BINARY MATRIX FACTORISATIONS

Pauli Miettinen

Tutorial @ ECML PKDD 2012

max planck institut
informatik

In the sleepy days when the provinces of France were still quietly provincial, matrices with Boolean entries were a favored occupation of aging professors at the universities of Bordeaux and Clermont-Ferrand. But one day…

Gian-Carlo Rota
Foreword to *Boolean matrix theory and applications* by K. H. Kim, 1982

PART I
DEFINITIONS AND THEORY

CONTENTS

MOTIVATING EXAMPLE

Images by John Tenniel, openclipart.org, and Wikipedia

max planck institut
informatik

TABLE OF FEATURES

long-haired	✔	✔	✘
well-known	✔	✔	✔
male	✘	✔	✔

BINARY MATRIX

long-haired	1	1	0
well-known	1	1	1
male	0	1	1

BOOLEAN MATRIX FACTORISATION

Alice & Bob: long-haired and well-known

Bob & Charles: well-known males

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

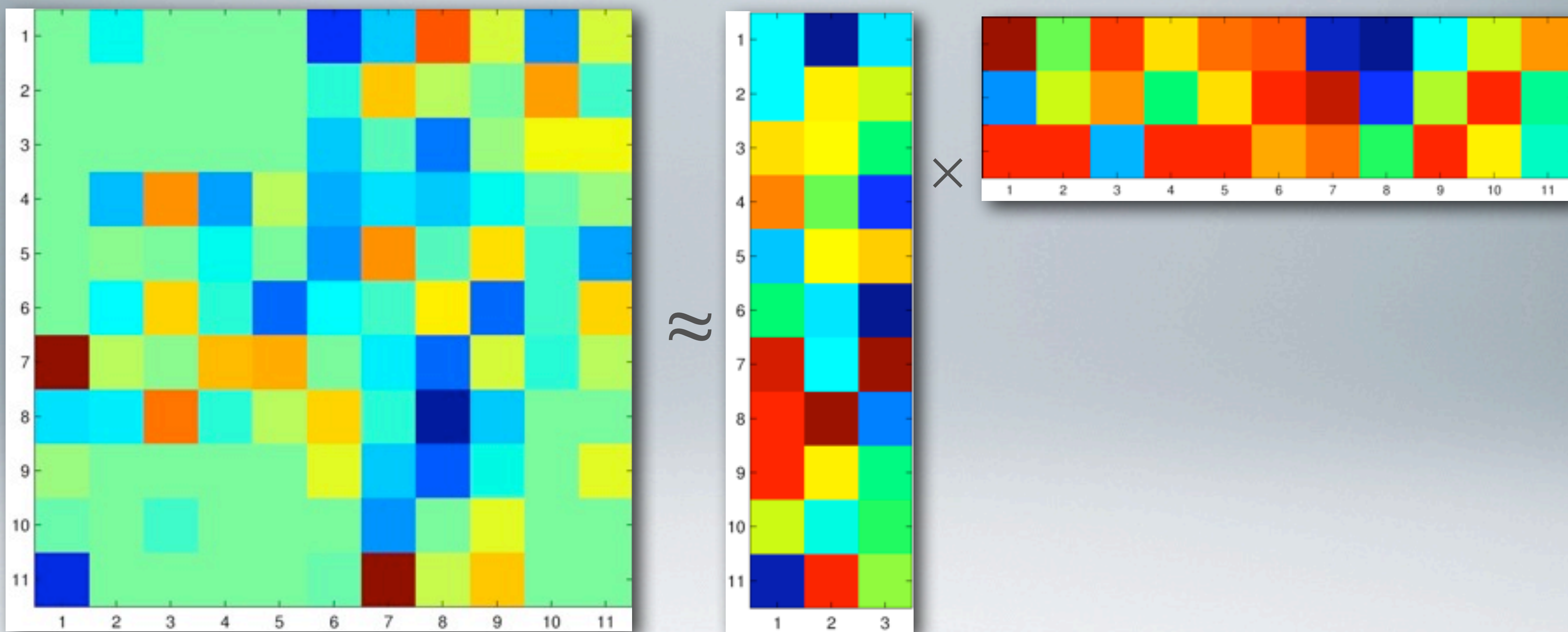
				A	B	C
long-haired	1	0		1	1	0
well-known	1	1	∘	0	1	1
male	0	1				

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

MODULO-2 EXAMPLE

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

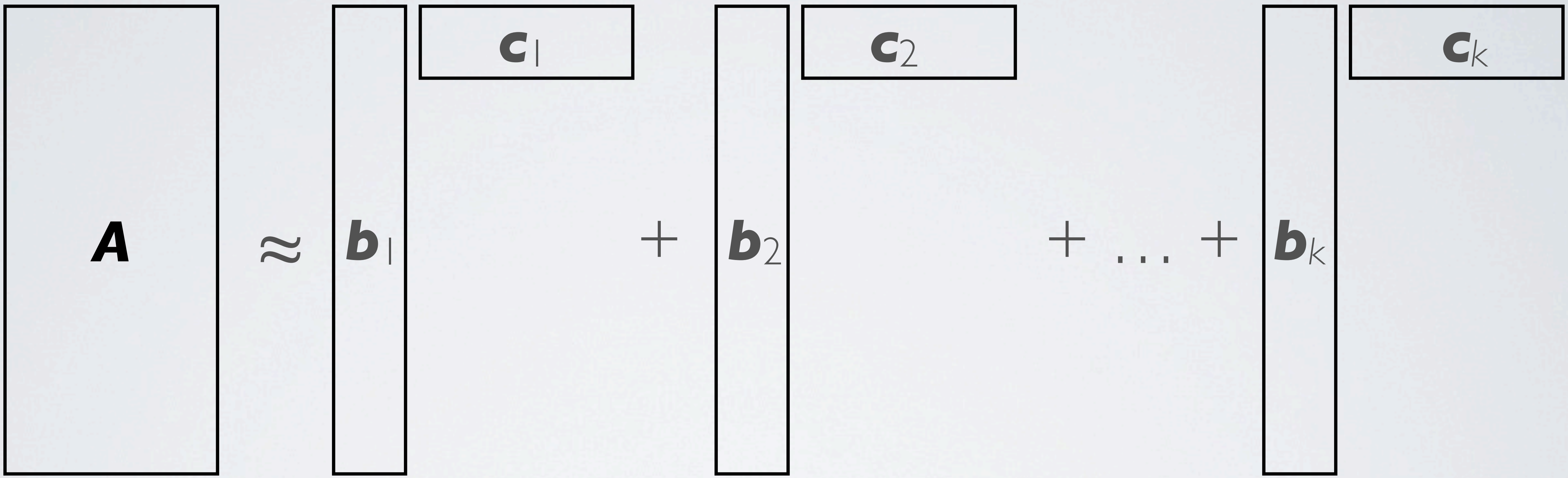
MATRIX FACTORISATIONS



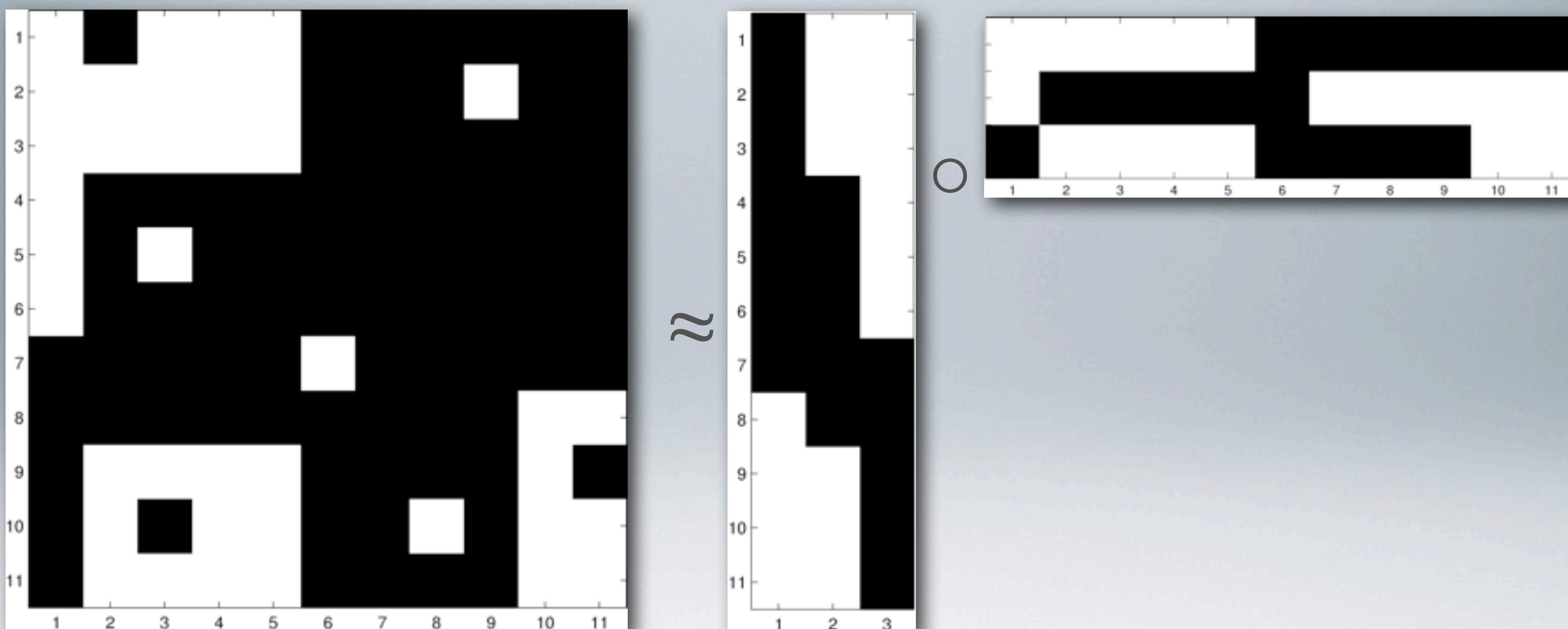
DEFINITION

- A **factorisation** of matrix $\boldsymbol{A}$ represents it as a product of two (or more) **factor matrices**: $\boldsymbol{A} = \boldsymbol{BC}$
  - $\boldsymbol{A}$ is n-by-m, $\boldsymbol{B}$ is n-by-k, and $\boldsymbol{C}$ is k-by-m
    - k is the **size** (or **rank**) of the factorisation
- Factorisation can be **exact** ($\boldsymbol{A} = \boldsymbol{BC}$) or **approximate** ($\boldsymbol{A} \approx \boldsymbol{BC}$)

K RANK-1 FACTORISATIONS



BINARY MATRIX FACTORISATIONS



BINARY MATRIX FACTORISATIONS

- All involved matrices (***A***, ***B***, and ***C***) are binary (0/1)
- Loss function is sum of absolute differences

$$|\mathbf{A} - \mathbf{B}\times\mathbf{C}| = \sum_{ij} |a_{ij} - (\mathbf{B}\times\mathbf{C})_{ij}|$$

  - Or squared Frobenius
- The **algebra** is different for different factorisations
  - We consider normal, modulo-2, and Boolean algebras

max planck institut informatik

NORMAL ALGEBRA

Binary matrix factorisation under $\mathbb{R}$ (RMF).
Given an *n*-by-*m* binary matrix $\boldsymbol{A}$ and integer *k*, find *n*-by-*k* and *k*-by-*m* binary matrices $\boldsymbol{B}$ and $\boldsymbol{C}$ such that $|\boldsymbol{A} - \boldsymbol{B}\times\boldsymbol{C}|$ is minimised.

- Algebra is normal ($1+1 = 2$)
  $\Rightarrow \boldsymbol{B}\times\boldsymbol{C}$ is not necessarily binary

BOOLEAN ALGEBRA

Boolean matrix factorisation (**BMF**).
Given an *n*-by-*m* binary matrix $\boldsymbol{A}$ and integer *k*, find *n*-by-*k* and *k*-by-*m* binary matrices $\boldsymbol{B}$ and $\boldsymbol{C}$ such that $|\boldsymbol{A} - \boldsymbol{B} \circ \boldsymbol{C}|$ is minimised.

- Algebra is Boolean ($1+1 = 1$)
  $\Rightarrow \boldsymbol{B} \circ \boldsymbol{C}$ is always binary

max planck institut informatik

MODULO-2 ALGEBRA

Binary matrix factorisation under modulo-2 algebra (**XMF**).
Given an *n*-by-*m* binary matrix ***A*** and integer *k*, find *n*-by-*k* and *k*-by-*m* binary matrices ***B*** and ***C*** such that $|\boldsymbol{A} - \boldsymbol{B} \otimes \boldsymbol{C}|$ is minimised.

- Algebra is modulo-2 ($1+1 = 0$)
  $\Rightarrow \boldsymbol{B} \otimes \boldsymbol{C}$ is always binary

max planck institut informatik

OTHER OPTIONS

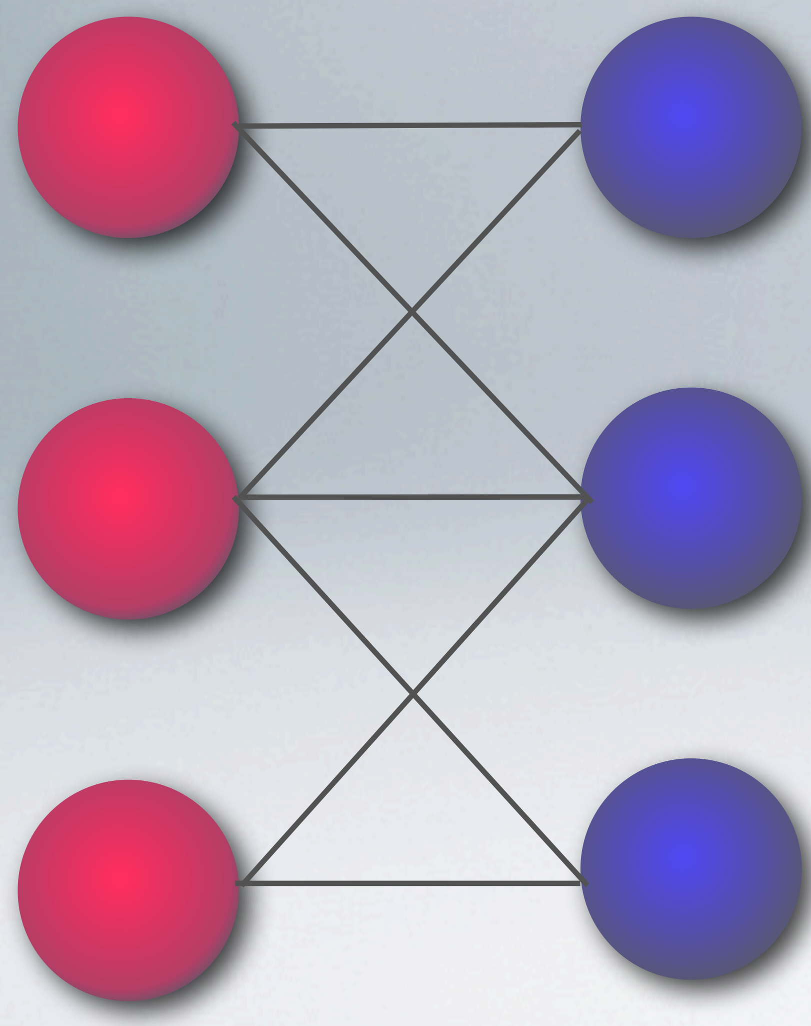
- Other definitions of underlying algebra are possible
- Example: define addition to be logical implication
    - Non-commutative
        - $\boldsymbol{A} + \boldsymbol{B} \neq \boldsymbol{B} + \boldsymbol{A}$
        - $(\boldsymbol{AB})^{\top} \neq \boldsymbol{B}^{\top}\boldsymbol{A}^{\top}$

