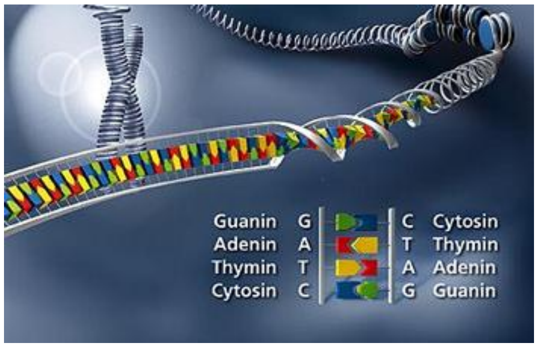
Forschungskolleg

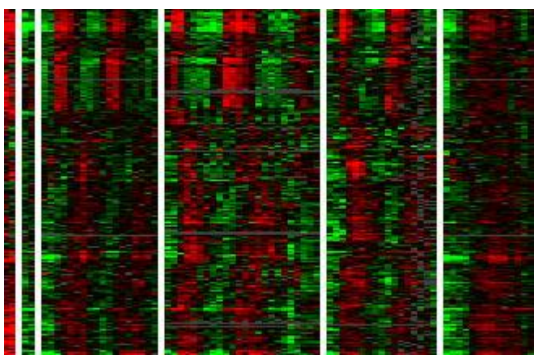
Data Analytics Methods and Techniques

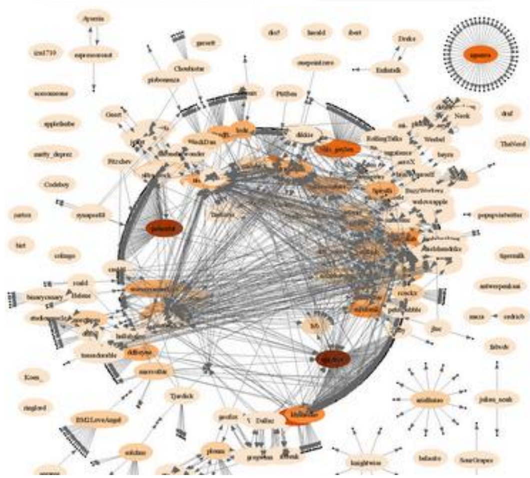
Martin Hahmann,
Gunnar Schröder,
Phillip Grosse

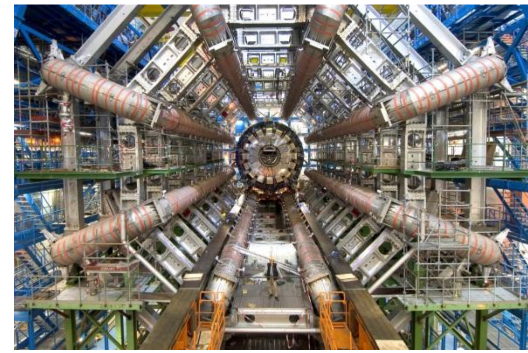
> Why do we need it?

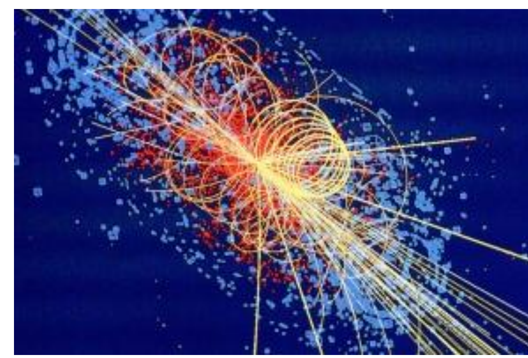
"We are drowning in data, but starving for knowledge!" John Naisbett











23 petabyte/second of raw data
1 petabyte/year

The Big Four

Classification

Association Rules

Prediction

23.11.

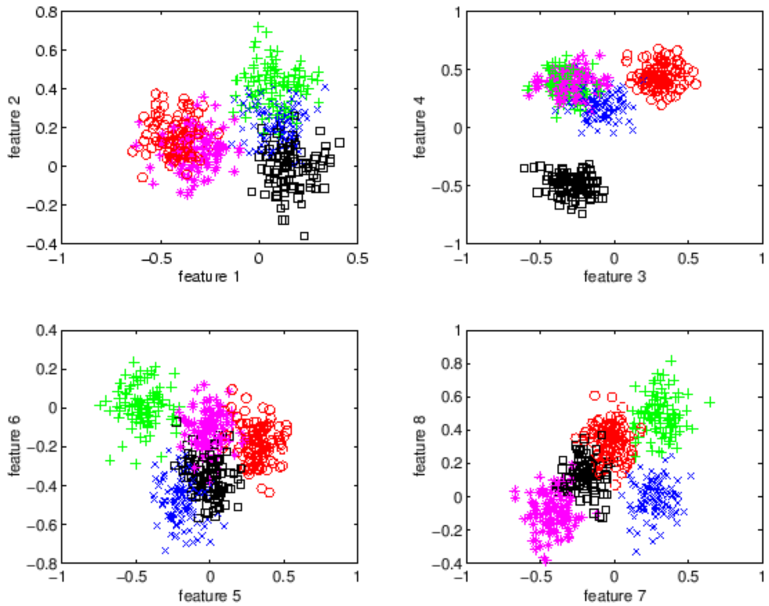
Clustering

Clustering

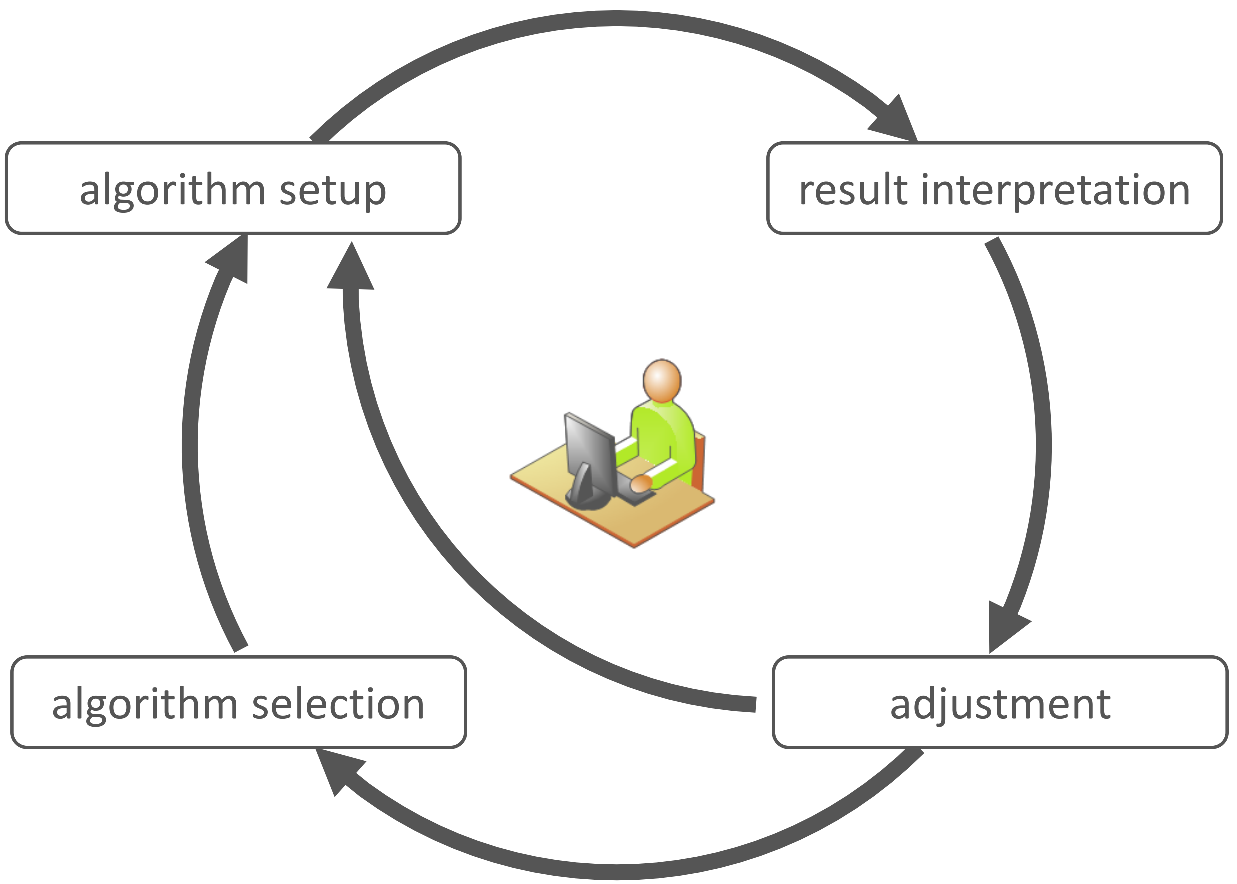
- automated grouping of objects into so called clusters
- objects of the same group are similar
- different groups are dissimilar

Problems

- applicability
- versatility
- support for non-expert users



> Clustering Workflow



>500

> Clustering Workflow

Which algorithm fits my data?

Which parameters fit my data?

How good is the obtained result?

How to improve result quality?

