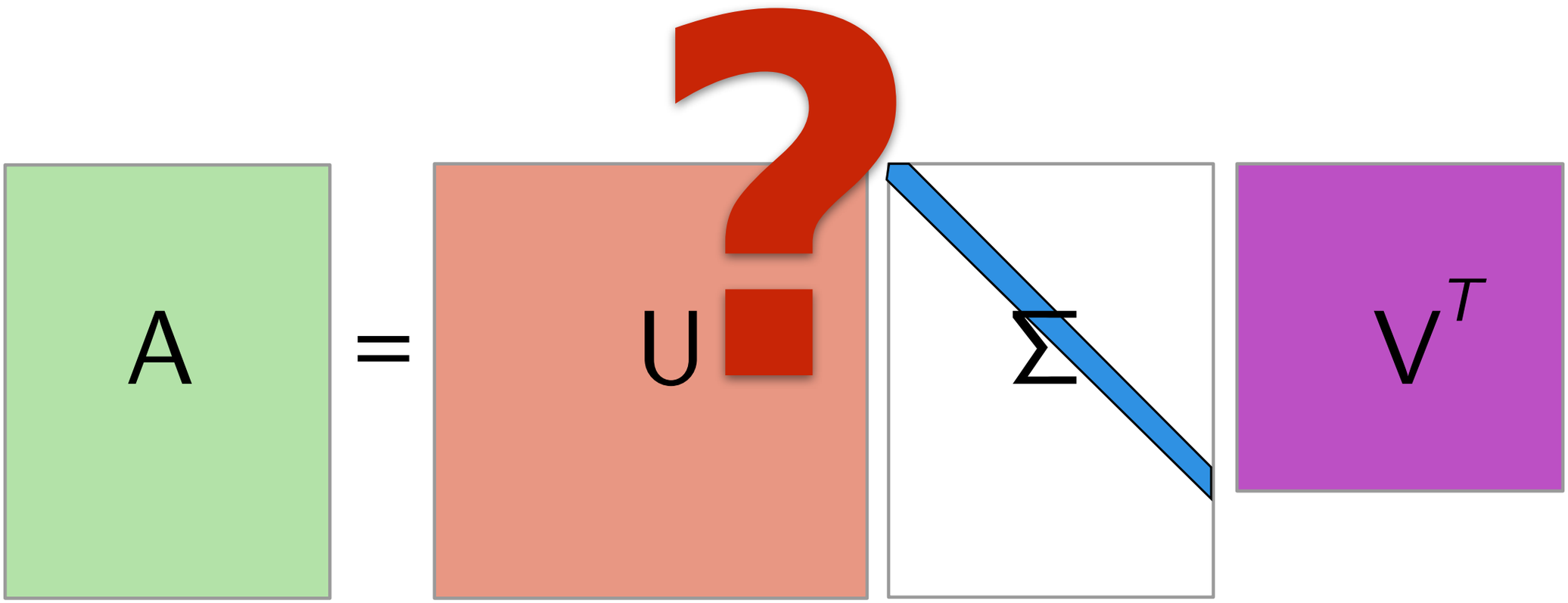
Matrix factorization models for patterns beyond blocks

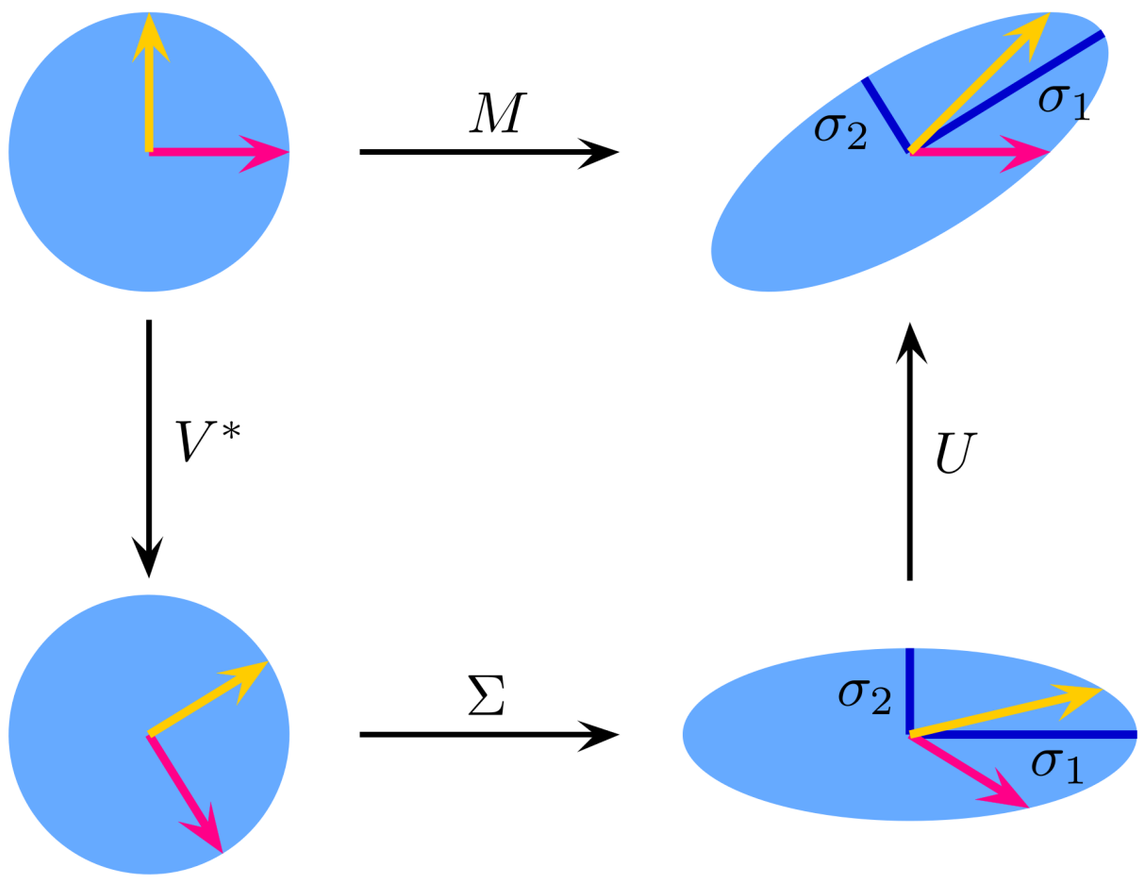
Pauli Miettinen

18 February 2016

What does a matrix factorization do?



