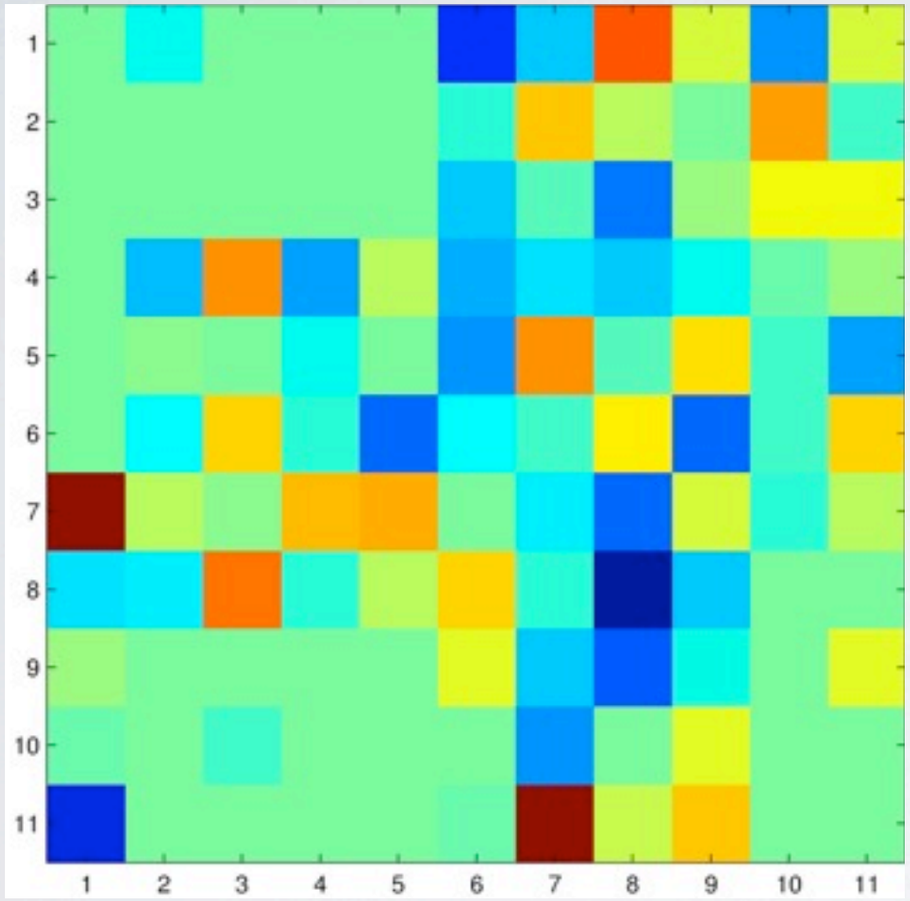
BOOLEAN MATRIX FACTORIZATIONS

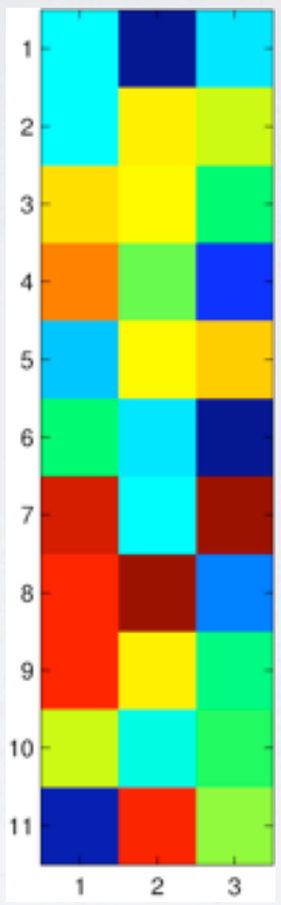
Pauli Miettinen

Leap day, 2012

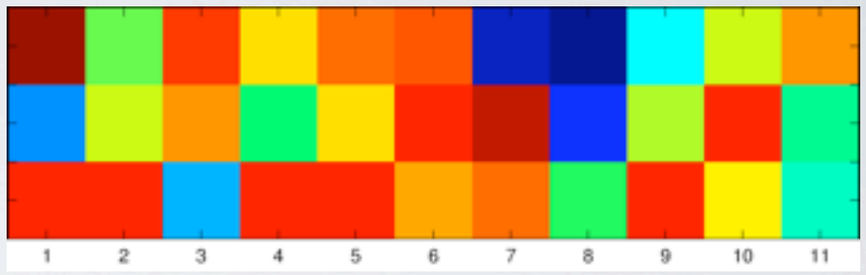
MATRIX FACTORIZATIONS



≈



×

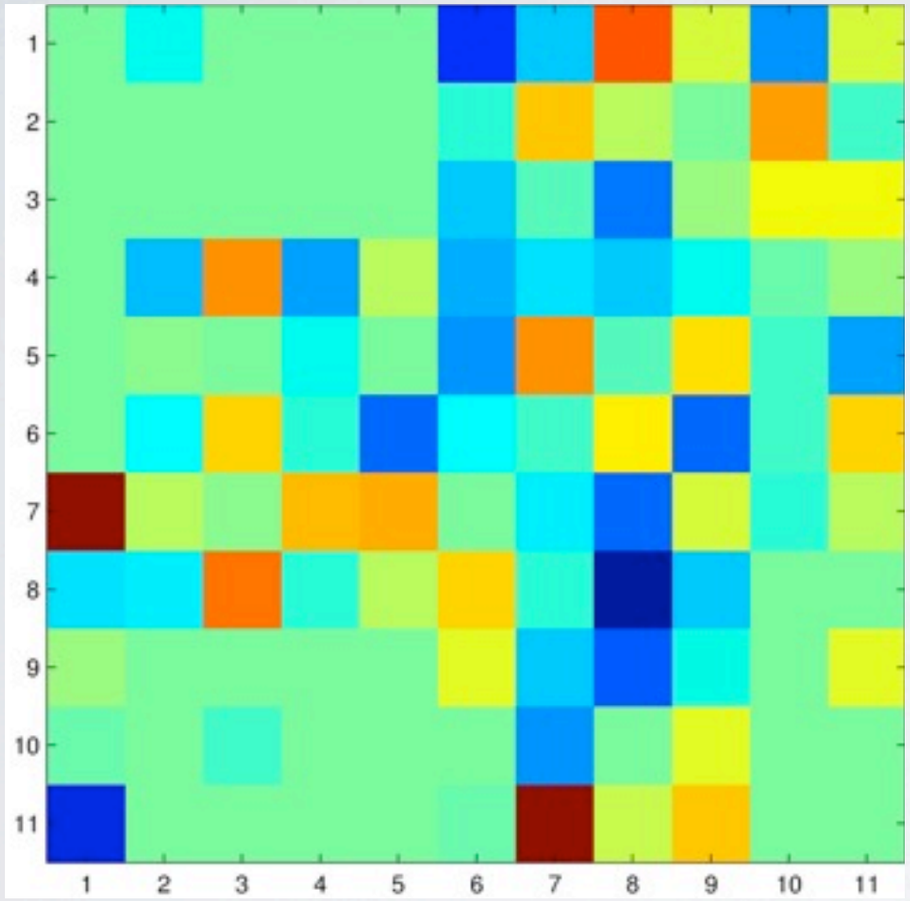


MATRIX FACTORIZATIONS

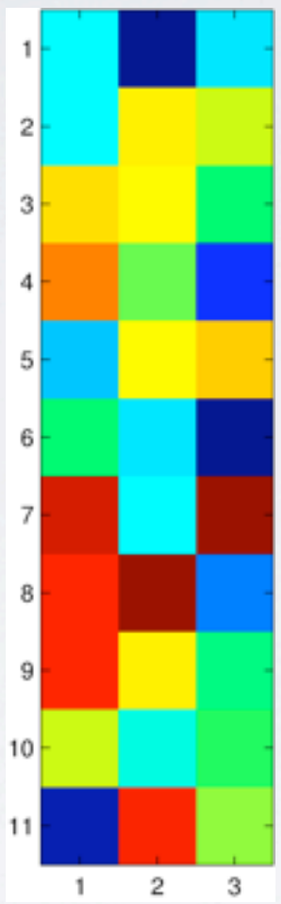
- A **factorization** of matrix ***X*** represents it as a product of two (or more) **factor matrices**: ***X*** = ***AB***
  - ***X*** is *n*-by-*m*, ***A*** is *n*-by-*k*, and ***B*** is *k*-by-*m*
    - *k* is the **size** (or **rank**) of the factorization
- Factorization can be **exact** (***X*** = ***AB***) or **approximate** (***X*** ≈ ***AB***)

mpii max planck institut informatik

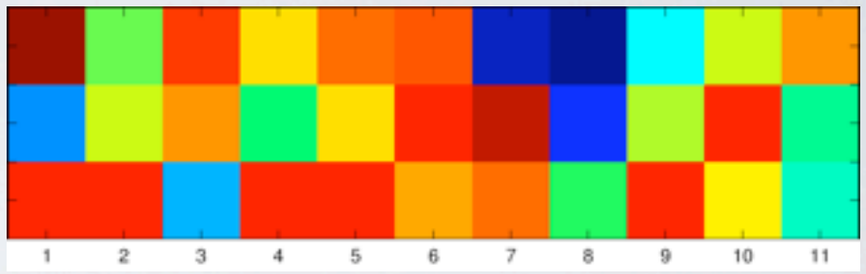
MATRIX FACTORIZATIONS



≈



×



Factor matrices

Rank = 3

max planck institut informatik

SOME LINEAR ALGEBRA

- A set of vectors is *linearly independent* if no vector in the set can be expressed as a linear combination of the others