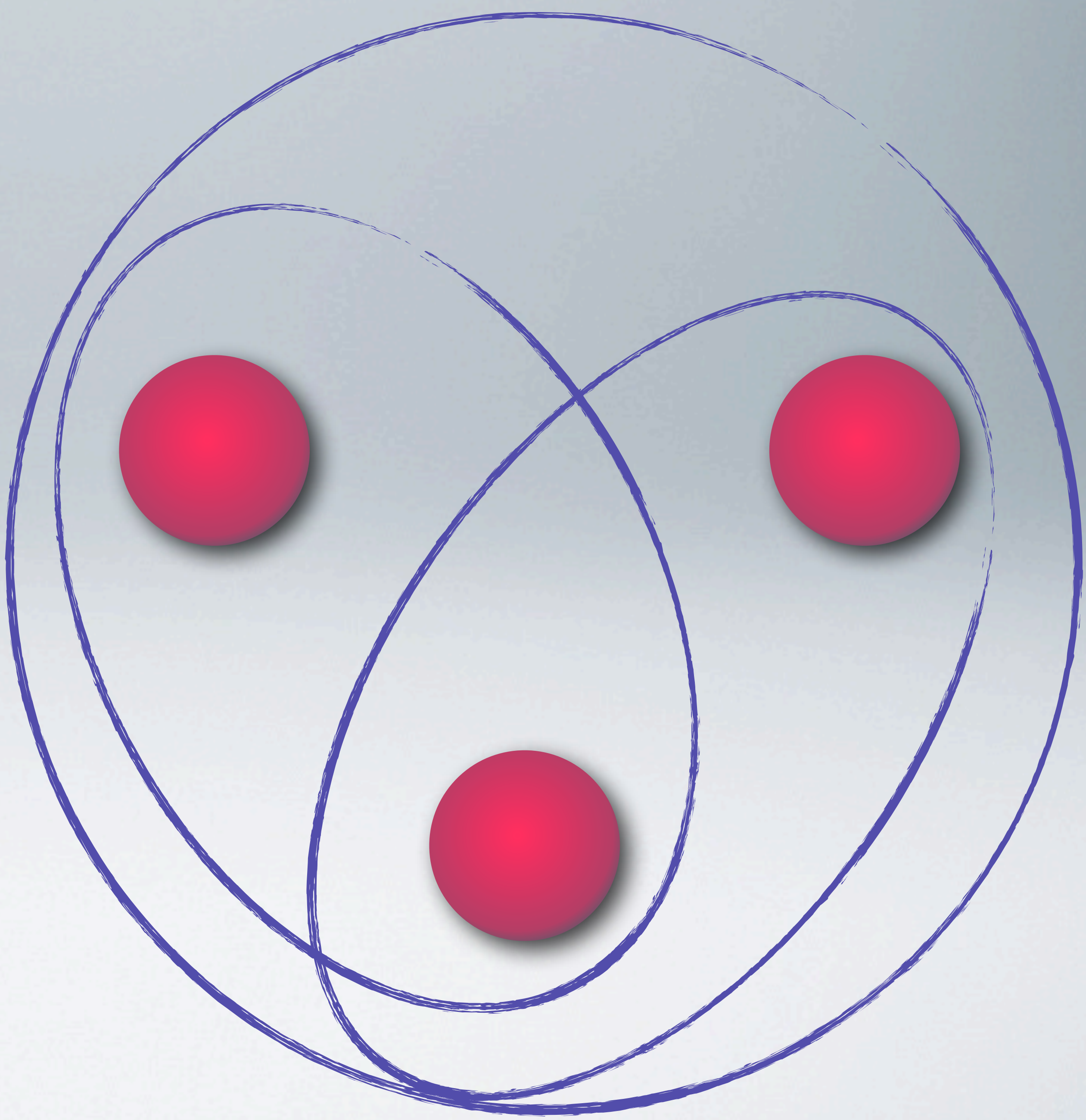
	0	1
0	1	1
1	0	1

COMPARISON

	RMF	BMF	XMF
Addition	1+1=2	1+1=1	1+1=0
Algebra	semi-ring	semi-ring	field
Closed?	not closed	closed	closed

DIFFERENT VIEWS OF BINARY DATA



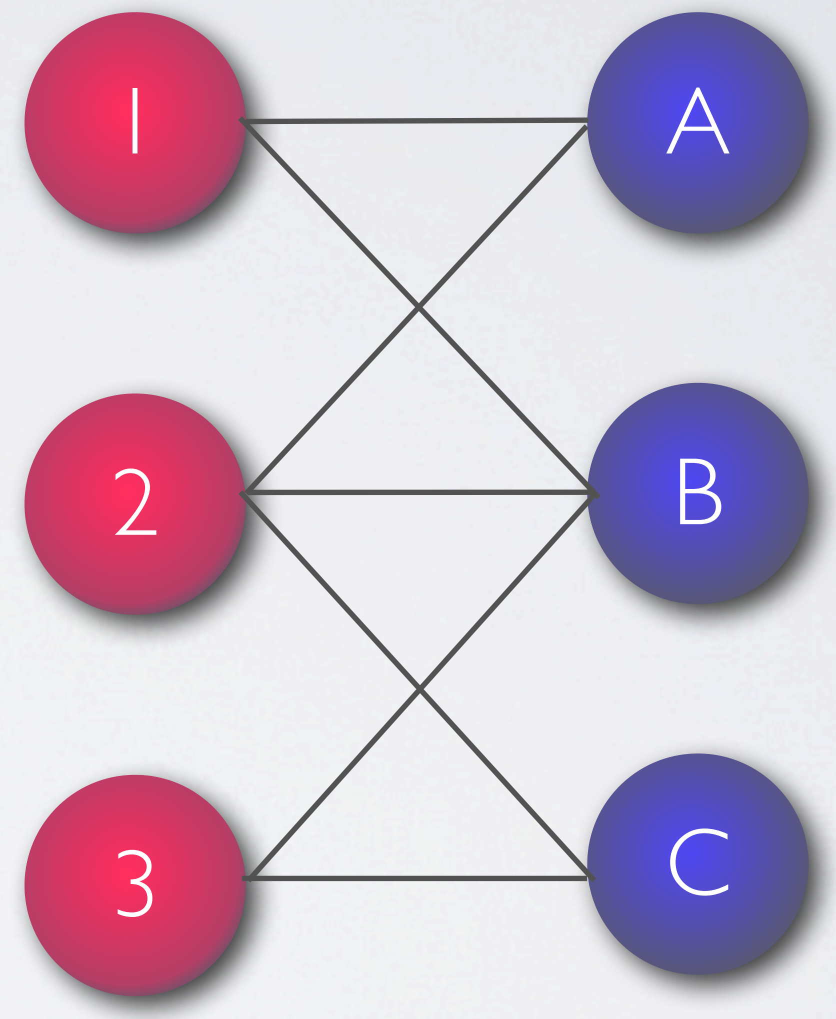


BIPARTITE GRAPHS

A

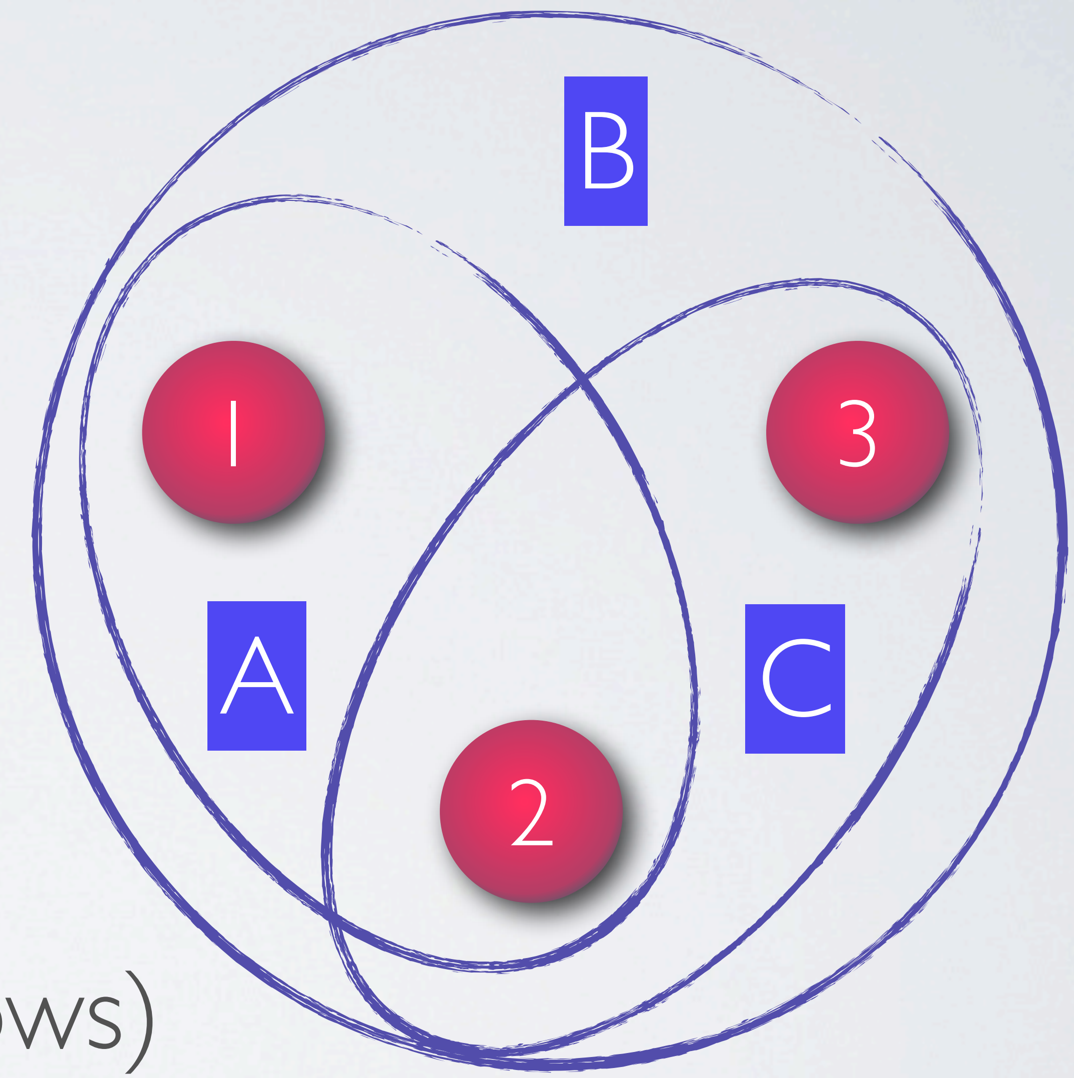
	A	B	C
1	1	1	0
2	1	1	1
3	0	1	1

$G(\boldsymbol{A})$



SETS AND COLLECTIONS

	A	B	C
1	1	1	0
2	1	1	1
3	0	1	1



$U(\boldsymbol{A})$ is an induced universe (rows)
$C(\boldsymbol{A})$ is an induced collection of sets (columns)

TILING & CLUSTERING AS MATRIX FACTORISATIONS

Image by Wikipedia user PJM

K-MEANS AS MATRIX FACTORISATION

- Given *m* data points (in $\mathbf{R}^n$), partition them in *k* clusters such that

  $$\sum_{i=1}^{k} \sum_{x_j \in C_i} \|\mathbf{x}_j - \boldsymbol{\mu}_i\|_2^2$$

  is minimised

- Equivalently, minimise $\|\boldsymbol{X} - \boldsymbol{MC}\|^2$, where
  - ***X*** is the data (*n*-by-*m*), ***M*** (*n*-by-*k*) has the centroids as its columns, and **C** (*k*-by-*m*) is a **cluster assignment matrix**
    - Each column of **C** has exactly one 1, and rest is 0s

TILING AS MATRIX FACTORISATION

- Maximum *k*-tiling: find at most *k* **tiles** such that the tiling has maximum area [1]
  - Data is binary matrix, tiles are submatrices full of 1s
  - Area of a tiling is the number of 1s in the data that belong to at least one tile
- We turn this to *minimum-error tiling*
  - Minimise the number of 1s in the data that do not belong to any tile

[1] F. Geerts et al., Tiling databases, in: DS '04, 77–122.

TILING AS MATRIX FACTORISATION

- We want to find factor matrices $\boldsymbol{A}$ and $\boldsymbol{B}$ such that $(\boldsymbol{AB})_{ij} = 1$ iff element (i, j) belongs to at least one tile
  - Minimise $|\boldsymbol{X} - \boldsymbol{AB}|$
- Single tile is an outer product of two binary vectors: $\boldsymbol{ab}^T$
  - $b_j = 1$ if an item j belongs to the tile; $a_i = 1$ if a transaction i belongs to the tile
- But how to combine the tiles?

COMBINING THE TILES

- The problem: $\sum_{i=1}^{k} \mathbf{a}_i \mathbf{b}_i^T$ is not necessarily binary
  - RMF: $|\boldsymbol{X} - \boldsymbol{AB}|$ will add an error every time $x_{ij} = 1$ belongs to more than one tile
  - BMF: don't count multiplicity (1+1 = 1)
  - XMF: consider parity (1+1 = 0)

RNF, BMF, AND XMF AS TILING

- Unlike tiling, all methods allow holes in the tiles
- BMF is otherwise like tiling
- RMF penalises for overlapping tiles
- XMF removes the overlapping part of pairs of tiles
  - For nested tiles, this would be removing exceptional areas

MATRIX RANKS

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

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

DEFINITIONS

Normal matrix rank.
The **rank** of a matrix $\boldsymbol{A}$, $\text{rank}_R(\boldsymbol{A})$, is the least integer k such that $\boldsymbol{A}$ can be expressed exactly with a decomposition of size k.

Boolean matrix rank.
The **Boolean rank** of a binary matrix $\boldsymbol{A}$, $\text{rank}_B(\boldsymbol{A})$, is the least integer k such that $\boldsymbol{A}$ can be expressed exactly with a Boolean decomposition of size k.

DEFINITIONS

Boolean matrix rank.
The **Boolean rank** of a binary matrix $\boldsymbol{A}$, $\mathrm{rank}_B(\boldsymbol{A})$, is the least integer k such that $\boldsymbol{A}$ can be expressed exactly with a Boolean decomposition of size k.

Modulo-2 matrix rank.
The **modulo-2 rank** of a binary matrix $\boldsymbol{A}$, $\mathrm{rank}_X(\boldsymbol{A})$, is the least integer k such that $\boldsymbol{A}$ can be expressed exactly with a modulo-2 decomposition of size k.

DEFINITIONS

Modulo-2 matrix rank.
The **modulo-2 rank** of a binary matrix ***A***, $\text{rank}_X(\boldsymbol{A})$, is the least integer *k* such that ***A*** can be expressed exactly with a modulo-2 decomposition of size *k*.

Binary matrix rank over normal algebra.
The **binary rank** of a binary matrix ***A***, $\text{rank}_N(\boldsymbol{A})$, is the least integer *k* such that ***A*** can be expressed exactly with a binary decomposition (with normal algebra) of size *k*.

EXAMPLE OF BOOLEAN RANK

$$\begin{pmatrix} 1 & 1 & 0 \\ 1 & 1 & 1 \\ 0 & 1 & 1 \end{pmatrix} \qquad \text{rank}_B(\boldsymbol{A}) = 2$$

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

max planck institut informatik

EXAMPLE OF XOR RANK

$\mathrm{rank}_X(\boldsymbol{A}) = 3$

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

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

EXAMPLE OF BINARY RANK

$$\begin{pmatrix} 1 & 1 & 0 \\ 1 & 1 & 0 \\ 0 & 1 & 1 \end{pmatrix} \qquad \text{rank}_N(\boldsymbol{A}) = 2$$

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

COMPARISON OF RANKS

- How do these ranks compare?
  - Is one always the smallest?
  - Is one always the largest?
  - How big the differences can be?
  - How about the normal rank?

max planck institut informatik

BOOLEAN VS NORMAL

- Incommensurable [1]
  - For some ***A***, $\text{rank}_R(\boldsymbol{A}) < \text{rank}_B(\boldsymbol{A})$
  - For some ***A***, $\text{rank}_R(\boldsymbol{A}) > \text{rank}_B(\boldsymbol{A})$
- Extrema:
  - Exists *n*-by-*n* matrix ***A***: $\text{rank}_B(\boldsymbol{A}) = \log_2(\text{rank}_R(\boldsymbol{A}))$ [1]
  - Exists *n*-by-*n* matrix ***A***, when $n \to \infty$: $\text{rank}_R(\boldsymbol{A}) = \text{rank}_B(\boldsymbol{A}) / 2$ [2]

As good as it gets

?