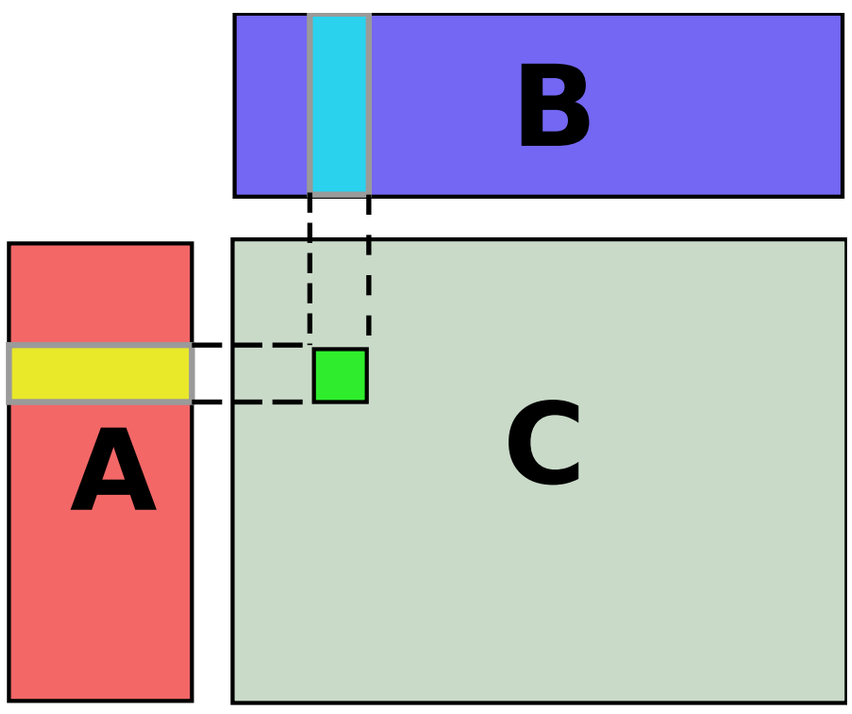
For SVD that's easy!



$$M = U \cdot \Sigma \cdot V^*$$

Inner-product interpretation

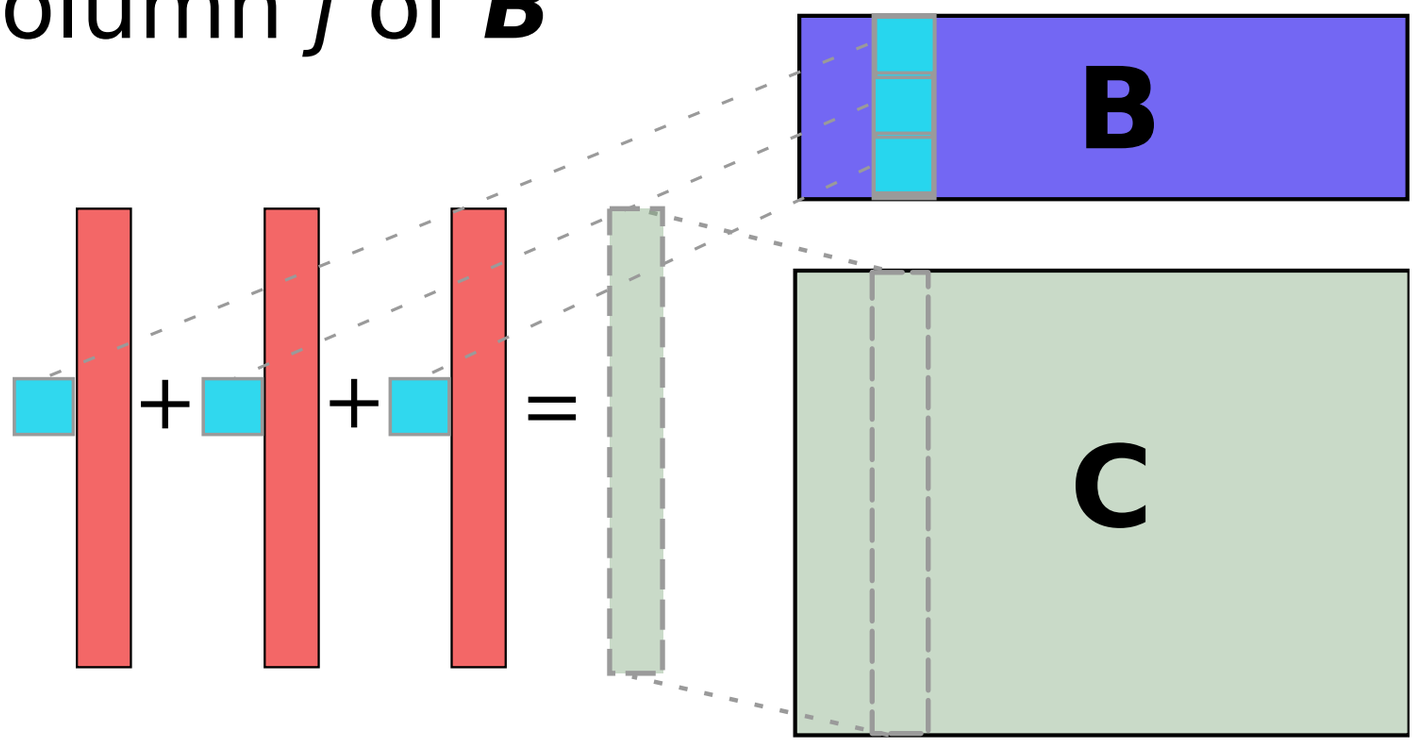
- Element $(\boldsymbol{AB})_{ij}$ is the inner product of row i of $\boldsymbol{A}$ and column j of $\boldsymbol{B}$



