Matrix completion

Vinayak Rao

Purdue University

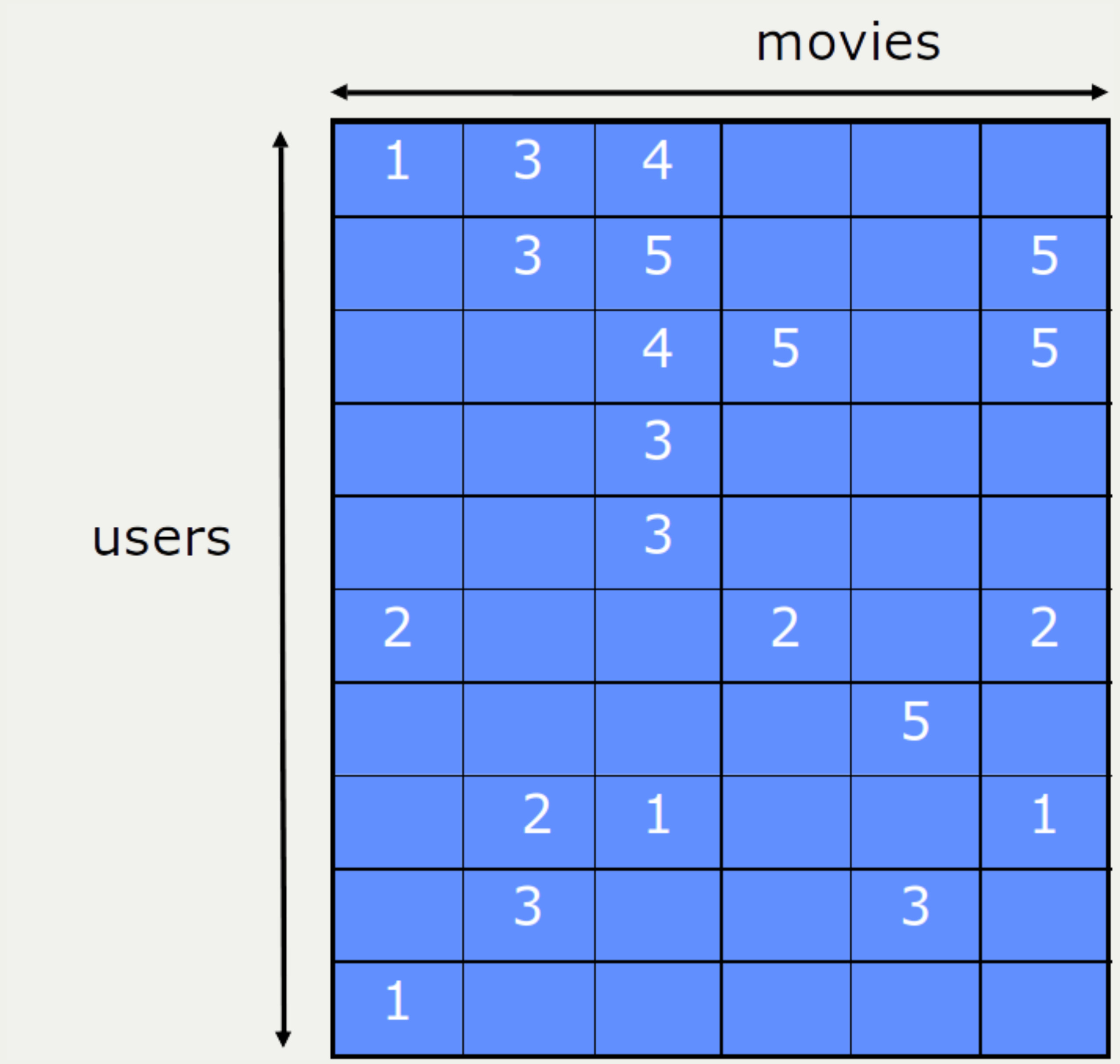
A wide-range of modern applications involve observations organized in the form of a matrix

A canonical example is a collaborative filtering/recommender system:

- The `Netflix prize' (Bennett and Lanning, 2007)

We have a large matrix of users vs movies/products

- element (i, j) is the rating user i gave to movie j



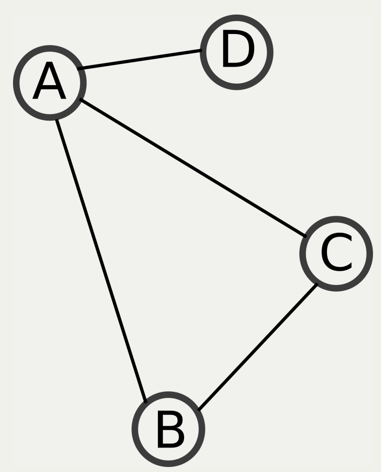
Adjacency matrices

Given a set of V nodes linked by edges

- E.g. users on a social media network with edges representing friendships

Represented by an adjacency matrix:

- $A_{ij} = 1$ if there is an edge between nodes i and j, else 0



$$A = \begin{bmatrix} 0 & 1 & 1 & 1 \\ 1 & 0 & 1 & 0 \\ 1 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 \end{bmatrix}$$

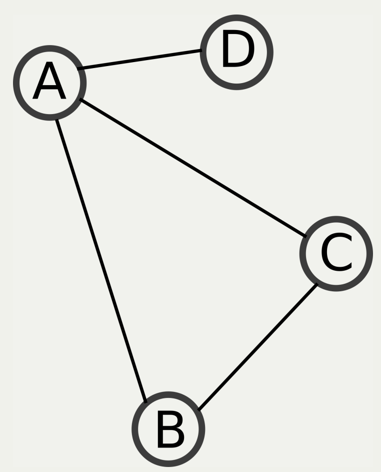
Adjacency matrices

Given a set of V nodes linked by edges

- E.g. users on a social media network with edges representing friendships

Represented by an adjacency matrix:

- $A_{ij} = 1$ if there is an edge between nodes i and j, else 0



$$A = \begin{bmatrix} 0 & 1 & 1 & 1 \\ 1 & 0 & 1 & 0 \\ 1 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 \end{bmatrix}$$

Can be extended to weighted graphs

tf-idf matrices

Another example (a weighted bipartite graph) is a term-document matrix from text analysis

