Completion (on tensors)

NATALI RUCHANSKY

Motivating Example: Netflix

What data does Netflix have?

- Users rate movies they have seen

Data represented in matrix form:

- Harry rates The Hobbit 5/5 but Gravity 2/5

		The Hobbit			**Gravity**	
	3	4	1	1	5	
Harry	2	$\boxed{5}$	1	1	$\boxed{2}$	Users
	3	3	4	3	1	
	5	4	5	2	1	

Movies

Motivating Example: Netflix

What does Netflix want to do?

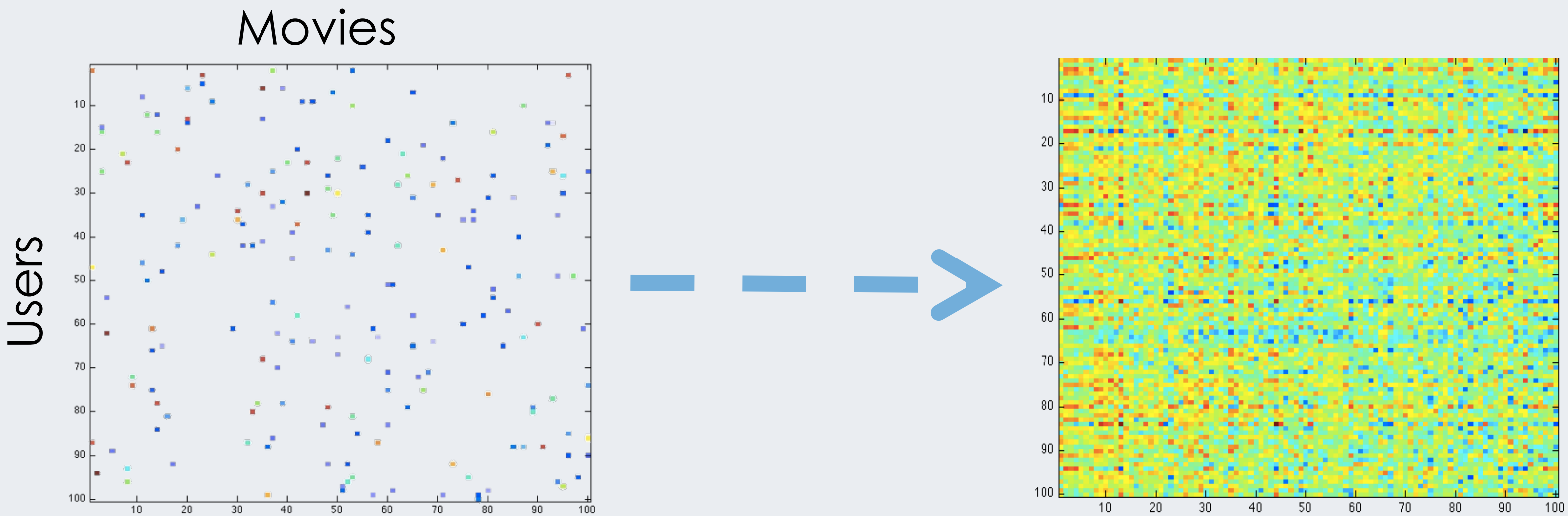
- Recommend you movie you would like – make you happy with the service

How can it do this?

- Knowing which movies Harry would rate 5/5
- Predict ratings it cannot see

			The Hangover		
	3	4	1	**?**	5
Harry	**?**	5	$\boxed{?}$	1	2
	3	3	4	3	1
	5	**?**	**?**	**?**	**?**

Netflix Challenge



Want to predict **all** ratings, but we know only 1% of the entries!