[1] S.D. Monson et al., A Survey of Clique and Biclique Coverings and Factorizations of (0,1)-Matrices, *Bull. ICA*. 14 (1995), 17–86.
[2] P. Kaski, personal communication.

BINARY VS THE OTHERS

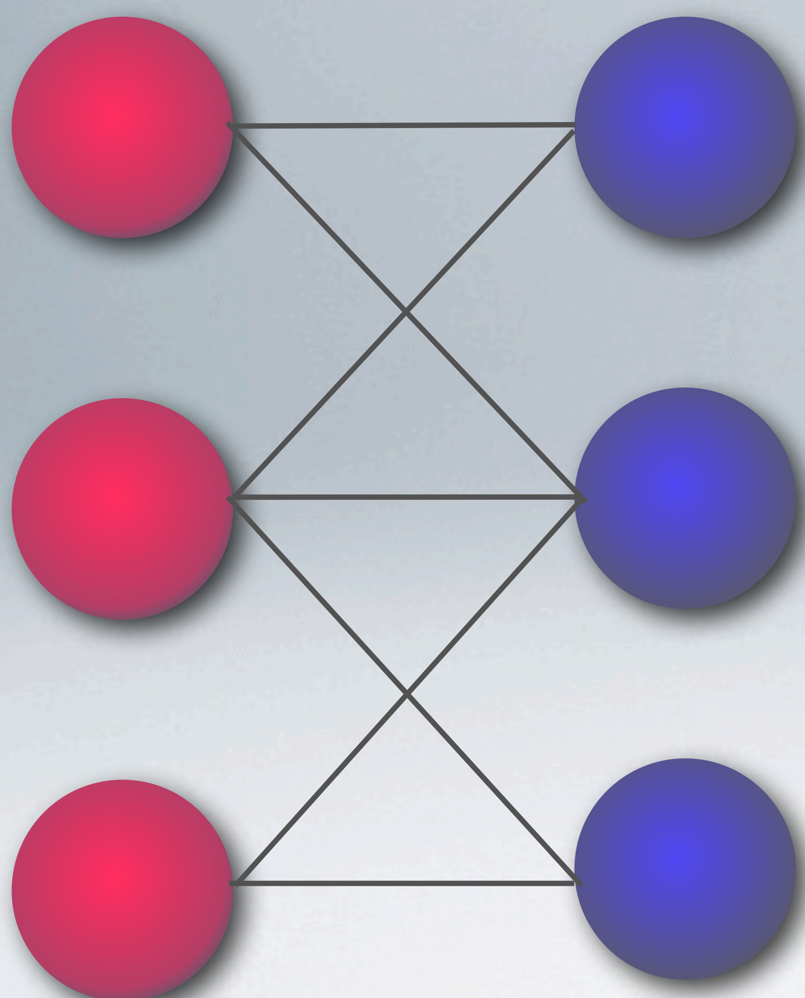
- Binary rank is always the biggest
  - $\text{rank}_N(\boldsymbol{A}) \geq \text{rank}_B(\boldsymbol{A})$ for all $\boldsymbol{A}$ [1]
  - $\text{rank}_N(\boldsymbol{A}) \geq \text{rank}_X(\boldsymbol{A})$ for all $\boldsymbol{A}$
    - All use binary numbers and binary doesn't allow overlap
  - $\text{rank}_N(\boldsymbol{A}) \geq \text{rank}_R(\boldsymbol{A})$ for all $\boldsymbol{A}$ [1]
    - Both use the same arithmetic

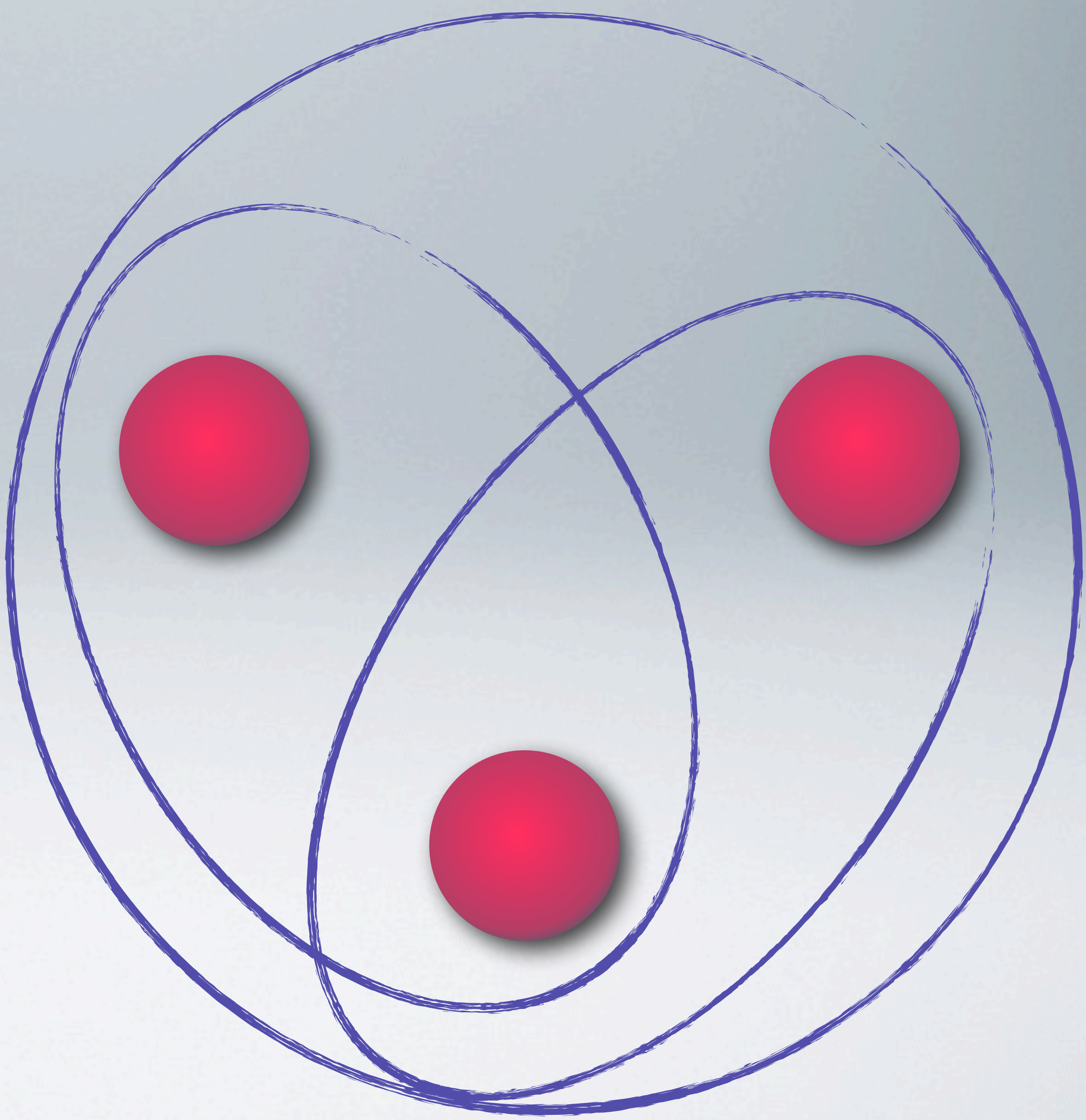
[1] D.A. Gregory, N.J. Pullman, Semiring rank: Boolean rank and nonnegative rank factorizations, *J. Combin. Inform. System Sci.* 8 (1983) 223–233.

SUMMARY

	Normal	Boolean	XOR	Binary
Normal	$=$	$\gtreqless$	$\gtreqless$	$\leq$
Boolean	$\gtreqless$	$=$	$\gtreqless$	$\leq$
XOR	$\gtreqless$	$\gtreqless$	$=$	$\leq$
Binary	$\geq$	$\geq$	$\geq$	$=$