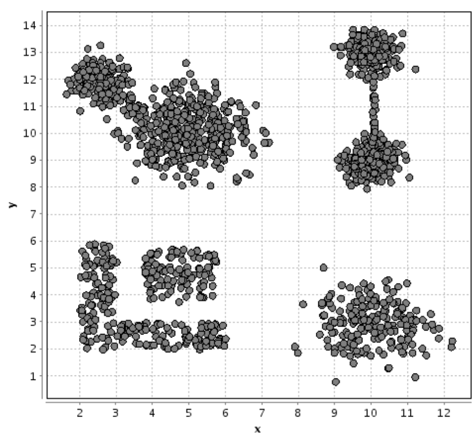
algorithm setup

algorithm selection

result interpretation

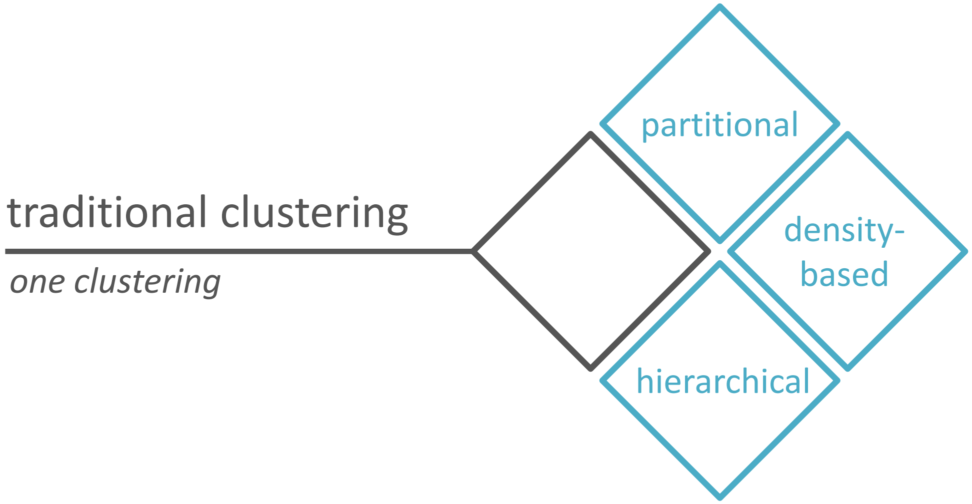
adjustment

Algorithmics



Feedback

> Traditional Clustering



Algorithmics

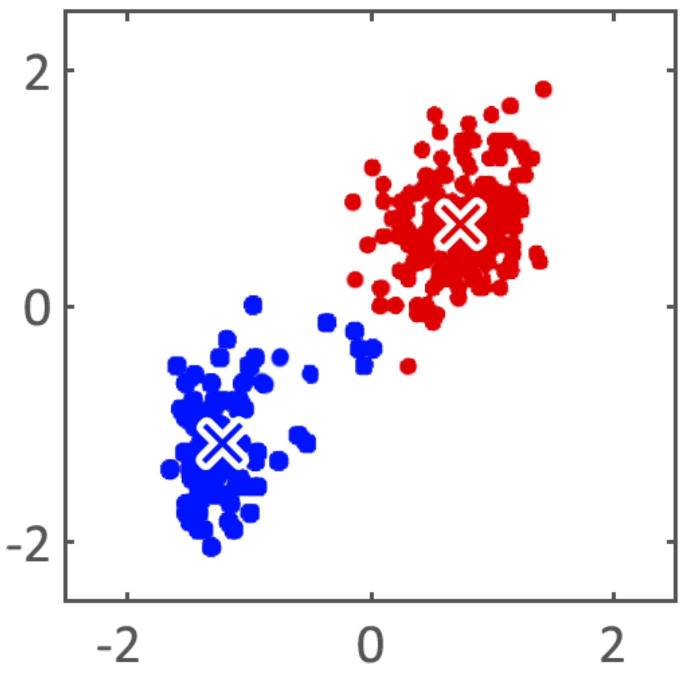
Feedback

Partitional Clustering

- clusters represented by prototypes
- objects are assigned to most similar prototype
- similarity via distance function

k-means[Lloyd, 1957]

- partitions dataset into *k* disjoint subsets
- minimizes sum-of-squares criterion
- parameters: *k, seed*



> Traditional Clustering

Partitional Clustering

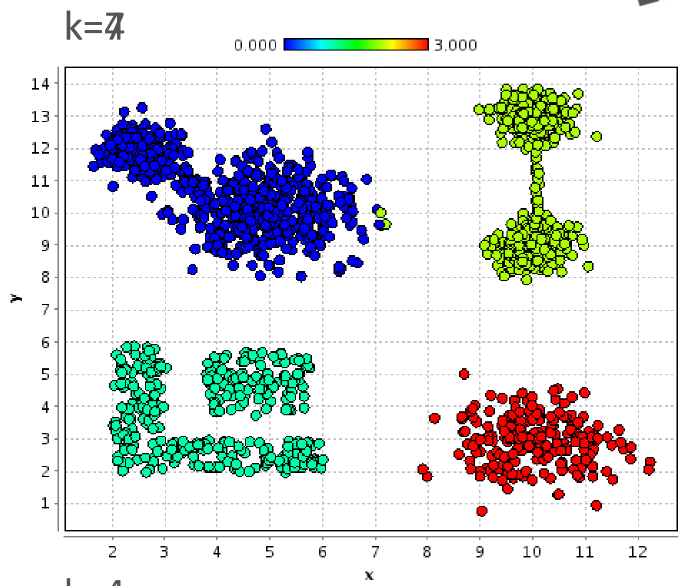
- clusters represented by prototypes
- objects are assigned to most similar prototype
- similarity via distancefunction

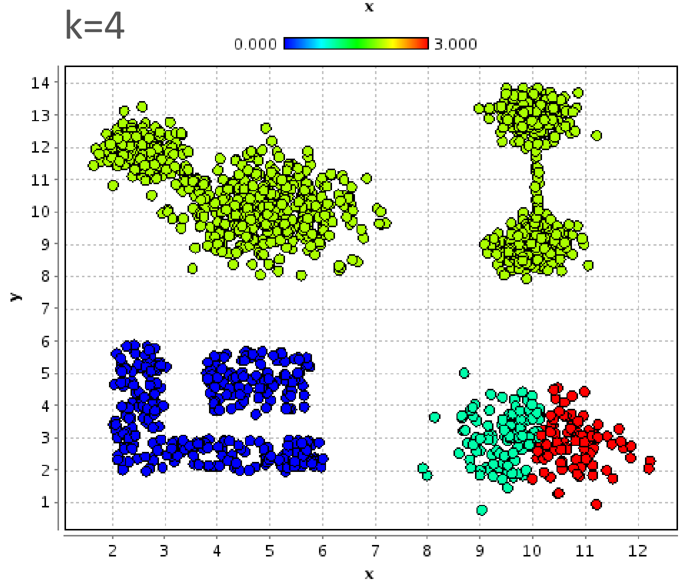
k-means[Lloyd, 1957]

- partitions dataset into *k* disjoint subsets
- minimizes sum-of-squares criterion
- parameters: *k, seed*

ISODATA[Ball & Hall, 1965]

- adjusts number of clusters
- merges, splits, deletes according to thresholds
- 5 additional parameters





Partitional Clustering

- clusters represented by prototypes
- objects are assigned to most similar prototype
- similarity via distancefunction

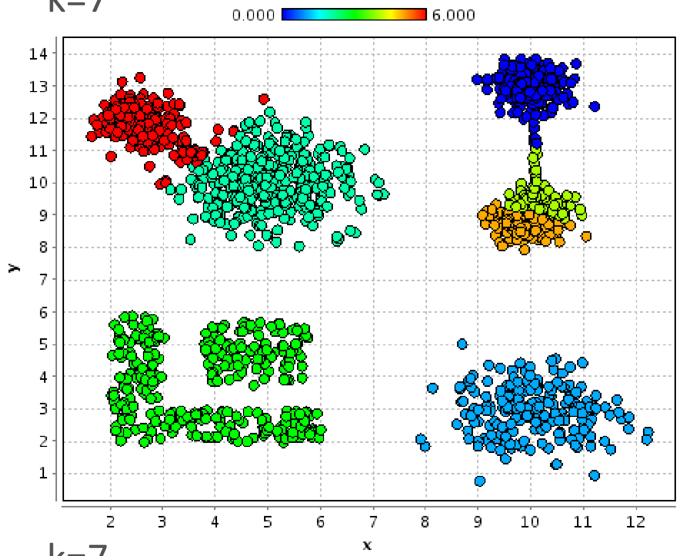
k-means[Lloyd, 1957]

- partitions dataset into *k* disjoint subsets
- minimizes sum-of-squares criterion
- parameters: *k, seed*

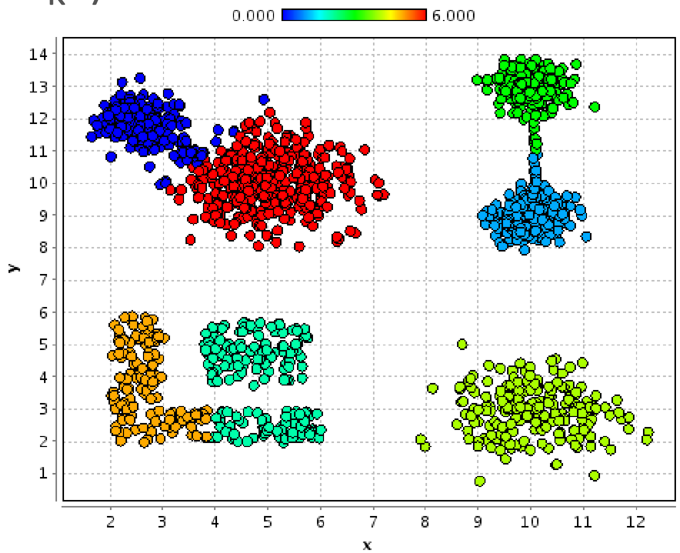
ISODATA[Ball & Hall, 1965]

