Segmented Nestedness in Binary Data

Esa Junttila & Petteri Kaski

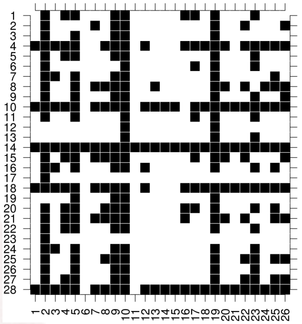
University of Helsinki & Aalto University & HIIT

Outline

1. Introduction:
   k-Nestedness in binary data
2. Complexity of discovery and algorithms
3. Choosing *k* with MDL
4. Experiments on synthetic data
5. ...and on real-world data

Introduction: Binary matrices

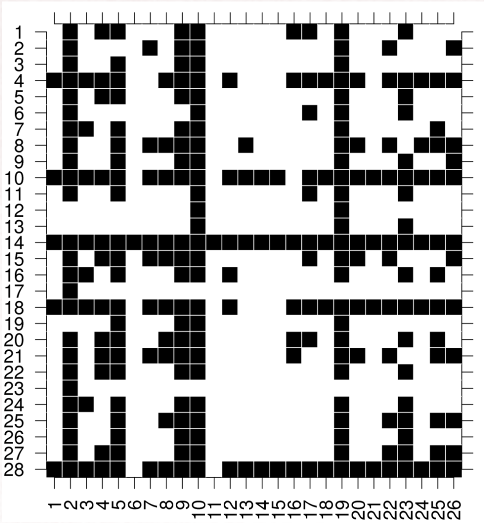
- Goal:
  Finding hidden structure in data
- We concentrate on datasets that come in the form of binary matrices

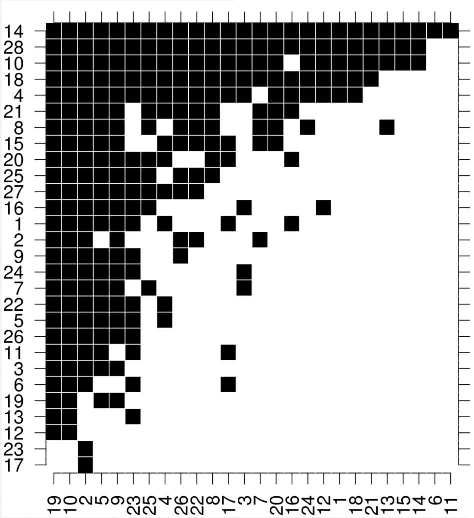


- Binary data occurs in ecology, paleontology, interaction and social networks, market-basket data of retail companies, …

Introduction: Reorderable patterns

- Reorganize data to uncover structure
- Permute the rows and columns:

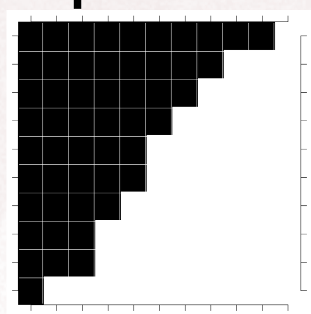


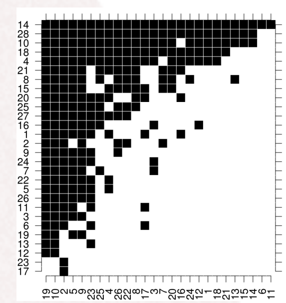


- Better interpretation/compression of data; understanding generative process

Introduction: Nestedness

- Nestedness describes hierarchies
- Examples of (almost) nested matrices:

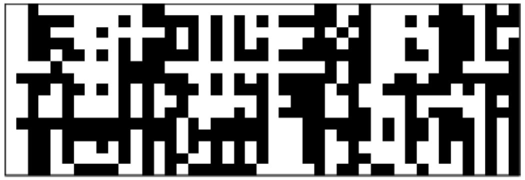




- Applications: ecology, hierarchies, ...
- For example: many species live at low altitudes, but few up in the mountains