$$\boldsymbol{c}_{ij} = \sum_{\ell=1}^{k} a_{i\ell} b_{\ell j}$$

Linear combination interpretation

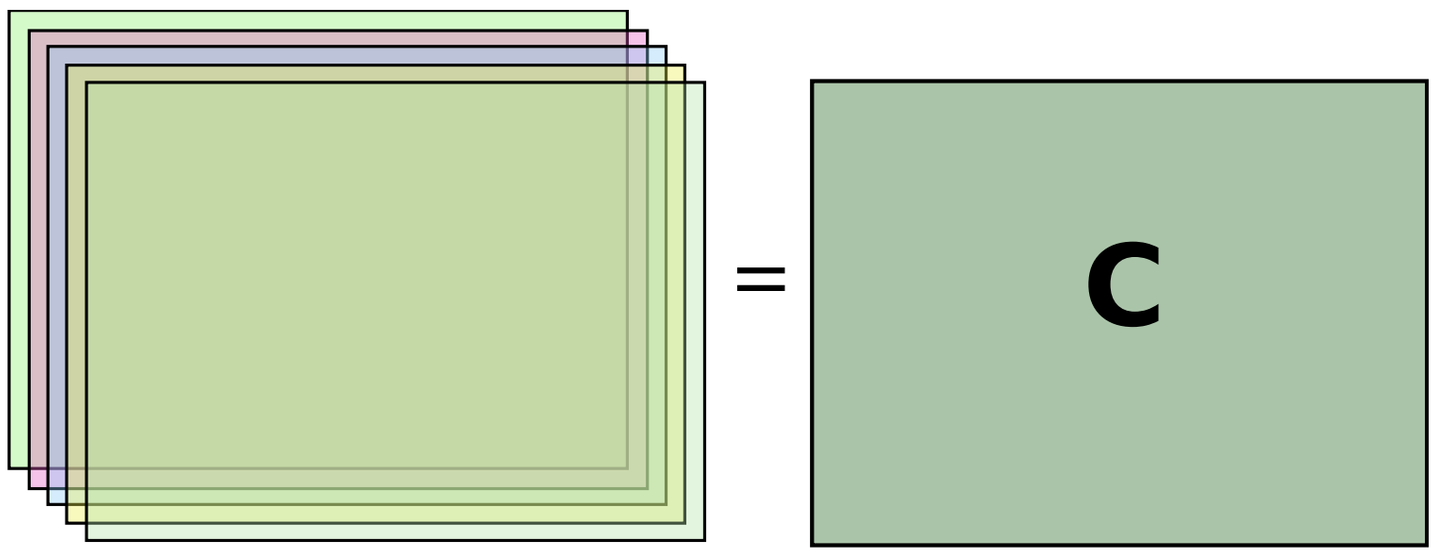
- Column *j* of ***AB*** is the linear combination of columns of ***A*** with the coefficients coming from column *j* of ***B***



$$\boldsymbol{C} = \left[\left[\sum_{\ell=1}^{k} b_{\ell 1} \boldsymbol{a}_\ell \right] \quad \left[\sum_{\ell=1}^{k} b_{\ell 2} \boldsymbol{a}_\ell \right] \cdots \left[\sum_{\ell=1}^{k} b_{\ell m} \boldsymbol{a}_\ell \right] \right]$$

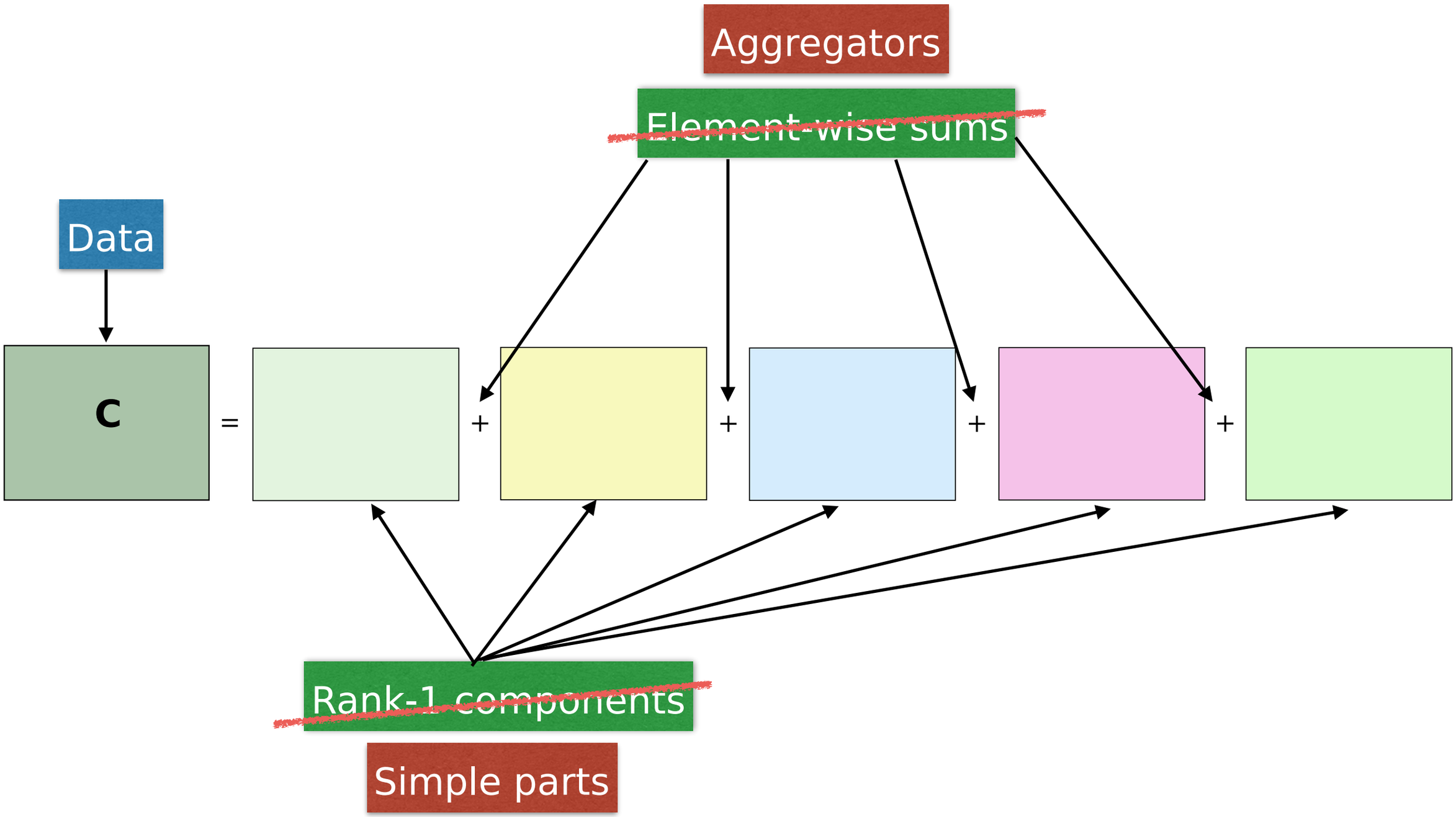
Component-wise interpretation

- Matrix $\boldsymbol{AB}$ is a sum of k matrices $\boldsymbol{a}_l\boldsymbol{b}_l^T$ obtained by multiplying the l-th column of $\boldsymbol{A}$ with the l-th row of $\boldsymbol{B}$



$$\boldsymbol{C} = \sum_{\ell=1}^{k} \boldsymbol{a}_\ell \boldsymbol{b}_\ell^T$$

Component-wise



On sums

- The summation operation in matrix factorization is just a type of aggregation function
- Other exist as well, and can be used:
  - Boolean OR, max, min, Łukasiewicz disjunction, ...
- **How you aggregate defines what kind of patterns you need to summarize the matrix**