- adjusts number of clusters
- merges, splits, deletes according to thresholds
- 5 additional parameters

k=7



k=7



Density-Based Clustering

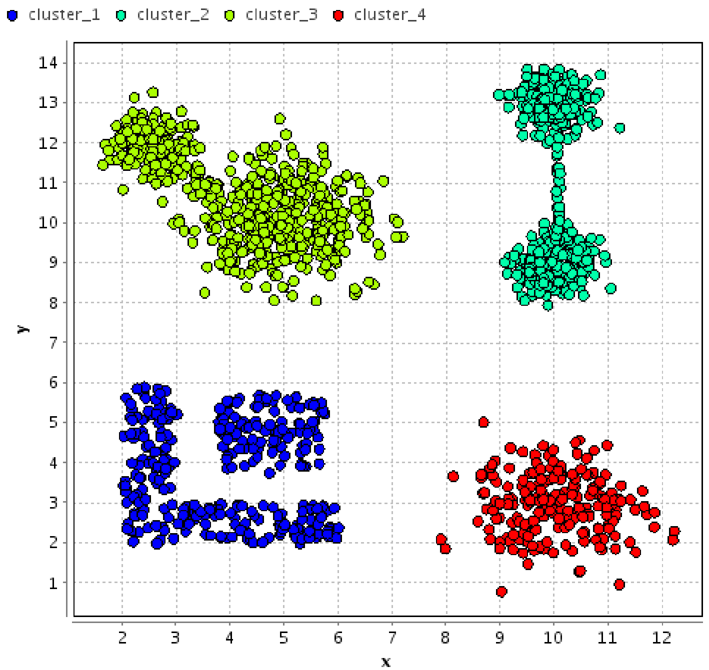
- clusters are modelled as dense areas separated by less dense areas
- density of an object = number of objects satisfying a similarity threshold

DBSCAN [Ester & Kriegel et al., 1996]

- dense areas → core objects
- object count in defined ε-neighbourhood
- connection via overlapping neighbourhood
- parameters: *ε, minPts*

DENCLUE[Hinneburg, 1998]

- density via gaussian kernels
- cluster extraction by hill climbing



ε=1.0; minPts=5

Density-Based Clustering

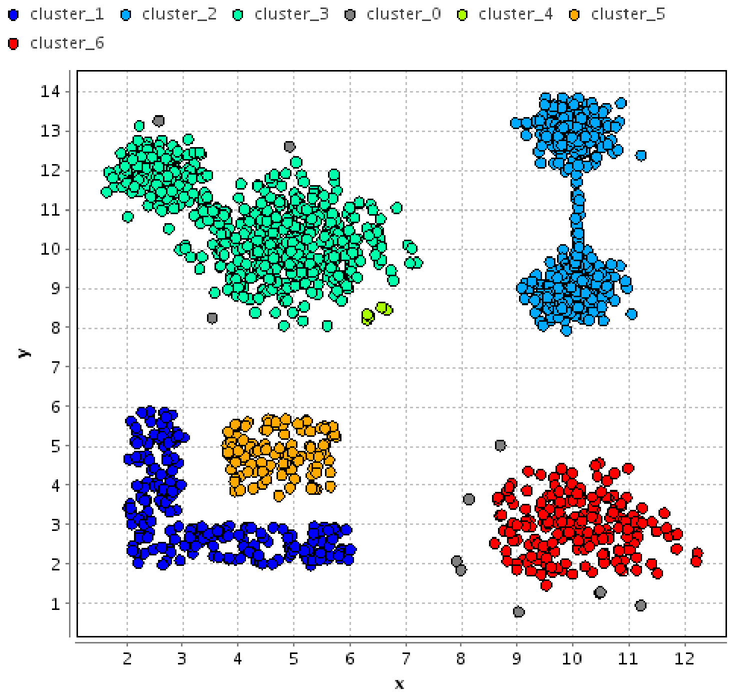
- clusters are modelled as dense areas separated by less dense areas
- density of an object = number of objects satisfying a similarity threshold

DBSCAN [Ester & Kriegel et al., 1996]

- dense areas → core objects
- object count in defined ε-neighbourhood
- connection via overlapping neighbourhood
- parameters: *ε, minPts*

DENCLUE[Hinneburg, 1998]

- density via gaussian kernels
- cluster extraction by hill climbing



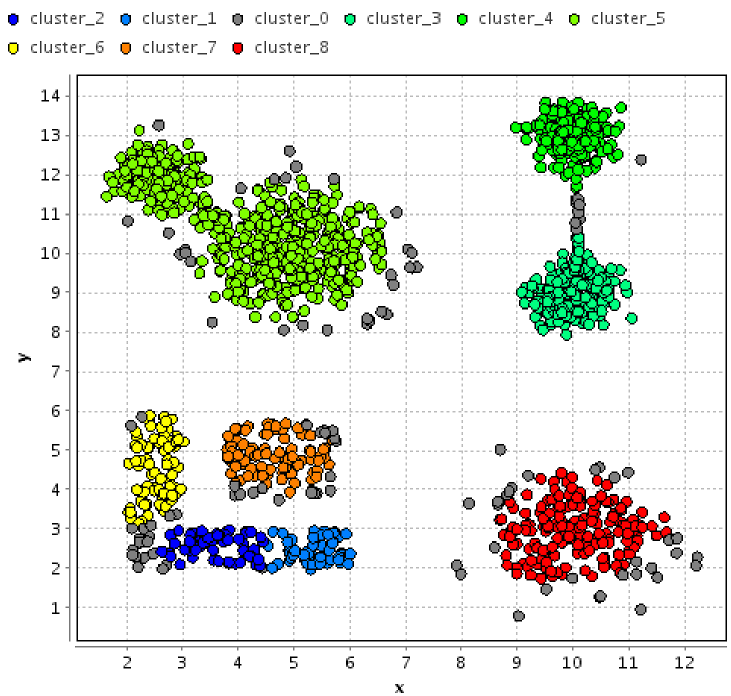
ε=0.5; minPts=5

Density-Based Clustering

- clusters are modelled as dense areas separated by less dense areas
- density of an object = number of objects satisfying a similarity threshold

DBSCAN [Ester & Kriegel et al., 1996]

- dense areas → core objects
- object count in defined ε-neighbourhood
- connection via overlapping neighbourhood
- parameters: *ε, minPts*



ε=0.5; minPts=20

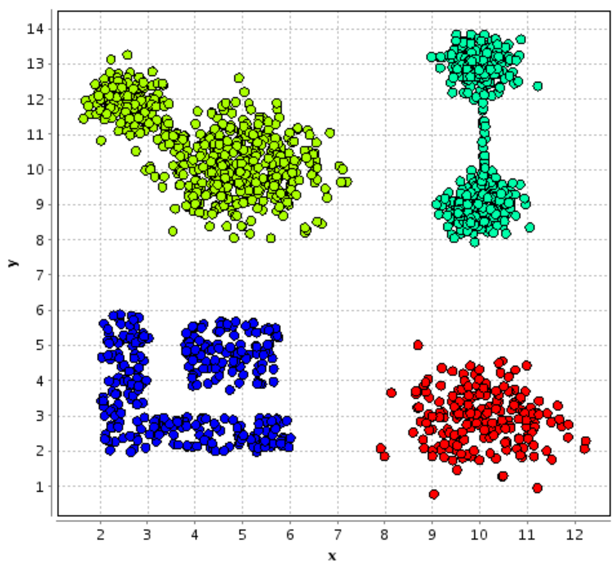
DENCLUE[Hinneburg, 1998]

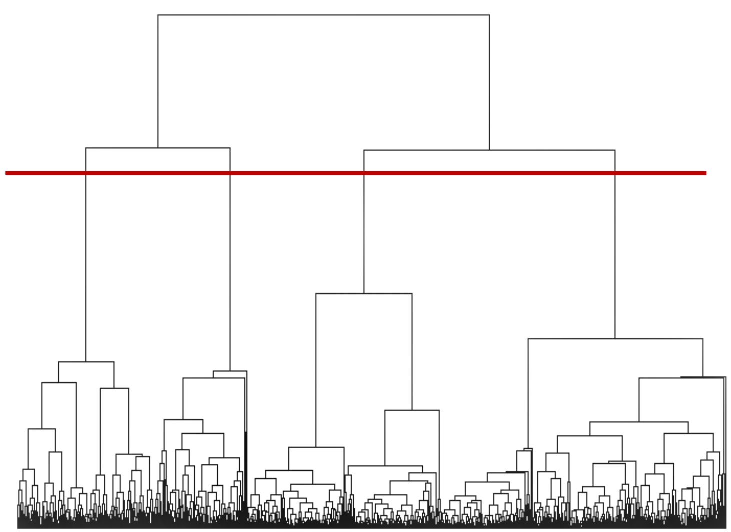
- density via gaussian kernels
- cluster extraction by hill climbing

Hierarchical Clusterings

- builds a hierarchy by progressively merging / splitting clusters
- clusterings are extracted by cutting the dendrogramm