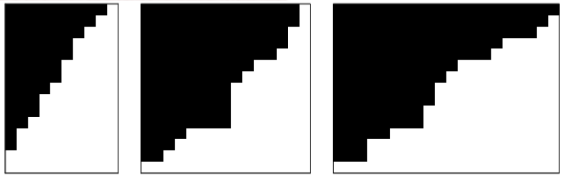
k-Nestedness

- *k*-nestedness: columns can be partitioned into *k* nested submatrices



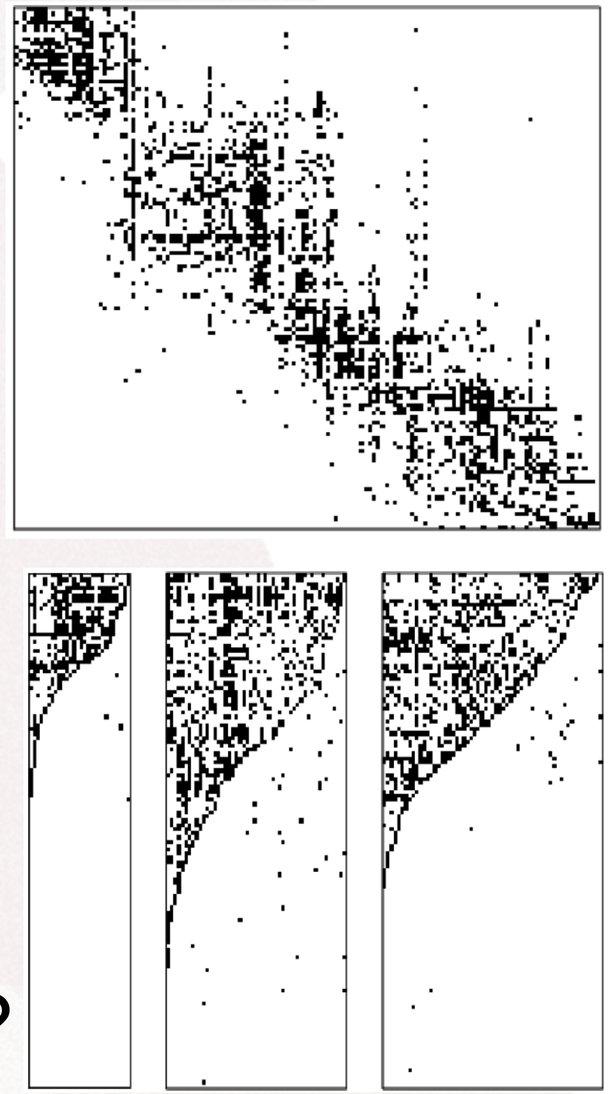
- Applications: course completion data, separate hierarchies in mammal data, data consists of several nested blocks

- Partition into *k* chains



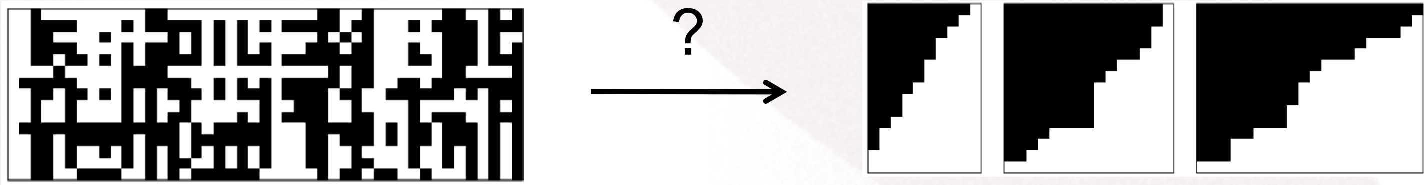
Real-world example: Paleontological data

- The data has 124 sites as rows and 139 genera (species) as columns
- Black dot (1s): fossil of a specific genus has been found at a site
- Presumably lots of missing 1s
- Species from three eras?