- A matrix $\boldsymbol{X}$ is orthogonal if and only if $\boldsymbol{X}\boldsymbol{X}^T = \boldsymbol{X}^T\boldsymbol{X} = \boldsymbol{I}$

- The **column rank** of a matrix is the number of linearly independent columns it has

  - Equals the row rank of the matrix
    ⇒ **the rank** of a matrix is its column rank = row rank

ON MATRIX RANK

- Matrix $\boldsymbol{X}$ has rank($\boldsymbol{X}$) = 1 iff $\boldsymbol{X} = \boldsymbol{a}\boldsymbol{b}^T$
  - Outer product of column vectors $\boldsymbol{a}$ and $\boldsymbol{b}$
- Matrix $\boldsymbol{X}$ has rank($\boldsymbol{X}$) ≤ k if it can be represented as a sum of k rank-1 matrices
  - Smallest such k is the rank of $\boldsymbol{X}$
- Equivalently, rank($\boldsymbol{X}$) ≤ k iff there is a rank-k factorization of $\boldsymbol{X}$
  - $\mathbf{X} = \sum_{i=1}^{k} \mathbf{a}_i \mathbf{b}_i^T = \mathbf{A}\mathbf{B}$

max planck institut informatik

MATRIX DISTANCES

- The Frobenius norm: $\|\mathbf{X}\|_F = \sqrt{\sum_{i=1}^{n} \sum_{j=1}^{m} x_{ij}^2}$
  - We drop the *F* in Frobenius for now…
- The sum of absolute values: $|\mathbf{X}| = \sum_{i=1}^{n} \sum_{j=1}^{m} |x_{ij}|$
  - If $\boldsymbol{X}$ is binary, $|\boldsymbol{X}| = ||\boldsymbol{X}||^2$

FAMOUS MATRIX FACTORIZATIONS

- Eigendecomposition: $\boldsymbol{X} = \boldsymbol{Q}\Lambda\boldsymbol{Q}^T$
  - $\boldsymbol{X}$ is square; $\boldsymbol{Q}$ is orthogonal with the eigenvectors of $\boldsymbol{X}$; Λ is diagonal and has the eigenvalues
- Singular value decomposition: $\boldsymbol{X} = \boldsymbol{U}\Sigma\boldsymbol{V}^T$
  - $\boldsymbol{U}$ and $\boldsymbol{V}$ are orthogonal, Σ is diagonal with the singular values
- Non-negative matrix factorization: $\boldsymbol{X} = \boldsymbol{W}\boldsymbol{H}$
  - All matrices are non-negative

OTHER FAMOUS MATRIX FACTORIZATIONS

- *k*-means clustering
- tiling databases

K-MEANS AS MATRIX FACTORIZATION

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

$$\sum_{i=1}^{k} \sum_{x_j \in C_i} \|x_j - \mu_i\|_2^2$$

is minimized — Distance of data to its cluster centroid

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

TILING AS MATRIX FACTORIZATION

- Maximum *k*-tiling: find at most *k* **tiles** such that the tiling has maximum area
  - 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*
  - Minimize the number of 1s in the data that do not belong to any tile

TILING AS MATRIX FACTORIZATION

- 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
  - Minimize $|\boldsymbol{X} - \boldsymbol{AB}|$
- 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?

max planck institut informatik

COMBINING THE TILES

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

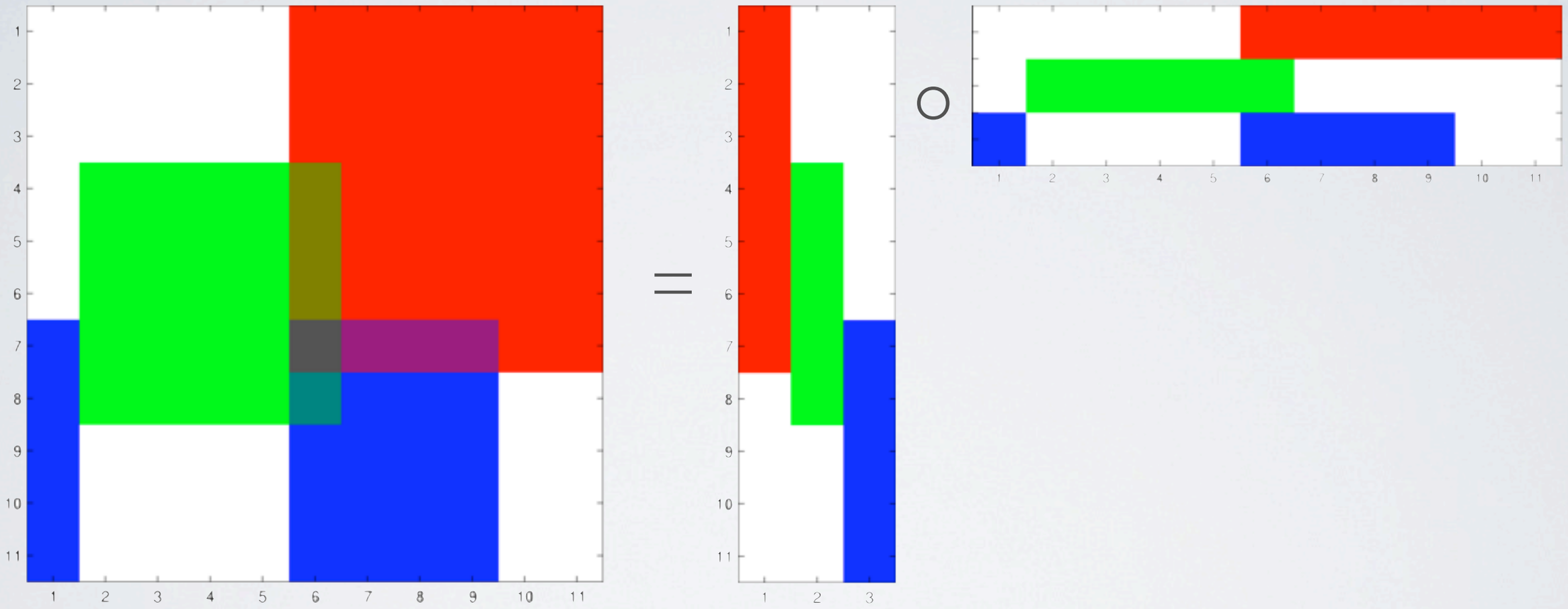
THE BOOLEAN MATRIX PRODUCT

- As normal matrix product, but with addition defined as $1+1=1$ (logical OR)
- Closed under binary matrices
- Corresponds to set union operation