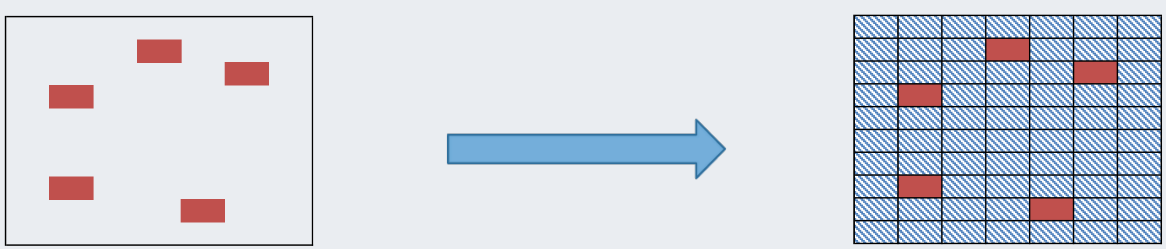
Matrix Completion

On Input:

Partially-observed matrix ***M***

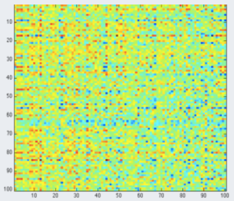
The Goal is to:

Find a **low-error completion** of the remaining entries



Notation

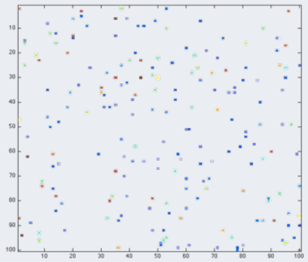
From a **ground truth** fully-known an $n \times m$ matrix T, we say only some of the values are observed.



Call the set of **observed locations**:

$$\Omega = \{(i, j) \text{ that are observed in } T\}$$

Define the **partially observed matrix** as:



$$M(i, j) = \begin{cases} T(i, j) \text{ if } (i, j) \in \Omega \\ 0 \text{ otherwise} \end{cases}$$

Approaches to Matrix Completion

Need to make *some* assumption.

- Currently the most popular is that ***T*** is *low rank*.

Find an estimate $\tilde{T}$ by solving the optimization problem:

$$\begin{aligned} &\text{minimize} && rank(\tilde{T}) \\ &\text{subject to} && M_{ij} = \tilde{T}_{ij} \quad (i,j) \in \Omega \end{aligned}$$

This problem is **NP-hard**.

Approaches to Matrix Completion

Need to make *some* additional assumption.

1. Low rank
2. Known rank

A proposed strategy called **OptSpace** takes an **SVD**-based approach:

$$\begin{aligned} &\text{minimize} && \left\|\tilde{T}\right\|_* \\ &\text{subject to} && M_{ij} = \tilde{T}_{ij} \quad (i,j) \in \Omega \end{aligned}$$

$$\left\|\tilde{T}\right\|_* = \sum_{k=1}^{n} \sigma_k(\tilde{T})$$

Characteristics

Such optimization approaches

- Output an estimate of the **whole** matrix $\tilde{T}$ on any input – any size Ω.
- Give no indication of of confidence
- Rely on randomly sampled Ω, i.e. the known entries are randomly dispersed

Completion Questions

If we want very small error between $\tilde{T}$ and T ...

1. How many entries does M need to have?
2. When there are not enough entries, how can I add them?

Lower Bound

Assuming that Ω is **random**, there does exist a bound that say:

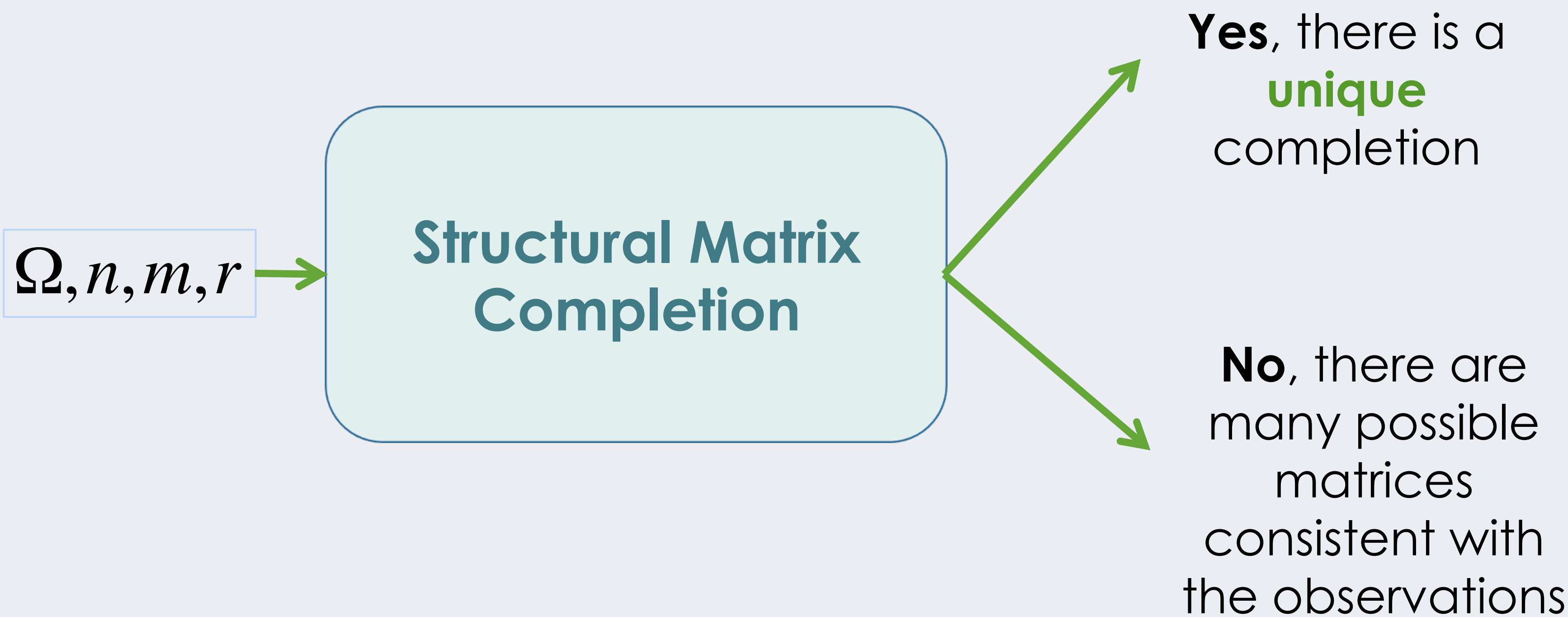
If there are at least $r\,n\log(n)$ *entries then w.h.p the completion will be accurate*

This is the only way to answer the questions.

Simply sample randomly up to threshold, and hope for the best…

Structural Methods

Recent methods take a more algebraic approach by analyzing the structure of **locations** of the observed entries.



Structural Methods Example

Consider the 2 x 2 matrix: $\begin{pmatrix} 1 & 2 \\ 2 & ? \end{pmatrix}$

If I tell you that **rank=1**, can you fill the missing element?

Yes! The only possible value is 4. i.e. $\begin{pmatrix} 1 & 2 \\ 2 & 4 \end{pmatrix}$

What about if I tell you **rank=2**?

Structural Method

- Explicitly state which entries can be recovered

- Provide means to answer the questions other methods could not
  1. How many entries does Ω need to have?
  2. When there are not enough entries, how can I add them?

Summary

Matrix Completion: Recover missing entries

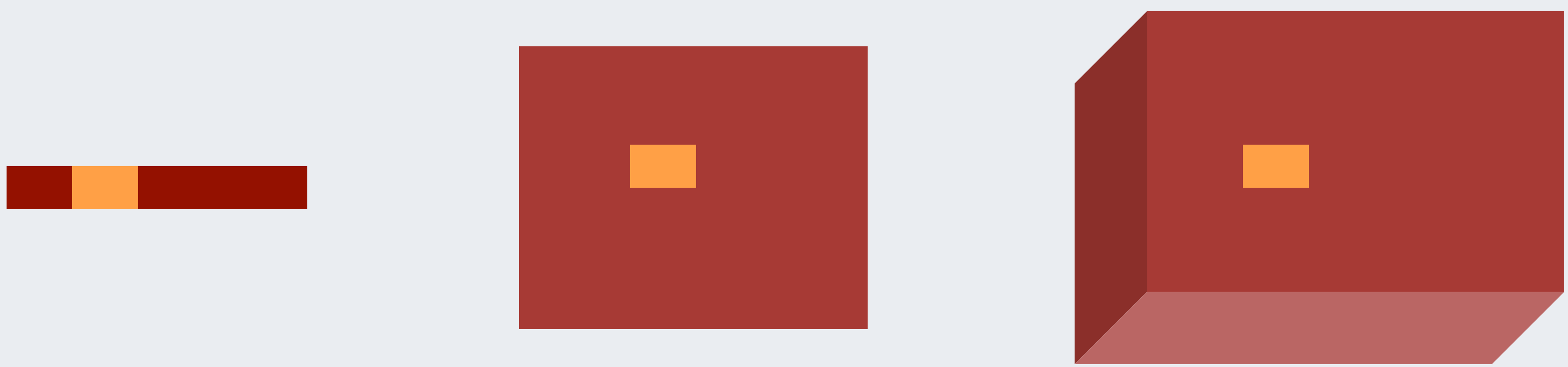
- Statistical approaches
  - Random Sampling
  - Optimization – always output a full estimate
- Structural approaches
  - Indicate which entries can be recovered exactly
  - Only recover those entries