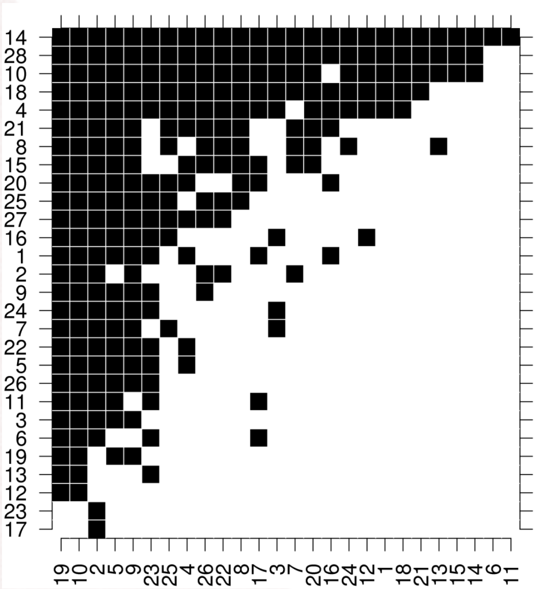
Recognition of k-nestedness

- Recognition problem: given a matrix *A* and *k*, decide whether *A* is *k*-nested
- We give a polynomial-time algorithm exists, assuming the data is perfect (free of noise and errors)



Noise and errors

- Real-world data have errors
- Distance between matrices *A* and *B*: the minimum number of flips that transform *A* to *B*
    - Hamming: both 1-to-0 and 0-to-1 flips
- Distance to nestedness is the minimum distance to a nested matrix



Finding a closest (k-)nested matrix

- Problem (Closest Nested):
  Find a nested matrix that is closest to a given matrix
  - Complexity: NP-hard relative to 0-to-1 flips, unknown for Hamming-distance
  - A heuristic algorithm
    “GreedyNested” [Mannila and Terzi, 2007]
- Problem (Closest *k*-Nested):
  Find a *k*-nested matrix closest to a given matrix

NP-hardness of "Closest k-Nested"

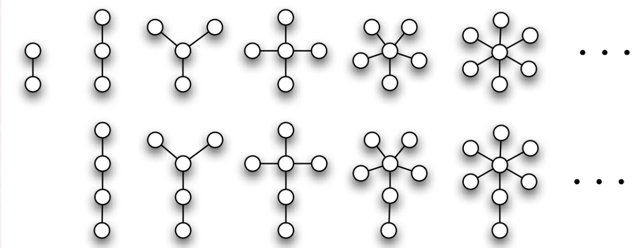
- In NP (as a decision problem)
- We show completeness:

  3SAT

  ≤ Daisy/Star Covering*

  ≤ Closest *k*-Nested

stars and daisies:



$$\begin{bmatrix} 1 & 1 & 1 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}, \begin{bmatrix} 1 & 1 & 1 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

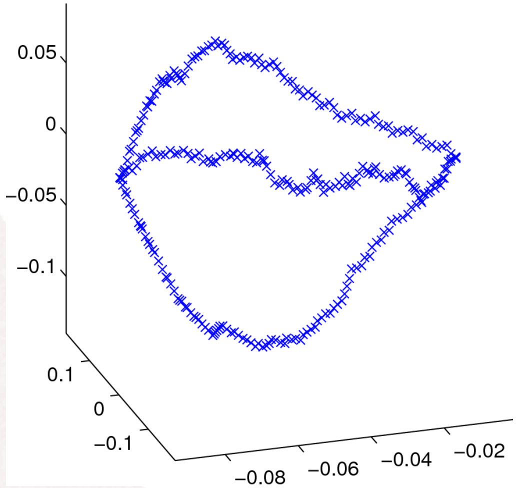
*: a relaxation of the Vertex Cover problem in triangle-free graphs

Heuristic algorithms

- Given the NP-hardness of Closest *k*-Nested, we settle for heuristics
- We introduce heuristics that
  - Take matrix *A* and positive *k* as input
  - Produce a *k*-partition of the columns
  - Use GreedyNested to assess distance

Heuristic: SVD-k-Baseline

- Singular value decomposition: $A=USV$
- Form a Euclidean space from k vectors in V with largest singular values
- Map the columns of A into the space
- Run k-means++ to produce a partition