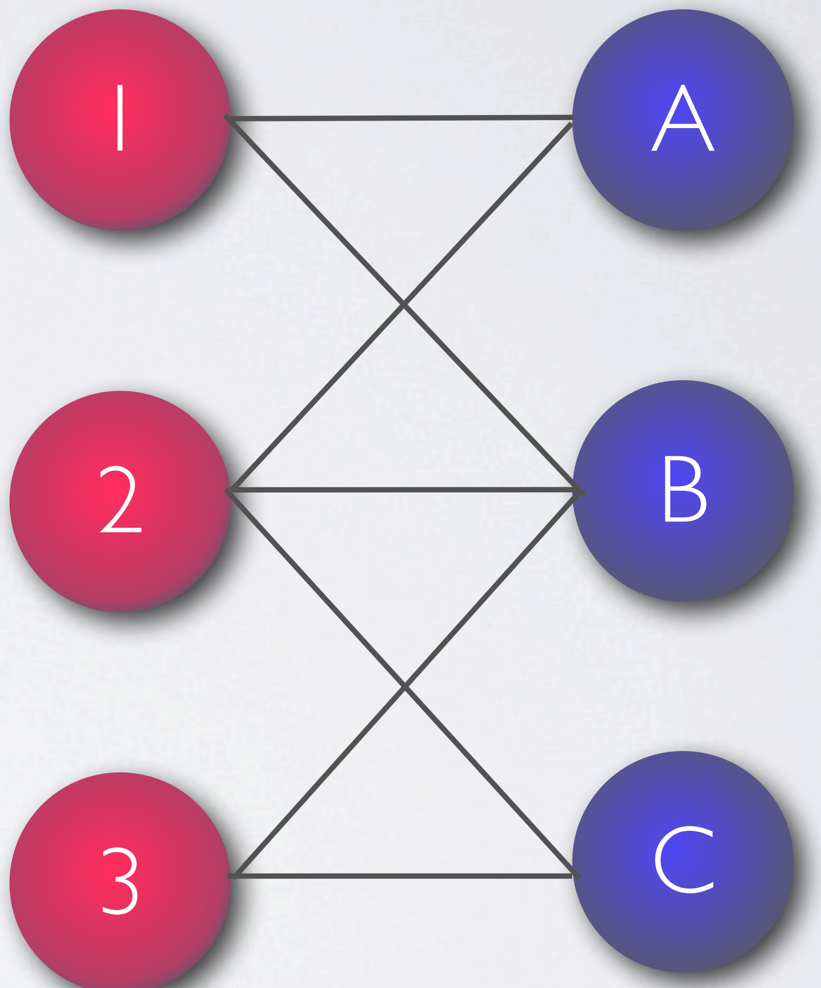
DIFFERENT VIEWS TO THE BOOLEAN RANK



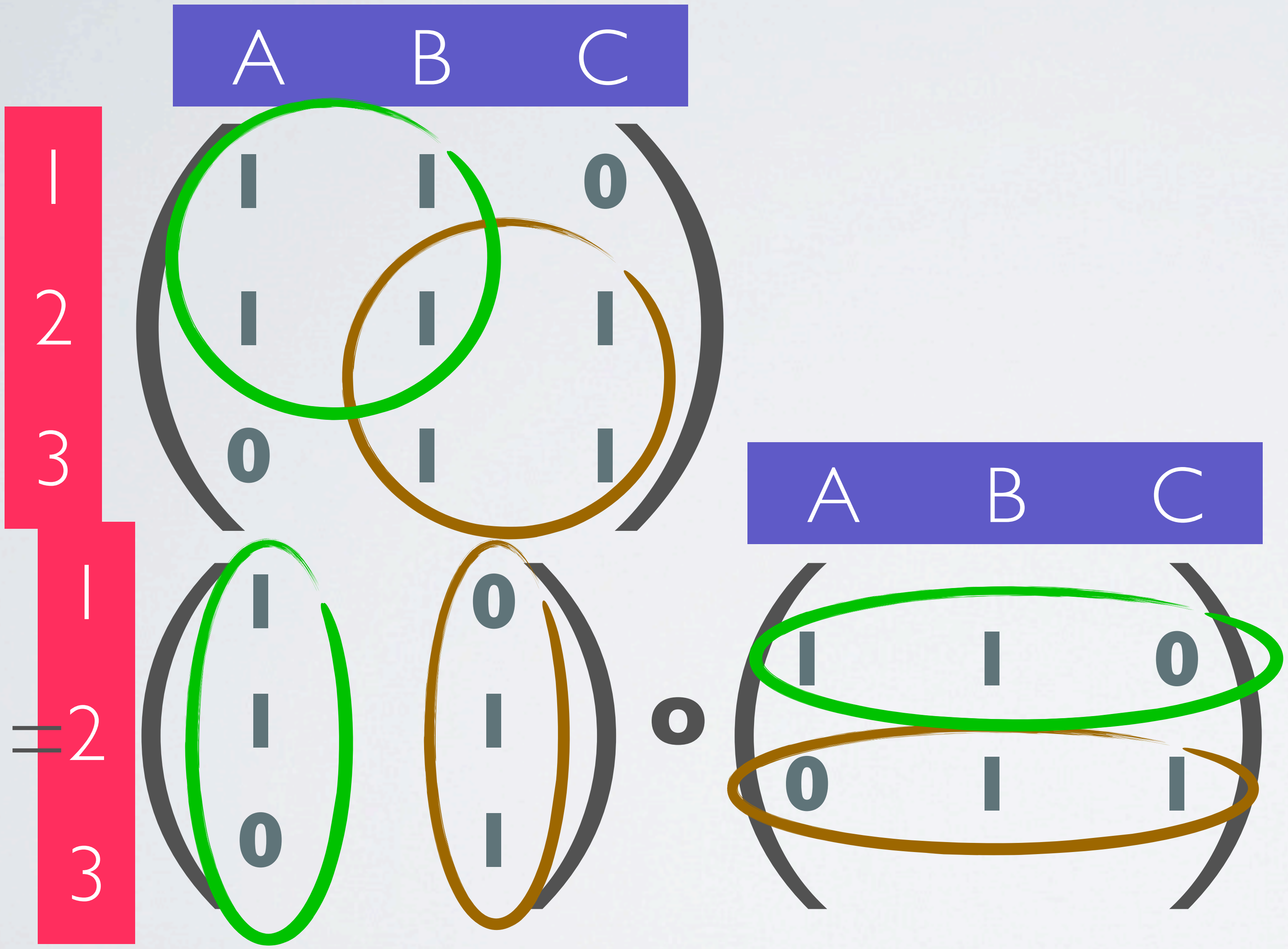


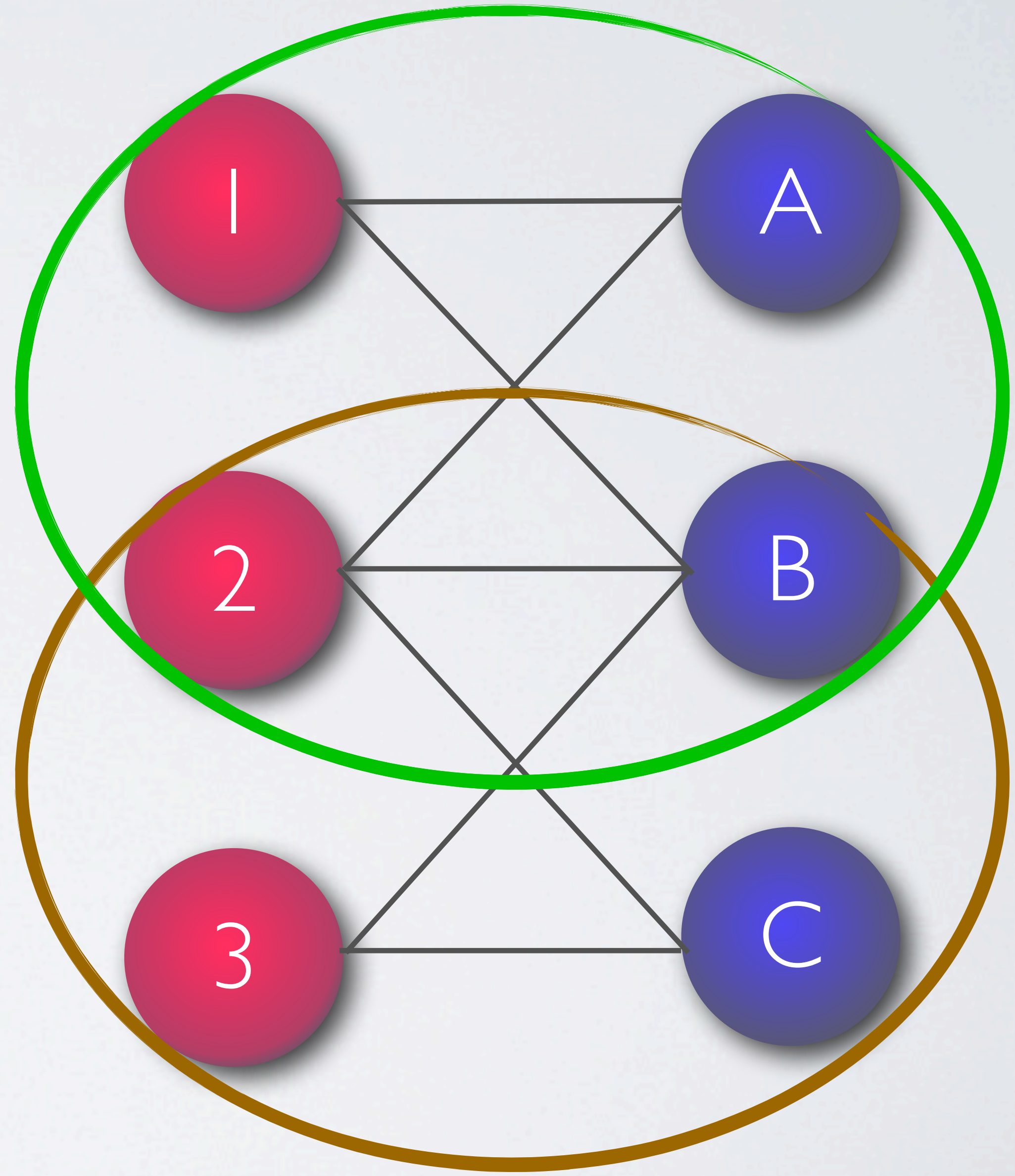
BOOLEAN RANK AND BICLIQUES

- The Boolean rank of a matrix ***A*** is **the least number of complete bipartite subgraphs needed to cover every edge** of the induced bipartite graph *G*(***A***)



BOOLEAN RANK AND BICLIQUES

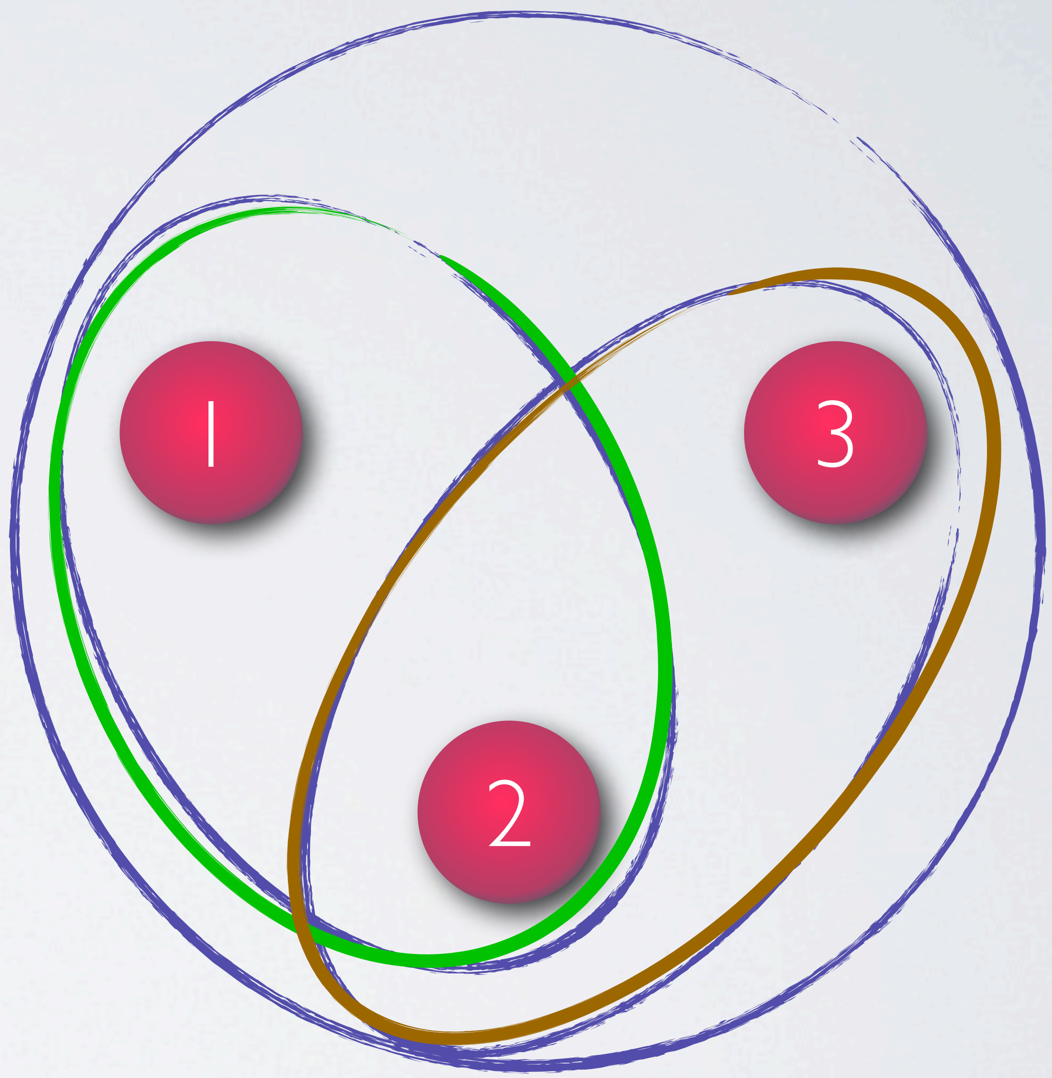




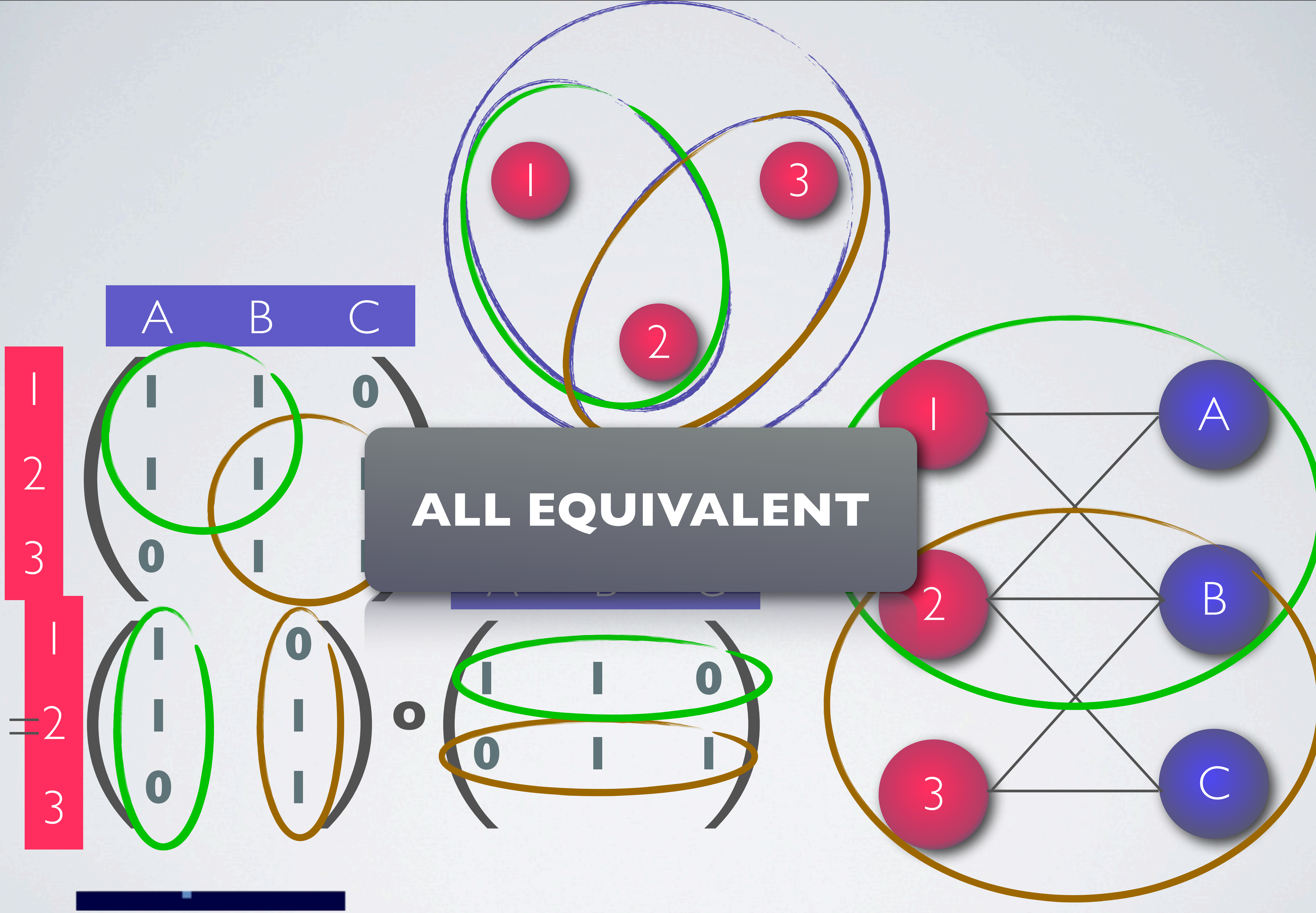
BOOLEAN RANK AND SETS

- The Boolean rank of a matrix ***A*** is **the least number of subsets of** *U*(***A***) **needed to cover every set** of the induced collection *C*(***A***)
  - For every *C* in *C*(***A***), if $\mathcal{S}$ is the collection of subsets, have subcollection $\mathcal{S}_C$ such that

$$\bigcup_{S \in \mathcal{S}_C} S = C$$



ALL EQUIVALENT



XOR AND BINARY

- XOR rank
  - Replace set union with symmetric difference and covering with parity
- Binary rank
  - Non-overlapping subsets / bicliques are sufficient, not necessary
    - Clustering

XOR RANK EXAMPLE

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

BINARY RANK EXAMPLE

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

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

A NOTE ON INVERSES

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

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

A NOTE ON INVERSES

- Every full-XOR-rank matrix has an inverse
  - Can be found e.g. using Gauss–Jordan elimination
- Only permutation matrices have an inverse in Boolean algebra [1]
- Only permutation matrices have **binary** inverses under normal algebra

[1] K.H. Kim, *Boolean matrix theory and applications*, Marcel Dekker, 1982, p. 105.

COMPUTATIONAL COMPLEXITY

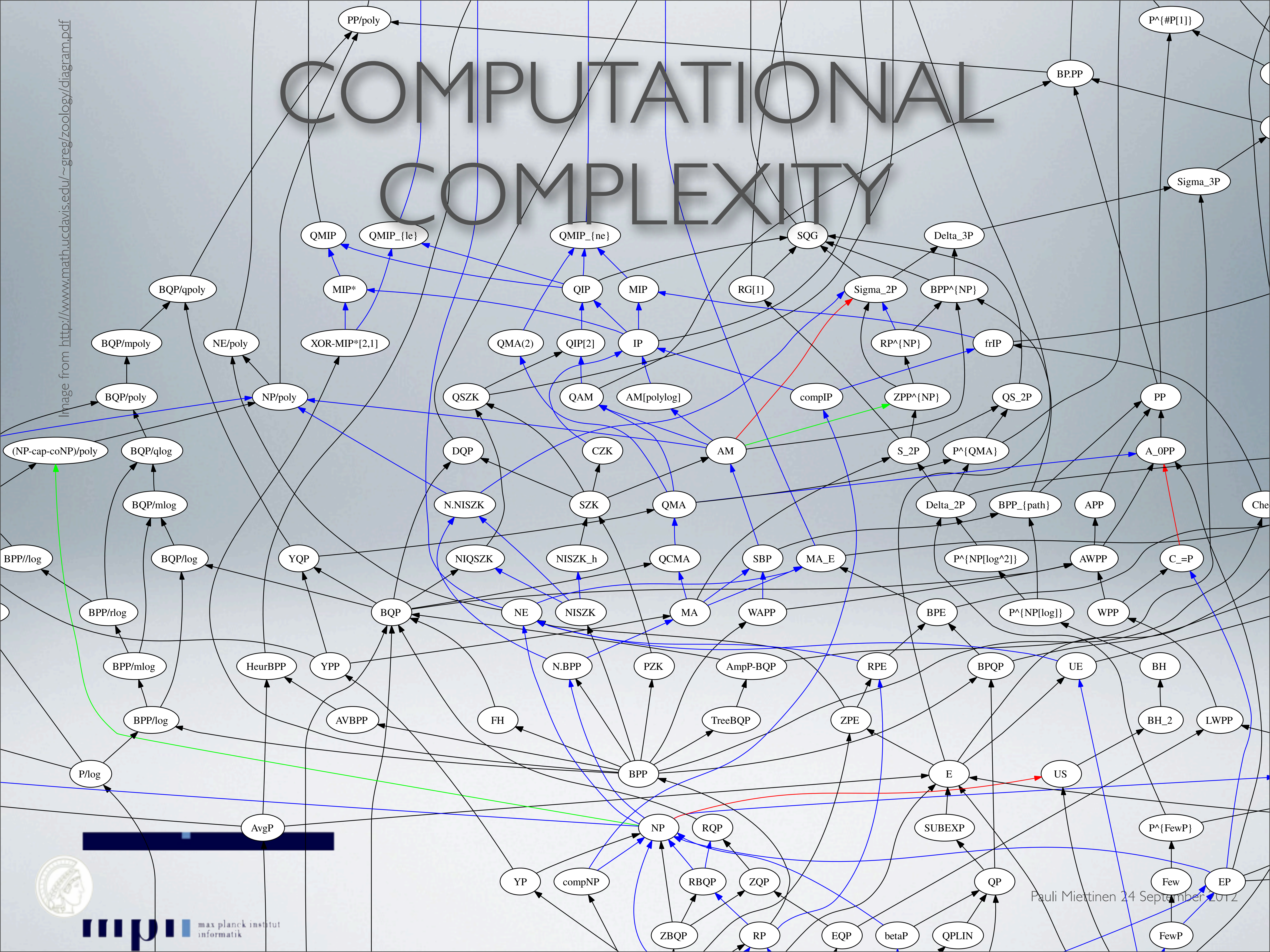


Image from http://www.math.ucdavis.edu/~greg/zoology/diagram.pdf

max planck institut informatik

FINDING THE RANKS

- XOR rank: polynomial time
  - Standard Gaussian elimination over modulo-2 arithmetic
- Boolean rank: NP-hard [1]
  - As hard to approximate as the clique ($\Omega(n^{1-\varepsilon})$ for all $\varepsilon > 0$) [2]
- Binary rank: Unknown
  - Restriction to non-overlapping factors is NP-hard (clustering) [3]

[1] D.S. Nau et al., A Mathematical Analysis of Human Leukocyte Antigen Serology, *Math. Biosci*. 40 (1978) 243–270.
[2] H.U. Simon, On approximate solutions for combinatorial optimization problems, *SIAM J. Discrete Math*. 3 (1990) 294–310.
[3] M. et al., The Discrete Basis Problem, *IEEE Trans. Knowl. Data En.* 20 (2008) 1348–1362.