We have a large matrix of documents vs tokens

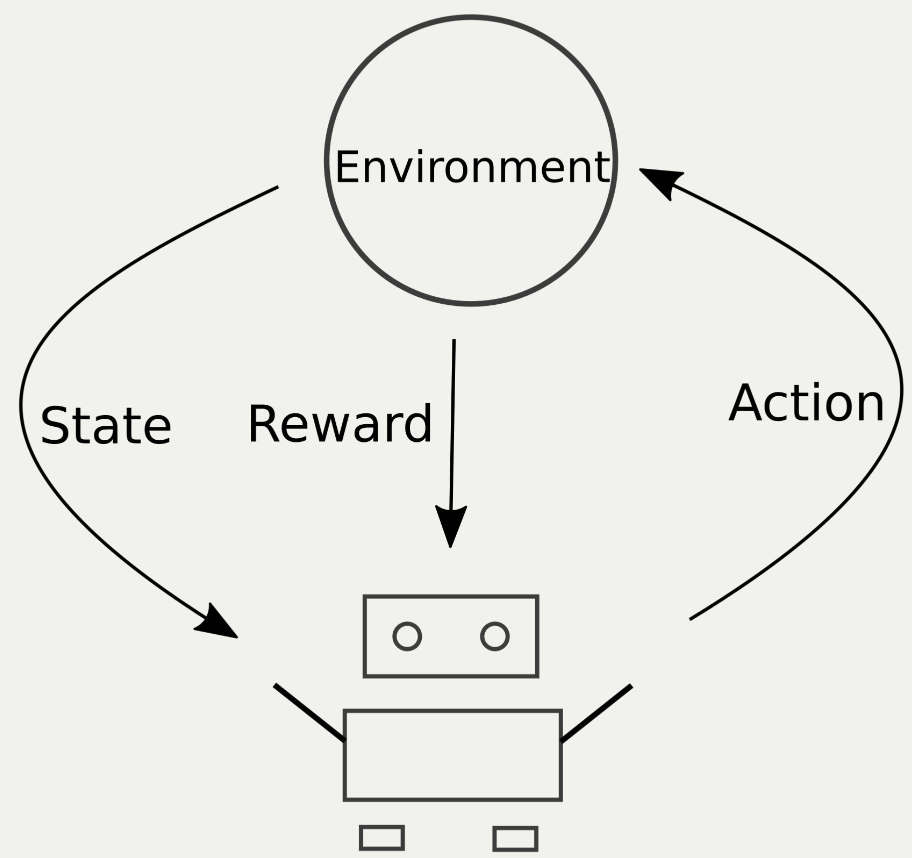
- element (i,j) is the frequency of token j in document i (or a related quantity)

Reward functions in reinforcement learning

An agent explores a state space $\mathcal{S}$ using a set of actions $\mathcal{A}$

The *reward function* is a $|\mathcal{S}| \times |\mathcal{A}|$ matrix of states versus actions

- element (i,j) gives the *reward* from taking action $i \in \mathcal{A}$ in state $j \in \mathcal{S}$



Panel data

Consists of N individuals observed over T time periods

- y_{ij} is the measured outcome for unit i at time t

In causal settings, we also have a binary treatment/control matrix

- $W_{ij} = 1$ if unit i received treatment at time j, else 0.

Thus, we have

$$Y = \begin{bmatrix} y_{11} & y_{12} & \cdots & y_{1T} \\ y_{21} & y_{22} & \cdots & y_{2T} \\ \vdots & \vdots & \ddots & \vdots \\ y_{N1} & y_{N2} & \cdots & y_{NT} \end{bmatrix} \quad W = \begin{bmatrix} 0 & 1 & \cdots & 1 \\ 1 & 0 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 1 & 0 & \cdots & 0 \end{bmatrix}$$

Often, we are interested in, but do not observe the entire matrix

- in recommender systems, we only observe user ratings on a sparse subset of movies
- in reinforncement learning, at any time, we have only taken a subset of actions in each state
- in a panel data, we only observe individual responses to treatment or control at any time

Panel data

Recall we had

$$Y = \begin{bmatrix} y_{11} & y_{12} & \cdots & y_{1T} \\ y_{21} & y_{22} & \cdots & y_{2T} \\ \vdots & \vdots & \ddots & \vdots \\ y_{N1} & y_{N2} & \cdots & y_{NT} \end{bmatrix} W = \begin{bmatrix} 0 & 1 & \cdots & 1 \\ 1 & 0 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 1 & 0 & \cdots & 0 \end{bmatrix}$$

We can write this as two potential outcomes matrices

$$Y(0) = \begin{bmatrix} y_{11} & ? & \cdots & ? \\ ? & y_{22} & \cdots & y_{2T} \\ \vdots & \vdots & \ddots & \vdots \\ ? & y_{N2} & \cdots & y_{NT} \end{bmatrix}, \quad Y(1) = \begin{bmatrix} ? & y_{12} & \cdots & y_{1T} \\ y_{21} & ? & \cdots & ? \\ \vdots & \vdots & \ddots & \vdots \\ y_{N1} & ? & \cdots & ? \end{bmatrix}$$

Matrix completion methods seek to impute missing values

- can help decide which product to recommend
- can help decide which action to take in which state
- can help impute potential outcomes to make causal inference

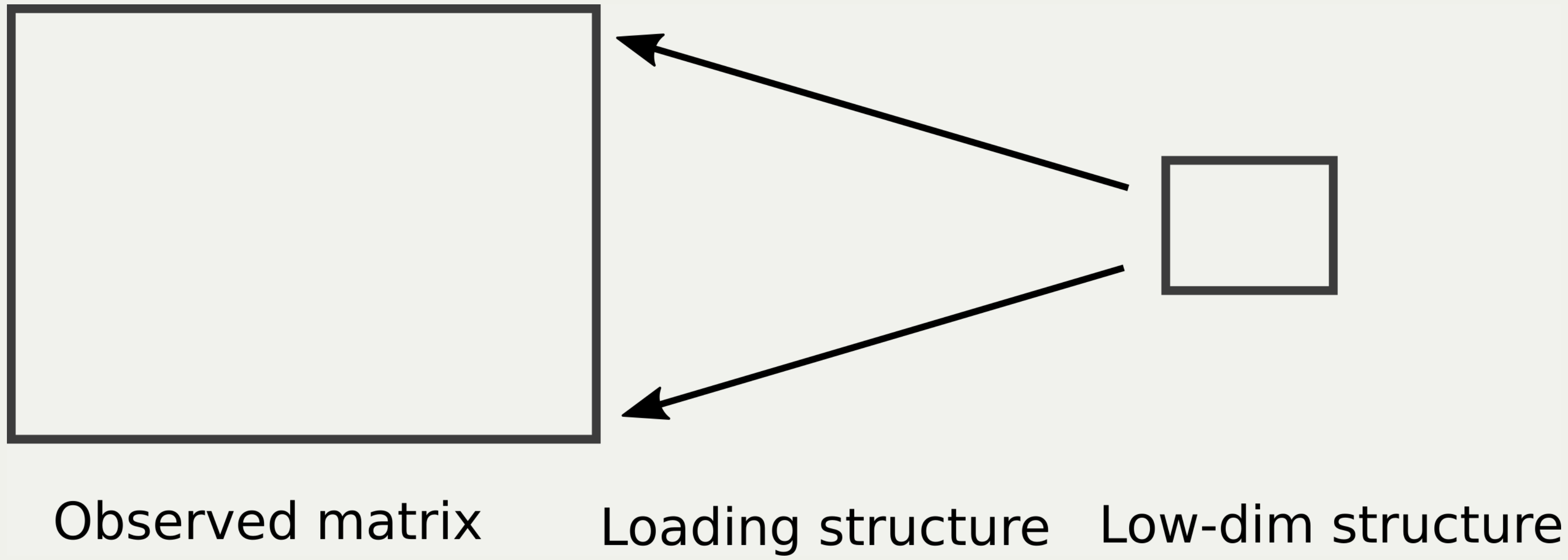
Matrix completion requires imposing structure on the underlying matirx