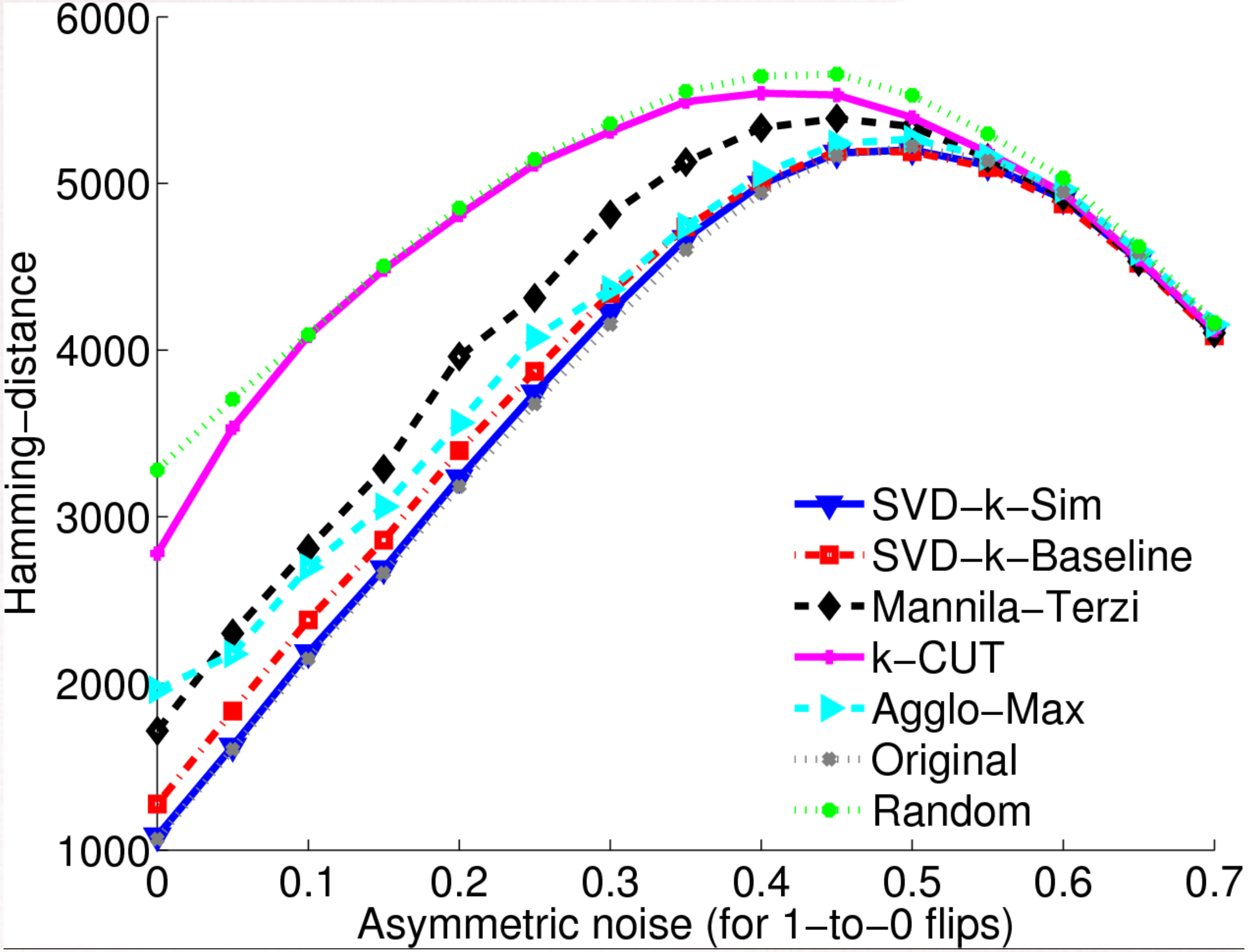
Heuristic: SVD-k-Sim

- k-means++ assumes Gaussian data
- Run SVD on a *similarity matrix*:
  - Do two columns *i*,*j* resemble each other more than expected by chance?
  - $S_{i,j}$: number shared 1s on columns *i*,*j*
  - Random variable *X:* number of shared 1s on two columns, given the sums
  - similarity(*i*,*j*) = 1/(1+exp(*C*)), where $C = (E(X_{i,j}) - S_{i,j}) / Var(X_{i,j})$

