Universiteit Utrecht

[Faculty of Science
Information and Computing Sciences]

Identifying the Components

Matthijs van Leeuwen, Jilles Vreeken & Arno Siebes
Algorithmic Data Analysis, Universiteit Utrecht

What's the deal?

- Most, if not all, databases are mixtures of samples from different distributions.
- Transactional data is no exception.
- Models that take this into account are generally superior to those that don't.
- How do we identify these components?

The problem

- Supermarket basket data
  - Mixture of different buying patterns.
  - *Retired people* buy different collections of items than *households with young children*. But, overlap exists.
- Goal
  - Find these groups of people, and
  - Give insight in their corresponding buying behaviours.

Isn't that ...

- Frequent itemset mining
  - Characterises data with patterns
  - But: does not find groups
- Clustering
  - Finds homogeneous groups
  - But: does not characterise and requires a distance measure
- Mixture modelling
  - Characterises and finds homogeneous groups
  - But: requires pre-defined distributions

The Plan

- Partition database *db* into $db_1,\ldots,db_n$ such that
  - Buying behaviour of each db_i is different
  - Each db_i is homogeneous in itself

How? By using compression

- Compression is
  - Low when data is a mixed bag
  - High when data is homogeneous
- Minimum Description Length principle
  - Lossless compression
  - Better compression $\Longleftrightarrow$ better model
- Our compressor: KRIMP
  - By MDL finds few patterns that characterise data
  - Models are called `code tables'

Formal Problem Statement

Find a partitioning $db_1,\ldots,db_k$ of database *db* and a set of associated code tables $CT_1,\ldots,CT_k$, such that the *total encoded size* of

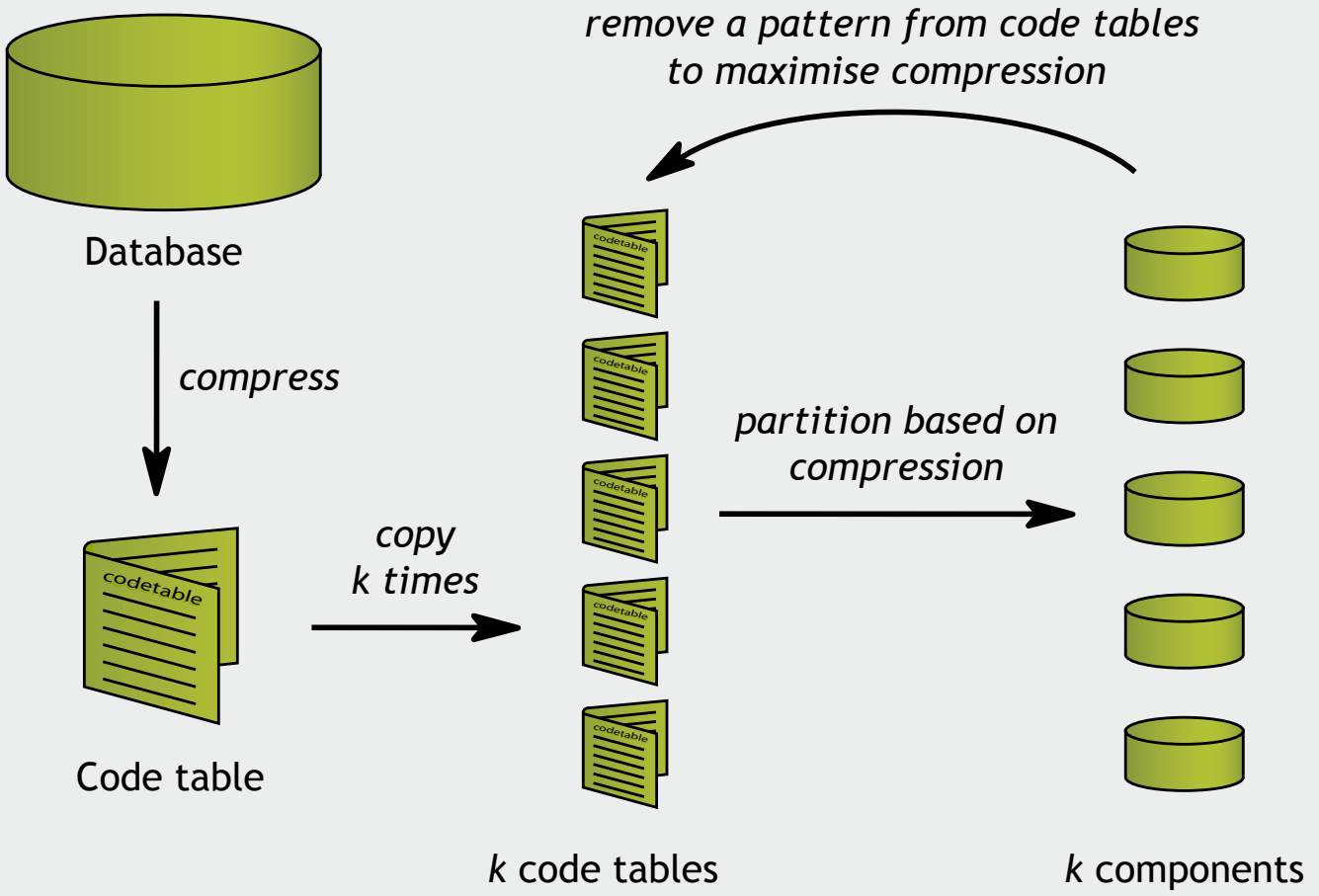
$$\sum_{i \in \{1,\ldots,k\}} L(CT_i, db_i)$$