How can we formalize that

- Someone who like "The Godfather" probably likes "The Godfather Part II"
- Someone who likes "Sharknado" probably won't like Tarkovsky's "The Mirror"

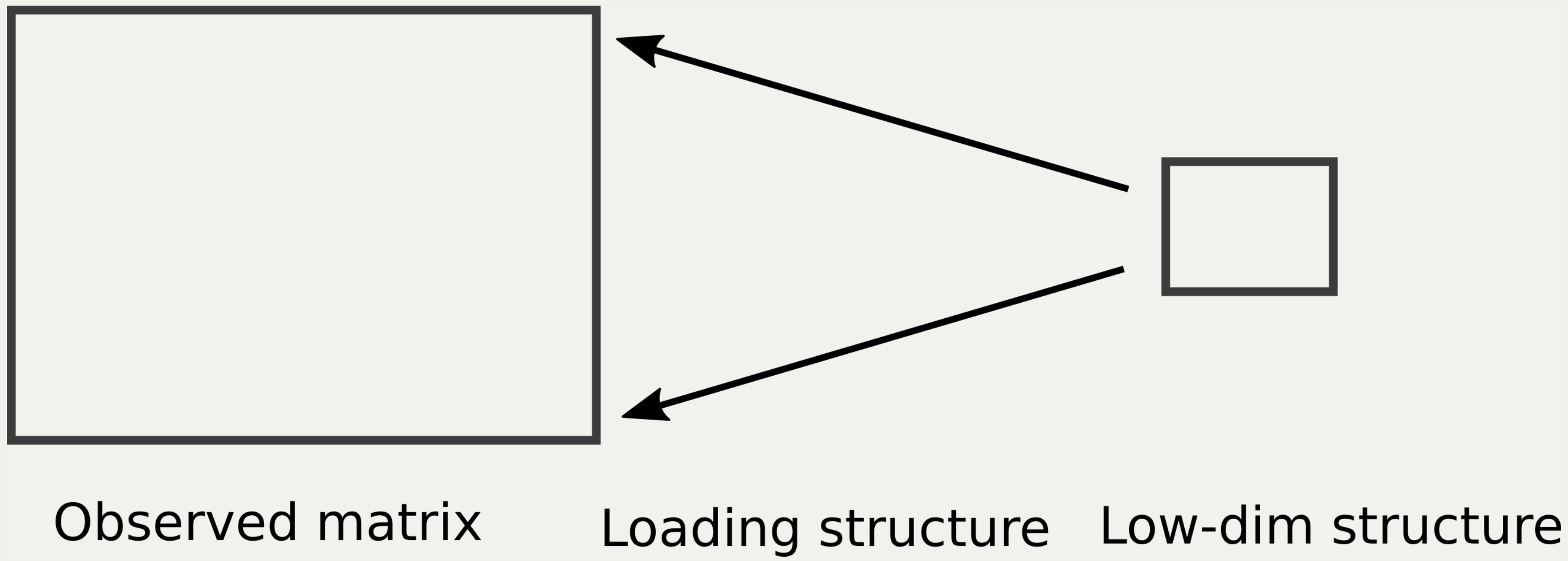
Note typically this is done without knowning details of the movies/users

- Based only on other entries in the matrix



Effectively summarizes an $m \times n$ matrix with a lower-dimensional representation

- use the partial observations to estimate this lower-dim structure
- use the lower-dim structure to impute missing elements



Effectively summarizes an $m \times n$ matrix with a lower-dimensional representation

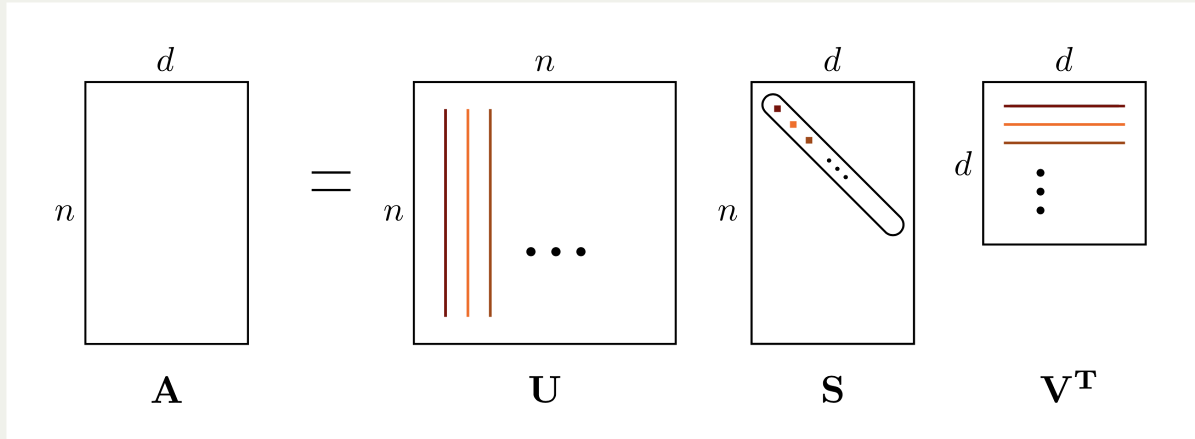
- use the partial observations to estimate this lower-dim structure
- use the lower-dim structure to impute missing elements

Different settings have different structure

- term-document matrices are sparse with lots of 0s (each document has only a small subset of all possible words in the vocabulary)
- movie-ratings matrices are dense but "low-rank"

The rank of a matrix is the number of:

- linearly independent rows
- linearly independent columns
- nonzero singular values



All of these are the same!

Problem setup

We observe a matrix X partially (and perhaps noisily) at locations $(i, j) \in \Omega$

- $\mathbf{X} \in \mathbb{R}^{m \times n}$ is the `true matrix'
- $Y_{ij} = X_{ij} + \epsilon_{ij}$
- $\Omega = \{0, 1\}^{m \times n}$ is a binary masking matrix.

Write $\mathcal{P}_\Omega Y$ for the matrix:

- $[\mathcal{P}_\Omega Y]_{ij} = Y_{ij}$ if $\Omega_{ij} = 1$
- $[\mathcal{P}_\Omega Y]_{ij} = ? \quad$ if $\Omega_{ij} = 0$

Given $\mathcal{P}_\Omega Y$, we want a reconstruction M that is as close as possible to X according to some metric.