BOOLEAN RANK AND TILING

- The Boolean rank of a matrix also tells us the minimum number of tiles needed to completely cover the matrix
- Minimum number of tiles can be approximated within $O(\log nm)$ [1, Thm. 2]
  - This requires an oracle that gives the largest-area tile [1]
- Without the oracle, the reduction requires exponential time
  - Except for certain sparse matrices…

[1] F. Geerts et al., Tiling databases, in: DS '04, 77–122.

MINIMUM-ERROR BMF

- NP-hard to approximate within any polynomially computable function [1]
  - Because it's NP-hard to recognise the zero-error case
- NP-hard to approximate within additive factor of $\max\{\sqrt[4]{n}, \sqrt[4]{m}\}$ [1]

[1] M., *Matrix Decomposition Methods for Data Mining: Computational Complexity and Algorithms*, PhD thesis, U. Helsinki, 2009.

MINIMUM-ERROR PROJECTIONS

- **Problem**: Given the data matrix $\boldsymbol{A}$ and one factor matrix ($\boldsymbol{B}$), find the other factor matrix ($\boldsymbol{C}$) that minimises the error
  - Per column: given a column vector $\boldsymbol{a}$ and a matrix $\boldsymbol{B}$, find a column vector $\boldsymbol{c}$ such that $\boldsymbol{a} \approx \boldsymbol{Bc}$
- ''Binary programming''
- Needed for alternating projections type algorithms (ALS)

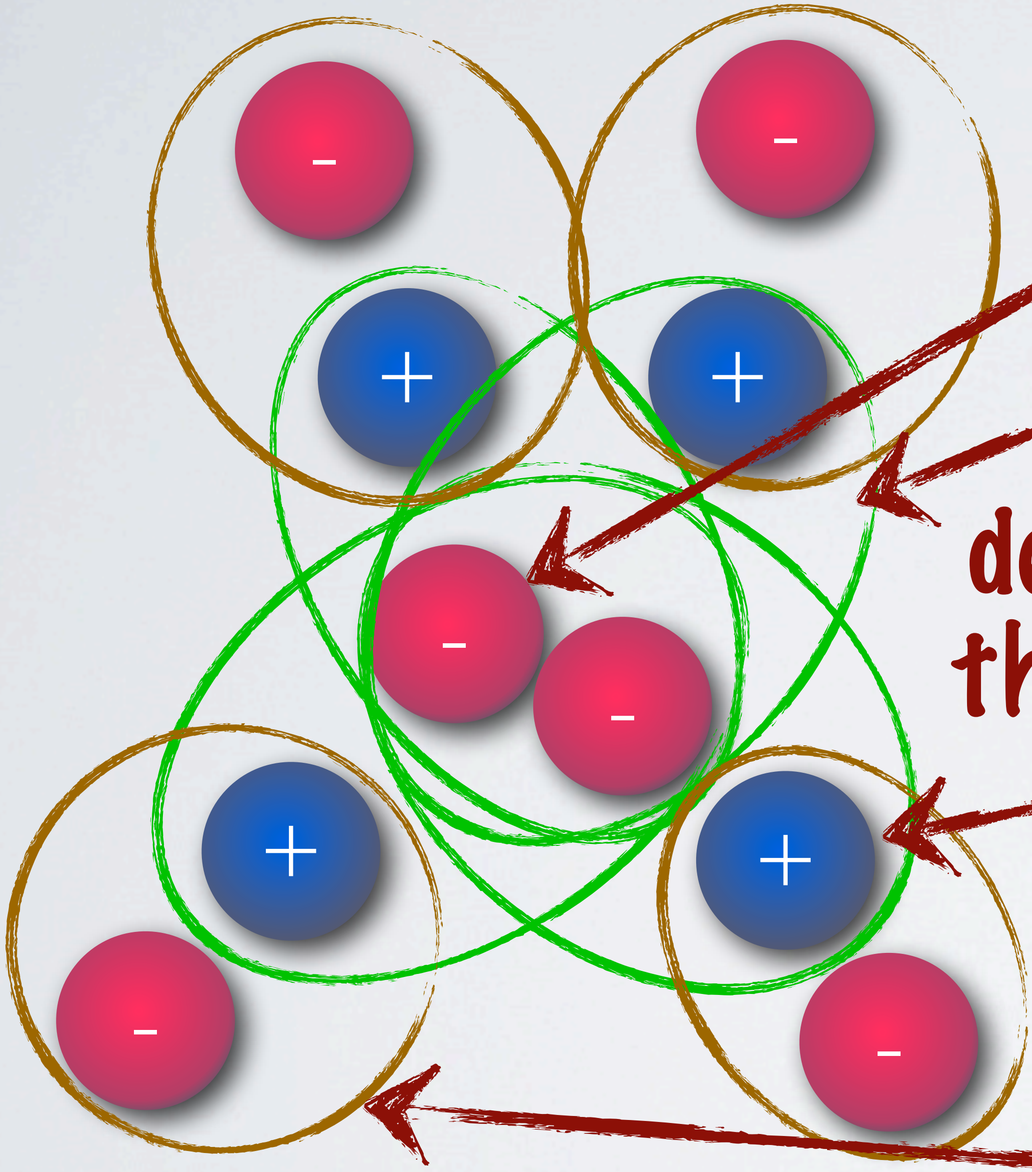
BOOLEAN PROJECTION, OR ±PSC

- The minimum-error projection under Boolean algebra is equivalent to the following problem

Positive-Negative Partial Set Cover (**±PSC**). Given a triple $(P, N, \mathcal{Q})$, where P and N are disjoint sets and $\mathcal{Q} \subseteq 2^{P \cup N}$, find a subcollection $\mathcal{D} \subseteq \mathcal{Q}$ that minimises $|P \setminus (\cup \mathcal{D})| + |N \cap (\cup \mathcal{D})|$.

EXAMPLE

defines the sets



defines the sign

a

0
0
1
1
1
1
0
0
0
0

B

1	1	1	1	0	0	0	0
1	1	1	1	0	0	0	0
1	0	0	0	1	0	0	0
0	1	0	0	0	1	0	0
0	0	1	0	0	0	1	0
0	0	0	1	0	0	0	1
0	0	0	0	1	0	0	0
0	0	0	0	0	1	0	0
0	0	0	0	0	0	1	0
0	0	0	0	0	0	0	1

COMPLEXITY OF ±PSC

- NP-hard to approximate within $\Omega(2^{\log^{1-\varepsilon}|P|})$ for any $\boldsymbol{\varepsilon} > 0$ [1]
- There exists a polynomial-time approximation algorithm that achieves $2\sqrt{[(|Q|+|P|) \log |P|]}$ approximation ratio [1,2]
⇒ In Boolean case, even simple projections are hard

[1] P. Miettinen, On the positive-negative partial set cover problem, *Inform. Process. Lett.* 108 (2008) 219–221.
[2] D. Peleg, Approximation algorithms for the Label-Cover$_{MAX}$ and Red-Blue Set Cover problems, *J. Discrete Alg.* 5 (2007) 55–64.

THE BINARY CASE

- The zero-error case is NP-hard
  - Simple reduction from Exact Cover by 3-sets (X3C)
- A variant is the Closest Vector problem (CVP), where columns of ***B*** have to be linearly independent and the vectors take integer values
  - CVP is NP-hard to approximate within $n^{1/\log\log n}$ [1]

[1] I. Dinur et al., Approximating CVP to Within Almost-Polynomial Factors is NP-Hard, *Combinatorica*. 23 (2003) 205–243.

THE MODULO-2 CASE

- The problem of finding binary vector $\boldsymbol{x}$ such that, for given $\boldsymbol{a}$ and $\boldsymbol{B}$, the Hamming distance between $\boldsymbol{a}$ and $\boldsymbol{B} \otimes \boldsymbol{x}$ is minimised, is known as the Closest Codeword problem
  - NP-hard to approximate to within any constant factor [1]
    - And quasi-NP-hard to approximate within $2^{\log^{\varepsilon} n}$ for $0 < \varepsilon < 1/2$
  - Admits polynomial-time $n/\log(n)$ factorisation [2]

[1] S. Arora et al., The Hardness of Approximate Optima in Lattices, Codes, and Systems of Linear Equations, in: FOCS '93, 724–733.
[2] N. Alon et al., Deterministic Approximation Algorithms for the Nearest Codeword Problem, in: APPROX RANDOM '09, 339–351.

SUMMARY

	RMF	BMF	XMF
Rank	?	NP-hard even to approximate	Polynomial
Min. error decomp.	?	NP-hard even to approximate	?
Closest projection	NP-hard	NP-hard to approx. $\Omega(2^{\log^{1-\varepsilon}\lvert P\rvert})$	NP-hard to approx. w/ constant factor
Projection approx.	?	$2\sqrt{[(\lvert Q\rvert+\lvert P\rvert) \times \log \lvert P\rvert]}$	$O(n/\log(n))$

OPEN PROBLEMS

?

RANKS

- **P1.1** What is the largest possible ratio $\text{rank}_B(\boldsymbol{A})/\text{rank}_R(\boldsymbol{A})$
  - Best known is 2
- **P1.2** What are the extrema of the XOR rank w.r.t. the other ranks?
  - It's incommensurable to normal and Boolean rank

max planck institut informatik

COMPLEXITY

- **P1.3** Is binary rank NP-hard to compute?
- **P1.4** Is RMF NP-hard?
  - Probably, given that NMF is [1]
- **P1.5** Is XMF NP-hard?
- **P1.6** What's the approximability of binary projections?
- **P1.7** What's the approximability of maximum similarity problems?

[1] S.A. Vavasis, On the Complexity of Nonnegative Matrix Factorization, *SIAM J. Optim.* 20 (2010) 1364–1377.

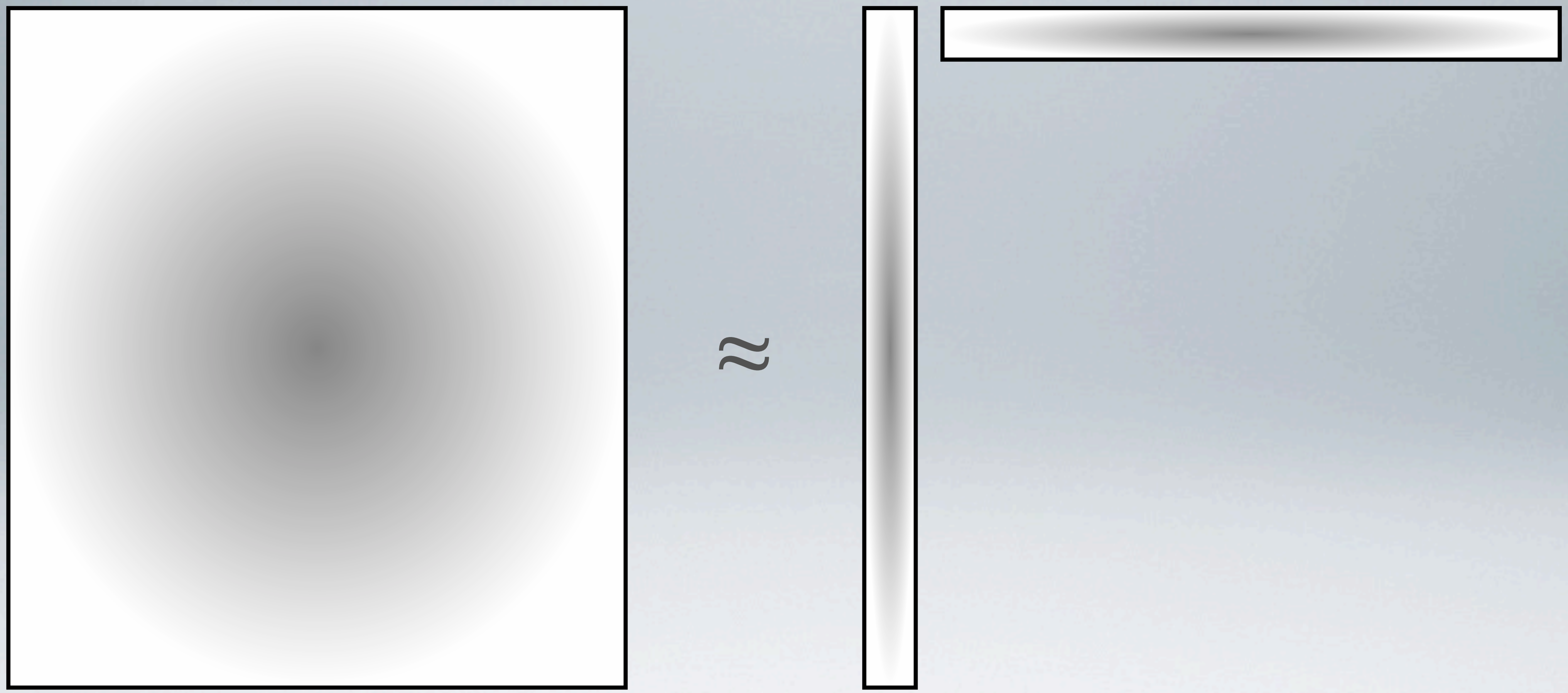
MISCELLANEOUS

- **P1.8** Are there meaningful (in data mining) definitions of the addition (or multiplication) not covered here?

PART II
ALGORITHMS AND EXTENSIONS

CONTENTS

RANK-1 DECOMPOSITIONS



RANK-1 DECOMPOSITIONS

- In rank-1 decompositions, addition doesn't matter
  - We can also use squared Frobenius for distance
- One could hope to use rank-1 approximations as building blocks for higher-rank decompositions
  - Problem: good rank-1 decomposition does not need to be a part of any good rank-2 decompositions

EXAMPLE

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

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

PROXIMUS

- The PROXIMUS algorithm [1] finds the binary rank-1 factorisation using iterative updates
  - To find ***b*** and ***c*** such that $\boldsymbol{A} \approx \boldsymbol{b}\boldsymbol{c}^T$, fix ***c*** and set

$$\mathbf{b}_i = \begin{cases} 1, & \text{if } 2(\boldsymbol{A}\mathbf{c})_i \geqslant \|\mathbf{c}\|_2^2 \\ 0, & \text{otherwise} \end{cases}$$

    and similarly for ***b*** fixed
- Proper initialisation is important

[1] M. Koyutürk, A. Grama, PROXIMUS: a framework for analyzing very high dimensional discrete-attributed datasets, in: KDD '03, 147–156.

IP, LP, AND MAX FLOW ALGORITHMS

- Minimum-error rank-1 binary factorisation can be presented as an integer programming
- Can be relaxed to a linear program that gives an upper bound for the error
  - This LP is totally unimodular ⇒ solution is binary
  - The solution is a 2-approximation
- A regularised version can be approximated with a max flow algorithm

B.-H. Shen et al., Mining Discrete Patterns via Binary Matrix Factorization, in: KDD '09, 757–765.

NORMAL ALGEBRA

$$\min \quad J(B, C) = \sum_{i,j} (\mathbf{A}_{ij} - (\mathbf{BC})_{ij})^2$$

$$\text{s.t.} \quad \mathbf{B}_{ij}^2 - \mathbf{B}_{ij} = 0$$

$$\mathbf{C}_{ij}^2 - \mathbf{C}_{ij} = 0$$

$$\sum_{i,j} \left(\mathbf{A}_{ij} - \left(\theta(\mathbf{B} - b)\theta(\mathbf{C} - c)\right)_{ij}\right)^2$$

PROXIMUS

- PROXIMUS uses rank-1 factorisations to make a hierarchical factorisation of the full data
  - Matrix rows are divided into two sets based on the column factor
  - Rank-1 decomposition is applied to those two sets separately (or recursion is stopped)
- Ensures that columns of ***B*** don't overlap ⇒ representation is binary

M. Koyutürk, A. Grama, PROXIMUS: a framework for analyzing very high dimensional discrete-attributed datasets, in: KDD '03, 147–156.

