solves tensor prediction with $m = n^{3/2-\delta} r$ observations

Epilogue:

New directions in linear inverse problems

Robust PCA [Candes et al.], [Chandrasekaran et al.], …

Can we recover a low rank matrix from sparse corruptions?

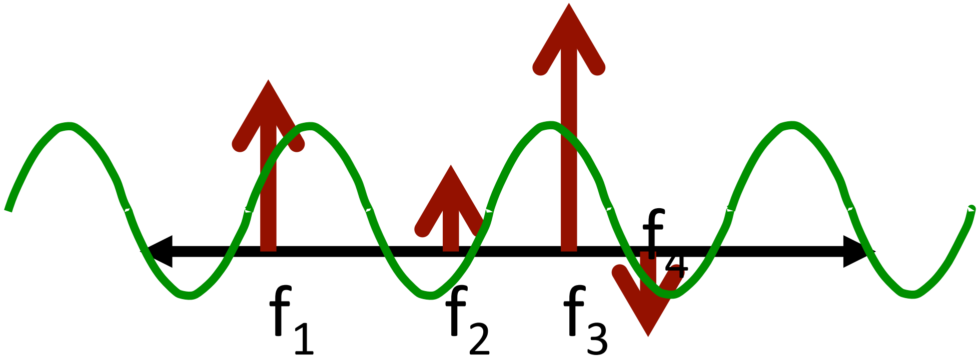
Robust PCA [Candes et al.], [Chandrasekaran et al.], …

Can we recover a low rank matrix from sparse corruptions?

Superresolution, compressed sensing off-the-grid

[Candes, Fernandez-Granda], [Tang et al.], …

Can we recover well-separated points from low-frequency measurements?



DISCUSSION

Convex programs are unreasonably effective for linear inverse problems!

DISCUSSION

Convex programs are unreasonably effective for linear inverse problems!

But we gave simple **linear inverse problems** that exhibit striking **gaps** between efficient and inefficient estimators

DISCUSSION

Convex programs are unreasonably effective for linear inverse problems!

But we gave simple **linear inverse problems** that exhibit striking **gaps** between efficient and inefficient estimators

Where else are there computational vs statistical tradeoffs?

DISCUSSION

Convex programs are unreasonably effective for linear inverse problems!

But we gave simple **linear inverse problems** that exhibit striking **gaps** between efficient and inefficient estimators

Where else are there computational vs statistical tradeoffs?

New Direction: Explore computational vs. statistical tradeoffs through the powerful **sum-of-squares** hierarchy