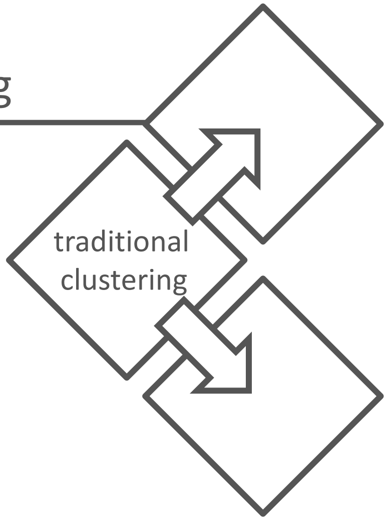
- bottom-up, *AGNES[Ward, 1963]*
- top-down,
  DIANA[Kaufmann & Rousseeuw, 1990]





multi-solution clustering

multiple clusterings

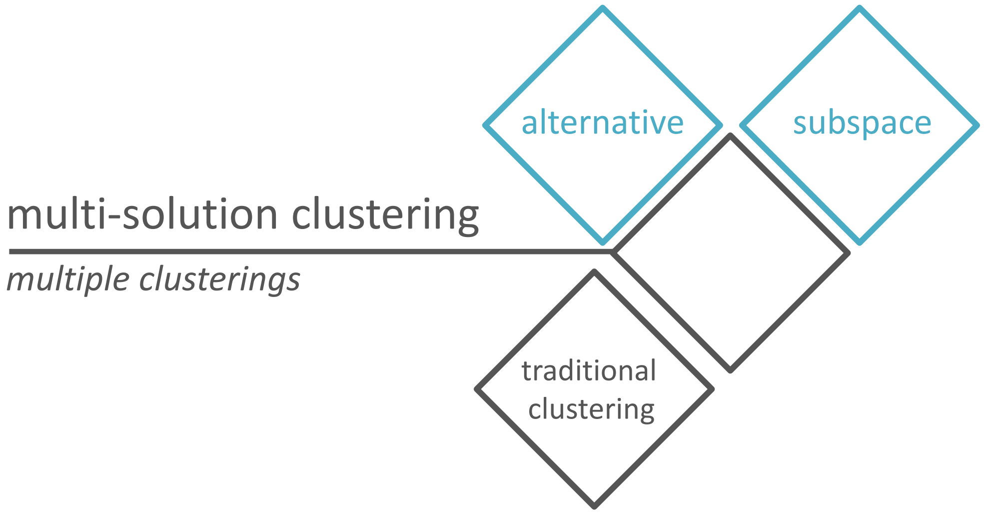


ensemble clustering

Algorithmics

Feedback

> Multi-Solution Clustering



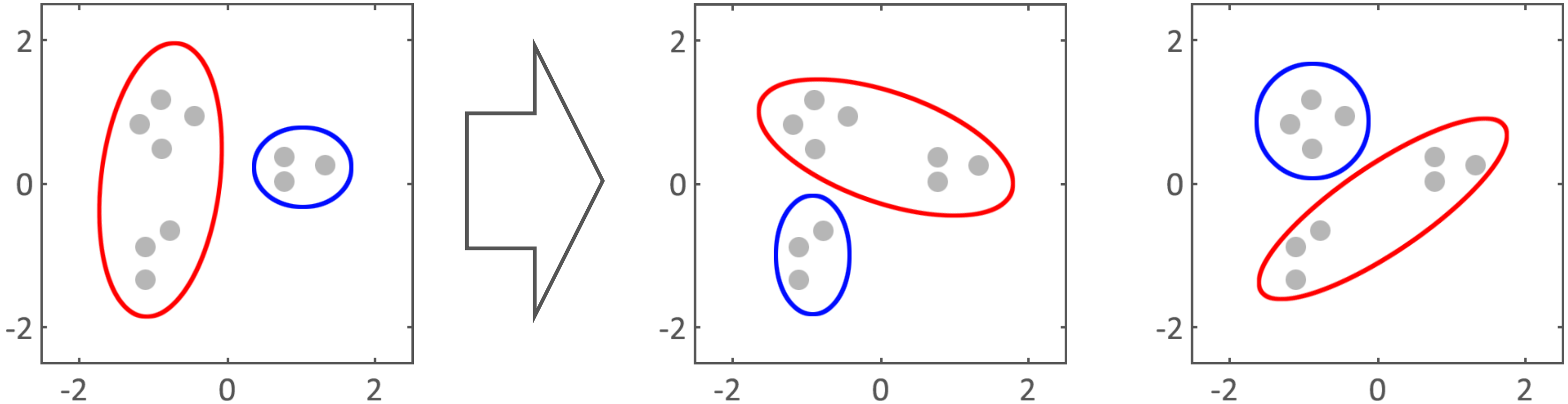
Algorithmics

Feedback

> Multi-Solution Clustering

Alternative Clustering

- generate initial clustering with traditional algorithm
- utilize given knowledge to find alternative clusterings
- generate a *dissimilar* clustering



initial clustering   alternative 1   alternative 2

Alternative Clustering: COALA[Bae & Bailey, 2006]

- generate alternative with same number of clusters
- high *dissimilarity* to initial clustering(cannot-link constraint)
- high *quality* alternative (objects in cluster still similar)
- trade-off between two goals controlled by parameter *w*
- if $d_{quality} < w * d_{dissimilarity}$ choose quality else dissimilarity

CAMI[Dang & Bailey, 2010]

- clusters as gaussian mixture
- quality by likelihood
- dissimilarity by mutual information

Orthogonal Clustering[Davidson & Qi, 2008]

- iterative transformation of database
- use of constraints

Subspace Clustering

- high dimensional data
- clusters can be observed in arbitrary attribute combinations (subspaces)
- detect multiple clusterings in different subspaces

CLIQUE [Agarawal et al. 1998]

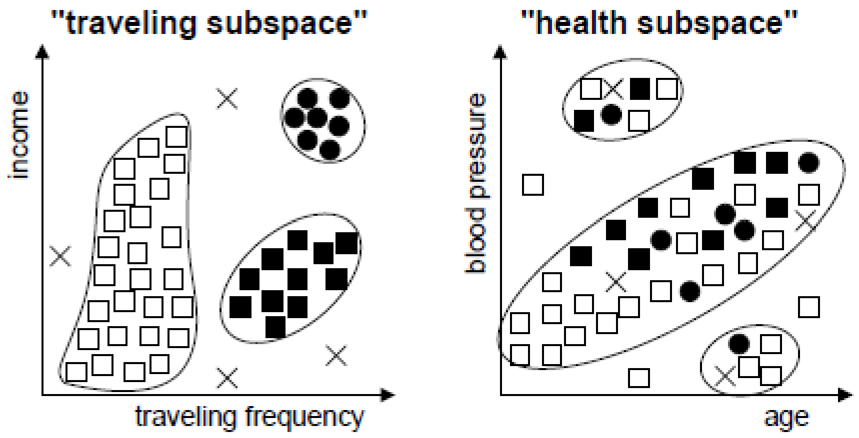
- based on grid cells
- find dense cells in all subspaces

FIRES[Kriegel et al.,2005]

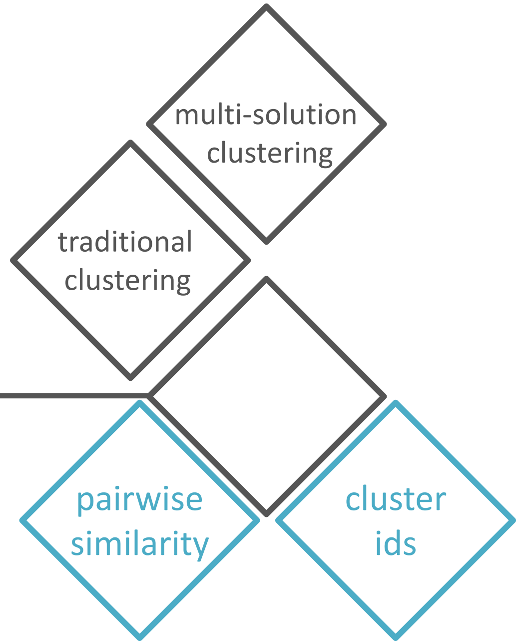
- stepwise merge of base clusters (1D)
- max. dimensional subspace clusters

DENSEST[Müller et al.,2009]

- estimation of dense subspaces
- correlation based (2D histograms)



> Ensemble Clustering



ensemble clustering

multiple base clusterings
one consensus clustering

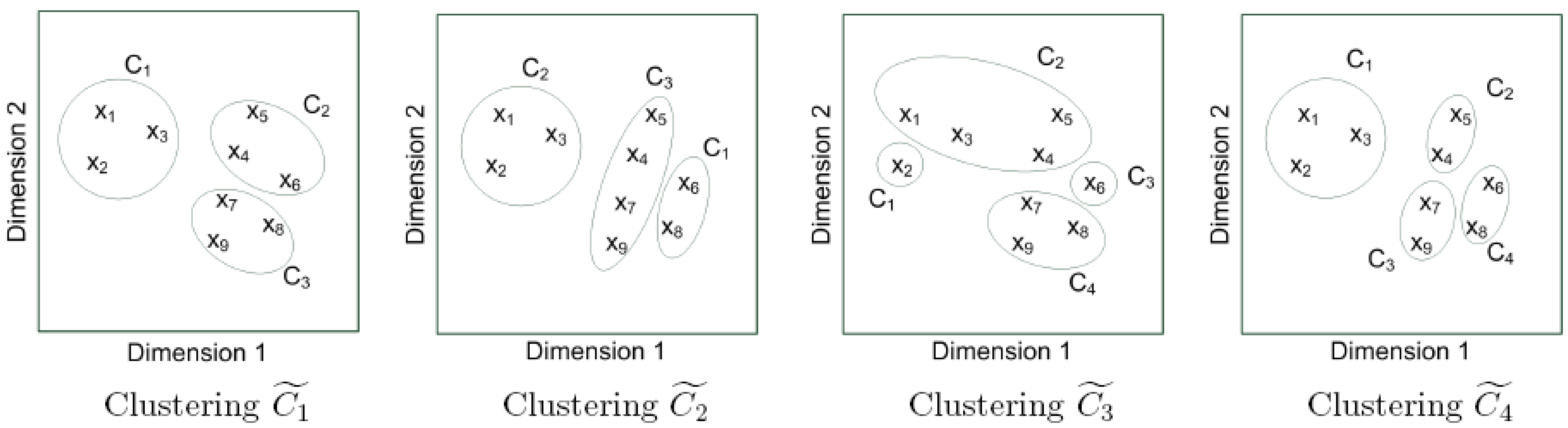
Algorithmics

Feedback

General Idea

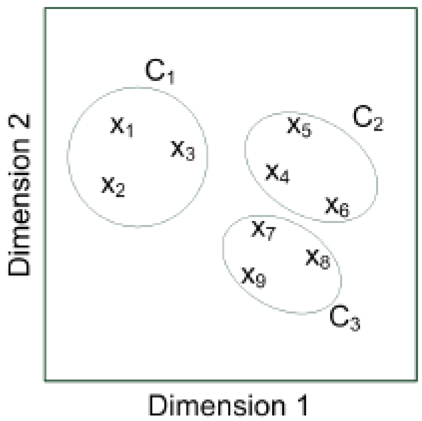
- generate multiple traditional clusterings
- employ different algorithms / parameters → ensemble
- combine ensemble to single robust consensus clustering