$$(\mathbf{X} \circ \mathbf{Y})_{ij} = \bigvee_{l=1}^{k} x_{il} y_{lj}$$

max planck institut informatik

THE BOOLEAN MATRIX PRODUCT

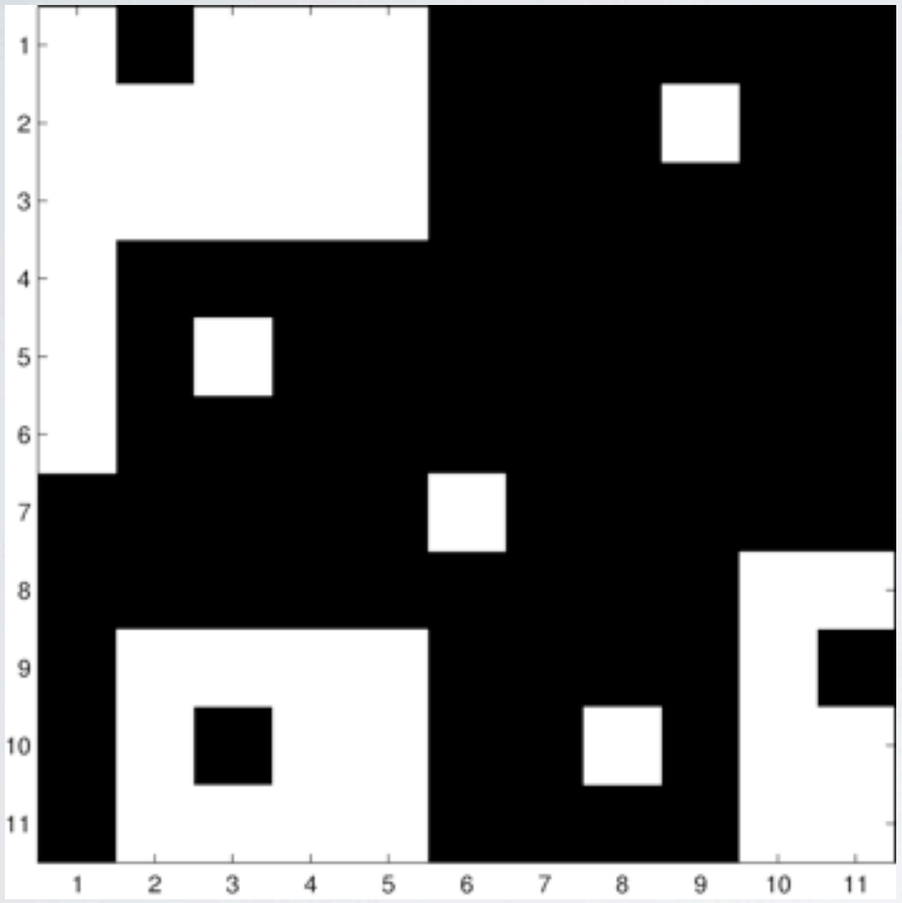


max planck institut
informatik

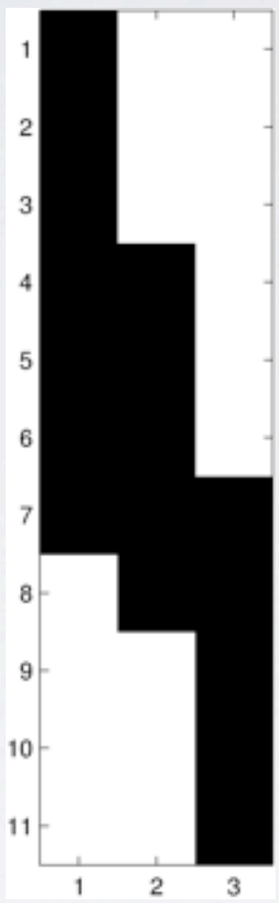
TILING REVISITED

- Given transaction data as an n-by-m binary matrix $\boldsymbol{X}$ and integer k, find binary matrices $\boldsymbol{A}$ (n-by-k) and $\boldsymbol{B}$ (k-by-m) such that if $(\boldsymbol{A} \circ \boldsymbol{B})_{ij} = 1$, then $\boldsymbol{X}_{ij} = 1$ and $|\boldsymbol{X} - \boldsymbol{A} \circ \boldsymbol{B}|$ is minimized
  - Requirement makes sure that tiles have only 1s that appear in the data
  - What happens if we remove this restriction?

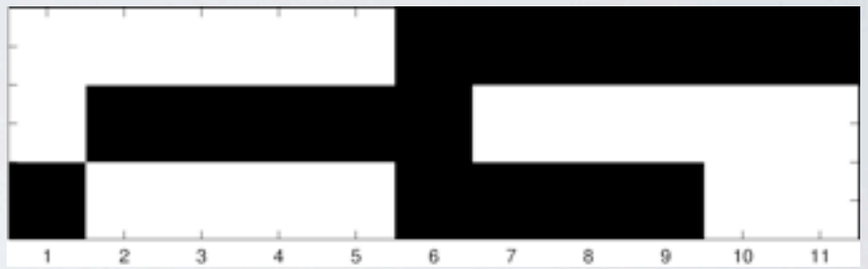
BOOLEAN MATRIX FACTORIZATIONS



≈



○



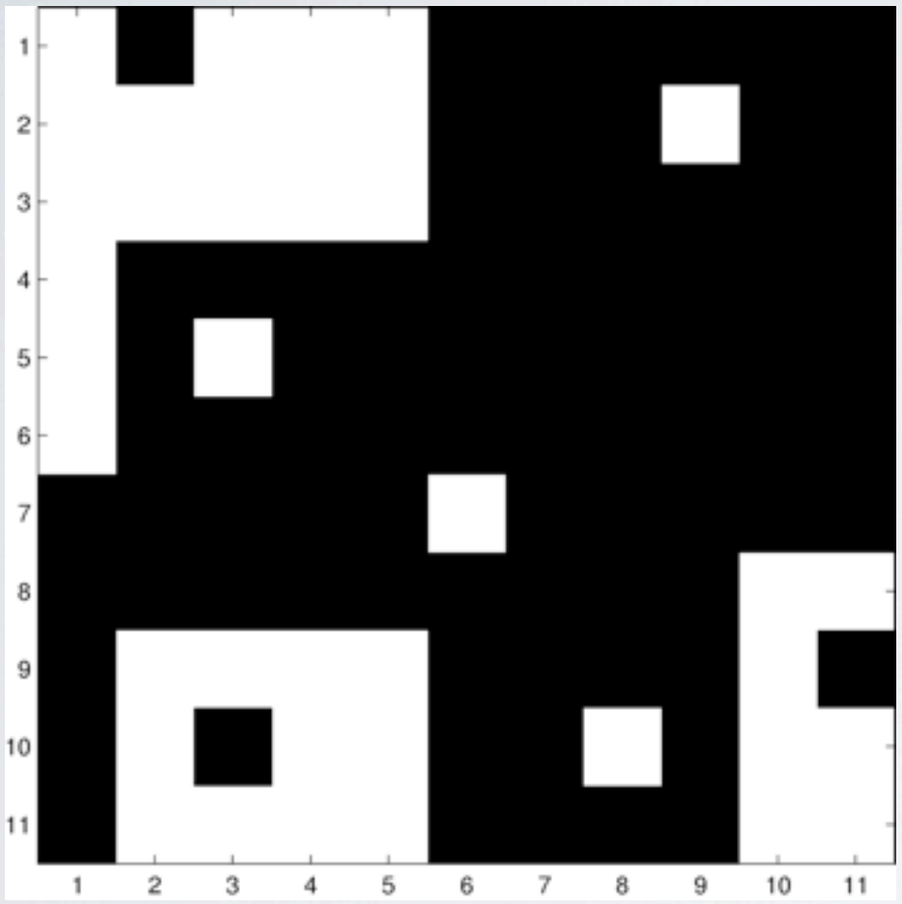
BOOLEAN MATRIX FACTORIZATIONS

Definition (BMF). Given an *n*-by-*m* binary matrix **A** and non-negative integer *k*, find *n*-by-*k* binary matrix **B** and *k*-by-*m* binary matrix **C** such that they minimize