Baseline model

Assume $X_{ij} = u_i + v_j$ for vectors $\mathbf{u} \in \mathbb{R}^m, \mathbf{v} \in \mathbb{R}^n$

$\min_{\mathbf{u},\mathbf{v}} \sum_{(i,j)\in\Omega}(Y_{ij} - (u_i + v_j))^2 + \lambda(\|\mathbf{u}\|_2^2 + \|\mathbf{u}\|_2^2)$

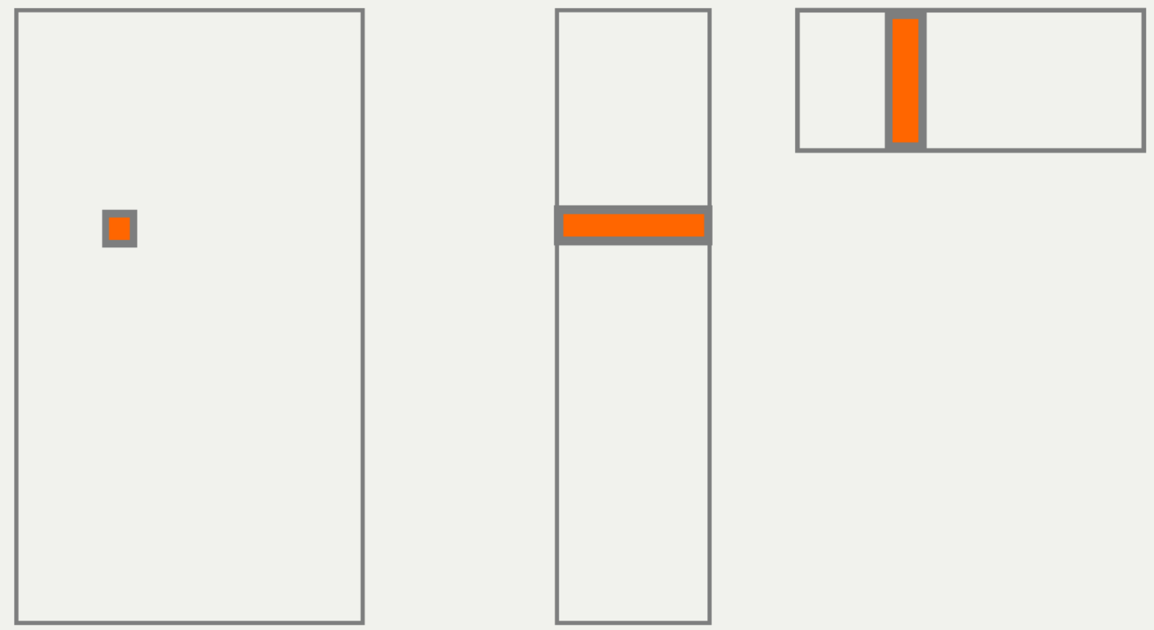
- How many parameters must we estimate?
- How can we interpret these parameters?

Matrix factorized model

The earlier model can be modified as $X_{ij} = u_i v_j$

- This is just a rank-1 approximation

A rank-R approximation takes the form $X_{ij} = \sum_{k=1}^{r} u_{ik} v_{kj}$

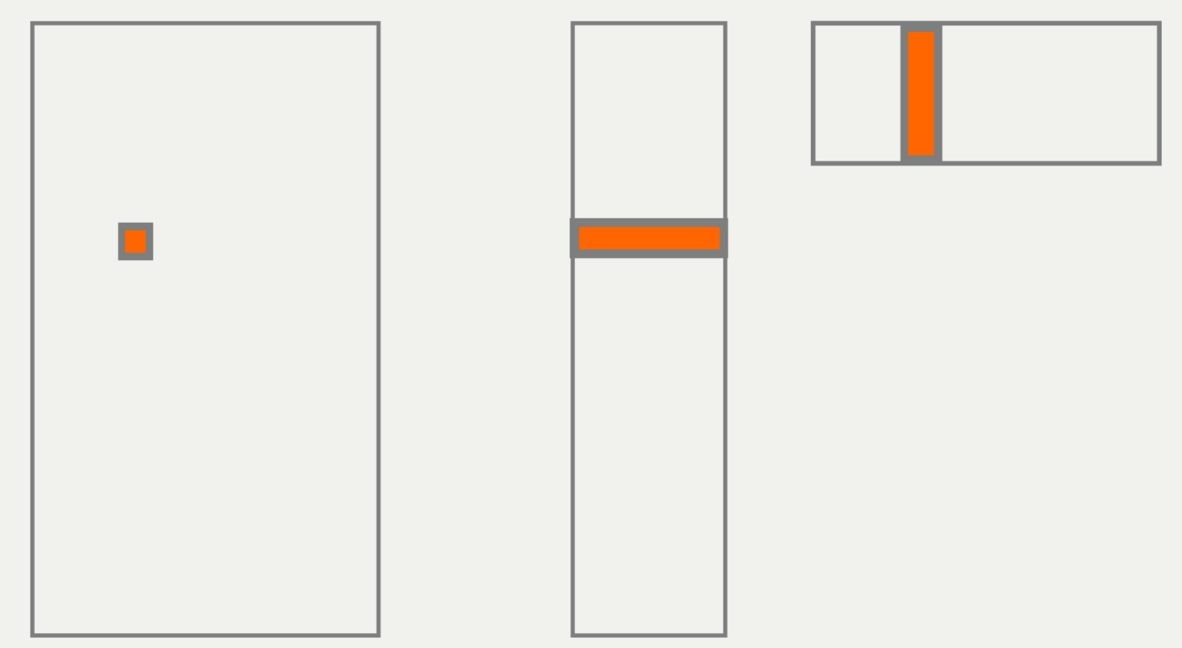


Matrix factorized model

The earlier model can be modified as $X_{ij} = u_i v_j$

- This is just a rank-1 approximation

A rank-R approximation takes the form $X_{ij} = \sum_{k=1}^{r} u_{ik} v_{kj}$



Interpretation:

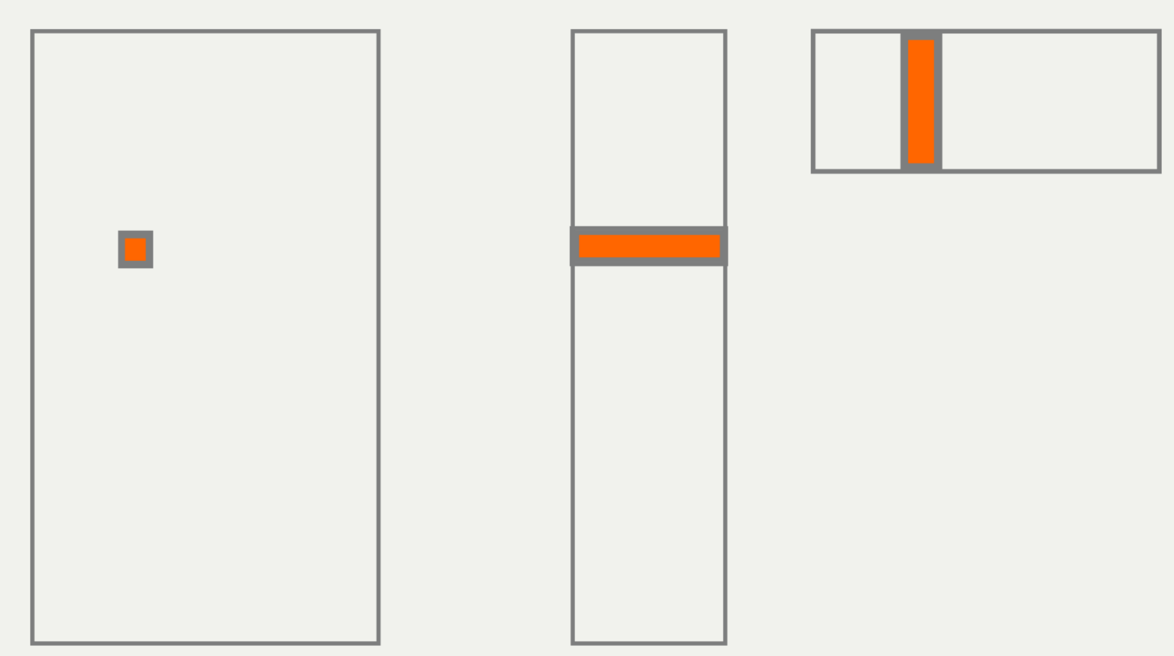
- each row (user) and column (movie) is embedded in an r-dim space
- Element (i, j) is inner-product of the two r-dim feature-vectors