- different consensus mechanisms
- [Strehl & Ghosh, 2002], [Gionis et al., 2007]



Pairwise Similarity

- a pair of objects belongs to: *the same* or *different* cluster(s)
- pairwise similarities represented as coassociation matrices



Clustering $\widetilde{C_1}$

	x1	x2	x3	x4	x5	x6	x7	x8	x9
x1	-	**1**	**1**	0	0	0	0	0	0
x2	-	-	**1**	0	0	0	0	0	0
x3	-	-	-	0	0	0	0	0	0
x4	-	-	-	-	**1**	**1**	0	0	0
x5	-	-	-	-	-	**1**	0	0	0
x6	-	-	-	-	-	-	0	0	0
x7	-	-	-	-	-	-	-	**1**	**1**
x8	-	-	-	-	-	-	-	-	**1**
x9	-	-	-	-	-	-	-	-	-

Pairwise Similarity

- a pair of objects belongs to: *the same* or *different* cluster(s)
- pairwise similarities represented as coassociation matrices

- simple example based on [Gionis et al., 2007]
- goal: generate consensus clustering with maximal similarity to ensemble

	x1	x2	x3	x4	x5	x6	x7	x8	x9
x1	-	1	1	0	0	0	0	0	0
x2	-	-	1	0	0	0	0	0	0
x3	-	-	-	0	0	0	0	0	0
x4	-	-	-	-	1	1	0	0	0
x5	-	-	-	-	-	1	0	0	0
x6	-	-	-	-	-	-	0	0	0
x7	-	-	-	-	-	-	-	1	1
x8	-	-	-	-	-	-	-	-	1
x9	-	-	-	-	-	-	-	-	-

	x1	x2	x3	x4	x5	x6	x7	x8	x9
x1	-	1	1	0	0	0	0	0	0
x2	-	-	1	0	0	0	0	0	0
x3	-	-	-	0	0	0	0	0	0
x4	-	-	-	-	1	0	1	0	1
x5	-	-	-	-	-	0	1	0	1
x6	-	-	-	-	-	-	0	1	0
x7	-	-	-	-	-	-	-	0	1
x8	-	-	-	-	-	-	-	-	0
x9	-	-	-	-	-	-	-	-	-

	x1	x2	x3	x4	x5	x6	x7	x8	x9
x1	-	0	1	1	1	0	0	0	0
x2	-	-	0	0	0	0	0	0	0
x3	-	-	-	1	1	0	0	0	0
x4	-	-	-	-	1	0	0	0	0
x5	-	-	-	-	-	0	0	0	0
x6	-	-	-	-	-	-	0	0	0
x7	-	-	-	-	-	-	-	1	1
x8	-	-	-	-	-	-	-	-	1
x9	-	-	-	-	-	-	-	-	-

	x1	x2	x3	x4	x5	x6	x7	x8	x9
x1	-	1	1	0	0	0	0	0	0
x2	-	-	1	0	0	0	0	0	0
x3	-	-	-	0	0	0	0	0	0
x4	-	-	-	-	1	0	0	0	0
x5	-	-	-	-	-	0	0	0	0
x6	-	-	-	-	-	-	0	1	0
x7	-	-	-	-	-	-	-	0	1
x8	-	-	-	-	-	-	-	-	0
x9	-	-	-	-	-	-	-	-	-

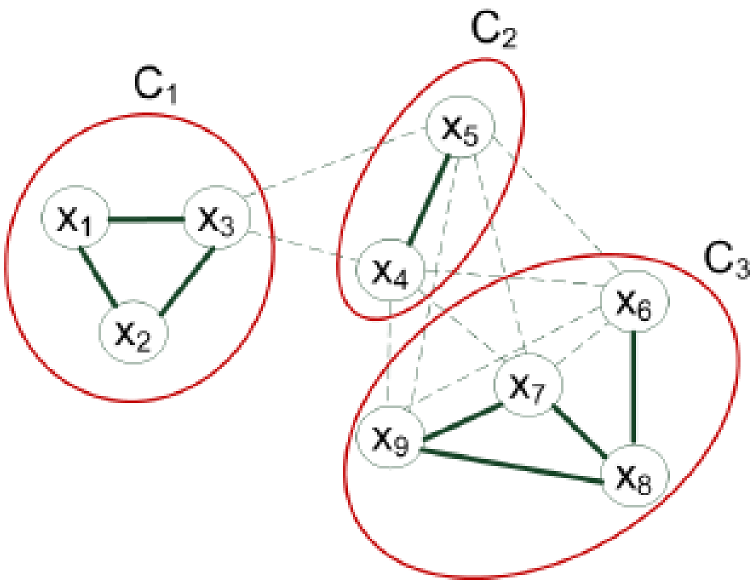
Consensus Mechanism

- majority decision
- pairs located in the same cluster in at least 50% of the ensemble, are also part of the same cluster in the consensus clustering

	x1	x2	x3	x4	x5	x6	x7	x8	x9
x1	-	**3/4**	**4/4**	-	-	-	-	-	-
x2	-	-	**3/4**	-	-	-	-	-	-
x3	-	-	-	-	-	-	-	-	-
x4	-	-	-	-	**4/4**	-	-	-	-
x5	-	-	-	-	-	-	-	-	-
x6	-	-	-	-	-	-	-	**3/4**	-
x7	-	-	-	-	-	-	-	**2/4**	**4/4**
x8	-	-	-	-	-	-	-	-	**2/4**
x9	-	-	-	-	-	-	-	-	-

Consensus Mechanism

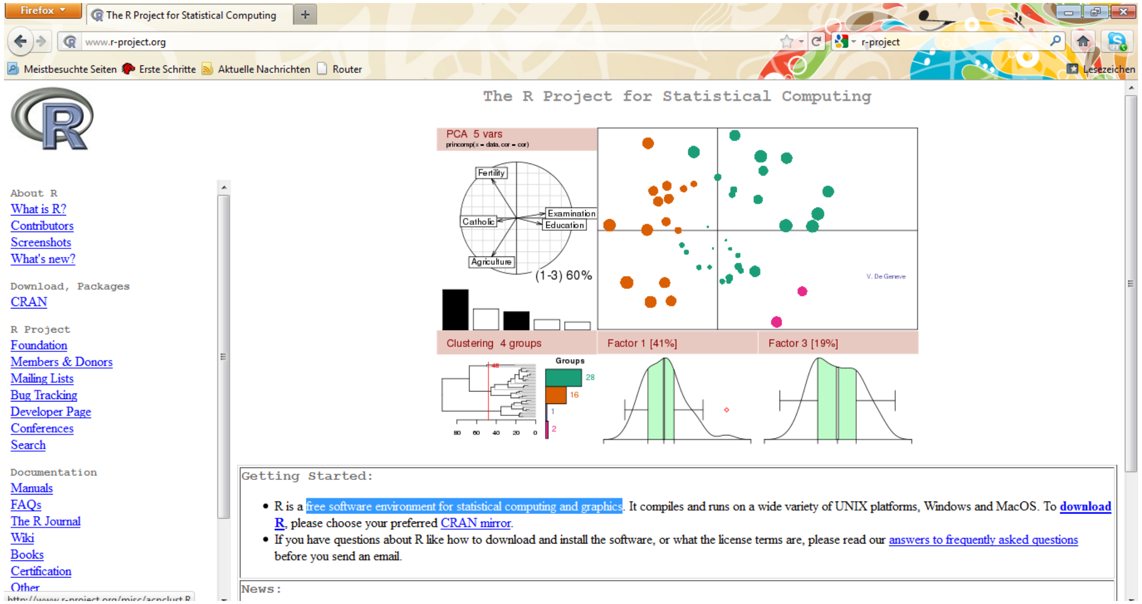
- majority decision
- pairs located in the same cluster in at least 50% of the ensemble, are also part of the same cluster in the consensus clustering
- more consensus mechanisms exist
- e.g. graph partitioning via METIS [Karypis et al., 1997]



Algorithm Integration in databases

Dipl.-Inf. Phillip Grosse
philipp.grosse@sap.com

- the R Project for Statistical Computing
- free software environment for statistical computing and graphics
- includes lots of standard math functions/libraries
- function- and object-oriented
- flexible/extensible



Vector: basic datastructure in R

```
> x <- 1            # vector of size 1
> x <- c(1,2,3)     # vector of size 3
> x <- 1:10         # vector of size 10
> "abc" -> y        # vector of size 1
```

Vector output

```
> x
[1] 1 2 3 4 5 6 7 8 9 10
> sprintf("y is %s", y)
[1] "y is abc"
```

Vector arithmetic

- element-wise basic operators (+ - * / ^)

```
> x <- c(1,2,3,4)
> y <- c(2,3,4,5)
> x*y
[1] 2 6 12 20
```