is minimised.

Two Approaches

- Two different ways of identifying components
- By Model
  - Start with a model for all data
  - Extract components from it
- By Data
  - Start with random components
  - Iteratively optimise

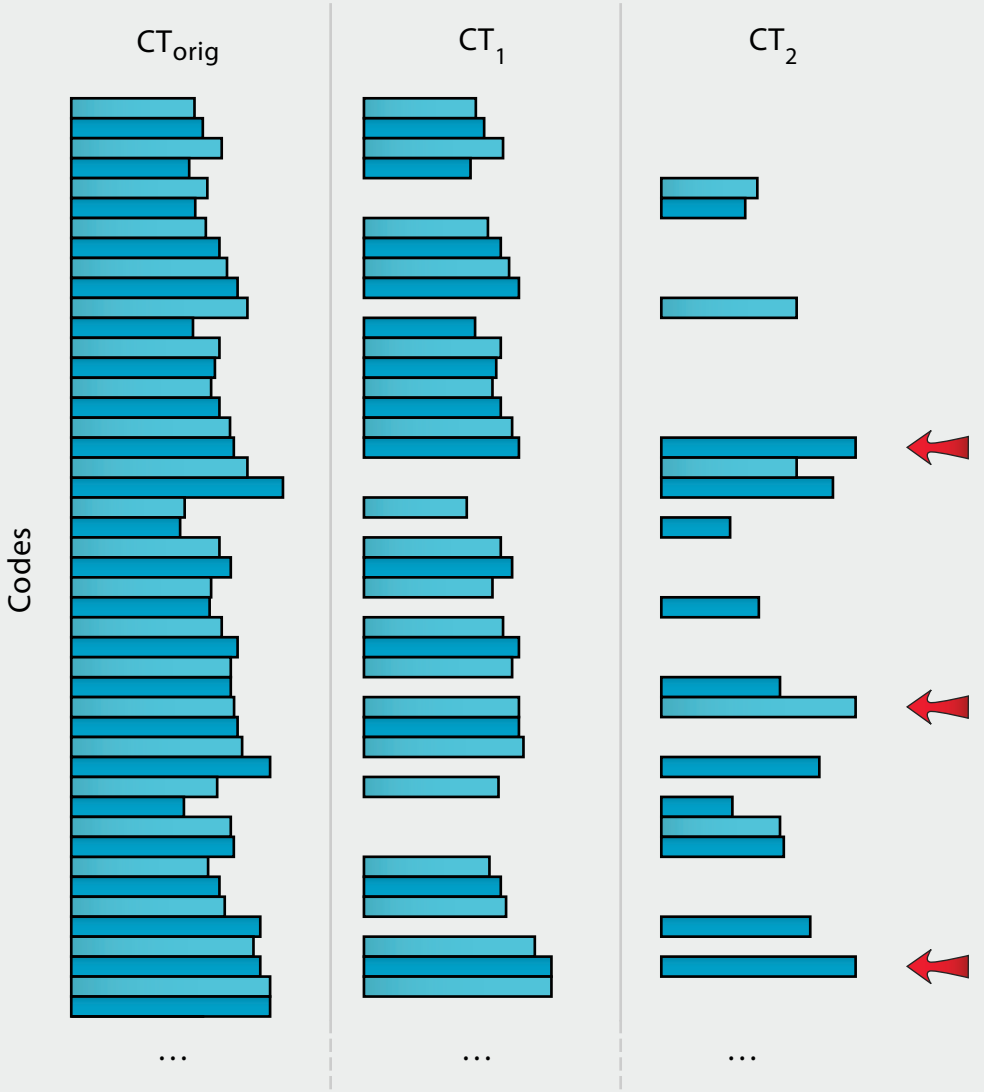
Model-driven Identification