Matrix factorized model

The earlier model can be modified as $X_{ij} = u_i v_j$

- This is just a rank-1 approximation

A rank-R approximation takes the form $X_{ij} = \sum_{k=1}^{r} u_{ik} v_{kj}$



Interpretation:

- each row (user) and column (movie) is embedded in an r-dim space
- Element (i, j) is inner-product of the two r-dim feature-vectors

Can add a link function if e.g. matrix entries are positive

Estimation problem: Find the feature matrices that best explain the observations

- $\min_{\mathbf{U},\mathbf{V}} \sum_{(i,j)\in\Omega}(Y_{ij} - (\mathbf{u}_i^T \mathbf{v}_j))^2$

Represent an $m \times n$ matrix with $mr + rn$ numbers

- For small r, $mr + rm << mn$

Estimation problem: Find the feature matrices that best explain the observations

- $\min_{\mathbf{U},\mathbf{V}} \sum_{(i,j)\in\Omega}(Y_{ij} - (\mathbf{u}_i^T \mathbf{v}_j))^2$

Represent an $m \times n$ matrix with $mr + rn$ numbers

- For small r, $mr + rm << mn$

By itself this problem is not well-posed. Why?

Estimation problem: Find the feature matrices that best explain the observations

- $\min_{\mathbf{U},\mathbf{V}} \sum_{(i,j)\in\Omega}(Y_{ij} - (\mathbf{u}_i^T \mathbf{v}_j))^2$

Represent an $m \times n$ matrix with $mr + rn$ numbers

- For small r, $mr + rm << mn$

By itself this problem is not well-posed. Why?

Typically regularize $\mathbf{U}, \mathbf{V}$

- L_2 penalty is typically, but others can be used (e.g. L_1 gives sparsity)

Given our observations, how do we solve for $\mathbf{U}, \mathbf{V}$?

- $\min_{\mathbf{U},\mathbf{V}} \sum_{(i,j)\in\Lambda}(Y_{ij} - (\mathbf{u}_i^T \mathbf{v}_j))^2 + \lambda(\|\mathbf{U}\|_2 + \|\mathbf{V}\|_2)$ is nonconvex

Given our observations, how do we solve for $\mathbf{U}, \mathbf{V}$?

- $\min_{\mathbf{U},\mathbf{V}} \sum_{(i,j)\in\Lambda}(Y_{ij} - (\mathbf{u}_i^T \mathbf{v}_j))^2 + \lambda(\|\mathbf{U}\|_2 + \|\mathbf{V}\|_2)$ is nonconvex

Alternating minimization:

- Starting with some initialization, solve for U given V and V given U
- Each step is convex and equivalent to solving regularized linear regression

Incoherence (Candes and Recht 2009)

Matrix completion methods typically impose low-rank structure

However, low rank structure is not sufficient

E.g. consider a rank-1 $N \times N$ matrix all of whose elements are 0 except for (1,N)

$$A = \begin{bmatrix} 0 & 0 & \cdots & 1 \\ 0 & 0 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & 0 \end{bmatrix} = \begin{bmatrix} 1 \\ 0 \\ \vdots \\ 0 \end{bmatrix} [0, 0, \cdots, 1]$$

Incoherence (Candes and Recht 2009)

Matrix completion methods typically impose low-rank structure

However, low rank structure is not sufficient

E.g. consider a rank-1 $N \times N$ matrix all of whose elements are 0 except for (1,N)

$$A = \begin{bmatrix} 0 & 0 & \cdots & 1 \\ 0 & 0 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & 0 \end{bmatrix} = \begin{bmatrix} 1 \\ 0 \\ \vdots \\ 0 \end{bmatrix} [0, 0, \cdots, 1]$$

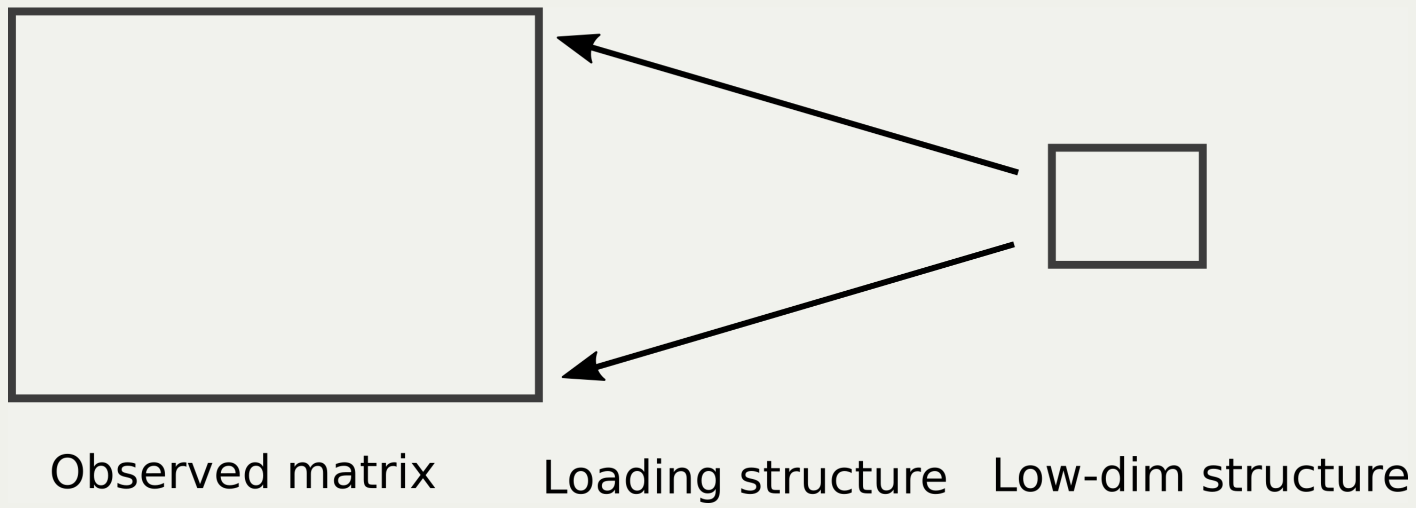
We have no hope of recovering some elements until we actually see them

A term you will often see in *incoherence*

- avoids situations like this

A term you will often see in *incoherence*

- avoids situations like this



Ensures the influence of each element in the matrix is similar

Equivalently, ensures the influence of the low-dimensional structure is spread across many elements of the observed matrix

Also important is the pattern of missingness

E.g. Suppose there are no obsevations in an entire row/column of a matrix

- Can we impute the missing values?