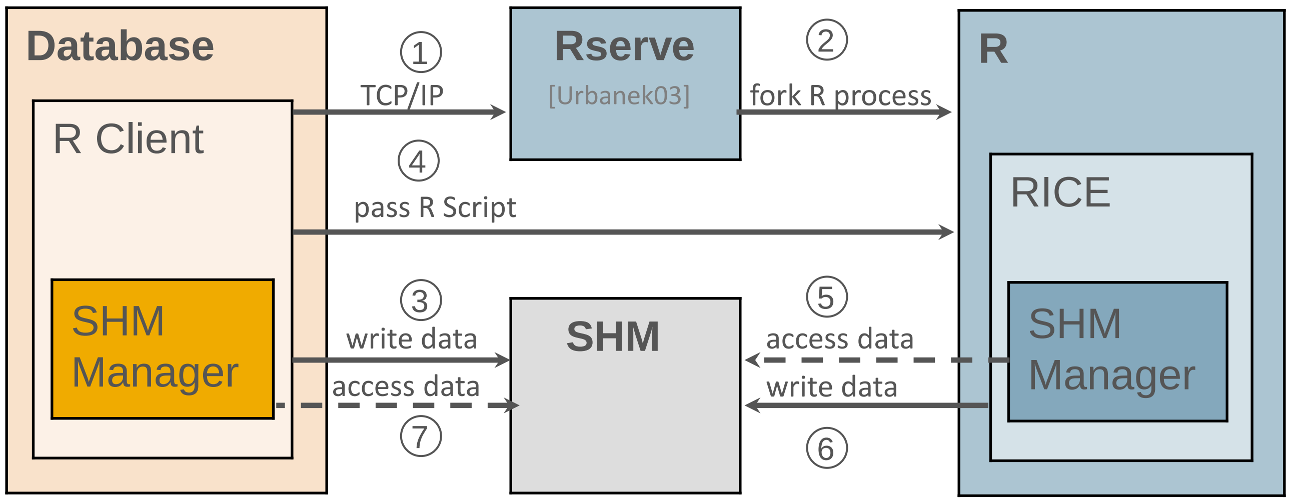
- vectorrecycling (short vectors)

```
> y <- c(2,7)
> x*y
[1] 2 14 6 28
```

- arithmetic functions

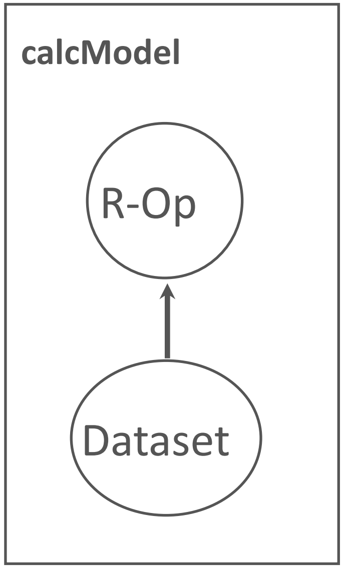
log, exp, sin, cos, tan, sqrt, min, max, range, length, sum, prod, mean, var

> R as HANA operator (R-OP)



> R scripts in databases

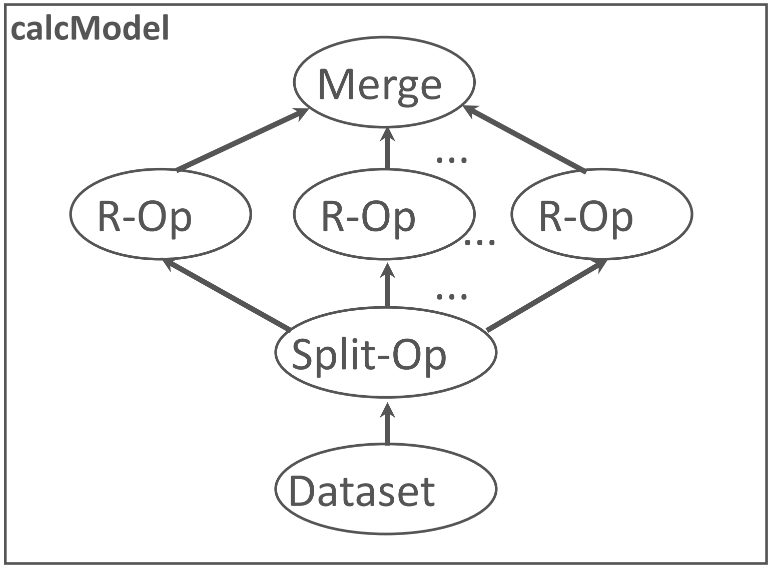

```
## Creates some dummy data
CREATE INSERT ONLY COLUMN TABLE Test1(A INTEGER, B DOUBLE);
INSERT INTO Test1 VALUES(0, 2.5);
INSERT INTO Test1 VALUES(1, 1.5);
INSERT INTO Test1 VALUES(2, 2.0);

CREATE COLUMN TABLE "ResultTable" AS TABLE "TEST1" WITH NO DATA;
ALTER TABLE "ResultTable"  ADD ("C" DOUBLE);

## Creates an SQL-script function including the R script
CREATE PROCEDURE ROP(IN input1 "Test1", OUT result "ResultTable" )
LANGUAGE RLANG AS
BEGIN
        A <- input1$A ;
        B <- input1$B ;
        C <- A * B;

        result <- cbind(A, B, C) ;
END;

## execute SQL-script function and retrieve result
CALL ROP(Test1, "ResultTable");
SELECT * FROM "ResultTable";
```



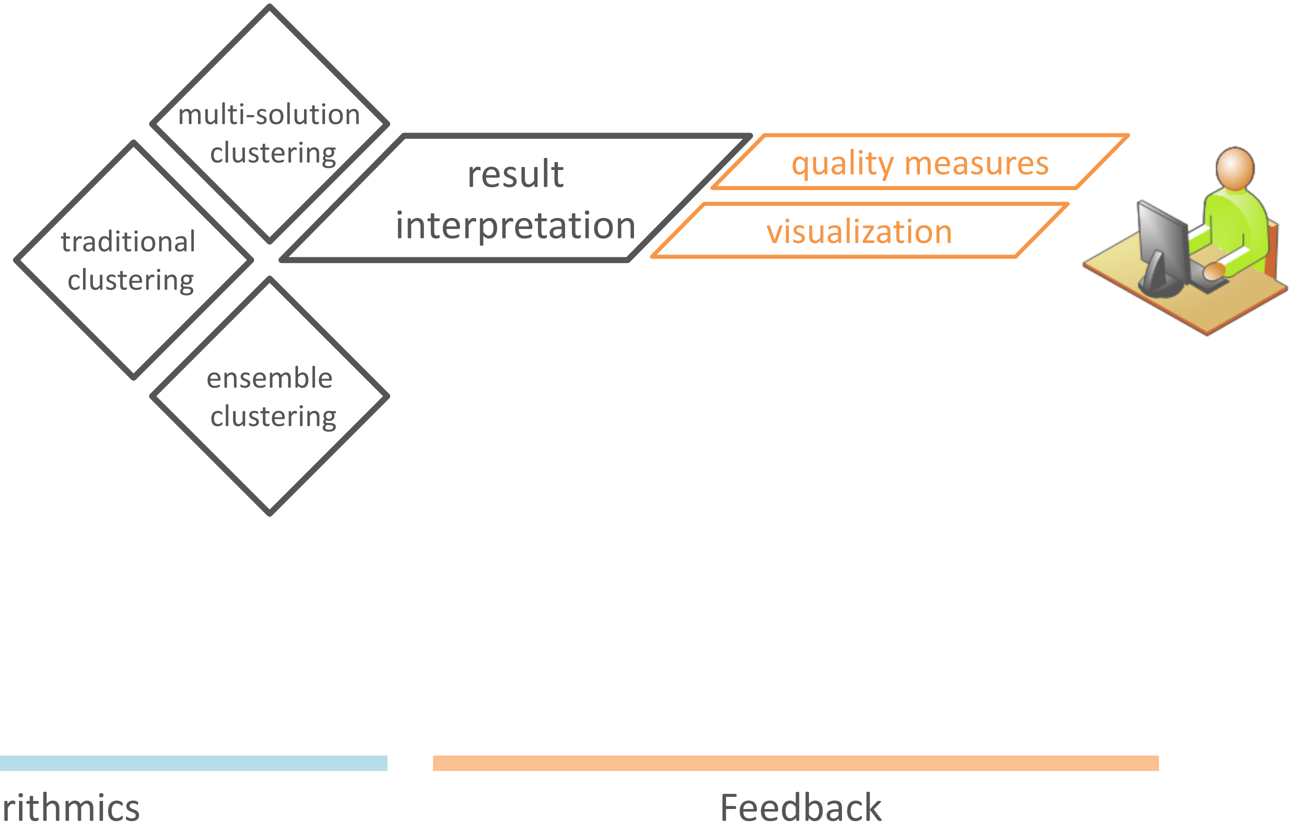
A	B	C
0	2.5	0.0
1	1.5	1.5
2	2.0	4.0

> parallel R operators

```
## call ROP in parallel
DROP PROCEDURE testPartition1;
CREATE PROCEDURE testPartition1 (OUT result "ResultTable")
AS  SQLSCRIPT
BEGIN
        input1 = select * from Test1;

        p = CE_PARTITION([@input1@],[],'''HASH 3 A''');
        CALL ROP(@p$$input1@, r);
        s = CE_MERGE([@r@]);

        result = select * from @s$$r@;
END;

CALL testPartition1(?);
```



A	B	C
0	2.5	0.0
1	1.5	1.5
2	2.0	4.0


multi-solution clustering
traditional clustering
ensemble clustering
result interpretation
quality measures
visualization
Algorithmics
Feedback

> Result Interpretation

Quality measures

- *„How good is the obtained result?“*
- Problem: **There is no universally valid definition of clustering quality**
- multiple approaches and metrics

Rand index [Rand , 1971]

- comparison to known solution

Dunns Index[Dunn, 1974]