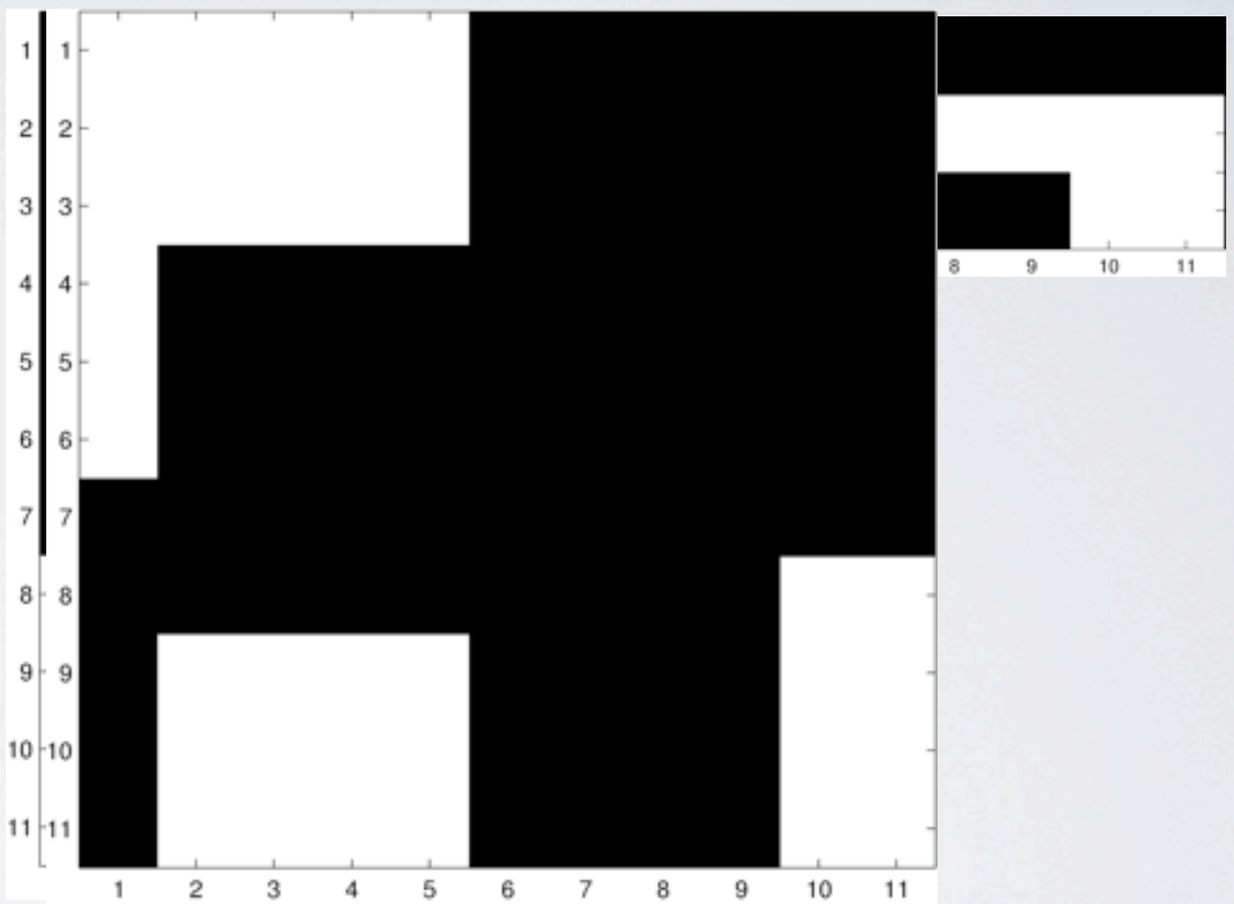
$$|\mathbf{A} \otimes (\mathbf{B} \circ \mathbf{C})| = \sum_{i,j} |a_{ij} - (\mathbf{B} \circ \mathbf{C})_{ij}|$$

max planck institut informatik

BOOLEAN MATRIX FACTORIZATIONS



≈



max planck institut informatik

WHAT ABOUT DATA MINING?

- Factors provide groups of objects that 'go together'
  - Everything is binary ⇒ factors are sets (unlike NMF or SVD)
- Factors can overlap (unlike clustering)
- Provides a global view (unlike frequent item sets)
- Allows missing ones and zeros (unlike tiling)

BMF: A DM EXAMPLE

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

BMF: A DM EXAMPLE

long-haired
well-known
male

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

BMF: A DM EXAMPLE

Alice & Bob: long-haired and well-known
Bob & Charles: well-known males

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

long-haired
well-known
male

A B C

SOME APPLICATIONS

- Explorative data mining
  - Factors tell something about the data
- Role mining
  - Naïve approach not very good
- Entity disambiguation / synonym finding
  - Allows synonymity and polysemy
  - Might need tensors

SOME THEORY

max planck institut
informatik

BOOLEAN RANK

Matrix rank. The **rank** of an *n*-by-*m* matrix $\boldsymbol{A}$ is the least integer *k* such that there exists *n*-by-*k* matrix $\boldsymbol{B}$ and *k*-by-*m* matrix $\boldsymbol{C}$ for which $\boldsymbol{A} = \boldsymbol{BC}$.

Boolean matrix rank. The **Boolean rank** of an *n*-by-*m* binary matrix $\boldsymbol{A}$ is the least integer *k* such that there exists *n*-by-*k* binary matrix $\boldsymbol{B}$ and *k*-by-*m* binary matrix $\boldsymbol{C}$ for which $\boldsymbol{A} = \boldsymbol{B} \circ \boldsymbol{C}$.

SOME PROPERTIES OF BOOLEAN RANK

- For some matrices, Boolean rank is higher than normal rank
  - Twice the normal rank is the biggest known difference
- For some matrices, Boolean rank is much smaller
  - Can be a logarithm of the normal rank
  - **Boolean matrix factorization can have smaller reconstruction error than SVD of same size**

AN EXAMPLE

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

Original matrix

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

Exact Boolean rank-2 decomposition

$$\approx \begin{pmatrix} 1/2 & 1/\sqrt{2} \\ 1/\sqrt{2} & 0 \\ 1/2 & -1/\sqrt{2} \end{pmatrix} \begin{pmatrix} \frac{\sqrt{2}+1}{2} & \frac{\sqrt{2}+2}{2} & \frac{\sqrt{2}+1}{2} \\ 1/\sqrt{2} & 0 & 1/\sqrt{2} \end{pmatrix}$$

The best approximate normal rank-2 decomposition

COMPUTATIONAL COMPLEXITY

- Approximating the Boolean rank is as hard as approximating the minimum chromatic number of a graph
  - Read: hard to even approximate
  - Except with some sparse matrices; more on that later

COMPUTATIONAL COMPLEXITY

- Finding minimum-error BMF is NP-hard
- NP-hard to approximate within any poly computable factor
  - Because best answer = 0 is NP-hard to recognize
- NP-hard to approximate within additive error of $n^{1/4}$

A SUBPROBLEM AND ITS COMPLEXITY

Basis Usage (BU). Given binary matrices $\boldsymbol{A}$ and $\boldsymbol{B}$, find a binary matrix $\boldsymbol{C}$ that minimizes $|\boldsymbol{A} - \boldsymbol{B} \circ \boldsymbol{C}|$.

- Corresponds to a problem where $\boldsymbol{A}$ and $\boldsymbol{C}$ are just column vectors
- Error NP-hard to approximate better than in superpolylogarithmic factor

$$\Omega\left(2^{\log^{1-\varepsilon}|\mathbf{a}|}\right)$$

AN ALGORITHM

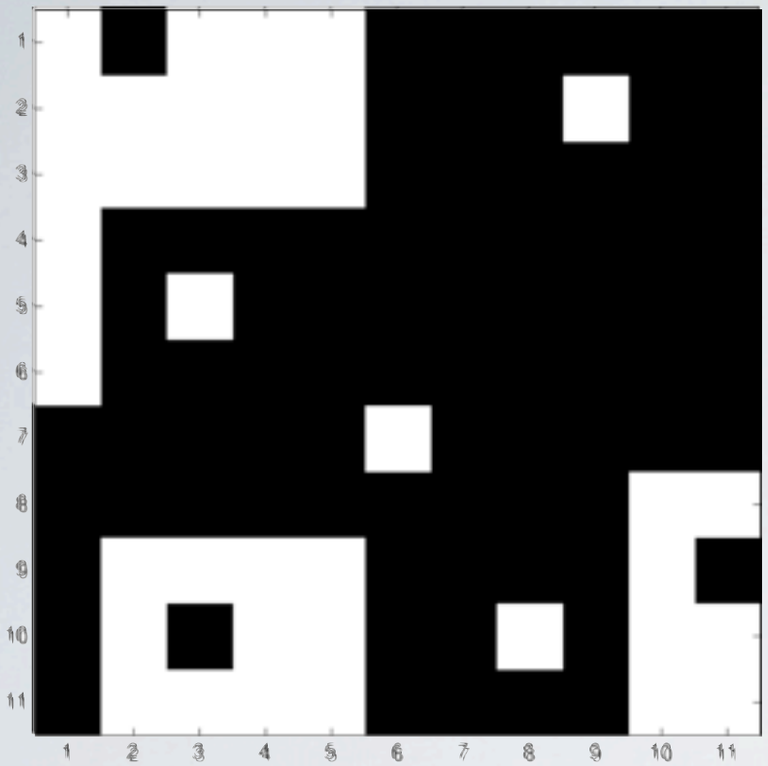
max planck institut
informatik

THE ASSO ALGORITHM

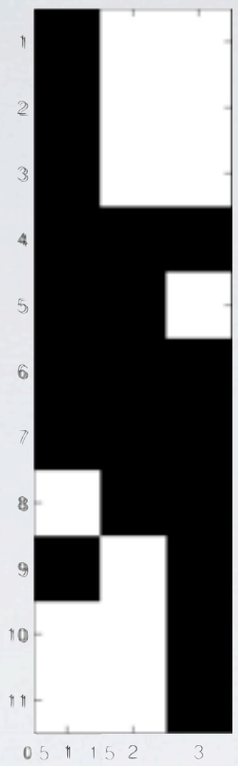
- Heuristic – too many hardness results to hope for good provable results in any case
- Intuition: If two columns share a factor, they have 1s in same rows
  - Noise makes detecting this harder
  - Pairwise row association rules reveal (some of) the factors