Also important is the pattern of missingness

E.g. Suppose there are no obsevations in an entire row/column of a matrix

- Can we impute the missing values?

Often assume the pattern of missingness is random

- E.g. a Bernoulli coin flip at each element

Also important is the pattern of missingness

E.g. Suppose there are no obsevations in an entire row/column of a matrix

- Can we impute the missing values?

Often assume the pattern of missingness is random

- E.g. a Bernoulli coin flip at each element

Is this realistic?

Rather than decomposing the matrix into U and V and regularizing these, one can directly regularize the reconstructed matrix

$$\operatorname{argmin} \operatorname{rank}(M) \quad s.t. \quad \mathcal{P}_\Omega M = \mathcal{P}_\Omega Y$$

Rather than decomposing the matrix into U and V and regularizing these, one can directly regularize the reconstructed matrix

$$\operatorname{argmin} \operatorname{rank}(M) \quad s.t. \quad \mathcal{P}_\Omega M = \mathcal{P}_\Omega Y$$

- Does not require assuming the rank of the matrix
- Can rigorously bound number of measurements required

Rather than decomposing the matrix into U and V and regularizing these, one can directly regularize the reconstructed matrix

$$\operatorname{argmin}\ \operatorname{rank}(M) \quad s.t. \quad \mathcal{P}_\Omega M = \mathcal{P}_\Omega Y$$

- Does not require assuming the rank of the matrix
- Can rigorously bound number of measurements required

Unfortunately, solving this is NP-hard

Can relax this is a number of ways. A common approach uses the nuclear norm $\|X\|_*$

- $\|X\|_* = \sum \sigma_i(X)$ is the sum of the singular values of X

Can relax this is a number of ways. A common approach uses the nuclear norm $\|X\|_*$

- $\|X\|_* = \sum \sigma_i(X)$ is the sum of the singular values of X

The nuclear norm is the tightest convex envelope of the rank function

Can relax this is a number of ways. A common approach uses the nuclear norm $\|X\|_*$

- $\|X\|_* = \sum \sigma_i(X)$ is the sum of the singular values of X

The nuclear norm is the tightest convex envelope of the rank function

This is a convex problem

- Can be solved in polynomial time use semidefinite programming

Given noisy measurements we can relax this as

$$\min \|X\|_* \quad s.t. \quad \|\mathcal{P}_\Omega X - \mathcal{P}_\Omega Y\| < \delta$$

Given noisy measurements we can relax this as

$$\min \|X\|_* \;\; s.t. \;\; \|\mathcal{P}_\Omega X - \mathcal{P}_\Omega Y\| < \delta$$

In Lagrangian form, this becomes

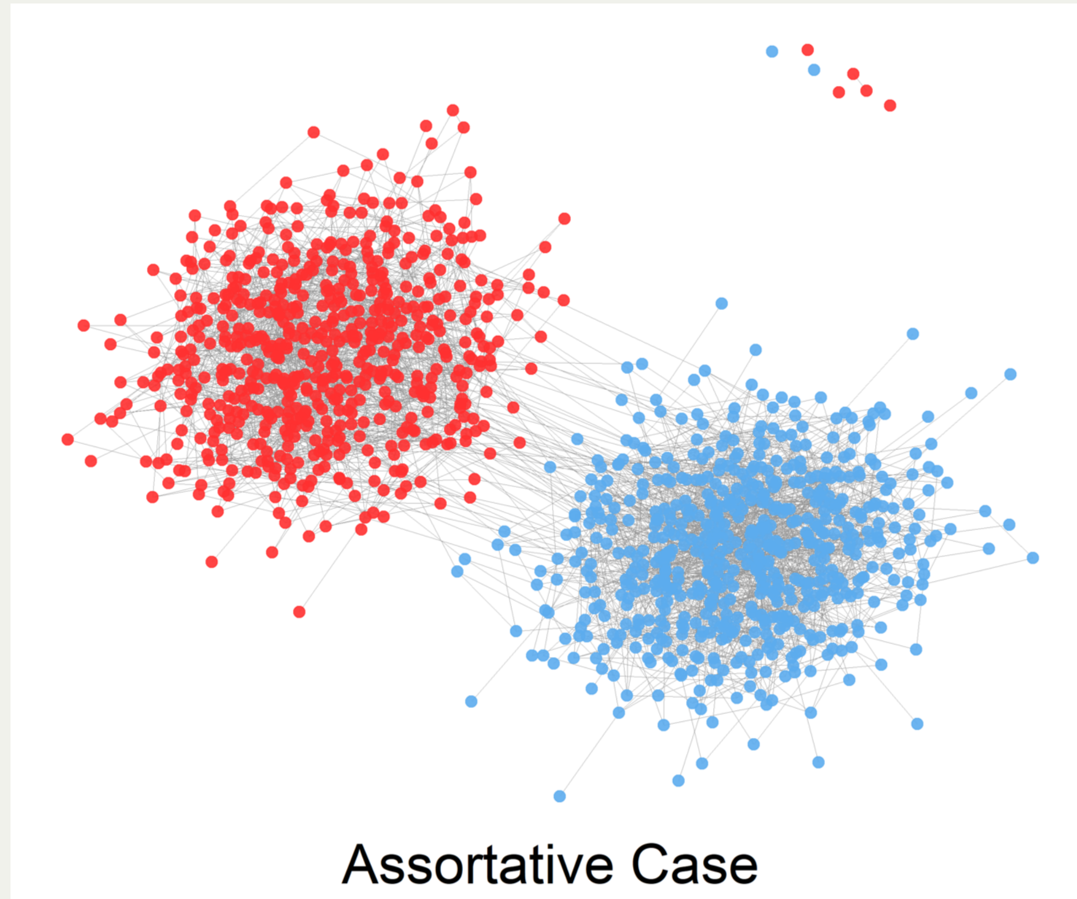
$$\min \|X\|_* + \lambda(\|\mathcal{P}_\Omega X - \mathcal{P}_\Omega Y\|)$$

Stochastic Block Models

Another class of methods proceed by *clustering* rows/columns of the observed matrix