Tensor Completion. Let’s talk about tensors…

Tensor

Multi-dimensional matrix

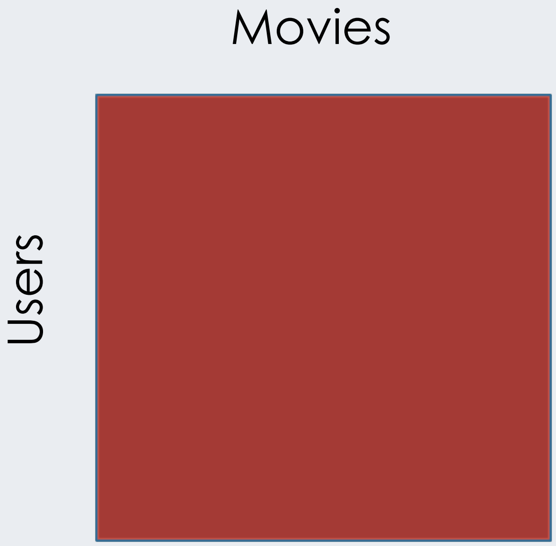
1-dim 2-dim 3-dim



A(i) A(i,j) A(i,j,k)

Tensors: Example

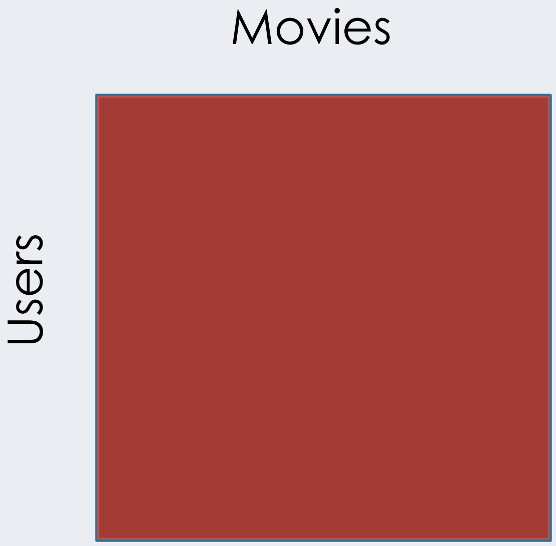
Netflix Matrix



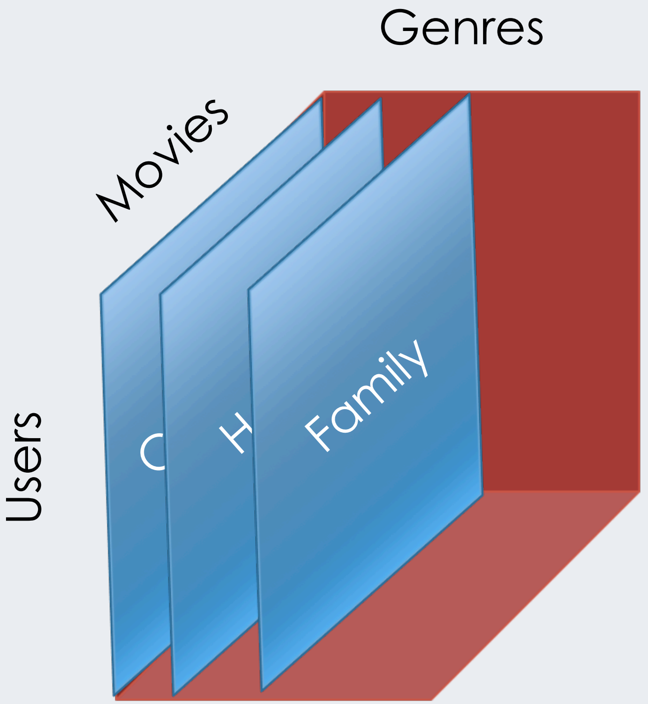
Netflix Tensor

Tensors: Example

Netflix Matrix



Netflix Tensor



Tensor Applications

- Traffic analysis (urban, internet)
- Recommender systems
- Biology
- Vision/Images
- Physics (immense application) e.g.