THE ASSO ALGORITHM

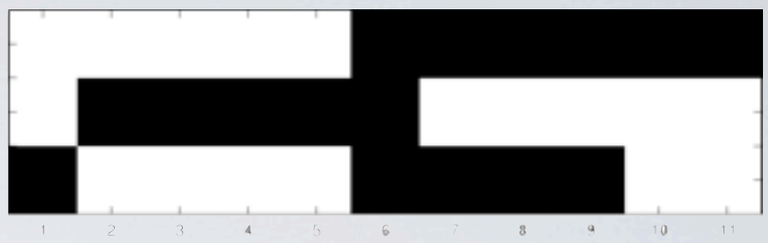
1. Compute pairwise association accuracies between rows of **A**
2. Round these (from a user-defined point *t*) to get a binary *n*-by-*n* matrix of candidate columns
3. Select greedily the candidate column that covers most of the not-yet covered 1s of **A**
4. Mark the 1s covered by the selected vector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

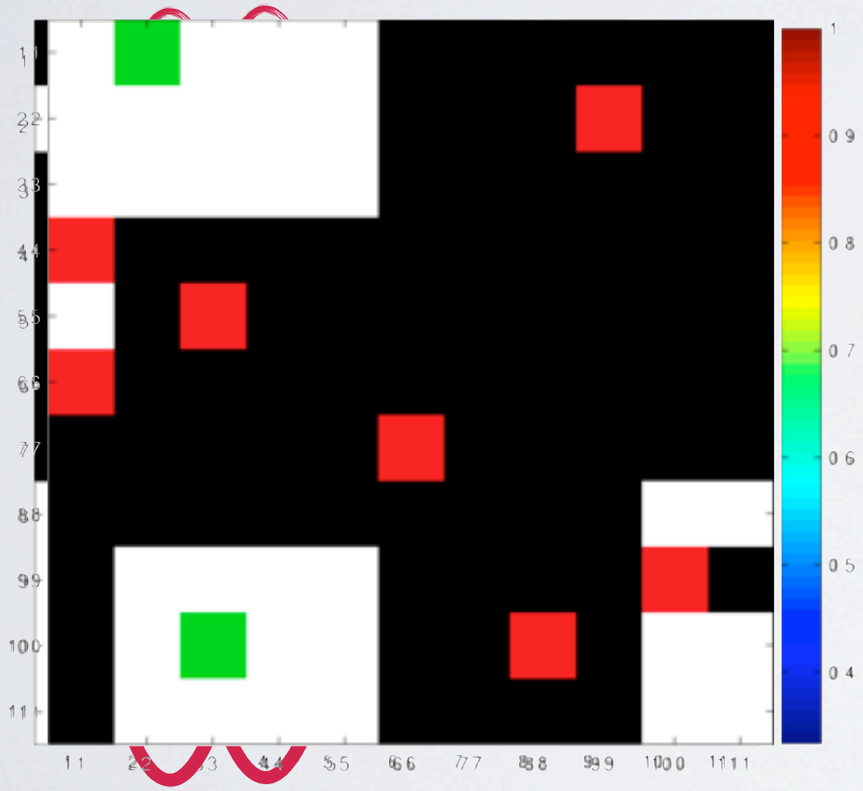


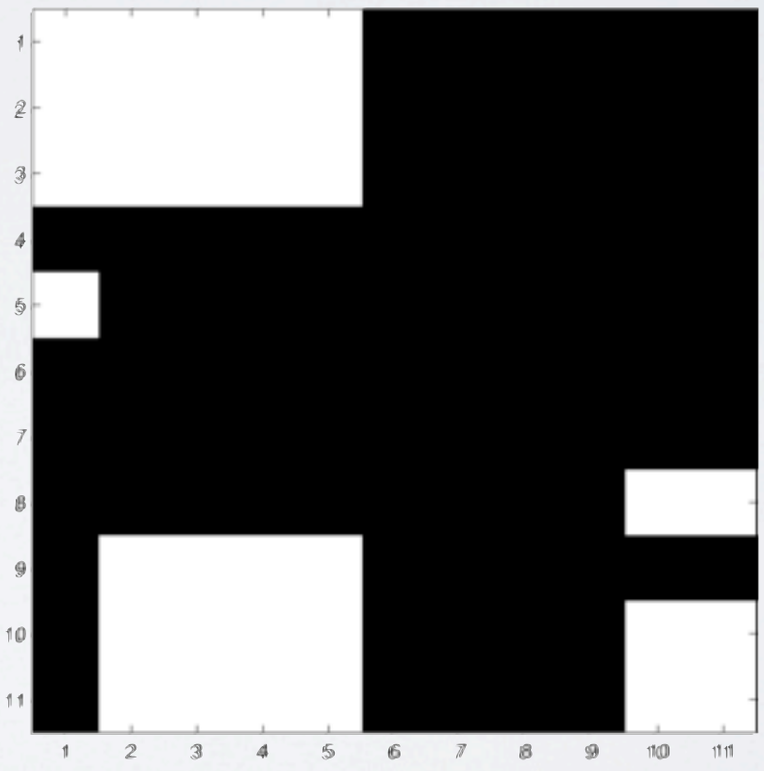
≈



∘







max planck institut informatik

SPARSE MATRICES

max planck institut
informatik

MOTIVATION

- Many real-world data are sparse
- With sparse input, we hope for sparse output (factors)
- Sparsity should also help with computational complexity
  - Less degrees of freedom

SPARSE FACTORIZATIONS

- Ideally, sparse matrices have sparse factors
  - Not true with many factorization methods
  - Sparse Boolean matrices have sparse decompositions

Theorem 1. For any n-by-m 0/1 matrix $\mathbf{A}$ of Boolean rank k, there exist n-by-k and k-by-m 0/1 matrices $\mathbf{B}$ and $\mathbf{C}$ such that $\mathbf{A}=\mathbf{B}\circ\mathbf{C}$ and $|\mathbf{B}|+|\mathbf{C}|\leq 2|\mathbf{A}|$.

max planck institut informatik

APPROXIMATING BOOLEAN RANK IN SPARSE MATRICES

- Intuition: Sparse matrices cannot have as complex structure as dense matrices – rank could be easier to approximate
- Recently, Belohlavek and Vychodil (2010) proposed a reduction to Set Cover, giving $O(\log n)$ approximation
  - Can yield exponential increase in instance size
- Sparsity helps!

max planck institut informatik

APPROXIMATING THE BOOLEAN RANK

- Sparsity is not enough; we need some structure in it
- An *n*-by-*m* 0/1 matrix **A** is *f(n)*-uniformly sparse, if all of its columns have at most *f(n)* 1s

Theorem 2. The Boolean rank of log*(n)*-uniformly sparse matrix can be approximated to within *O(*log*(m))* in time $\tilde{O}(m^2 n)$.