E.g. consider a network of N nodes with an $N \times N$ adjacency matrix A

- $A_{ij} = 1$ if nodes i and j are connected

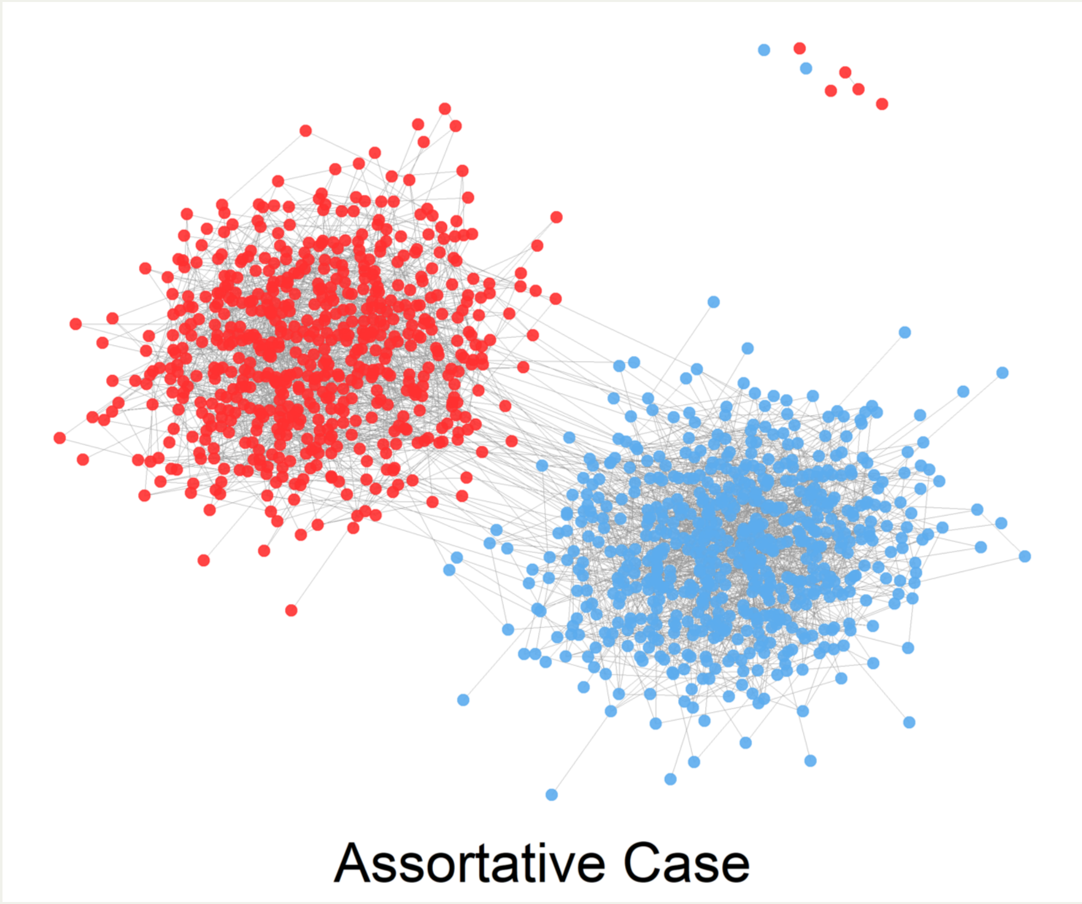


(wikipedia)

Stochastic block models assume each node belongs to 1 of $K << N$ clusters

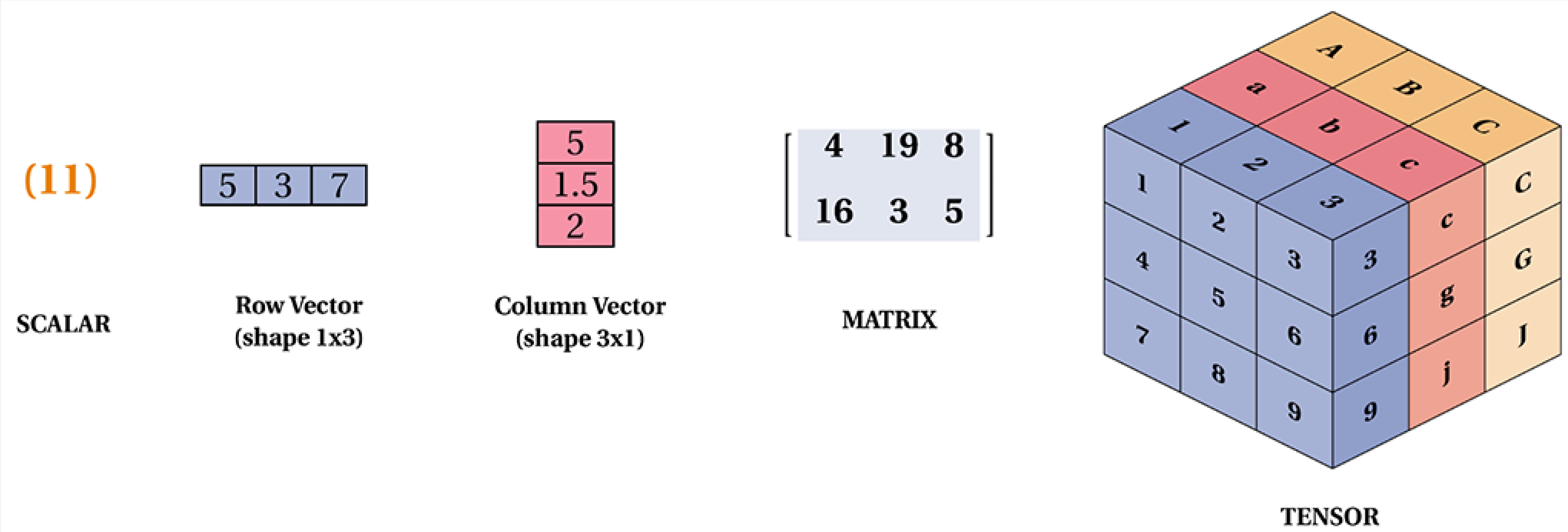
- A $K \times K$ connectivity matrix gives edge probabilities between clusters

Given a partial observation of A, estimate the cluster assignments and connectivity matrix



(wikipedia)