RMF AND NMF

Boundedness [1]. If $\boldsymbol{X}$ is a matrix taking values from [0,1] and if $\boldsymbol{X}$ admits a rank-k factorisation to nonnegative matrices, then there exists a nonnegative rank-k factorisation such that no value in the factor matrices is larger than 1.

[1] Z.-Y. Zhang et al., Binary matrix factorization for analyzing gene expression data, *Data Min. Knowl. Discov.* 20 (2010) 28–52.

NON-LINEAR PROGRAMMING

$$\begin{aligned} \min \quad & J(B, C) = \sum_{i,j} (\mathbf{A}_{ij} - (\mathbf{BC})_{ij})^2 \\ \text{s.t.} \quad & \mathbf{B}_{ij}^2 - \mathbf{B}_{ij} = 0 \\ & \mathbf{C}_{ij}^2 - \mathbf{C}_{ij} = 0 \end{aligned}$$

Solved by minimising (alternatively for ***B*** and ***C***):

$$\sum_{i,j} (\mathbf{A}_{ij} - (\mathbf{BC})_{ij})^2 + \frac{1}{2}\lambda\big((\mathbf{B}_{ij}^2 - \mathbf{B}_{ij}) + (\mathbf{C}_{ij}^2 - \mathbf{C}_{ij})\big)$$

Z.-Y. Zhang et al., Binary matrix factorization for analyzing gene expression data, *Data Min. Knowl. Discov.* 20 (2010) 28–52.

THRESHOLD METHOD

- Change the objective to $\sum_{i,j}\left(\mathbf{A}_{ij} - \left(\theta(\mathbf{B} - b)\theta(\mathbf{C} - c)\right)_{ij}\right)^2$
  - $\theta(\boldsymbol{X})$ is the (element-wise) Heaviside function
- Can be optimised using gradient descent after the Heaviside is replaced with $\phi(x) = 1/\left(1 + e^{-\lambda x}\right)$

Z.-Y. Zhang et al., Binary matrix factorization for analyzing gene expression data, *Data Min. Knowl. Discov.* 20 (2010) 28–52.

BOOLEAN ALGEBRA

Images by Wikipedia users Arab Ace and Sheilalau

THE BOOLEAN PROJECTION

- Peleg's algorithm approximates within $2\sqrt{[(k+a(\log a)]}$ [1]
  - a is the maximum number of 1s in ***A***'s columns
- Optimal solution
  - Either an $O(2^k knm)$ exhaustive search [1], or an integer program [2]
- Greedy algorithm: select each column of ***B*** if it improves the residual error [1]

[1] M., *Matrix Decomposition Methods for Data Mining: Computational Complexity and Algorithms*, PhD thesis, U. Helsinki, 2009.
[2] H. Lu et al., Optimal Boolean Matrix Decomposition: Application to Role Engineering, in: ICDE '08, 297–306.

THE ASSO ALGORITHM

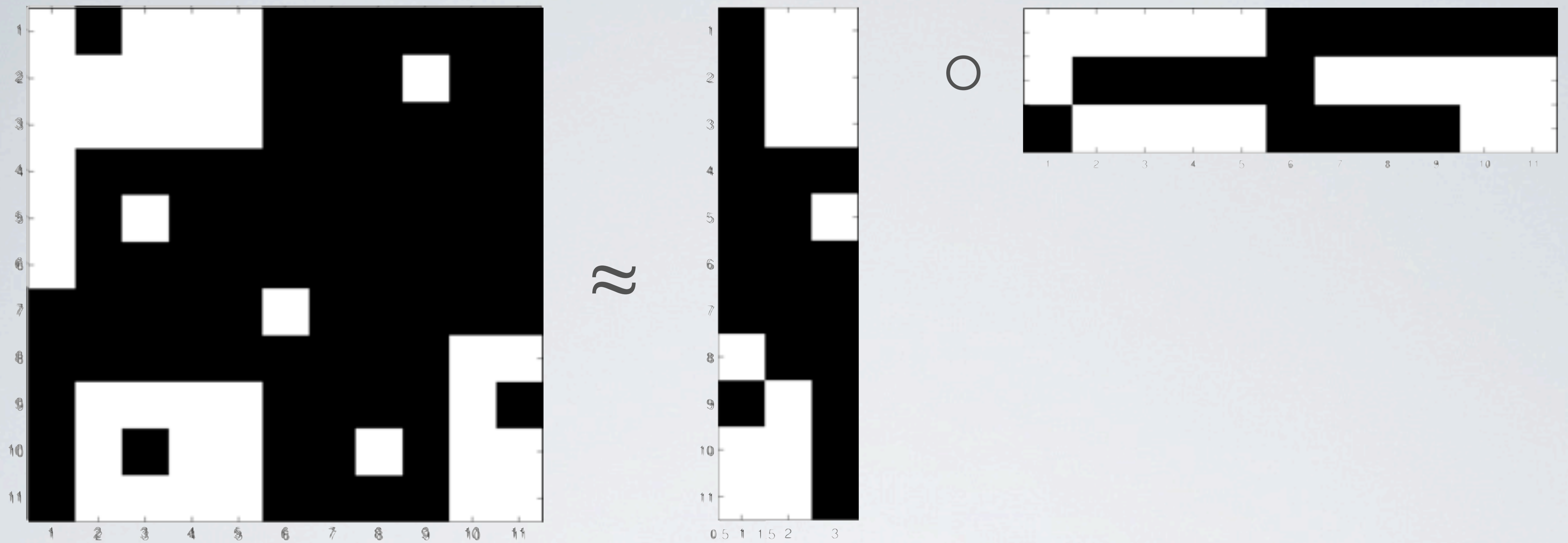
- Heuristic – too many hardness results to hope for good provable results in any case
- **Intuition**: If two columns share a factor, they have 1s in same rows
  - Noise makes detecting this harder
  - Pairwise row association rules reveal (some of) the factors

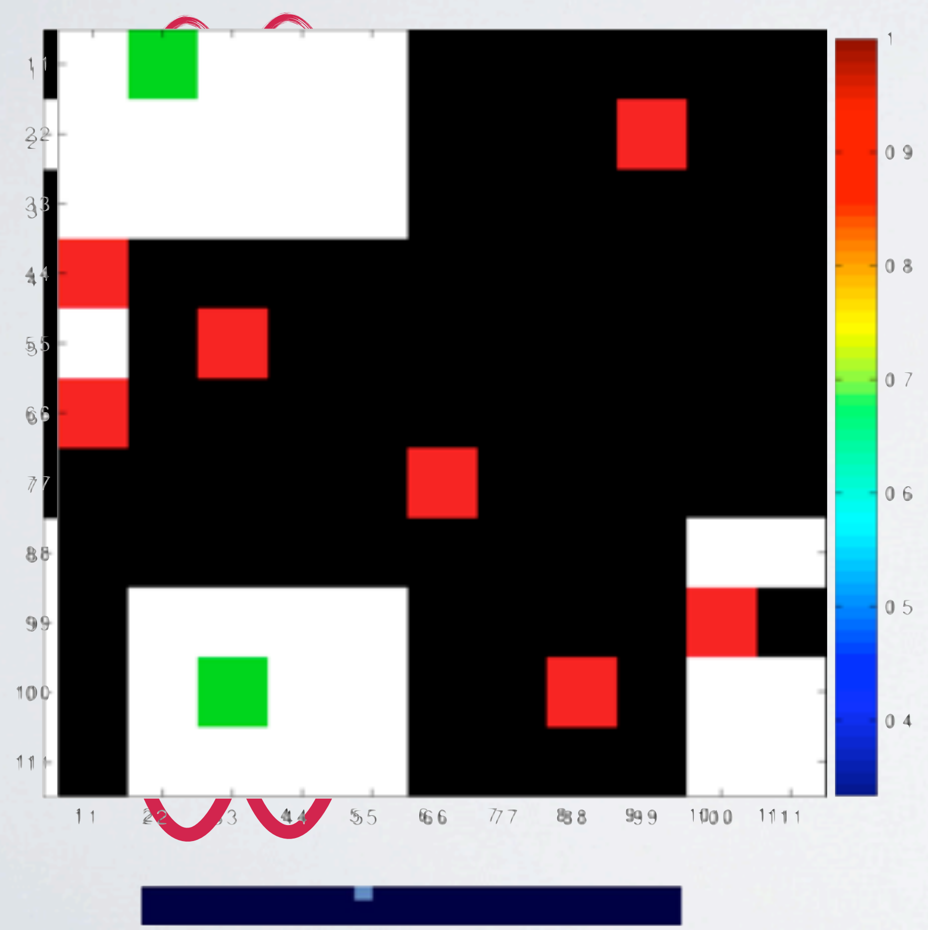
M. et al., The Discrete Basis Problem, *IEEE Trans. Knowl. Data En.* 20 (2008) 1348–1362.

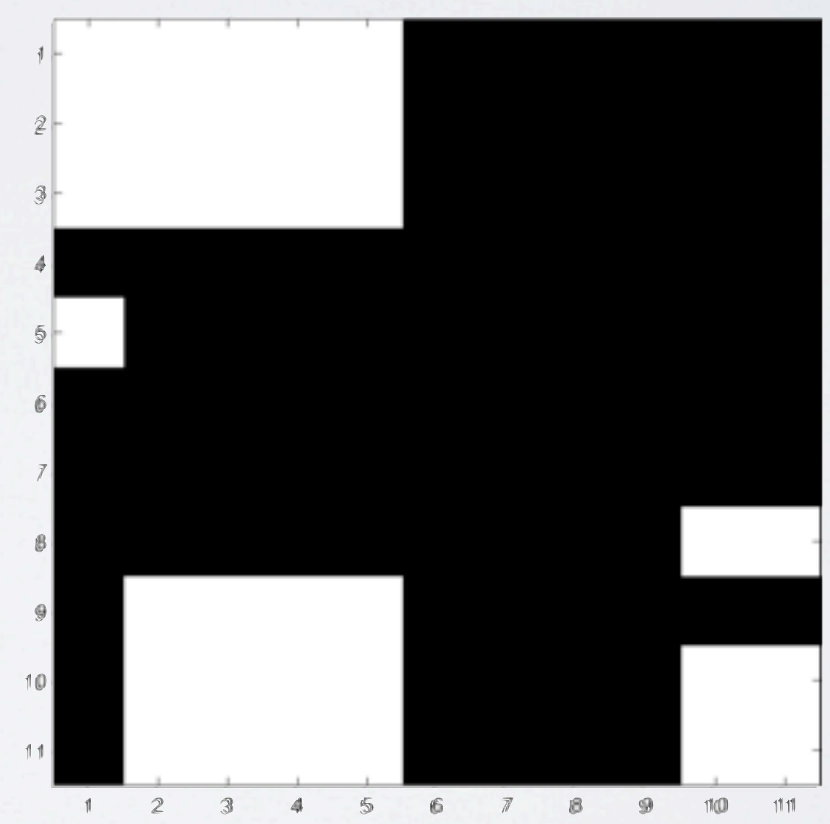
THE ASSO ALGORITHM

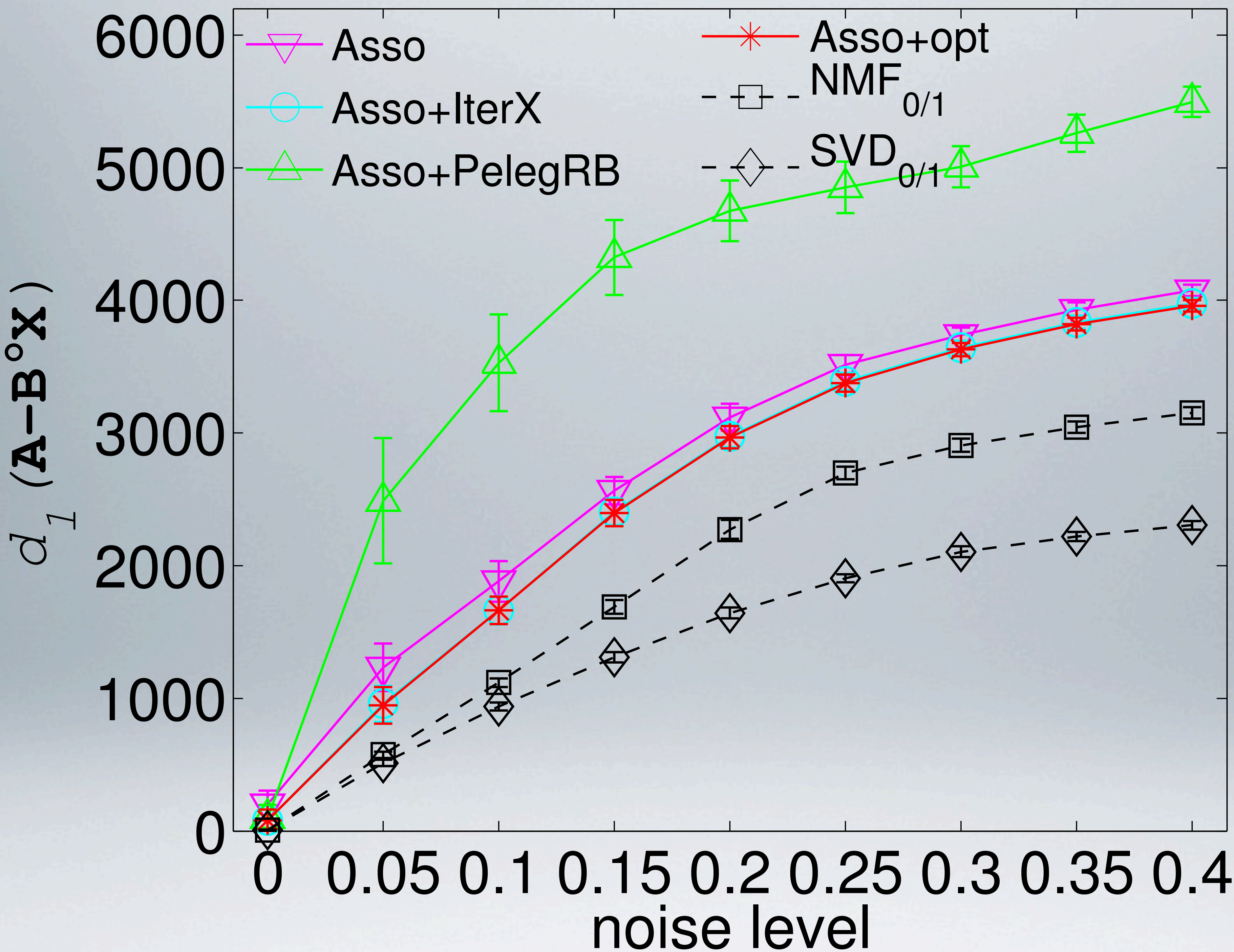
1. Compute pairwise association accuracies between rows of **A**
2. Round these (from a user-defined point *t*) to get a binary *n*-by-*n* matrix of candidate columns
3. Select greedily the candidate column that covers most of the not-yet covered 1s of **A**
4. Mark the 1s covered by the selected vector and return to 3 or quit if enough factors have been selected

M. et al., The Discrete Basis Problem, *IEEE Trans. Knowl. Data En.* 20 (2008) 1348–1362.









M., *Matrix Decomposition Methods for Data Mining: Computational Complexity and Algorithms*, PhD thesis, U. Helsinki, 2009, p. 72.