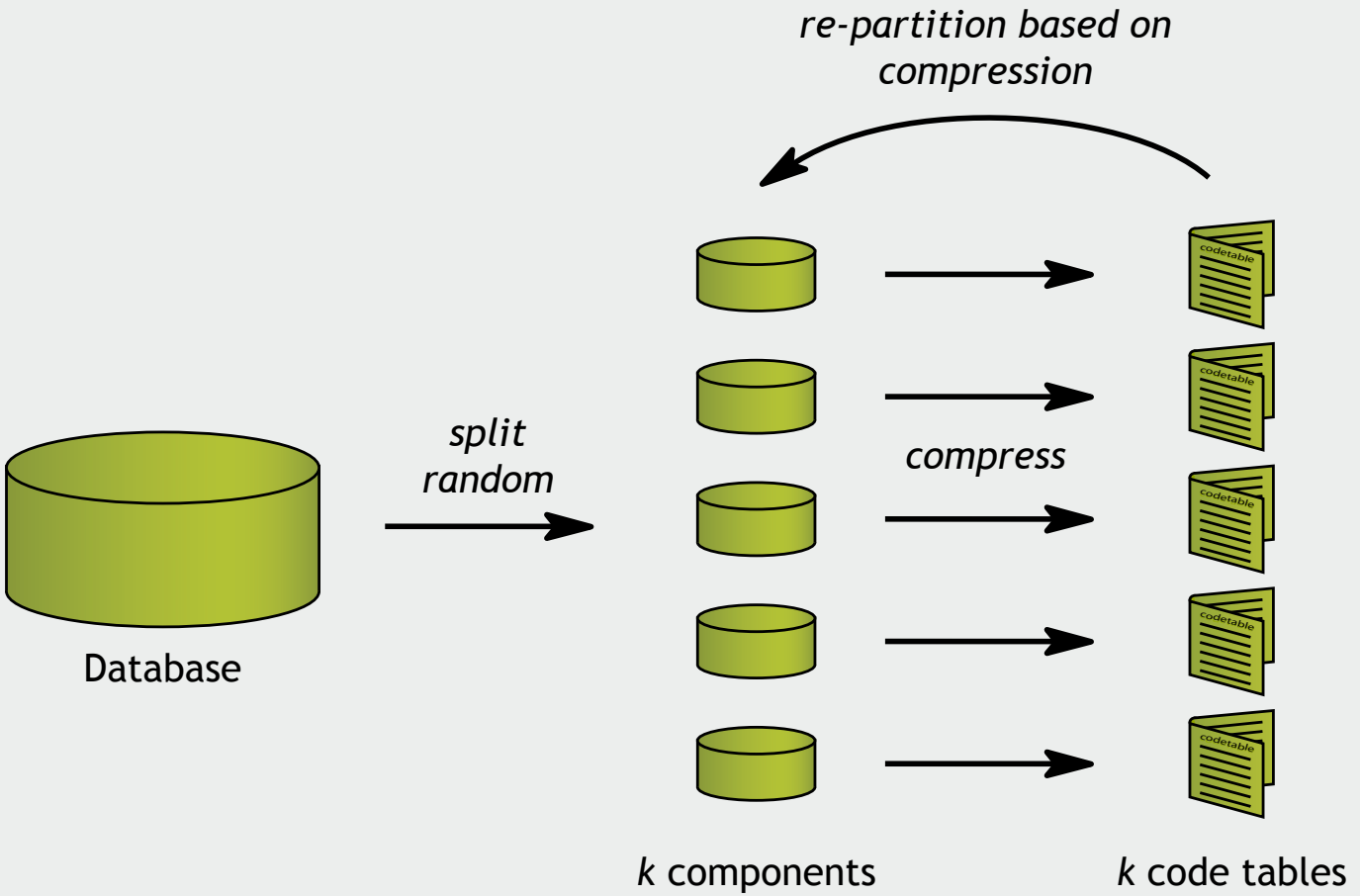
Model-driven Results

Dataset	#rows	Optimal *k*	Compression gain (%)	Purity (%)	
				base	obtained
Anneal	898	19	18.1	76.2	80.8
Chess (kr-k)	28056	6	14.5	16.2	18.2
Mushroom	8124	12	25.7	51.8	88.2
Nursery	12960	14	11.7	33.3	45.0