Example: Subtropical Algebras

- A **subtropical algebra** is a semiring over the non-negative reals with the addition being the max-operator
  - A.k.a. max-times algebra
  - Related to the **tropical algebra** ($\boldsymbol{R} \cup \{-\infty\}$, max, +) (a.k.a. max-plus algebra)

S. Karaev & P.M. *Capricorn: An Algorithm for Subtropical Matrix Factorization*, SDM ’16

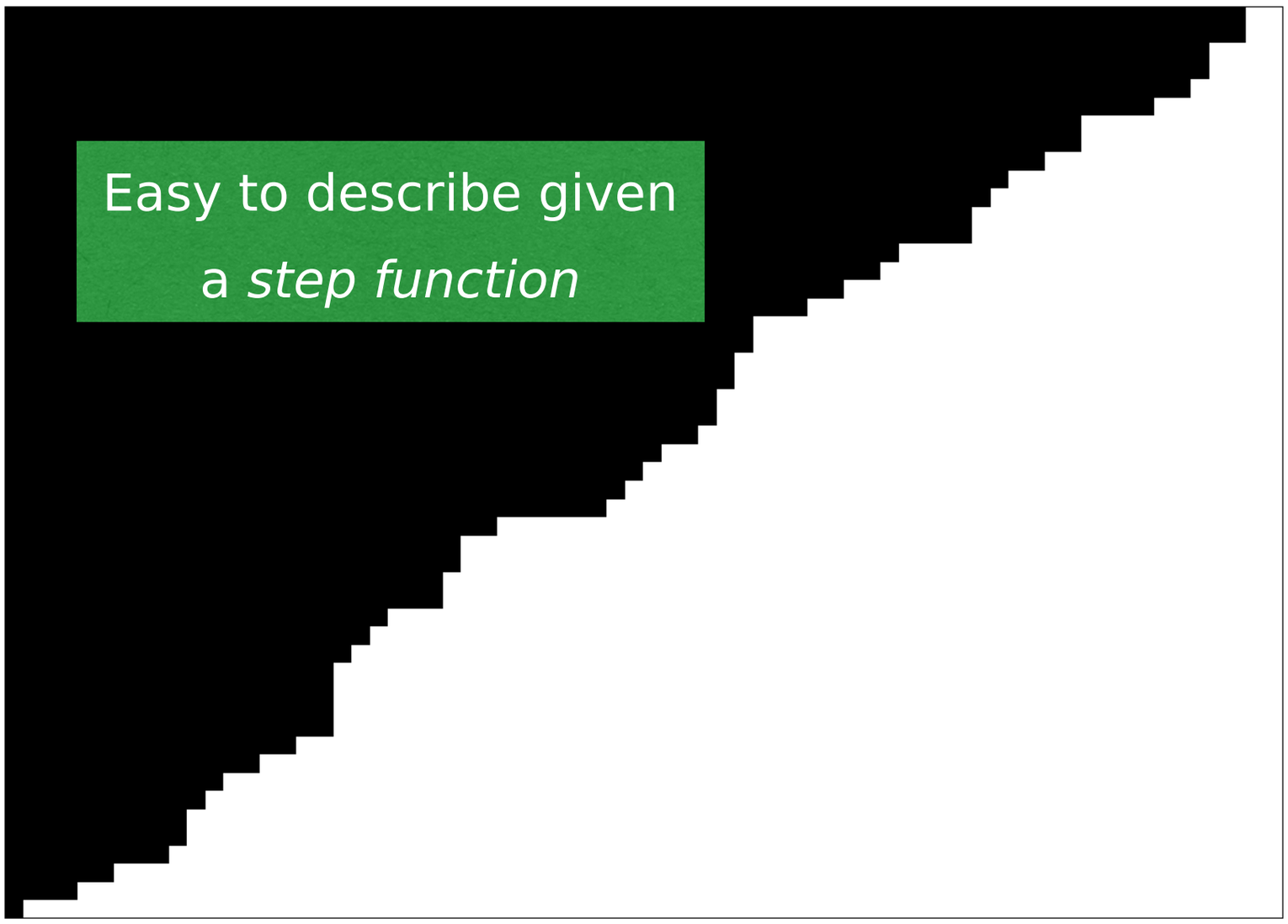
Intuition

- Nonnegative matrix factorization (NMF) gives "parts of whole" interpretation of the data
- Subtropical algebra gives "winner takes it all" interpretation

$$\max_{\ell=1}^{k}\{\boldsymbol{A}_{i\ell}\boldsymbol{B}_{\ell j}\}$$

- The largest element determines the aggregate value

Simple(?) parts



On products

- The vector outer product can be re-defined to handle all kinds of “simple to describe” matrices
  - Call these **rank-1** matrices
- **The type of outer product determines the shape of your patterns**

P.M. *Generalized Matrix Factorizations as a Unifying Framework for Pattern Set Mining: Complexity Beyond Blocks*, ECMLPKDD ’15

Generalized outer products

- Rank-1 matrix = outer product of two vectors
  - $\boldsymbol{A} = \boldsymbol{x}\boldsymbol{y}^T$
- Define **generalized outer product**

$$o(\boldsymbol{x}, \boldsymbol{y}, \theta) \in \mathbb{R}^{n \times m}$$

Vectors

Parameters

  - $o(\boldsymbol{x}, \boldsymbol{y}, \theta)_{ij} = x_i y_j$ or 0

Example: biclique core

$$o\left(\begin{bmatrix}1\\1\\1\\1\\1\end{bmatrix}, [\,1\;1\;1\;1\;1\,], \{1,2\}\right) = \begin{bmatrix}0&0&1&1&1\\0&0&1&1&1\\1&1&0&0&0\\1&1&0&0&0\\1&1&0&0&0\end{bmatrix}$$

Rows that belong to the pattern

Columns that belong to the pattern

The core

Generalized decompositions

- Recall, $\boldsymbol{X} \approx \boldsymbol{AB} = \boldsymbol{a}_1\boldsymbol{b}_1^T + \boldsymbol{a}_2\boldsymbol{b}_2^T + \cdots + \boldsymbol{a}_k\boldsymbol{b}_k^T$ is a decomposition of $\boldsymbol{X}$
- The **generalized decomposition** of $\boldsymbol{X}$ is

$$\boldsymbol{X} \approx \boldsymbol{F}_1 \boxplus \boldsymbol{F}_2 \boxplus \cdots \boxplus \boldsymbol{F}_k, \quad \boldsymbol{F}_i = o(\boldsymbol{x}_i, \boldsymbol{y}_i, \theta_i)$$

  - $\boxplus$ is the addition in the underlying algebra
    - sum, AND, OR, XOR, ...