max planck institut informatik

NON-UNIFORMLY SPARSE MATRICES

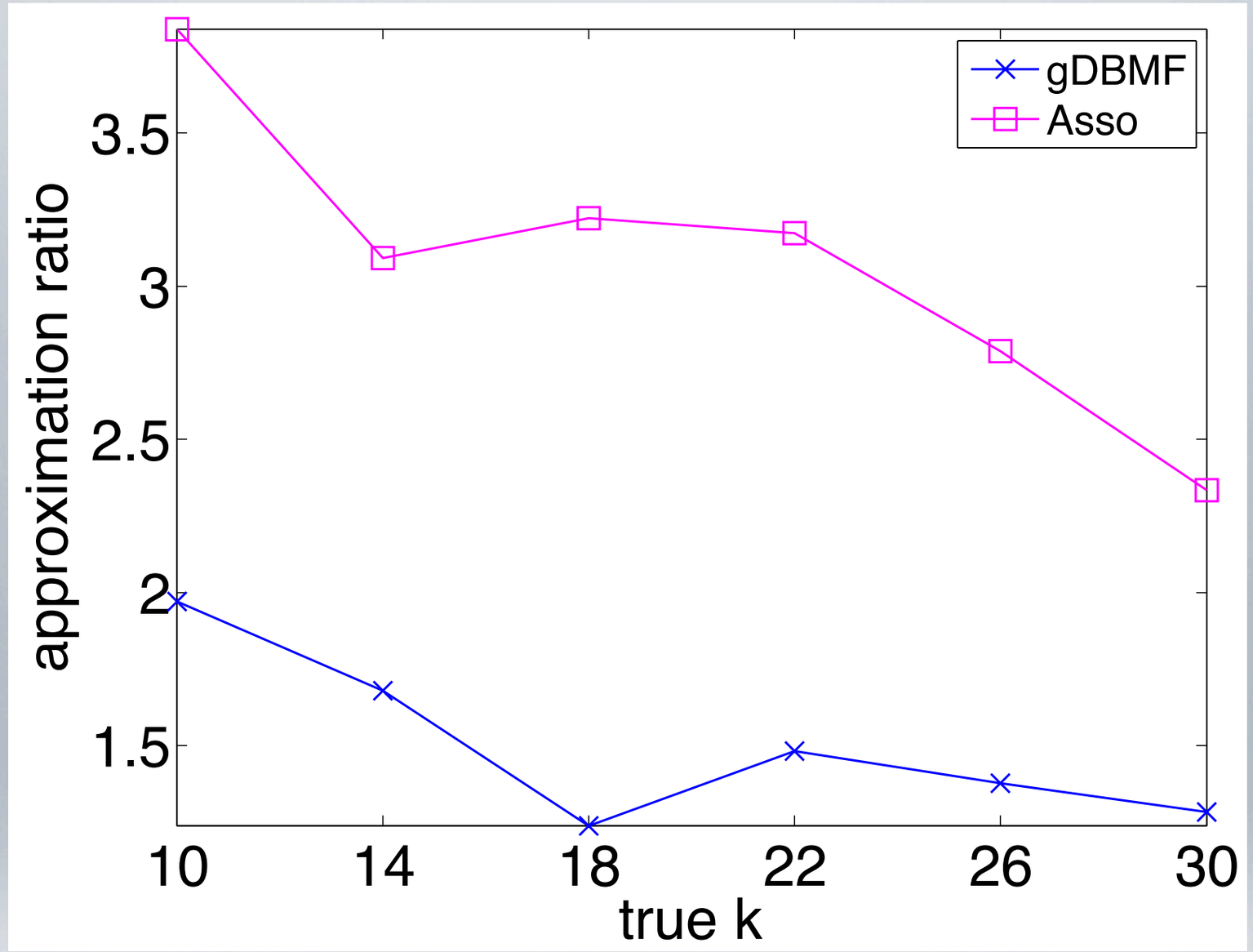
- Uniform sparsity is very restricted; what can we do
  - Trade non-uniformity with approximation accuracy

Theorem 3. If there are at most $\log(m)$ columns with more than $\log(n)$ 1s, then we can approximate the Boolean rank in polynomial time to within $O(\log^2(m))$.

APPROXIMATING DOMINATED COVERS

Theorem 4. If n-by-m 0/1 matrix $\mathbf{A}$ is $O(\log n)$-uniformly sparse, we can approximate the best dominated k-cover of $\mathbf{A}$ by $e/(e-1)$ in polynomial time.

- Dominated k-cover: The rank is k and if $(\mathbf{B} \circ \mathbf{C})_{ij} = 1$, then $\mathbf{A}_{ij} = 1$
  - This is tiling!



APPROXIMATING THE RANK

MODEL ORDER SELECTION

max planck institut
informatik

HOW DO I KNOW WHAT *K* TO USE?

Definition (BMF). Given an *n*-by-*m* binary matrix **A** and non-negative integer *k*, find *n*-by-*k* binary matrix **B** and *k*-by-*m* binary matrix **C** such that they minimize

$$|\mathbf{A} \otimes (\mathbf{B} \circ \mathbf{C})| = \sum_{i,j} |a_{ij} - (\mathbf{B} \circ \mathbf{C})_{ij}|$$

N.B. This is nothing special to BMF!

PRINCIPLES OF GOOD *K*

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

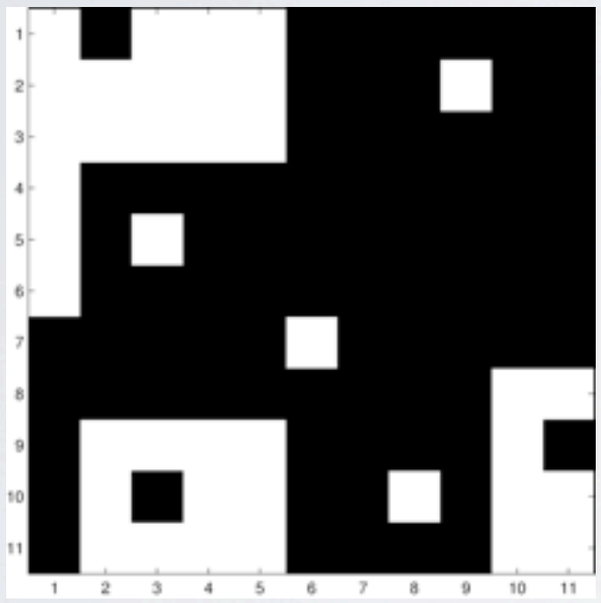
ENTER MDL

THE MINIMUM DESCRIPTION LENGTH PRINCIPLE

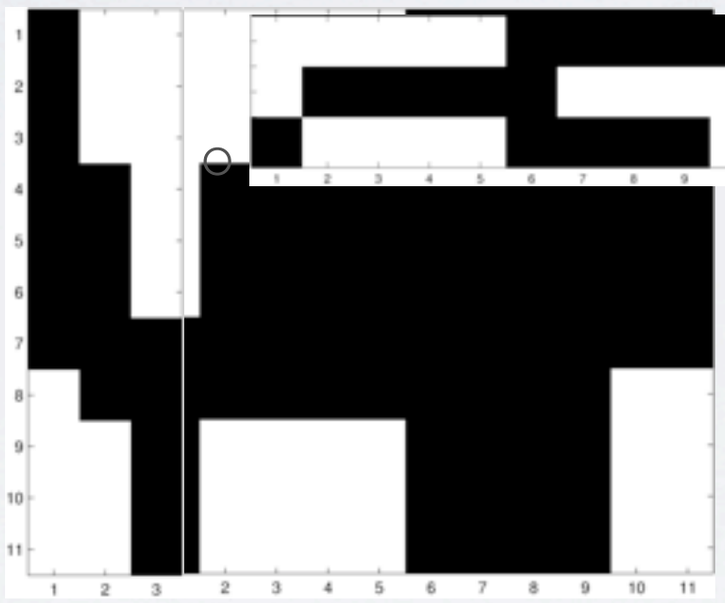
- Selecting k = model order selection problem
- The best model (order) is the one that allows us to represent the data with least number of bits
- **Intuition:** Using factor matrices to represent the BMF structure in the data saves space, but using them to represent noise wastes space