Anneal



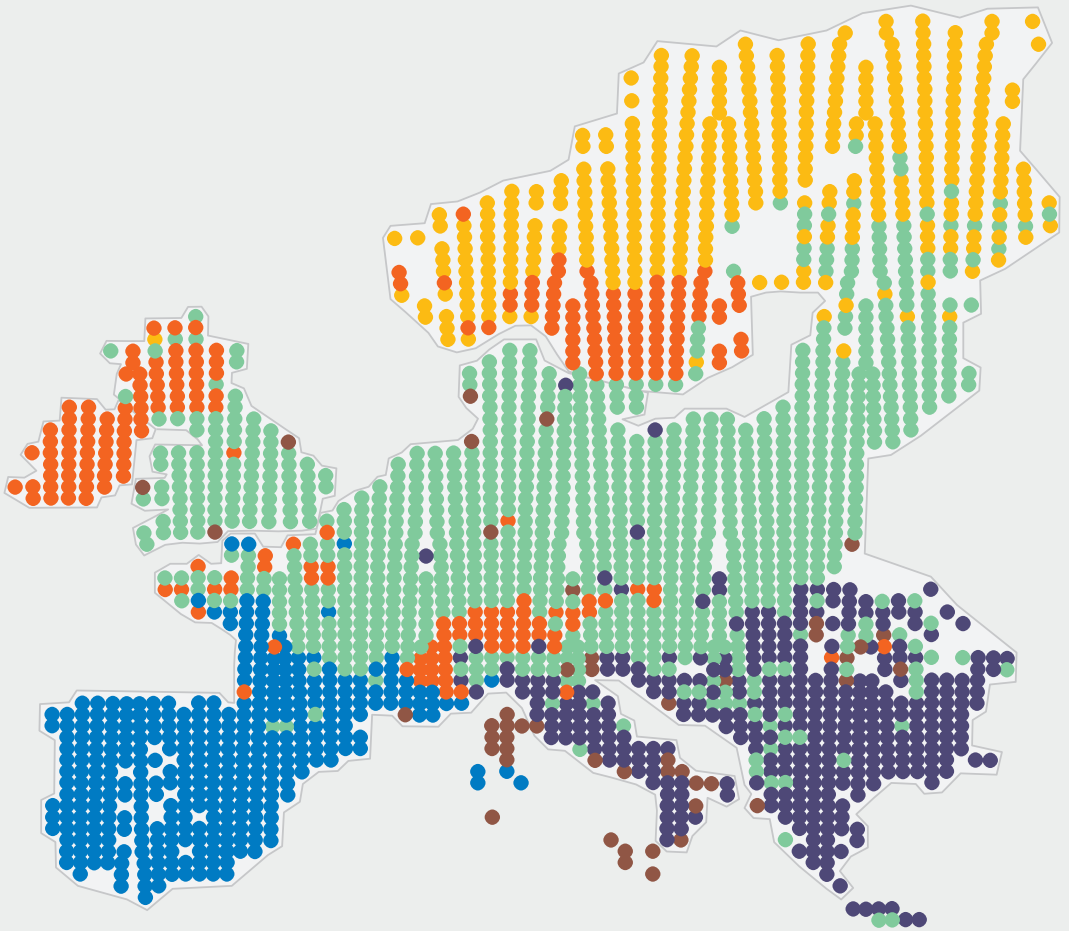
Data-driven Identification



Data-driven Results

Dataset	#rows	Optimal *k*	Compression gain (%)	Purity (%)	
				base	obtained
Adult	48842	177	40.3	76.1	82.2
Anneal	898	2	4.8	76.2	76.2
Chess (kr-k)	28056	13	18.2	16.2	17.8
Mammals	2183	6	46.2	-	-

Mammals



k = 6, optimal.

Conclusions

- Compression identifies database components
  - Each sample distribution characterised by a code table
  - No prior knowledge required
  - No distance measure required
  - Optimal *k* determined by MDL
- Two orthogonal algorithms
  - Model-driven and data-driven
  - Experiments show that the identified components are present in, and characteristic for, the database

Thank you for your attention!