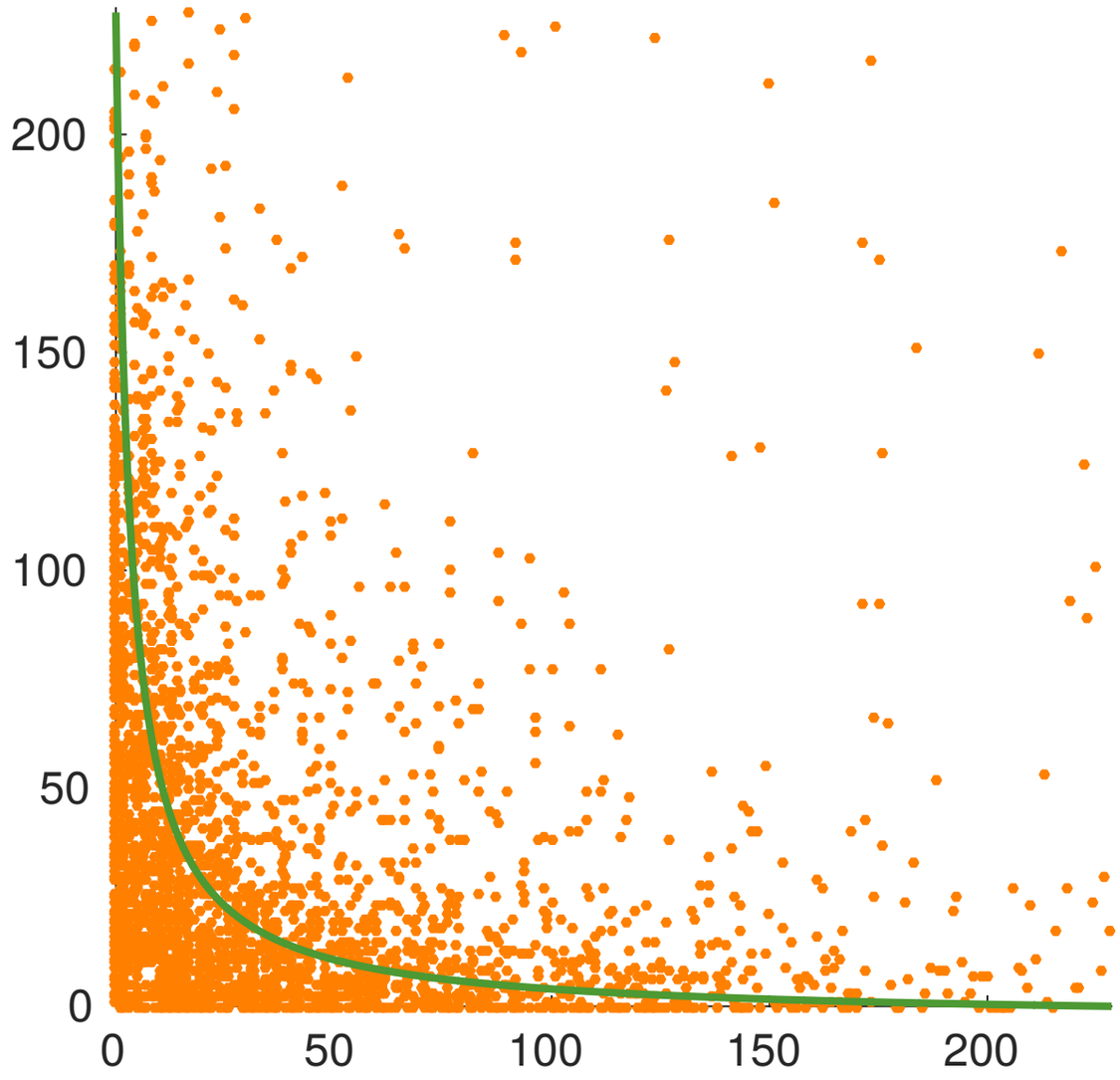
How hard can it be…

- …to find the maximum-circumference pattern?
- I.e. given $\boldsymbol{A}$, find $\boldsymbol{x}$, $\boldsymbol{y}$, and θ s.t. $o(\boldsymbol{x}, \boldsymbol{y}, \theta) \in \boldsymbol{A}$ and you maximize $|\boldsymbol{x}| + |\boldsymbol{y}|$
  - If o is hereditary and the pattern can have infinitely many distinct rows and columns, NP-hard
  - If there's only fixed number of distinct rows or columns, the problem is in P
  - If $\boldsymbol{x} = \boldsymbol{y}$ is required, then it's almost always NP-hard

How hard can it be…

- …to select the smallest subset that gives an exact summarization?
- I.e. given a set $S = \{\boldsymbol{F}_i : \text{rank}(\boldsymbol{F}_i) = 1\}$, $\boxplus_{\boldsymbol{F} \in S} \boldsymbol{F} = \boldsymbol{X}$, find the the smallest $C \subseteq S$ s.t. $\boxplus_{\boldsymbol{F} \in C} \boldsymbol{F} = \boldsymbol{X}$
  - NP-hard for $\boxplus \in \{\text{AND, OR, XOR}\}$
  - hard to approximate within $\ln(n)$ for OR and within superpolylogarithmic for XOR