THE PANDA ALGORITHM

- **Intuition**: every good factor has a noise-free core
- Two-phase algorithm:
    1. Find error-free core pattern (maximum area itemset/tile)
    2. Extend the core with noisy rows/columns
- The core patterns are found using a greedy method
    - The 1s already belonging to some factor/tile are removed from the residual data where the cores are mined

C. Lucchese et al., Mining Top-K Patterns from Binary Datasets in presence of Noise, in: SDM '10, 165–176.

EXTENDING CORES IN PANDA

- The cores are extended in a greedy manner
  - A new column is added to a row factor in ***c***
  - All rows not yet in the corresponding column factor ***b*** are tried
- As extending a core always covers some 0s, the quality is decided by trying to minimise the number of 1s in factors ***b*** and ***c*** plus the noise

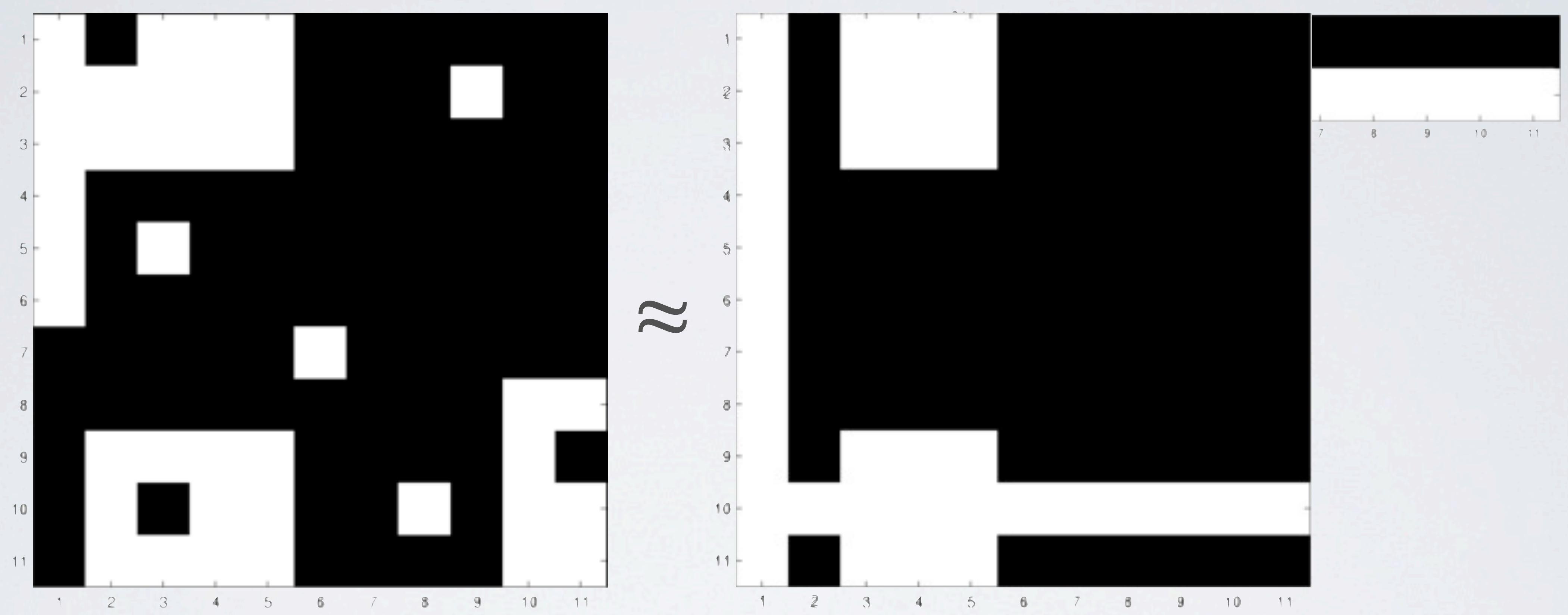
C. Lucchese et al., Mining Top-K Patterns from Binary Datasets in presence of Noise, in: SDM '10, 165–176.

NOTES ON PANDA

- Can automatically choose the rank of the decomposition
  - Parameter-free
- Uses sorting to speed up the computation
  - Consider the most promising candidates first
- Can be randomised

C. Lucchese et al., Mining Top-K Patterns from Binary Datasets in presence of Noise, in: SDM '10, 165–176.

EXAMPLE



MODULO-2 ALGEBRA

X

NO SPECIAL ALGORITHMS

- That I'm aware of, at least
- One could truncate any rank-*k* decomposition
  - No guarantees on quality, might cause more error than the trivial decomposition
  - No Eckart–Young theorem

SELECTING THE RANK

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

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

PRINCIPLES OF GOOD *K*

- **Goal:** Separate noise from structure
- We assume data has correct type of structure
  - There are k factors explaining the structure
  - Rest of the data does not follow the structure (noise)
- But how to decide where structure ends and noise starts?

WHAT HAS BEEN DONE BEFORE?

- Model order selection for matrix factorisations is studied before (mostly with SVD/PCA)
- Methods such as Guttman–Kaiser criterion [see 1] or Cattell's scree test [2] are not very good
  - Poor performance and need for subjective decisions

[1] K.A. Yeomans, P.A. Golder, The Guttman–Kaiser criterion as a predictor of the number of common factors, *The Statistician* 31 (1982) 221–229.
[2] R.B. Cattell, The Scree Test For The Number Of Factors, *Multivar. Behav. Res.* 1 (1966) 245–276.

CROSS VALIDATION

- Idea: hold part of the data, learn a model on the remaining, and fit the model to the withheld data
- Problems with matrix factorisations:
    - If we hold out only rows (or columns), no cost for fitting higher-order factorisations
    - If we hold out both, fitting the model becomes hard
    - Bi-cross-validation [1] does that, but requires singular data matrix and optimal projections

[1] A.B. Owen, P.O. Perry, Bi-cross-validation of the SVD and the nonnegative matrix factorization, *Ann. Appl. Stat.* 3 (2009) 564–594.

MINIMUM TRANSFER COST PRINCIPLE

- A variation of cross validation
- The withheld rows are mapped to their closest pairs in training data
    - For evaluation, the rows are represented using the representation of their pairs in training data
      ⇒ Penalises for over-fitting

M. Frank et al., The Minimum Transfer Cost Principle for Model-Order Selection, in: ECML PKDD '11, 423–438.

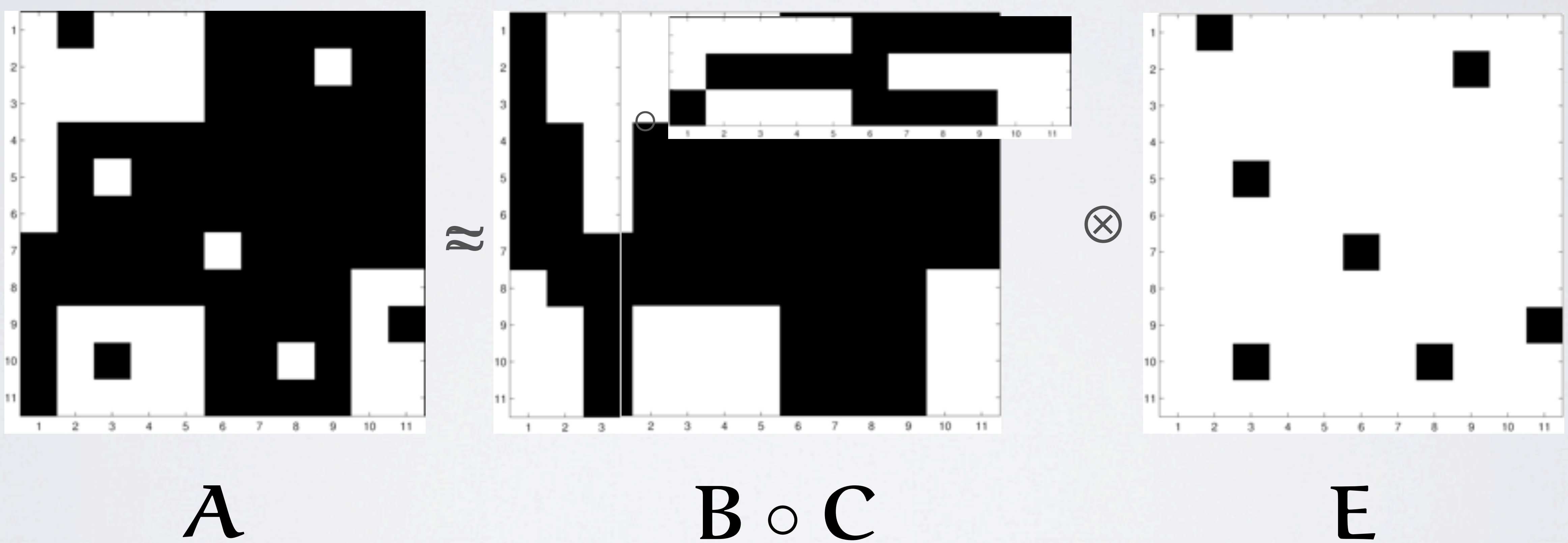
MINIMUM DESCRIPTION LENGTH PRINCIPLE

- The best model (order) is the one that allows you to explain your data with least number of bits
  - Two-part (crude) MDL: the cost of model $L(\mathcal{H})$ plus the cost of data given the model $L(D \mid \mathcal{H})$
- Problem: how to do the encoding
  - Has been done for BMF [1], similar encodings work for other binary factorisations

[1] M., J. Vreeken, Model Order Selection for Boolean Matrix Factorization, in: KDD '11, 51–59.

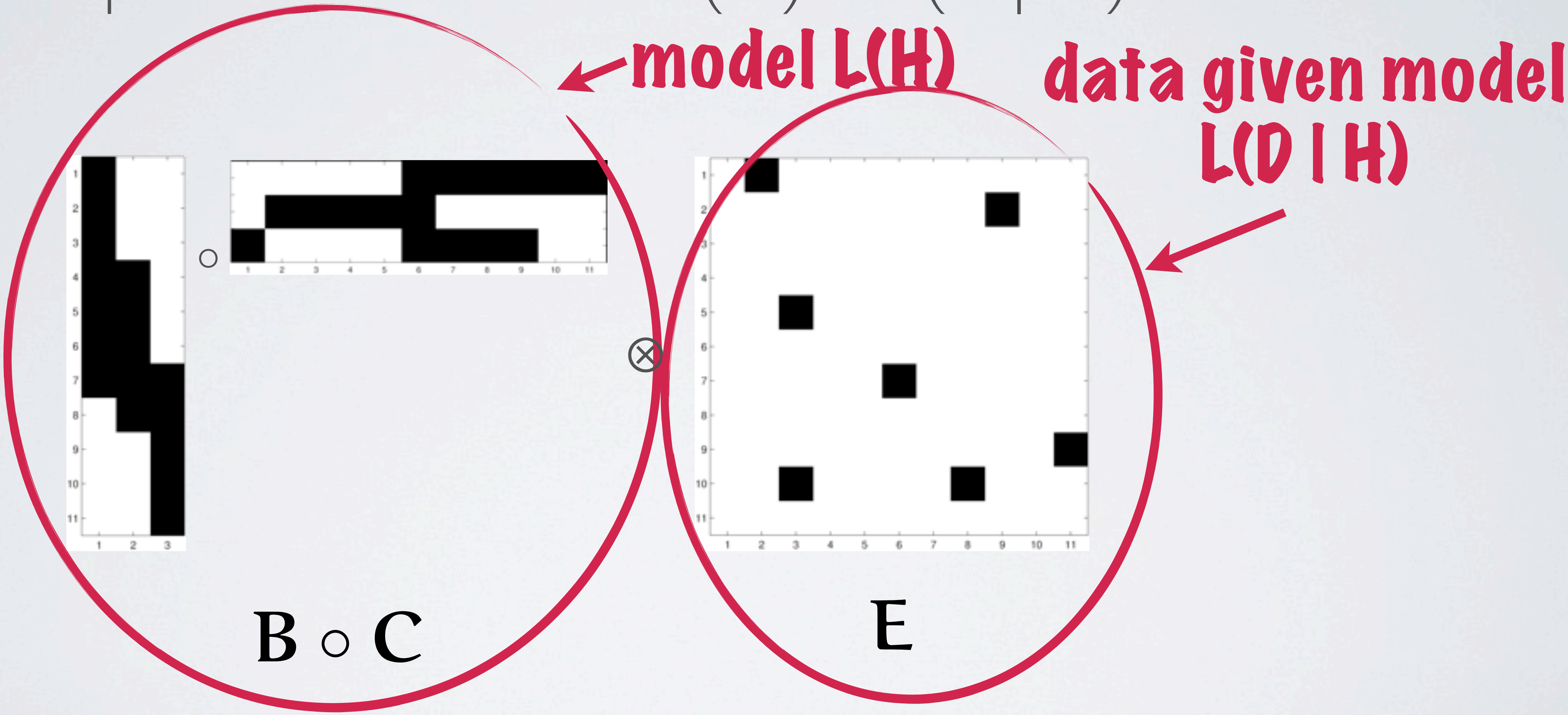
FITTING BMF TO MDL

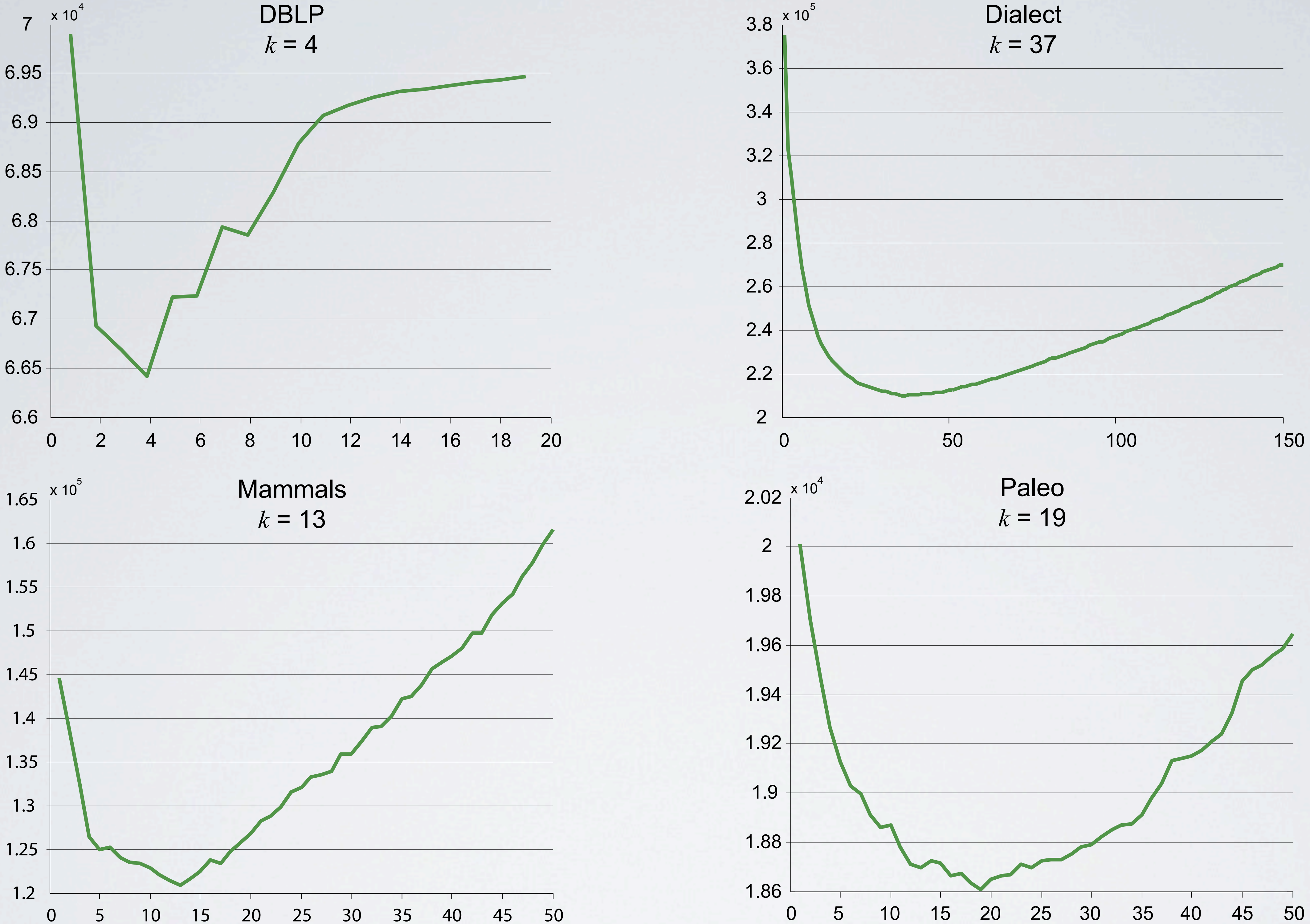
- MDL requires exact representation



FITTING BMF TO MDL

- Two-part MDL: minimise $L(\mathcal{H}) + L(D \mid \mathcal{H})$





EXAMPLE: ASSO & MDL

M., J. Vreeken, Model Order Selection for Boolean Matrix Factorization, in: KDD '11, 51–59.