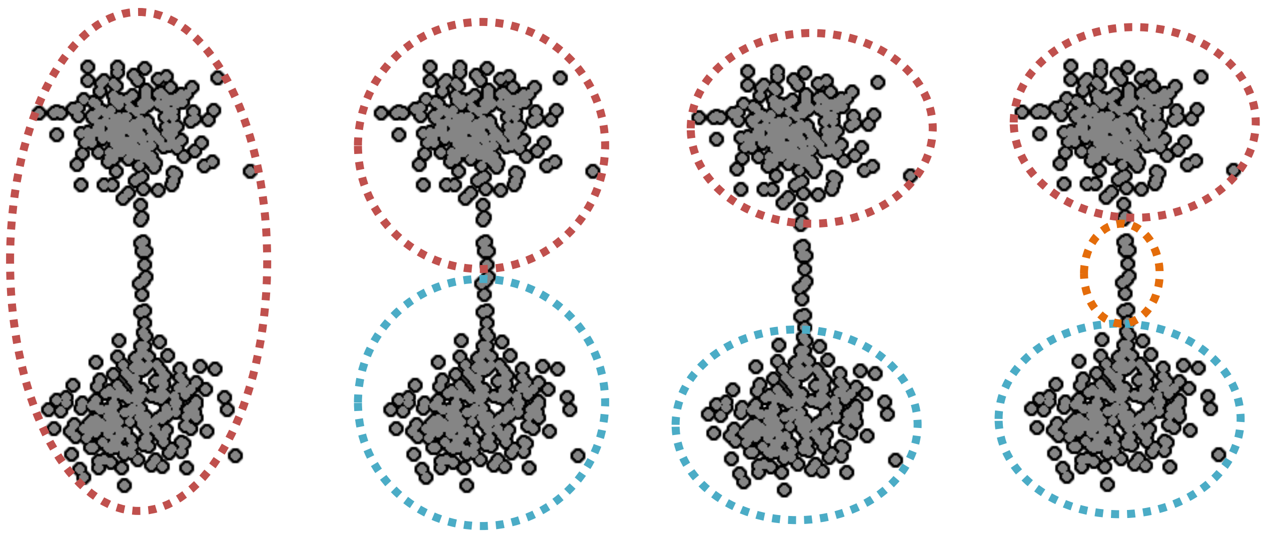
- ratio intercluster / intracluster distances
- higher score is better

Davis Bouldin Index[Davies & Bouldin, 1979]

- ratio intercluster / intracluster distances
- lower score is better

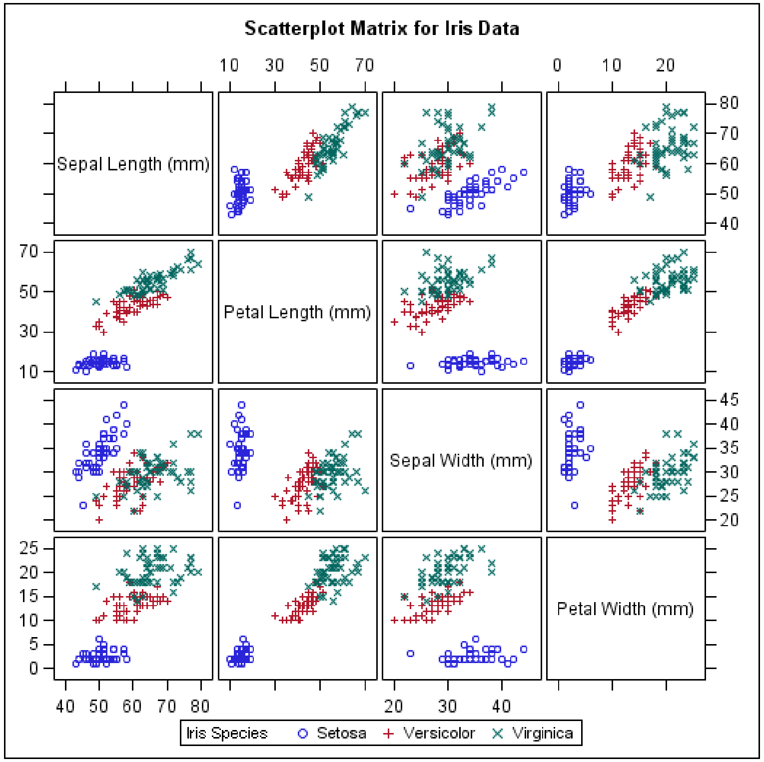
Visualization

- use visual representations to evaluate clustering quality
- utilize perception capacity of humans
- result interpretation left to user → subjective quality evaluation
- communicate information about structures
- data-driven → visualize whole dataset

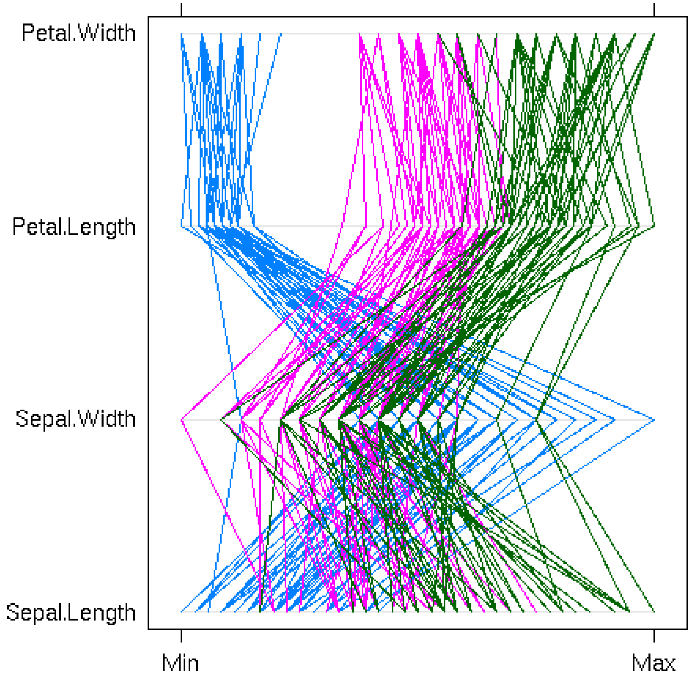


Visualization: examples

- all objects and dimensions are visualized → scalability issues for large scale data
- display multiple dimensions on a 2D screen