Example: Hyperbolic blocks

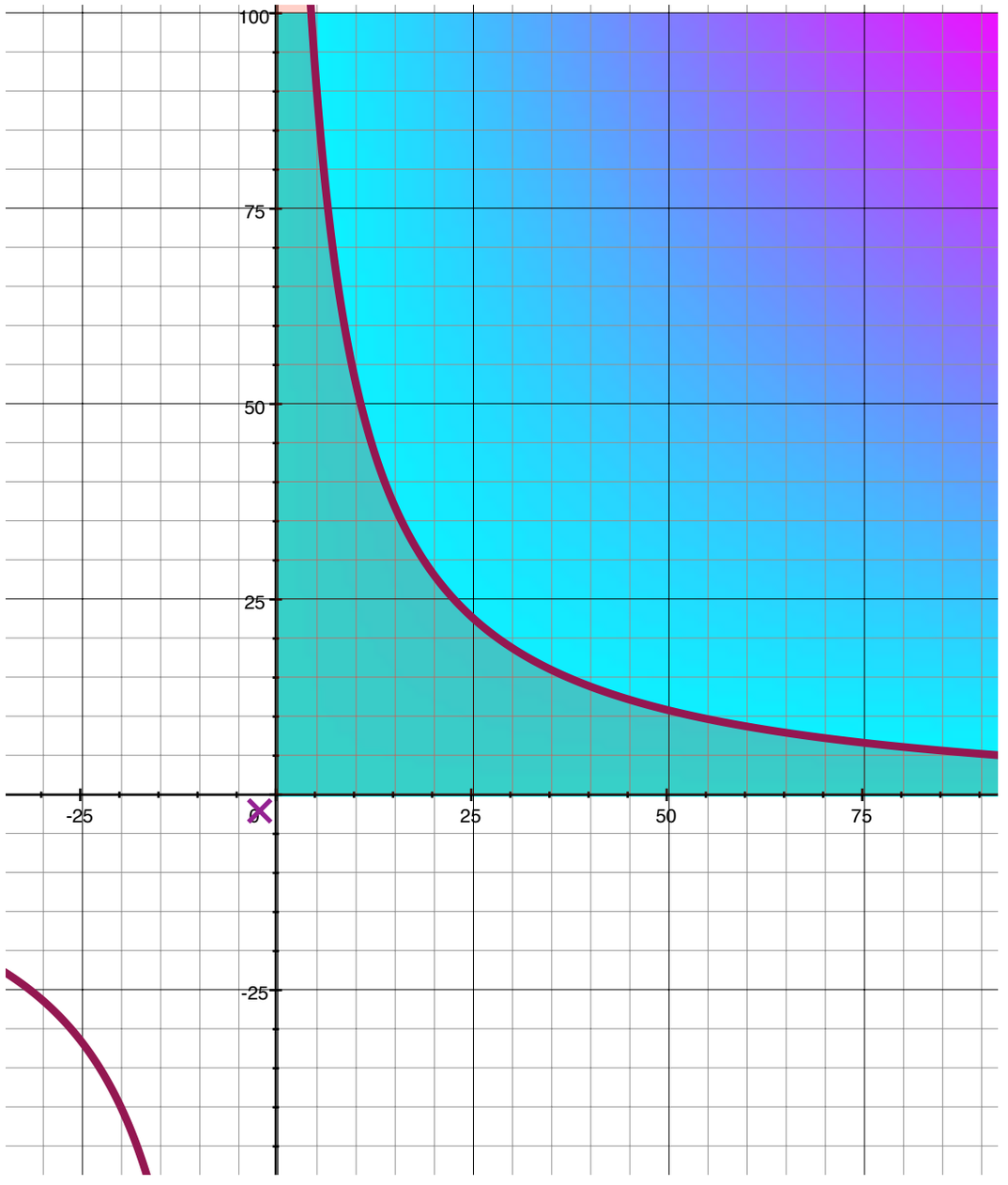


S. Metzler, S. Günnemann & P.M. *Hyperbolae Are No Hyperbole: Modelling Communities that Are Not Cliques*, arXiv, 2016

Hyperbolic communities

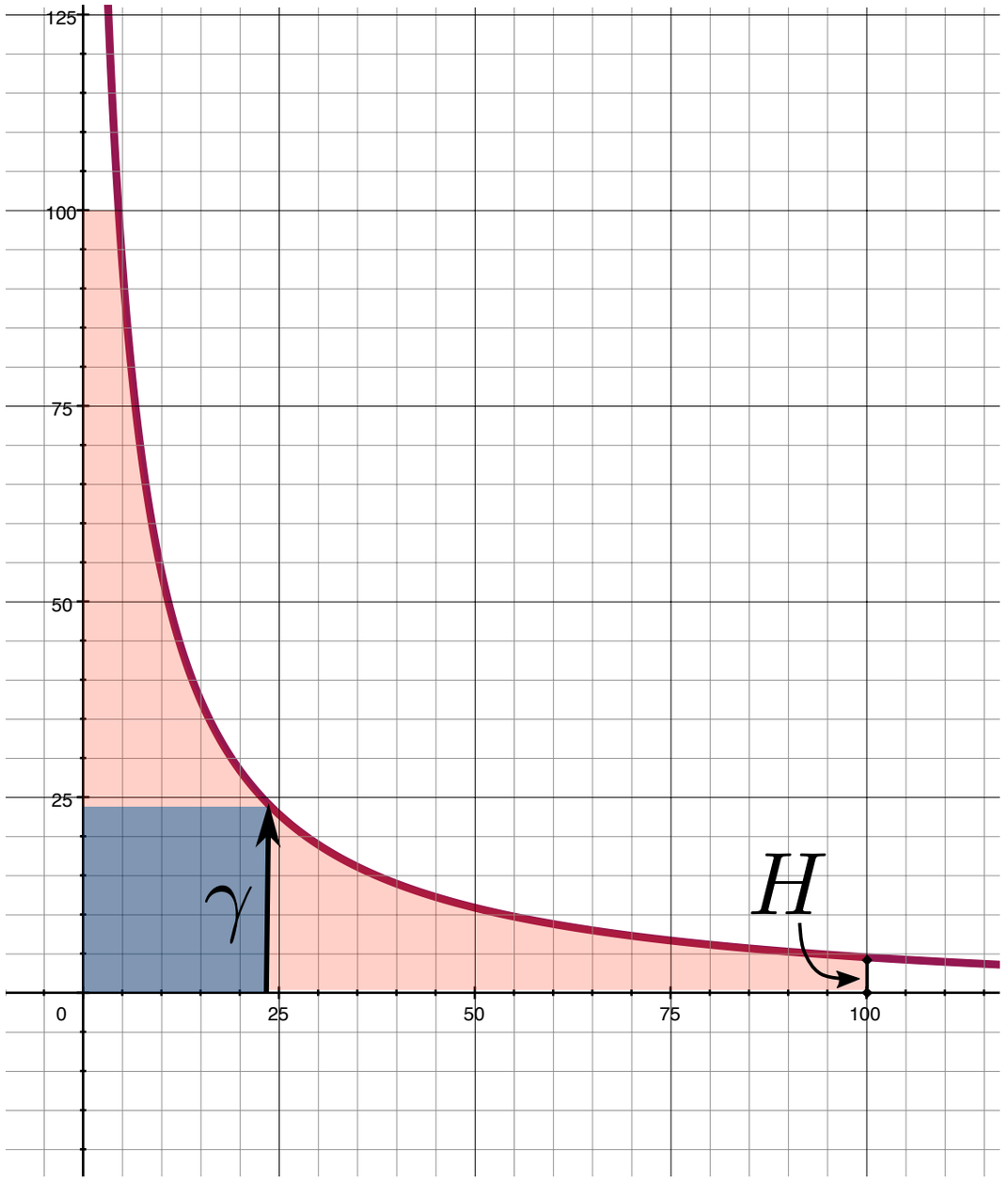
- Most communities have more structure than in a clique
  - Not all edges are equally probable
- We model these using a hyperbola