Other heuristics

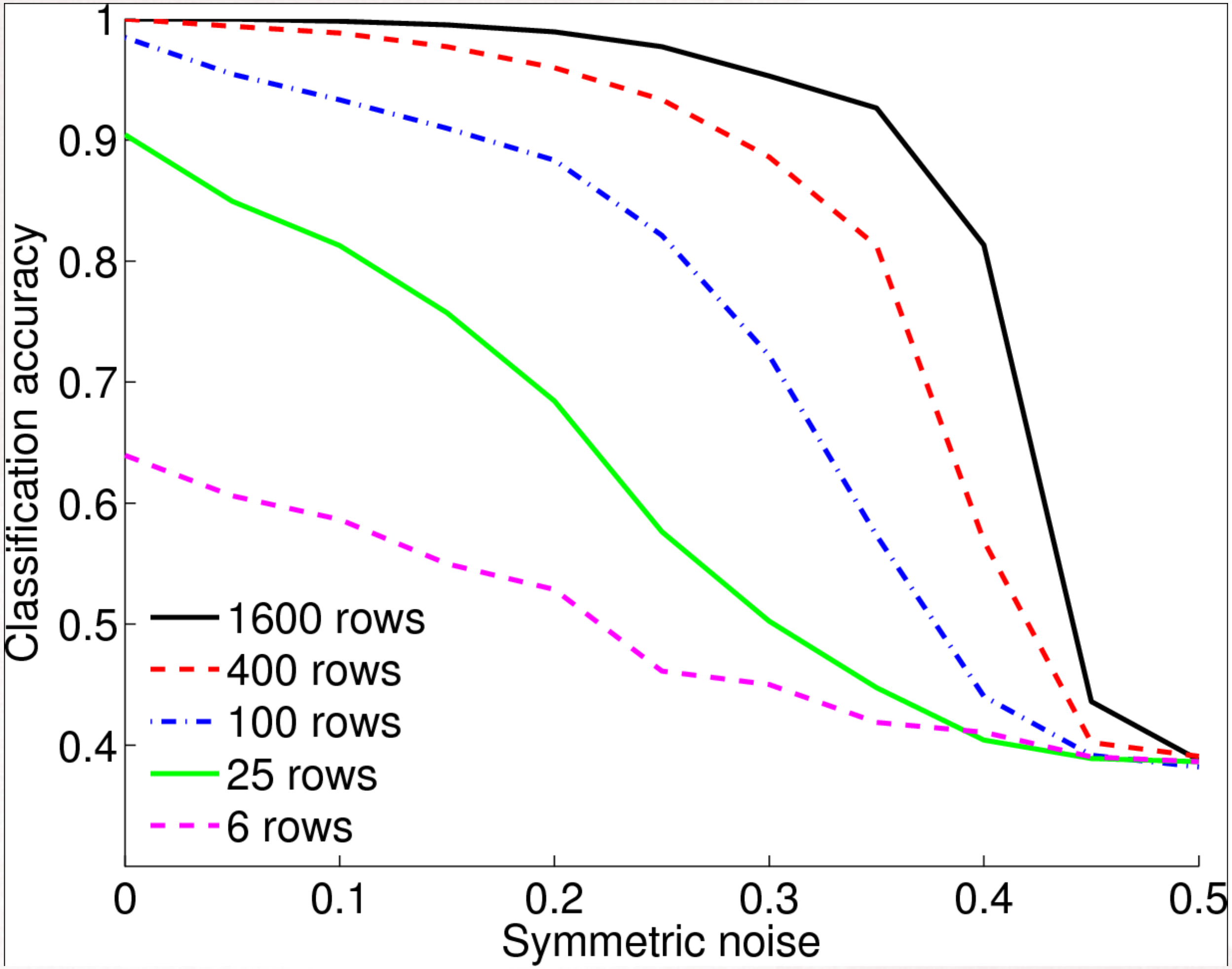
- Mannila–Terzi: Start with all columns in one part; reorder by spectral ordering and split until *k* parts
- *k*-Cut: construct a graph on columns where edge-weights are conflict-distances between the two columns
- Agglomerative clustering on the graph
- Benchmarks: (1) a random partition, (2) the original ground-truth partition

Synthetic data: Distance to k-nestedness with increasing noise



3-nested synthetic data,
150x150 matrices,
block sizes 25, 50, 75,
averages from 30 samples,
noise Pr(0→1)=0.1