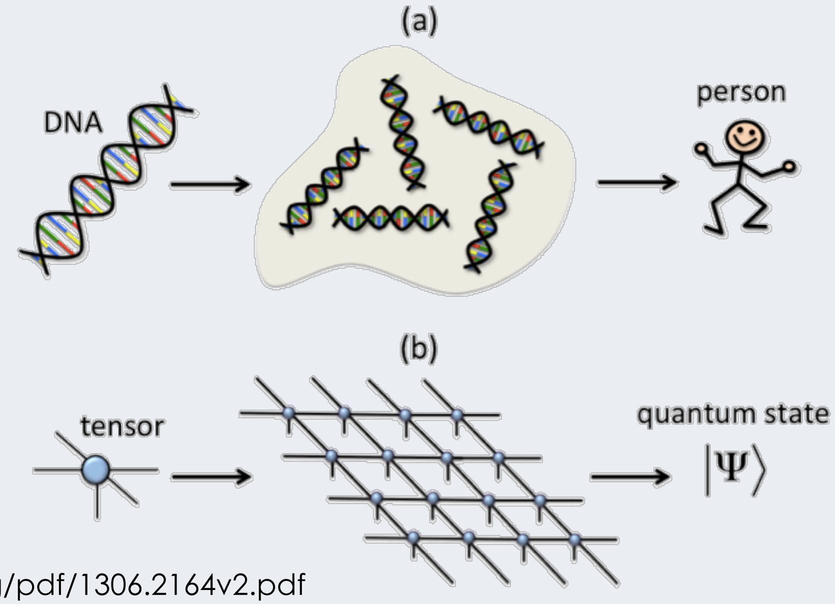


Image Source: http://arxiv.org/pdf/1306.2164v2.pdf

Tensors are hard

Most Tensor Problems are NP-Hard

CHRISTOPHER J. HILLAR, Mathematical Sciences Research Institute
LEK-HENG LIM, University of Chicago

We prove that multilinear (tensor) analogues of many efficiently computable problems in numerical linear algebra are NP-hard. Our list here includes: determining the feasibility of a system of bilinear equations, deciding whether a 3-tensor possesses a given eigenvalue, singular value, or spectral norm; approximating an eigenvalue, eigenvector, singular vector, or the spectral norm; and determining the rank or best rank-1 approximation of a 3-tensor. Furthermore, we show that restricting these problems to symmetric tensors does not alleviate their NP-hardness. We also explain how deciding nonnegative definiteness of a symmetric

Tensors are 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

Example 1: Rank

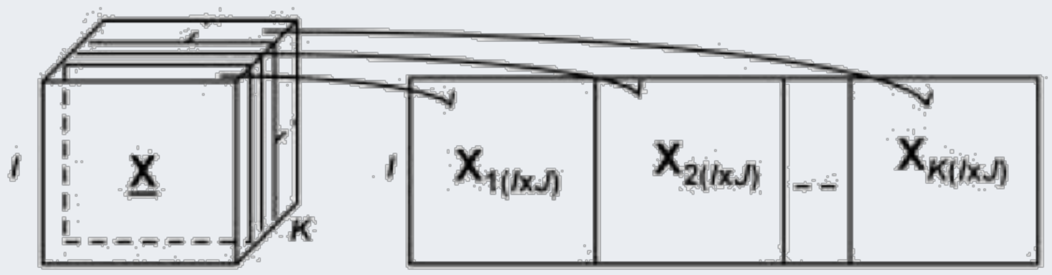
- Matrix Rank:
  - # independent row vectors
  - How many vectors are needed to represent the whole matrix?
  - For m x n M, rank(M)≤max(m,n)