SPARSE MATRICES

MOTIVATION

- Many real-world binary matrices are sparse
- Representing sparse matrices with sparse factors is desirable
  - Saves space, improves usability, …
- Sparse matrices should be computationally easier

SPARSE FACTORISATIONS

- Any binary matrix ***A*** that admits rank-*k* BMF has factorisation to matrices ***B*** and ***C*** such that the total number of 1s in ***B*** and ***C*** is at most twice that of ***A*** [1]
  - Can be extended to approximate factorisations
  - Tight result (consider a case when ***A*** has exactly one 1)
  - Holds also for exact RMF factorisations

[1] M., Sparse Boolean Matrix Factorizations, in: ICDM '10, 935–940.

APPROXIMATING THE BOOLEAN RANK

- Recall: we have log(*n*) approximation given an oracle
- We say *n*-by-*m* binary matrix ***A*** is log(*n*) uniformly sparse if each column of A has at most log(*n*) 1s

Theorem [1]**.** The Boolean rank of a log(*n*) uniformly sparse binary matrix A can be approximated within log(*n*).

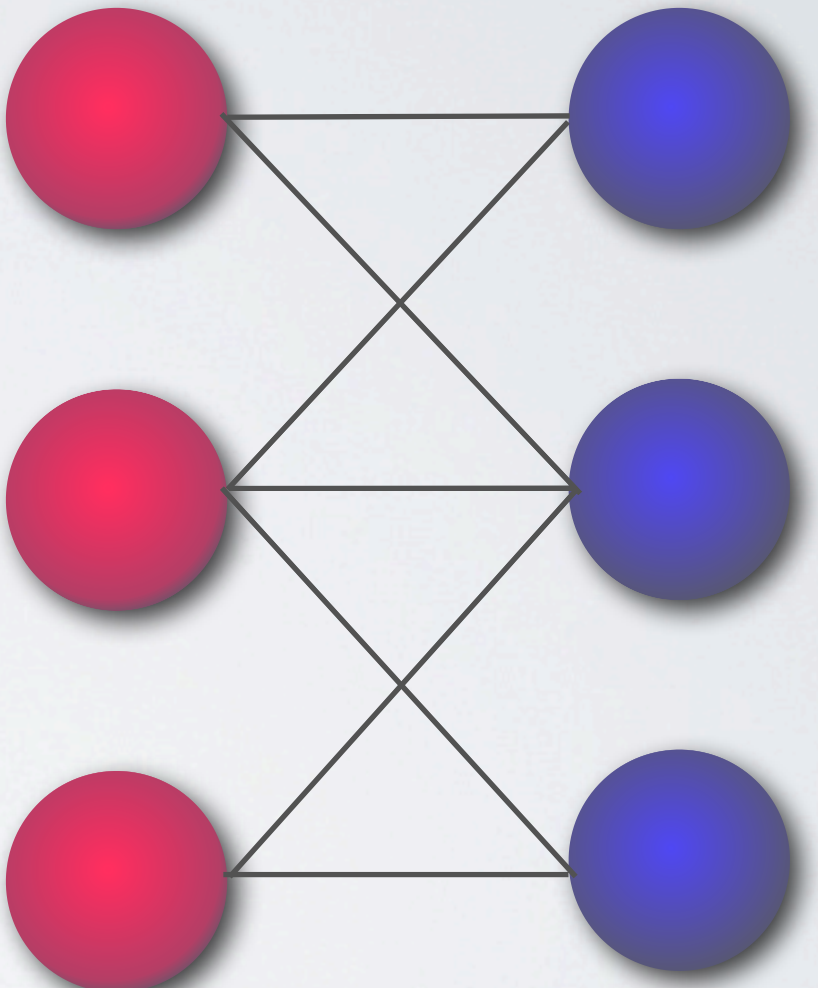
[1] M., Sparse Boolean Matrix Factorizations, in: ICDM '10, 935–940.

PROOF

- Each RHS node has ≤ log(n) neighbours
  ⇒ Optimum solution needs

  ≥ n/log(n) bicliques

- If we use n bicliques we get
  n/OPT ≤ n/(n/log(n))
  = log(n) □



EXTENSIONS

- We can approximate the Maximum k-tiling for $\log(n)$ uniformly sparse matrices within $e/(e-1)$
- If we have at most $\log(n)$ columns that have more than $\log(n)$ 1s, we can still approximate the rank within $\log^2(n)$
  - Both results require more complex reduction to the Set Cover problem [1]
  - Will also work on dense matrices, but will take exponential time

[1] R. Bělohlávek, V. Vychodil, Discovery of optimal factors in binary data via a novel method of matrix decomposition, *J. Comput. Syst. Sci.* 76 (2010) 3–20.

OPEN PROBLEMS

?

ALGORITHMS

- **P2.1** Are there good algorithms for XMF?
- **P2.2** Can we use the sparsity to really help us?

MODEL ORDER SELECTION

- **P2.3** How hard is it to minimise the MDL score directly?
  - Depends on the encoding, obviously
- **P2.4** Can we use binary methods to predict missing values and would these be better than continuous methods?


PP/poly
P^{#P[1]}
BP.PP
MP
PH
SF_4
Sigma_3P
AmpMP
1
2
3
The End
SUBEXP
P^{FewP}
QP
Few
EP
EQP
betaP
QPLIN
FewP
BPP

BIBLIOGRAPHY

Image by Lars Aronsson, Wikipedia

BACKGROUND

Beasley, L.B. & Guterman, A.E., 2005. Rank inequalities over semirings. *Journal of the Korean Mathematical Society*, 42(2), pp.223–241.

Beasley, L.B. & Pullman, N.J., 1984. Boolean-rank-preserving operators and Boolean-rank-1 spaces. *Linear Algebra and its Applications*, 59, pp.55–77.

Beasley, L.B. & Pullman, N.J., 1988. Semiring rank versus column rank. *Linear Algebra and its Applications*, 101, pp.33–48.

Cayley, A., 1858. A memoir on the theory of matrices. *Philosophical transactions of the Royal society of London*, 148, pp.17–37.

Doherty, F.C.C., Lundgren, J.R. & Siewert, D.J., 1999. Biclique covers and partitions of bipartite graphs and digraphs and related matrix ranks of {0, 1}-matrices. *Congressus Numerantium*, 136, pp. 73–96.

Gregory, D.A. & Pullman, N.J., 1983. Semiring rank: Boolean rank and nonnegative rank factorizations. *Journal of Combinatorics, Information & System Sciences*, 8(3), pp.223–233.

Kim, K.H., 1982. *Boolean matrix theory and applications*, New York: Marcel Dekker.

Miettinen, P., 2009. *Matrix Decomposition Methods for Data Mining: Computational Complexity and Algorithms*. Department of Computer Science, University of Helsinki.

Monson, S.D., Pullman, N.J. & Rees, R., 1995. A Survey of Clique and Biclique Coverings and Factorizations of (0,1)-Matrices. *Bulletin of the ICA,* 14, pp.17–86.

Peirce, C.S., 1873. Description of a notation for the logic of relatives, resulting from an amplification of the conceptions of Boole's calculus of logic. *Memoirs of the American academy of arts and sciences*, 9(2), pp.317–378.

COMPLEXITY

Alon, N., Panigrahy, R. & Yekhanin, S., 2009. Deterministic Approximation Algorithms for the Nearest Codeword Problem. In 12th Intl. Workshop on Approximation Algorithms for Combinatorial Optimization Problems and 13th Intl. Workshop on Randomization and Computation. Springer Berlin Heidelberg, pp. 339–351.

Amaldi, E. & Kann, V., 1995. The complexity and approximability of finding maximum feasible subsystems of linear relations. *Theoretical Computer Science*, 147(1-2), pp.181–210.

Amilhastre, J., Vilarem, M.-C. & Janssen, P., 1998. Complexity of Minimum Biclique Cover and Minimum Biclique Decomposition for Bipartite Domino-free Graphs. *Discrete Applied Mathematics*, 86(2-3), pp.125–144.

Arora, S. et al., 1993. The Hardness of Approximate Optima in Lattices, Codes, and Systems of Linear Equations. In 34th Annual IEEE Symposium on Foundations of Computer Science. IEEE, pp. 724–733.

Binkele-Raible, D. et al., 2010. Exact exponential-time algorithms for finding bicliques. *Information Processing Letters*, 111(2).

Carr, R.D. et al., 2000. On the Red-Blue Set Cover Problem. In 11th Annual ACM-SIAM symposium on Discrete algorithms. pp. 345–353.

Dinur, I. et al., 2003. Approximating CVP to Within Almost-Polynomial Factors is NP-Hard. *Combinatorica*, 23(2), pp.205–243.

Hochbaum, D.S., 1998. Approximating clique and biclique problems. *Journal of Algorithms*, 29(1), pp. 174–200.

Miettinen, P., 2009. *Matrix Decomposition Methods for Data Mining: Computational Complexity and Algorithms*. Department of Computer Science, University of Helsinki.

Miettinen, P., 2008. On the positive-negative partial set cover problem. *Information Processing Letters*, 108(4), pp.219–221.

Nau, D.S. et al., 1978. A Mathematical Analysis of Human Leukocyte Antigen Serology. *Mathematical Biosciences*, 40, pp.243–270.

Orlin, J., 1977. Contentment in graph theory: covering graphs with cliques. *Indagationes Mathematicae*, 80(5), pp.406–424.

Peeters, R., 2003. The maximum edge biclique problem is NP-complete. *Discrete Applied Mathematics*, 131(3), pp.651–654.

Peleg, D., 2007. Approximation algorithms for the Label-Cover$_{\text{MAX}}$ and Red-Blue Set Cover problems. *Journal of Discrete Algorithms*, 5(1), pp.55–64.

Simon, H.U., 1990. On approximate solutions for combinatorial optimization problems. *SIAM Journal on Discrete Mathematics*, 3(2), pp.294–310.

Stockmeyer, L.J., 1975. The Set Basis Problem is NP-complete, IBM Thomas J. Watson Research Center.

ALGORITHMS

Bělohlávek, R. & Vychodil, V., 2010. Discovery of optimal factors in binary data via a novel method of matrix decomposition. *Journal of Computer and System Sciences*, 76(1), pp.3–20.

Koyutürk, M. & Grama, A., 2003. PROXIMUS: a framework for analyzing very high dimensional discrete-attributed datasets. In 9th ACM SIGKDD international conference on Knowledge discovery and data mining.

Lu, H., 2011. *Boolean matrix decomposition and extension with applications*. Rutgers University.

Lu, H., Vaidya, J. & Atluri, V., 2008. Optimal Boolean Matrix Decomposition: Application to Role Engineering. In 24th IEEE International Conference on Data Engineering. pp. 297–306.

Lucchese, C., Orlando, S. & Perego, R., 2010. Mining Top-K Patterns from Binary Datasets in presence of Noise. In 2010 SIAM International Conference on Data Mining. pp. 165–176.

Miettinen, P. et al., 2008. The Discrete Basis Problem. *IEEE Transactions on Knowledge and Data Engineering*, 20(10), pp.1348–1362.

Peleg, D., 2007. Approximation algorithms for the Label-Cover$_{MAX}$ and Red-Blue Set Cover problems. *Journal of Discrete Algorithms*, 5(1), pp.55–64.

Shen, B.-H., Ji, S. & Ye, J., 2009. Mining Discrete Patterns via Binary Matrix Factorization. In 15th ACM SIGKDD international conference on Knowledge discovery and data mining. New York, New York, USA: ACM Press, pp. 757–765.

Zhang, Z.-Y. et al., 2010. Binary matrix factorization for analyzing gene expression data. *Data Mining and Knowledge Discovery*, 20(1), pp.28–52.

MISCELLANEOUS

Bělohlávek, R. & Vychodil, V., 2010. Discovery of optimal factors in binary data via a novel method of matrix decomposition. *Journal of Computer and System Sciences*, 76(1), pp.3–20.

Cattell, R.B., 1966. The Scree Test For The Number Of Factors. *Multivariate Behavioral Research*, 1(2), pp.245–276.

Couturier, J.-F. & Kratsch, D., 2012. Bicolored independent sets and bicliques. *Information Processing Letters*, 112(8-9), pp.329–334.

Frank, M., Chehreghani, M.H. & Buhmann, J.M., 2011. The Minimum Transfer Cost Principle for Model-Order Selection. In D. Hutchison et al., eds. 2011 European Conference on Machine Learning and Knowledge Discovery in Databases – Part III. pp. 423–438.

Miettinen, P., 2010. Sparse Boolean Matrix Factorizations. In 10th IEEE International Conference on Data Mining. pp. 935–940.

Miettinen, P. & Vreeken, J., 2011. Model Order Selection for Boolean Matrix Factorization. In 20th ACM SIGKDD international conference on Knowledge discovery and data mining. pp. 51–59.

Owen, A.B. & Perry, P.O., 2009. Bi-cross-validation of the SVD and the nonnegative matrix factorization. *Annals of Applied Statistics*, 3(2), pp.564–594.

Yeomans, K.A. & Golder, P.A., 1982. The Guttman–Kaiser criterion as a predictor of the number of common factors. *The Statistician*, 31(3), pp.221–229.

RELATED WORK

Besson, J. et al., 2006. Constraint-Based Mining of Fault-Tolerant Patterns from Boolean Data. In 4th International Workshop on Knowledge Discovery in Inductive Databases. pp. 55–71.

Geerts, F., Goethals, B. & Mielikäinen, T., 2004. Tiling databases. In 7th International Conference on Discovery Science. pp. 77–122.

Gionis, A., Mannila, H. & Seppänen, J.K., 2004. Geometric and Combinatorial Tiles in 0–1 Data. In 8th European Conference on Principles and Practice of Knowledge Discovery in Databases. pp. 173–184.

Hochbaum, D.S., 1998. Approximating clique and biclique problems. *Journal of Algorithms*, 29(1), pp. 174–200.

Kontonasios, K.-N. & De Bie, T., 2010. An information-theoretic approach to finding informative noisy tiles in binary databases. In 2010 SIAM International Conference on Data Mining. pp. 153–164.

Lempel, A., 1975. Matrix Factorization over GF(2) and Trace-Orthogonal Bases of GF(2^n). *SIAM Journal on Computing,* 4(2), pp.175–186.

Lu, H. et al., 2012. Constraint-Aware Role Mining Via Extended Boolean Matrix Decomposition. *IEEE Transactions on Dependable and Secure Computing*.

Lu, H. et al., Extended Boolean Matrix Decomposition. In 9th IEEE International Conference on Data Mining. IEEE, pp. 317–326.

Miettinen, P., 2012. On Finding Joint Subspace Boolean Matrix Factorizations. In 2012 SIAM International Conference on Data Mining. pp. 954–965.

Miettinen, P., 2008. The Boolean Column and Column-Row Matrix Decompositions. *Data Mining and Knowledge Discovery*, 17(1), pp.39–56.

Pensa, R.G. & Boulicaut, J.-F., 2005. Towards fault-tolerant formal concept analysis. In 2005 Advances in Artificial Intelligence. pp. 212–223.

Streich, A. et al., 2009. Multi-assignment clustering for Boolean data. In 26th Annual International Conference on Machine Learning.

Vaidya, J. et al., 2009. Edge-RMP: Minimizing administrative assignments for role-based access control. *Journal of Computer Security*, 17(2), pp.211–235.

Wicker, J., Pfahringer, B. & Kramer, S., 2012. Multi-Label Classification Using Boolean Matrix Decomposition. In 27th Annual ACM Symposium on Applied Computing. New York, New York, USA: ACM Press, pp. 179–186.

Xiang, Y. et al., 2011. Summarizing transactional databases with overlapped hyperrectangles. *Data Mining and Knowledge Discovery*, 23(2), pp.215–251.