Synthetic data: Discovered k-nested partition vs ground truth

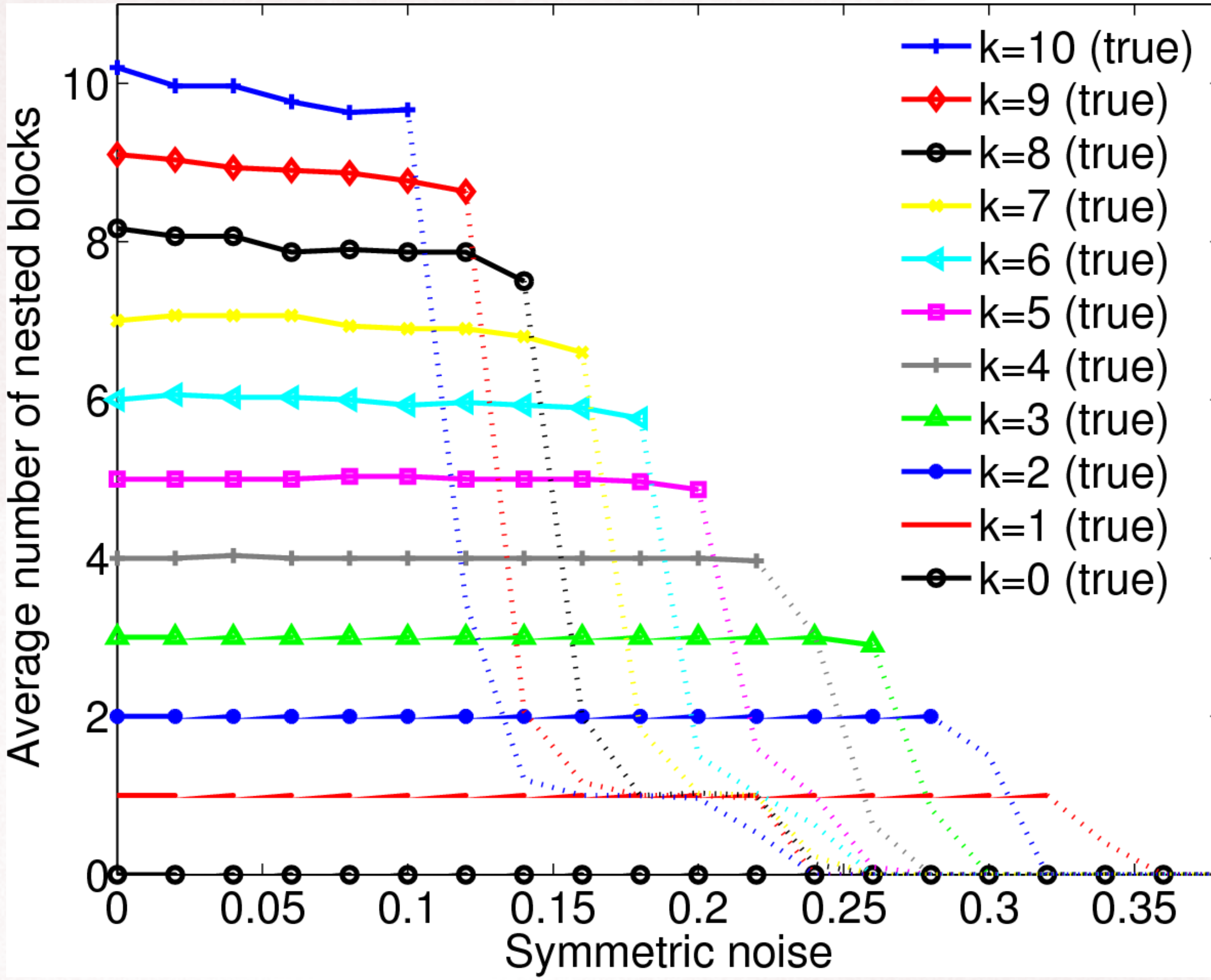


3-nested synthetic data,
r x 150 matrices,
block sizes 25, 50, 75,
averages from 50 samples

Choosing k with MDL

- Given a dataset, how to choose *k*?
- Minimum description length (MDL):
  - Choose a model (value of *k*) that minimizes the description length of the data *A*, measured in bits
- We use two model families:
  - assuming non-nestedness and independence of 1s
  - assuming *k*-nestedness for some *k*