FITTING BMF TO MDL

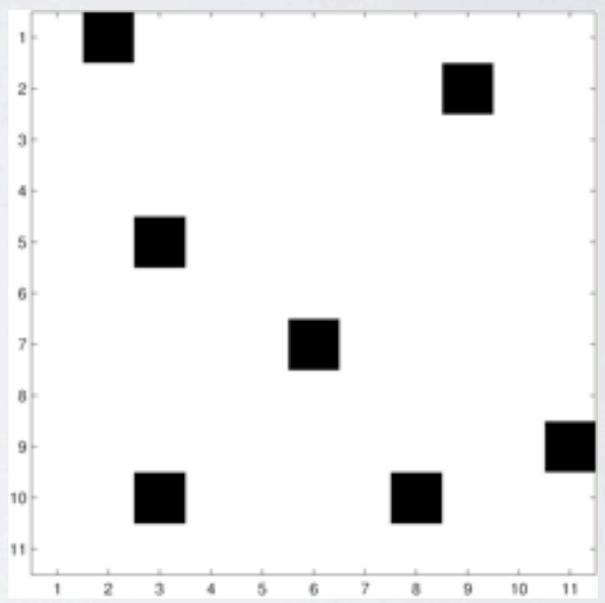
- MDL requires exact representation



$\approx$



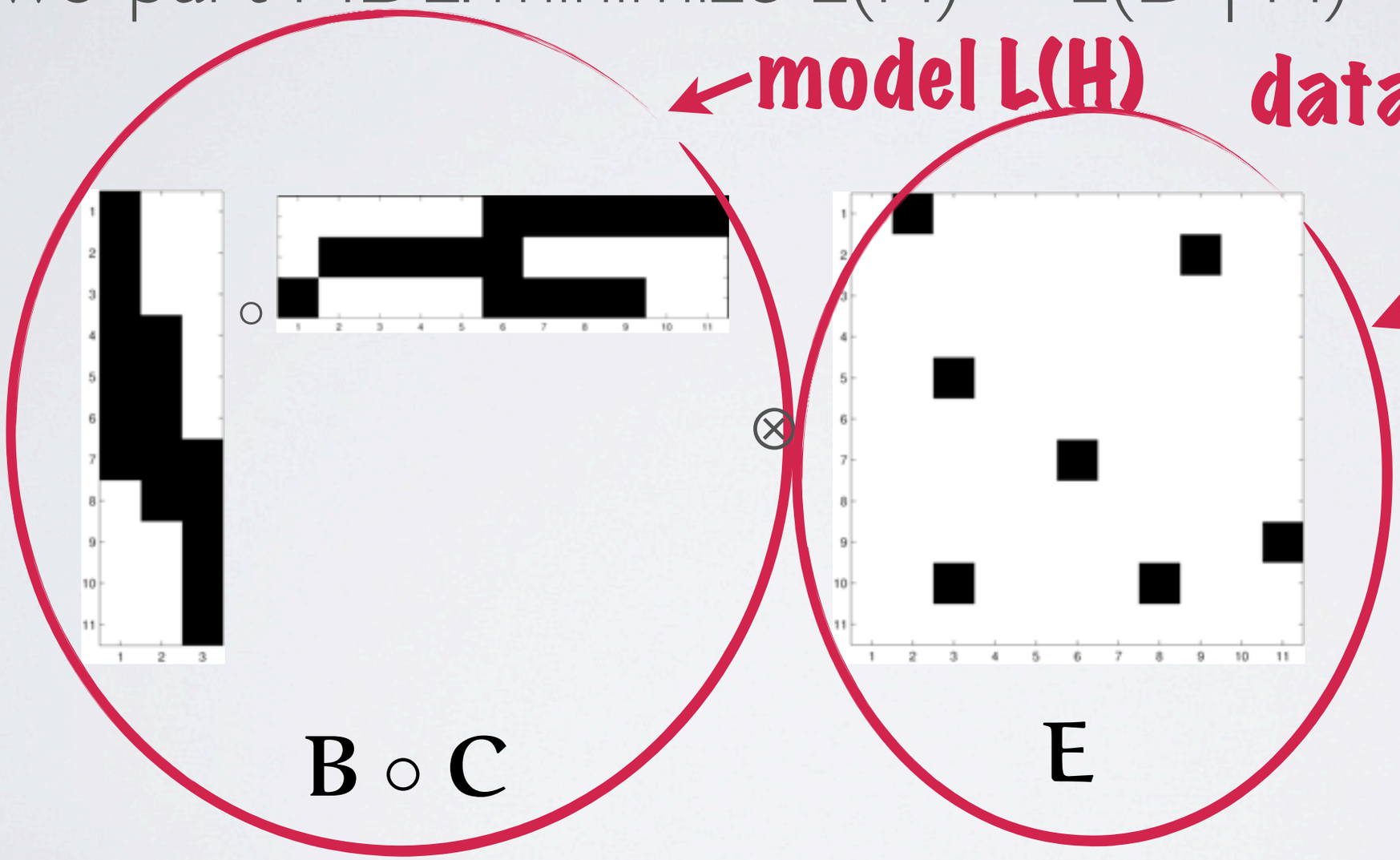
$\otimes$



A $B \circ C$ E

FITTING BMF TO MDL

- Two-part MDL: minimize L(H) + L(D | H)



ENCODING THE MODEL

- Model includes factor matrices **B** and **C** and their dimensions (*n*, *m*, and *k*)
- Each factor (row of **B** and column of **C**) is encoded using an optimal prefix code

$$(\mathbf{B}) = \mathrm{k}\log\mathrm{n} - \sum_{\mathrm{i}=1}^{\mathrm{k}} \left(|\mathbf{b}_\mathrm{i}| \log \frac{|\mathbf{b}_\mathrm{i}|}{\mathrm{n}} + (\mathrm{n} - |\mathbf{b}_\mathrm{i}|) \log \frac{\mathrm{n} - |\mathbf{b}_\mathrm{i}|}{\mathrm{n}} \right.$$

HOW HARD CAN IT BE?

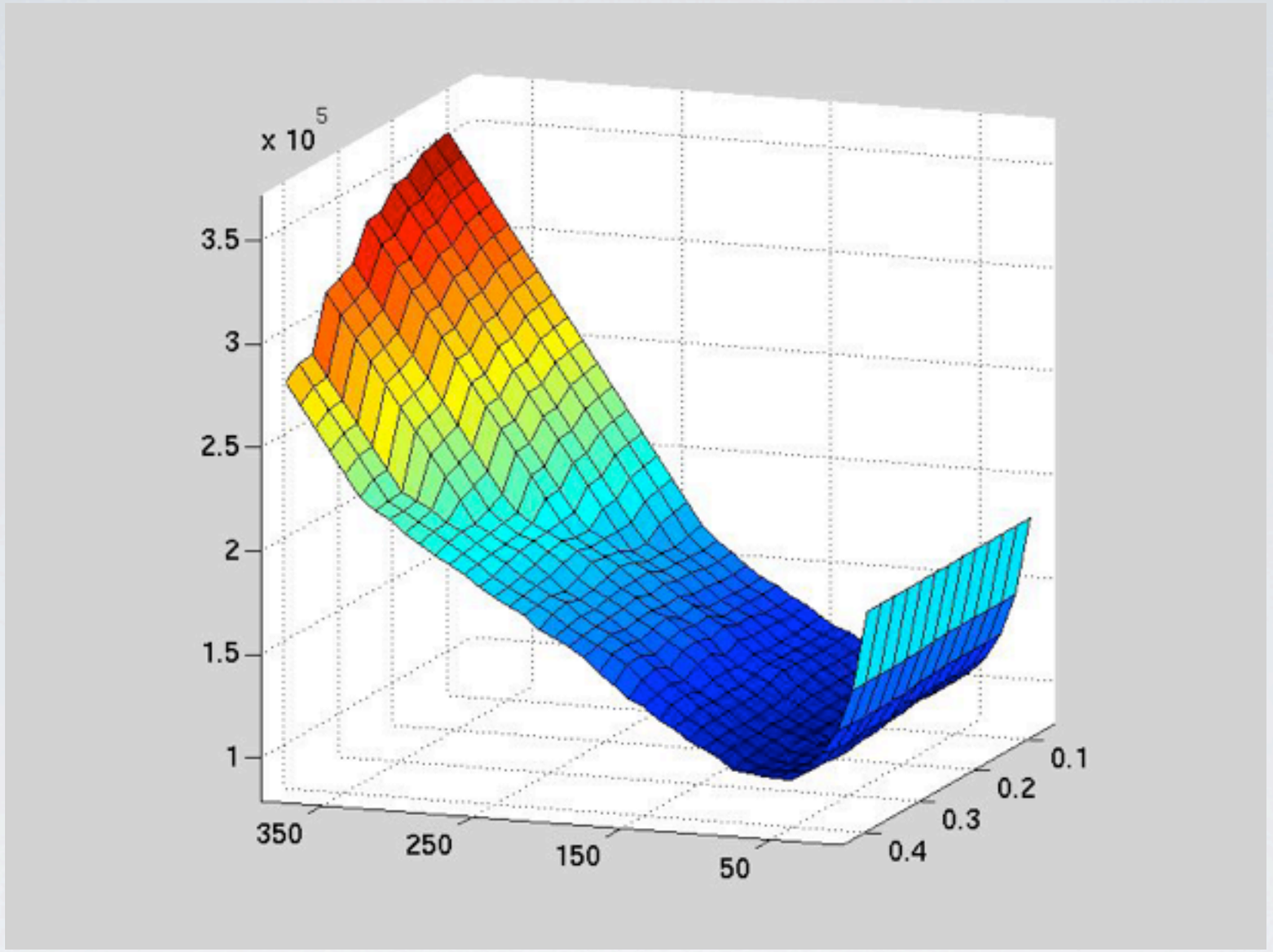
- MDL itself is an approximation of Kolmogorov complexity
- Finding minimum-error BMF is NP hard (even to approximate)
- But how hard it is to find the MDL-optimal decomposition?
  - Not necessarily minimum-error decomposition
  - Hardness depends on encoding
  - We know that there exists an encoding for which it is NP-hard to find the MDL-optimal decomposition