Tensor completion

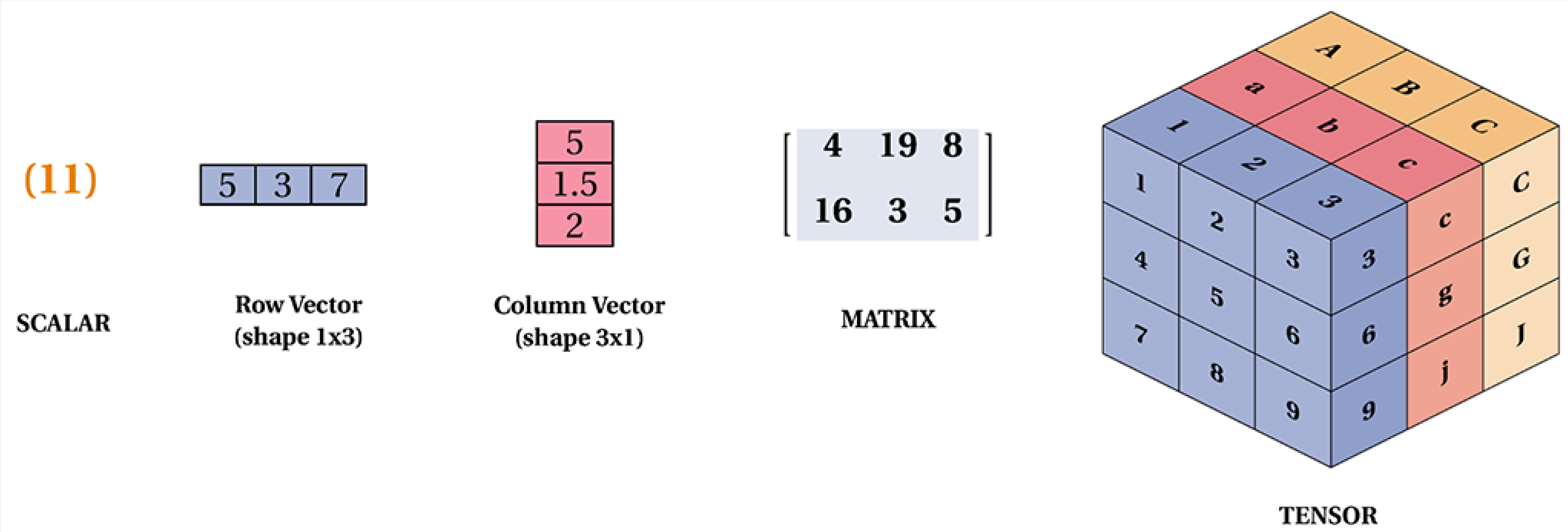


(tensorflownet.readthedocs.io)

Tensors are multidimensional arrays that generalize matrices

- A matrix is a second-order tensor
- For an order-k tensor X, we index elements as $X_{i_1 i_2 \cdots, i_k}$

Tensor completion



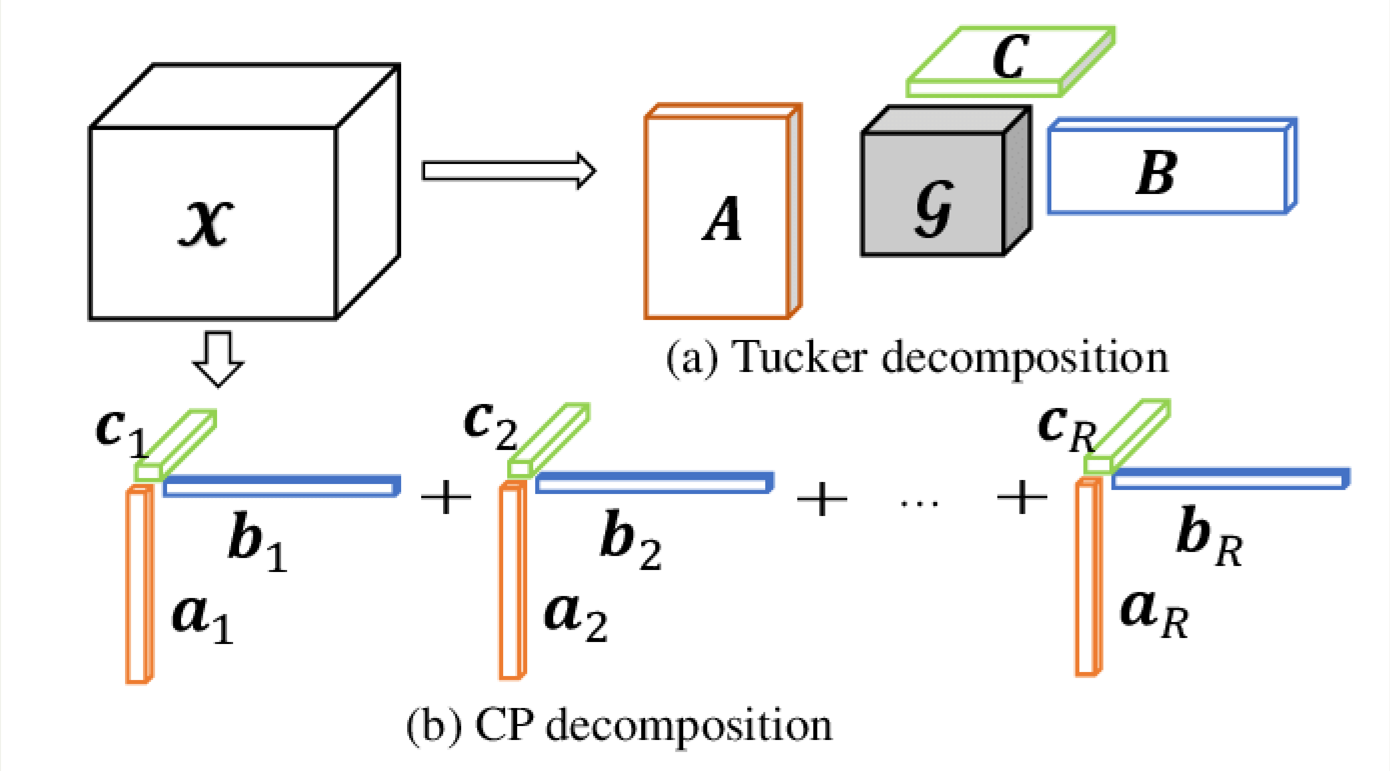
(tensorflownet.readthedocs.io)

Tensors are multidimensional arrays that generalize matrices

- A matrix is a second-order tensor
- For an order-k tensor X, we index elements as $X_{i_1 i_2 \cdots, i_k}$

Applications:

- video data, dynamic graphs
- potential outcome matrices
- probability tables



(Jiang et al 2017)

There are different notions of tensor rank and tensor decomposition

- Tucker decomposition
- PARAFAC decomposition
- Higher-order SVD

  Theory and computation for tensor completion is significantly more challenging

There is a massive and very active literature studying and extending these models to more complex and realistic applications

- Incorporate constraints into the solution (e.g. underlying matrix is positive/positive-definite)
- Incorporate side-information about the rows/columns
- Incorporate more realistic mechanisms for missing data
- Extend to higher-dimensional structures like tensors

There is also a massive literature with an algorithmic focus. Methods include

- Spectral methods
- Dual methods
- Stochastic gradient descent
- EM and MCMC

There is also lots of theoretical work

- Lower bounds on number of samples required for recovery
- Properties of various relaxations
- Convergence properties of various algorithms

Bennett, James, and Stan Lanning. "The netflix prize." Proceedings of KDD cup and workshop. Vol. 2007. 2007.

Candès, Emmanuel J., and Recht, Benjamin. "Exact matrix completion via convex optimization." Foundations of Computational mathematics 9.6 (2009): 717-772.

Jiang, Tai-Xiang, et al. "A novel nonconvex approach to recover the low-tubal-rank tensor data: When t-SVD meets PSSV." arXiv preprint arXiv:1712.05870 (2017).

Athey, Susan, et al. "Matrix completion methods for causal panel data models." Journal of the American Statistical Association 116.536 (2021): 1716-1730.

Athey, Susan, and Guido W. Imbens. "Machine learning methods that economists should know about." Annual Review of Economics 11 (2019): 685-725.

Ramlatchan, Andy, et al. "A survey of matrix completion methods for recommendation systems." Big Data Mining and Analytics 1.4 (2018): 308-323.

Song, Qingquan, et al. "Tensor completion algorithms in big data analytics." ACM Transactions on Knowledge Discovery from Data (TKDD) 13.1 (2019): 1-48.