$$\boldsymbol{A}_{ij} = [(i + p)(j + p) \leq \theta]$$



An alternative model

- Fix a **core size** and **tail height**
  - Core size γ = the point where the curve passes the diagonal (i, i)
  - Tail height H = the point at which the curve exits the community
- p and θ can be computed from γ and H, and vice versa



Outer product formulation

- Generalized outer product is natural via the hyperbola's equation

$$\boldsymbol{A}_{ij} = [(i + p)(j + p) \leq \theta]$$

- Hence, given parameters, finding the largest community is NP-hard (because clique)
- Given the subgraph, finding the parameters is easy
- Given a collection of communities, selecting some of them is hard

The link function

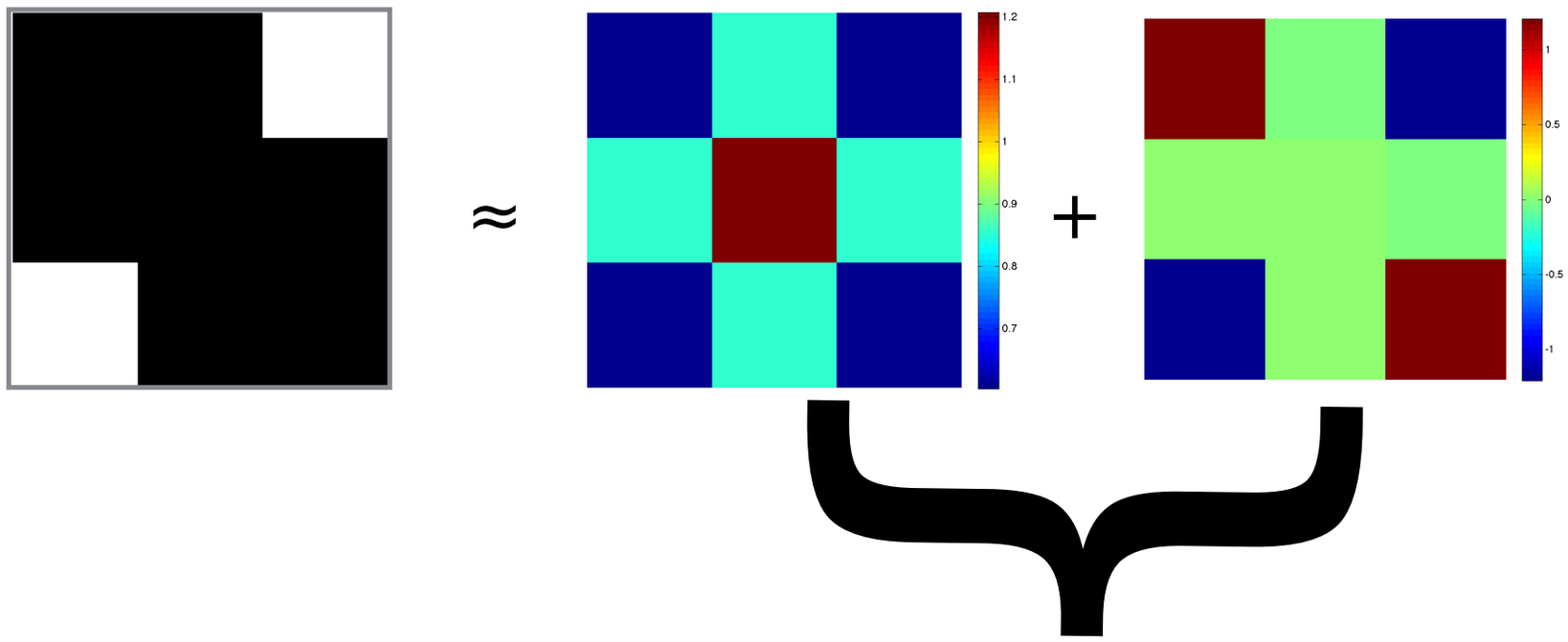
- Most common way of doing a “non-linear matrix factorization” is to apply a **link function** to the product
  - $f(\boldsymbol{AB}) = \boldsymbol{C}$
  - In *generalized linear models*, f is the link between the linear model and the non-linear response
  - E.g. the logistic function $(1 + \exp(-\boldsymbol{AB}))^{-1}$

The threshold link function

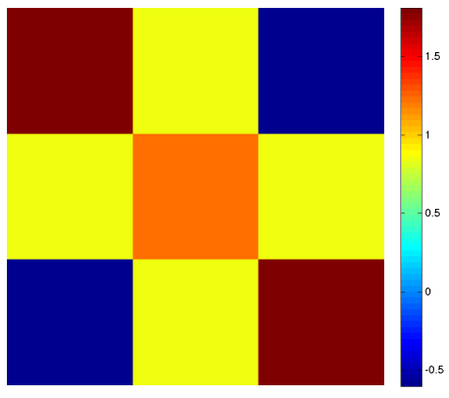
- $\mathrm{thr}_\tau(\boldsymbol{A}) = (\mathrm{thr}_\tau(a_{ij}))$, where $\mathrm{thr}_\tau(x) = \begin{cases} 0 & \text{if } x < \tau \\ 1 & \text{if } x \geq \tau \end{cases}$
  - The $\mathrm{sign}(x)$ is special variant of this
- How does such a link function behave?