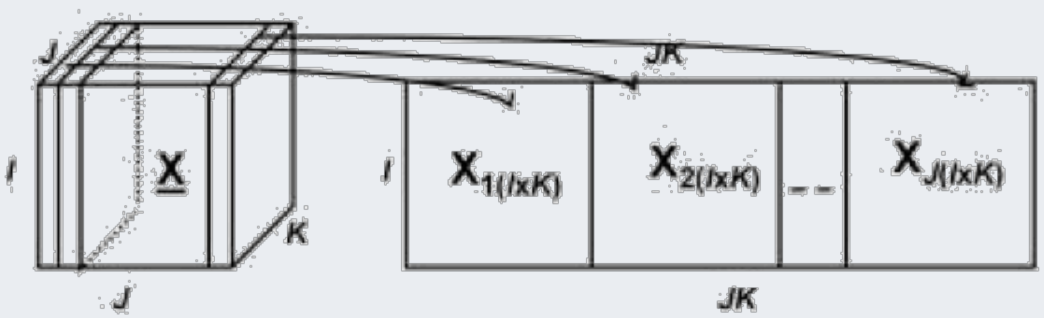
- Tensor Rank
  - Definition 1:

    Smallest r such that T is the sum of r rank-1 tensors B
  - Definition 2:

    A tuple $(r_1, \ldots, r_n)$ *where each* r_i *is the rank of the unfolding*

Tensor Unfloding





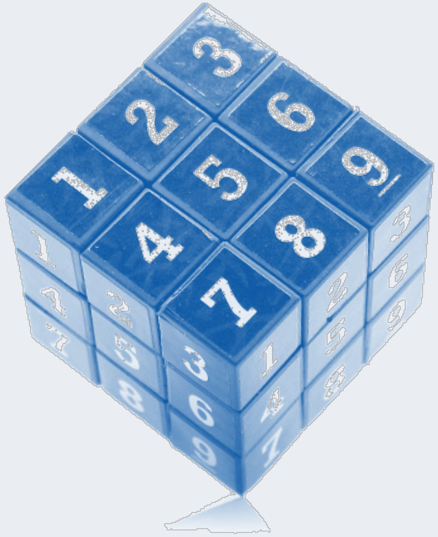
Example 2: SVD

- Matrix SVD:
  - Decomposition into $\mathbf{M} = \mathbf{U}\boldsymbol{\Sigma}\mathbf{V}^*$
- Tensor SVD??

Tensor Completion

Given a partially observed tensor

Reconstruct the missing values



Existing Tensor Completion Approaches

- Translate directly

$$\underset{\mathbf{X}\in\mathcal{T}}{\textbf{minimize}} \quad f\left(\text{n-rank}(\mathbf{X})\right) \quad \textbf{s. t.} \quad \mathcal{A}\left(\mathbf{X}\right) = b,$$

<u>Recall:</u>

$$\begin{array}{ll} \text{minimize} & rank(\tilde{T}) \\ \text{subject to} & M_{ij} = \tilde{T}_{ij} \quad (i,j) \in \Omega \end{array}$$

Problem setting 3.1 (**Low-n-rank tensor recovery**).

$$\underset{\mathbf{X}\in\mathcal{T}}{\textbf{minimize}} \quad \sum_{i=1}^{N} \text{rank}\left(X_{(i)}\right) \quad \textbf{s. t.} \quad \mathcal{A}\left(\mathbf{X}\right) = b$$

Problem setting 3.2 (**Tensor completion**).

$$\underset{\mathbf{X}\in\mathcal{T}}{\textbf{minimize}} \quad \sum_{i=1}^{N} \text{rank}\left(X_{(i)}\right) \quad \textbf{s. t.} \quad \mathbf{X}_{\Omega} = \mathbf{T}_{\Omega}$$

Problem setting 3.3 (**Low-n-rank tensor pursuit**).

$$\underset{\mathbf{X}\in\mathcal{T}}{\textbf{minimize}} \quad \sum_{i=1}^{N} ||X_{(i)}||_{*} \quad \textbf{s. t.} \quad \mathcal{A}(\mathbf{X}) = b.$$

Tensor Completion Question

- Can the ideas from structural matrix completion be applied to tensors?
- What kinds of bounds exist for the amount of information needed?
- Are there special cases that make tensors easier?
- How does rigidity translate to tensors?

Thank You!