USING ASSO WITH MDL

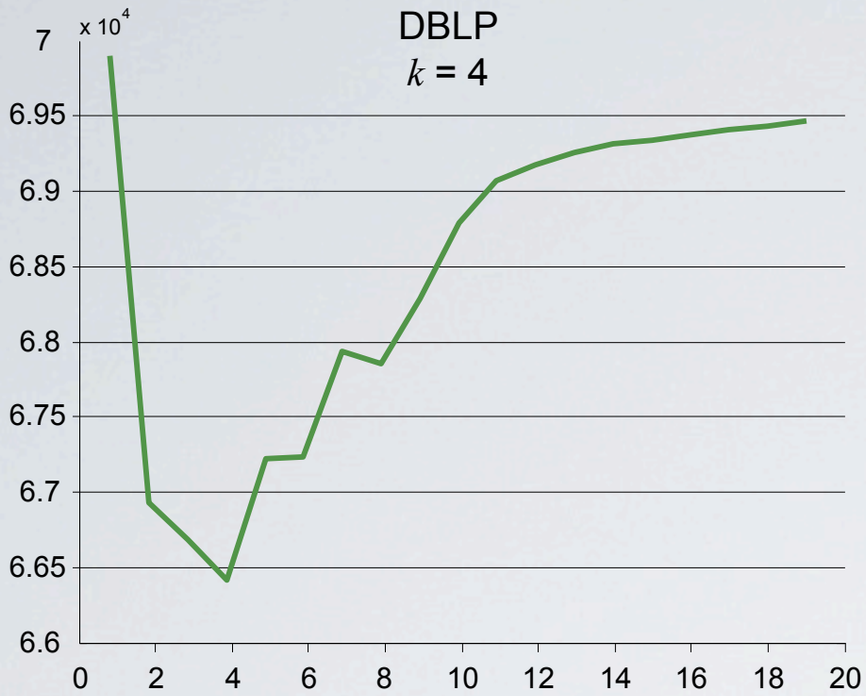
- **The Good**
  - Asso is hierarchical and deterministic
    - The k^{th} factor does not change the previous $k - 1$ factors
- **The Bad**
  - Asso is heuristic
- **The Ugly**
  - Asso requires extra parameter t — but MDL can be used to find this, too

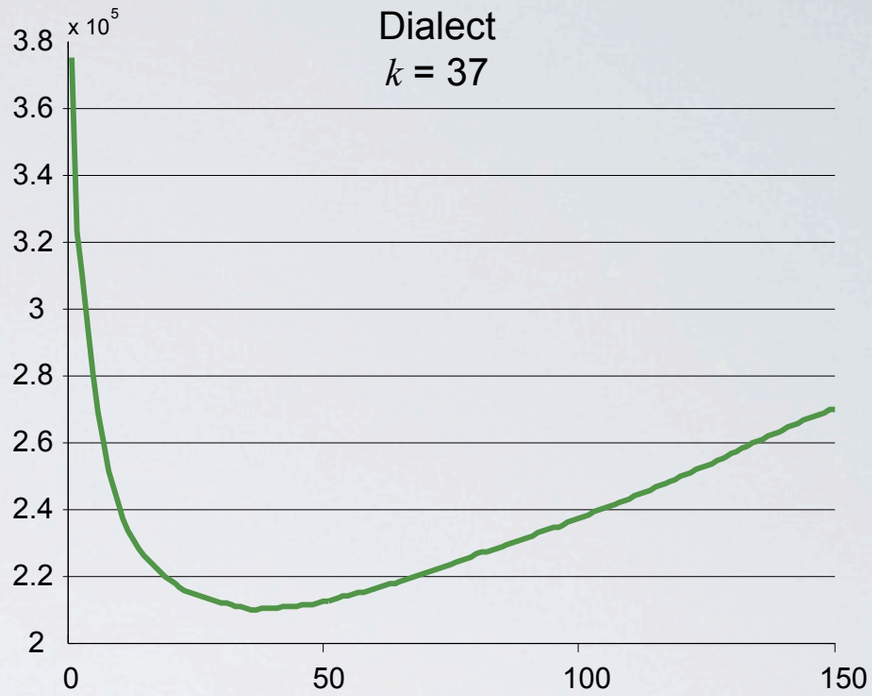
HASN'T THIS BEEN DONE BEFORE?

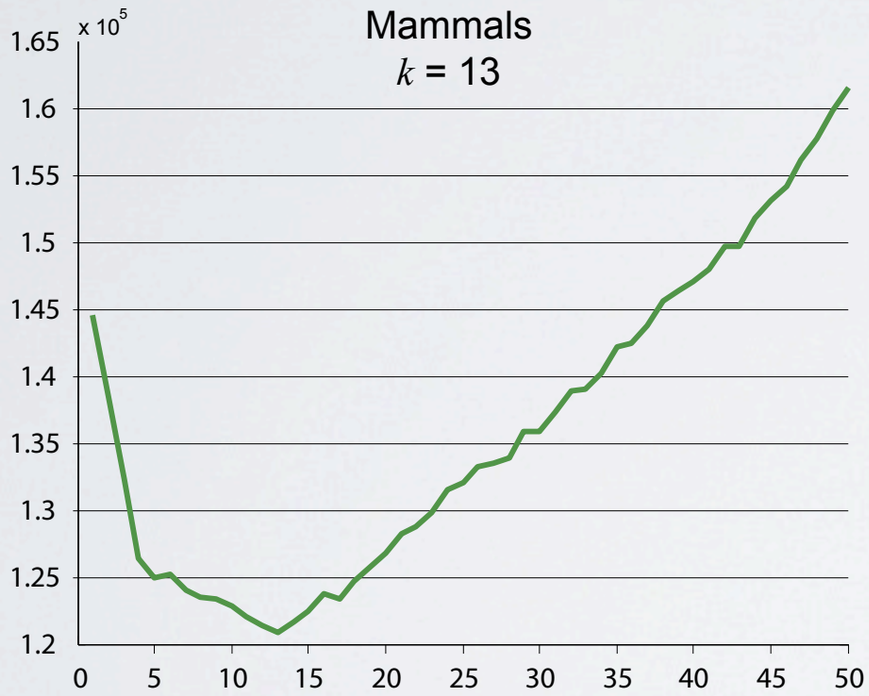
- Model order selection for matrix factorizations is studied before (mostly with SVD/PCA)
- Methods such as Guttman–Kaiser criterion (c. 1950) or Cattell's scree test (1966) are not suitable
  - Poor performance and need for subjective decisions
- Cross validation doesn't work, either
  - Well-known problem with matrix factorizations

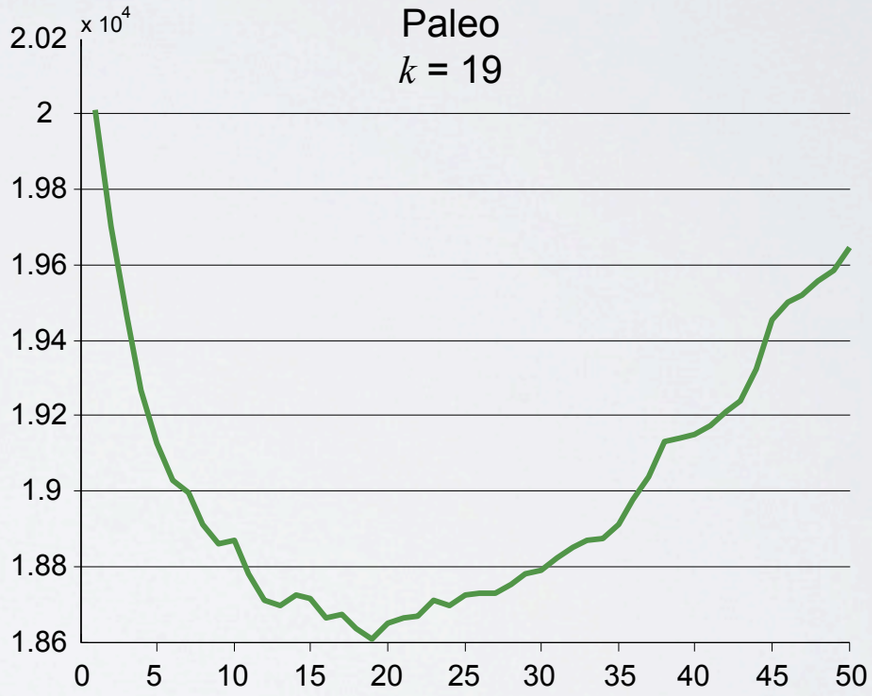


THE DNA DATA









REAL-WORLD DATA

max planck institut informatik

FUTURE WORK

- Binary tensors
- Maximize similarity vs. minimize dissimilarity
- Solve BMF via LP optimization
  - And better algorithms in general
- Joint subspaces
- …

max planck institut
informatik

CONCLUSIONS

- BMF is a strong data mining technique
  - If your data are binary, consider BMF
- Computationally hard, but sparsity helps
- Model order selection can be solved with MDL
  - Irrespective of algorithm used
- Lots of things to do...