Joint work with Rainer Gemulla & Stefan Neumann, including discussions with Shay Moran

An example



$$\begin{pmatrix} 1.10 & 0.85 & 0.10 \\ 0.85 & 1.21 & 0.85 \\ 0.10 & 0.85 & 1.10 \end{pmatrix}$$



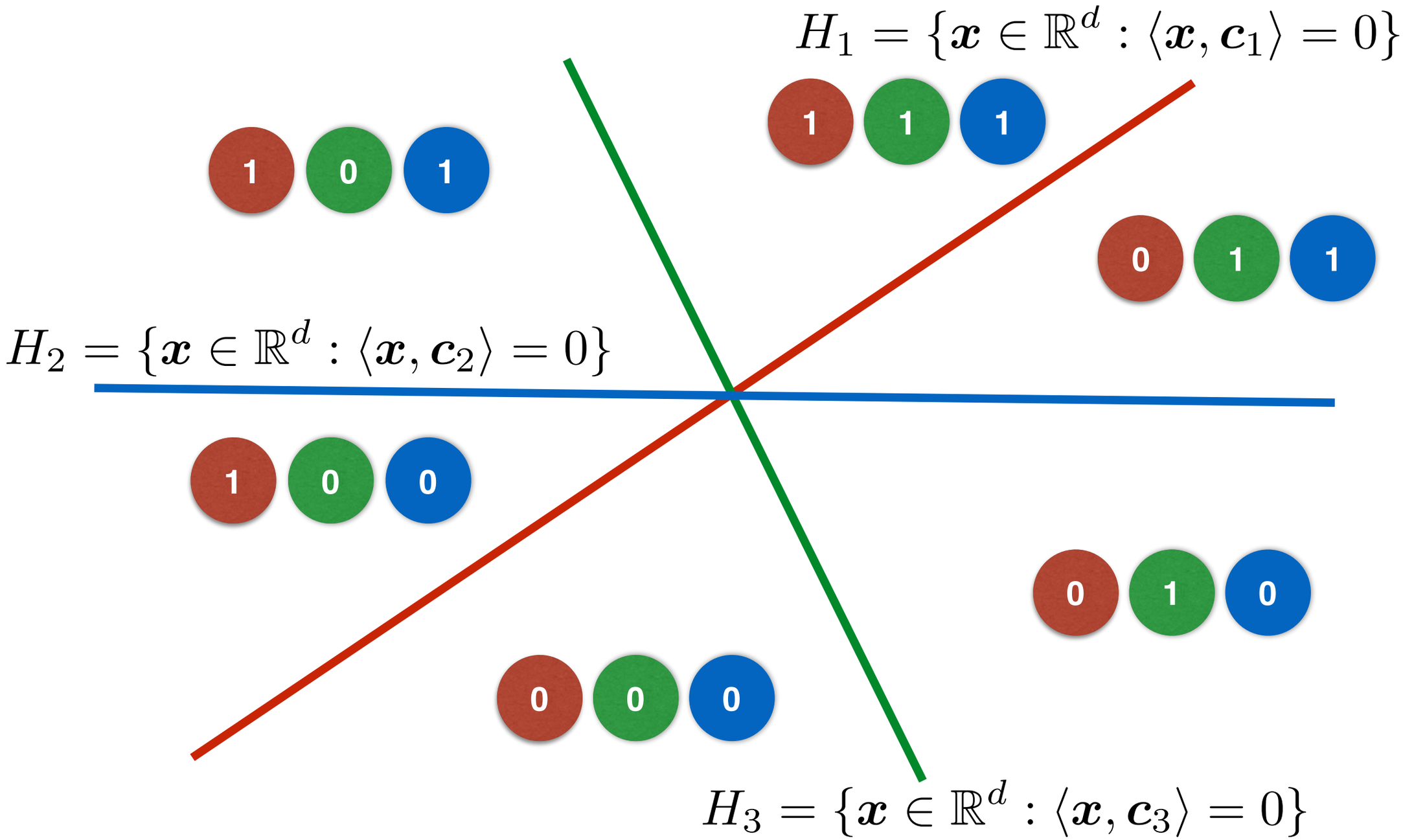
Why link/threshold?

- The link function encodes our knowledge of the data
  - The statistics, the distribution
    - Or just that the data is binary
- Understanding the threshold link gives us new type of simple patterns
  - And has lots of connections to other fields...

Rounding rank

- The **rounding rank** of a 0/1 matrix $\boldsymbol{B}$ (w.r.t. threshold τ) is the least k s.t. there exists rank-k real-valued matrix $\boldsymbol{A}$ for which $\text{thr}_\tau(\boldsymbol{A}) = \boldsymbol{B}$.
- The **sign rank** of a –1/1 matrix $\boldsymbol{B}$ is its rounding rank w.r.t. $\tau = 0$ (*mutatis mutandis*)

Geometric interpretation



Another example

$$\begin{pmatrix} 1 & 1 & 1 \\ 1 & 1 & 0 \\ 1 & 0 & 0 \end{pmatrix}$$

$$\begin{pmatrix} 4 \\ 2 \\ 1 \end{pmatrix} \begin{pmatrix} 1 & 1/2 & 1/4 \end{pmatrix} = \left(\begin{pmatrix} 4 & 2 & 1 \\ 2 & 1 & 1/2 \\ 1 & 1/2 & 1/4 \end{pmatrix} \right) = \begin{pmatrix} 1 & 1 & 1 \\ 1 & 1 & 0 \\ 1 & 0 & 0 \end{pmatrix}$$

Matrix is nested if and only if it has (nonnegative) rounding rank 1

Some comments

- Computing the rounding rank is NP-hard, but can be outside NP
  - For some matrices ***B***, the witness ***A*** has doubly exponential values
- Sign rank 3 is equivalent to the *existential theory of the reals*
- Changing the rounding threshold changes the rank by at most 1
  - rounding rank ≤ sign rank ≤ rounding rank + 1
- Requiring non-negative factor matrices increases the rank by at most 2

Potential algorithms

- Truncated, rounded SVD
  - Bad for computing the rank, good for fixed rank
- Nuclear norm optimization
  - Not very good
- Logistic PCA
  - Very good, but slow
- Randomly project to a subspace, then solve linear program
  - Fast, OK on rank and error

Conclusions

- Matrix factorizations are sort of mixture models
  - Aggregations of simpler parts
- How do you aggregate, and what is simple, can be defined differently to find different patterns
  - Sub-tropical algebras, hyperbolic blocks, …
  - Generalized outer products allow generalizing many existing results
- Rounding rank is connected to many interesting problems in data analysis and machine learning