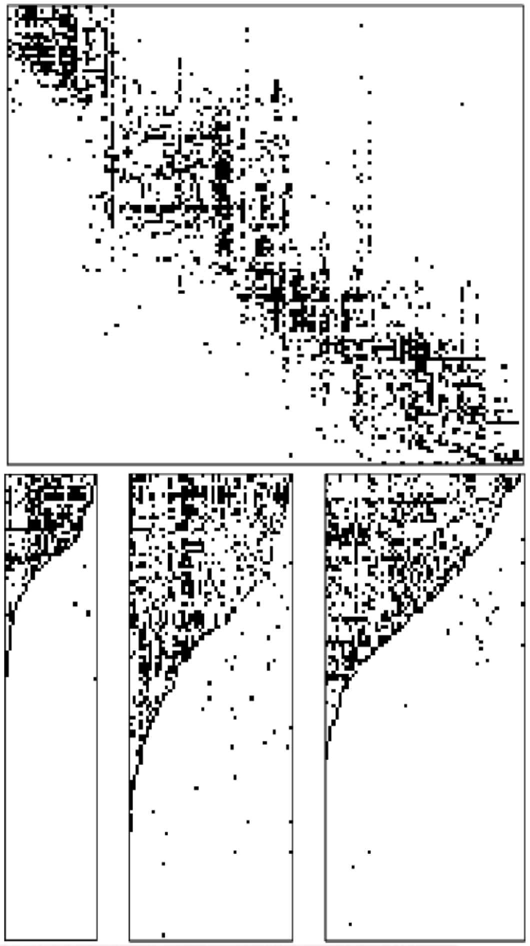
Synthetic data: Robustness of MDL in recovering “true” *k*



k-nested synthetic data,
150x150 matrices,
k equal-size block,s
averages from 30 samples,
0 ~ uniform model

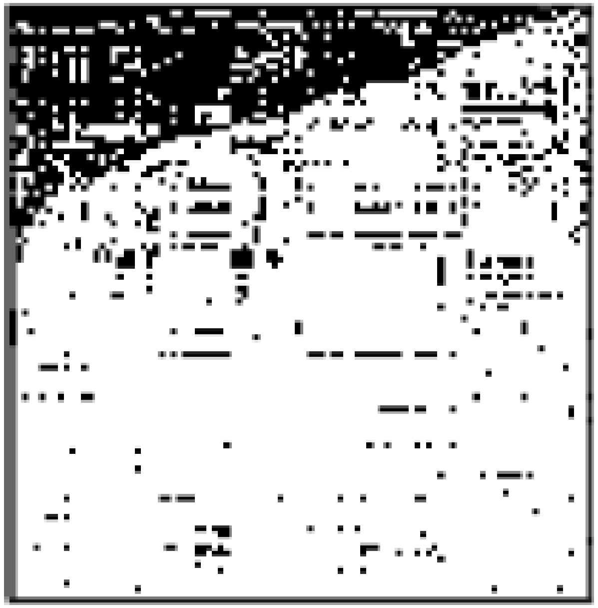
Experimental results: MDL and real-world data

- MDL: paleontological data is 3-nested.



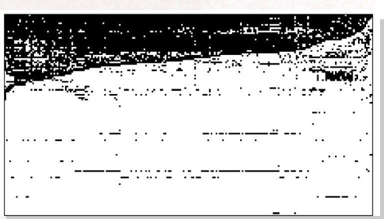
Experimental results: MDL and real-world data

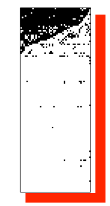
- Mammals data (presence/absence)
- 124 rows: mammal species
- 2179 columns: 40km × 40km locations in Europe.



Experimental results: Mammals in Europe

- MDL considers the data 16-nested





Conclusions

- *k*-nestedness describes multiple hierarchies that occur in real-world data
- Recognizing *k*-nested patterns can be done in polynomial-time (noise-free), but the noisy case is NP-hard
- Heuristic SVD-algorithm finds almost *k*-nested patterns in noisy data reliably
- By using an MDL-model, we can retrieve *k* automatically (up to a noise threshold)