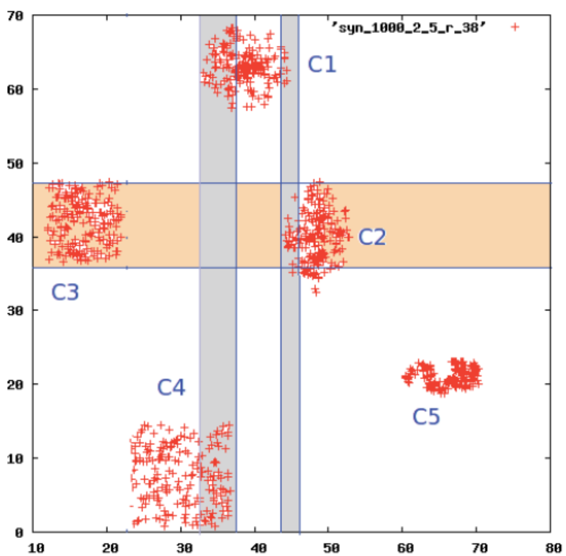
scatterplot matrix

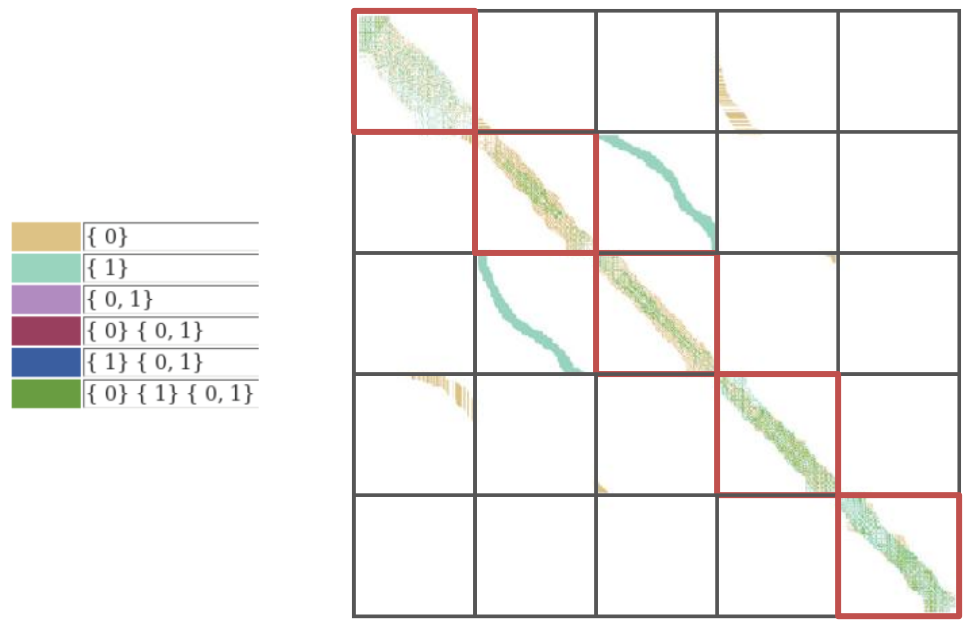


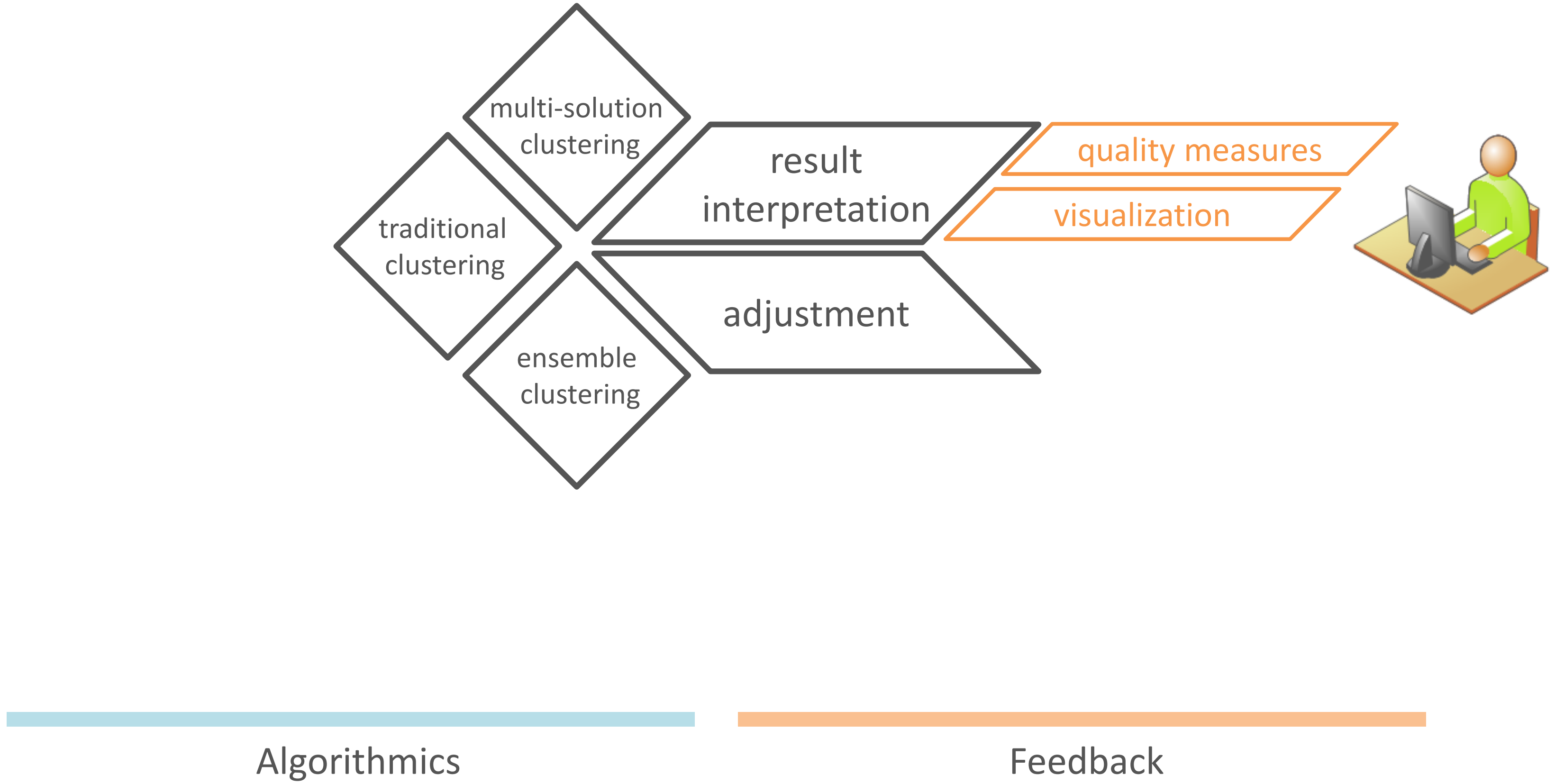
parallel coordinates[Inselberg, 1985]

Visualization: Heidi Matrix[Vadapali et al.,2009]

- visualize clusters in high dimensional space
- show cluster overlap in subspaces
- pixel technique based on k-nearest neighbour relations
- pixel= object pair, color= subspace






multi-solution clustering
traditional clustering
ensemble clustering
result interpretation
adjustment
quality measures
visualization
Algorithmics
Feedback

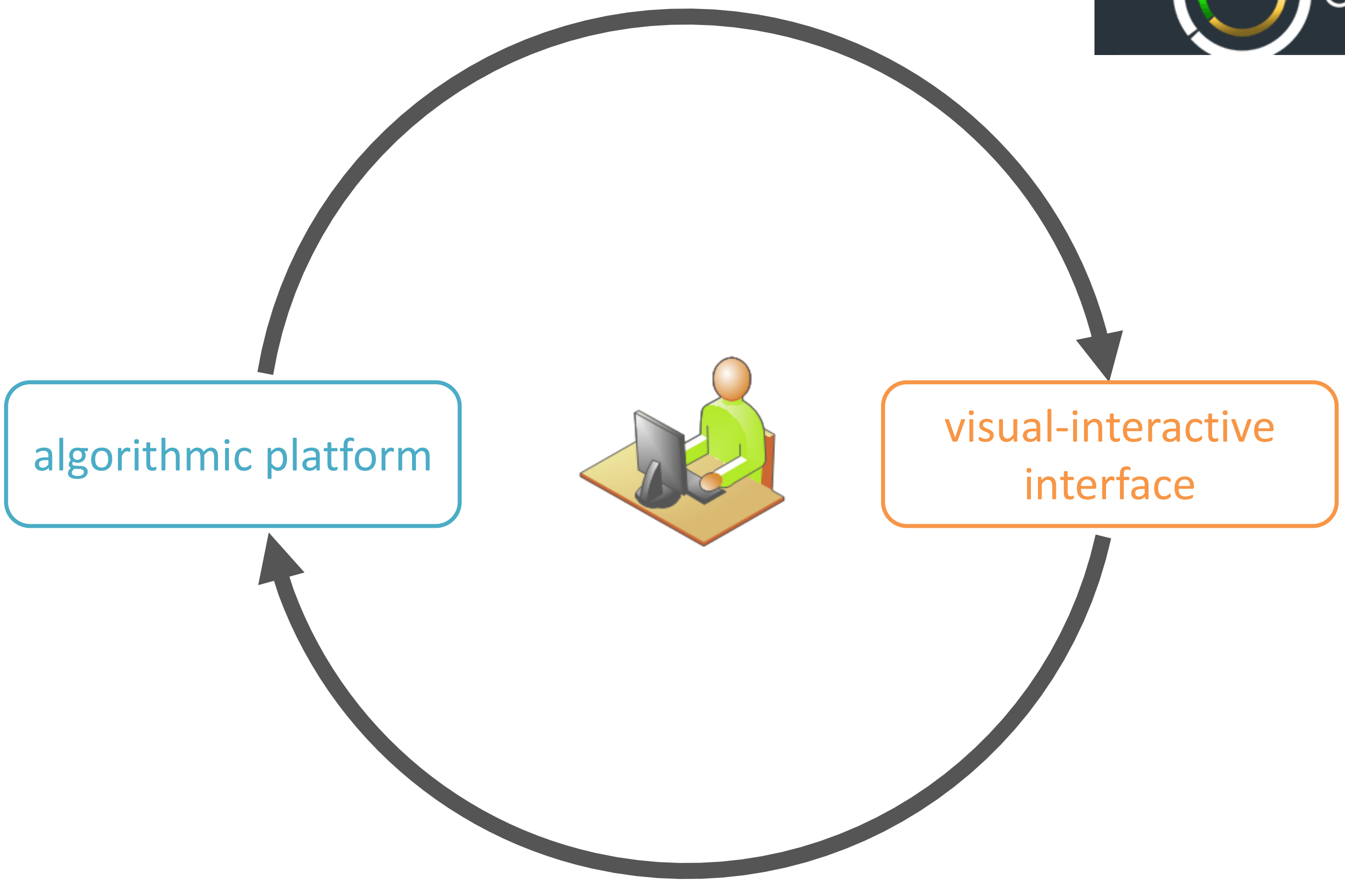
Result adjustment

- best practise: adjustment via re-parameterization or algorithm swap
- to infer modifications from result interpretation
- adjust low-level parameters
- only indirect

Clustering with interactive Feedback [Balcan & Blum, 2008]

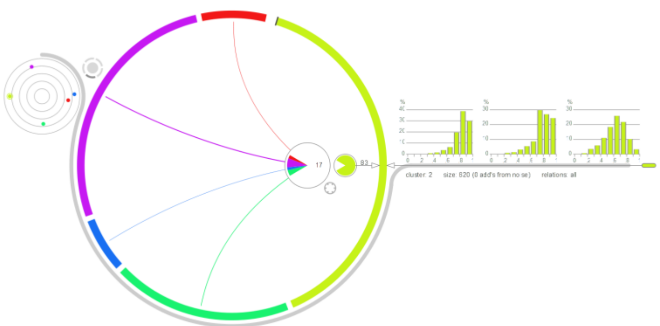
- theoretical proof of concept
- user adjustsments via high-level feedback: *split* and *merge*
- limitations: 1d data, target solution available to the user

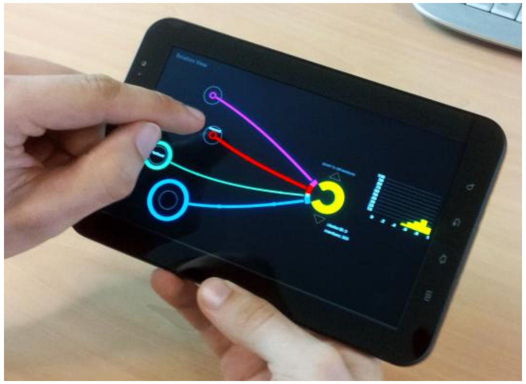
> Our Research Area

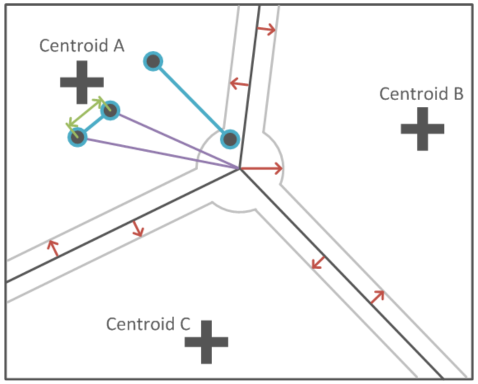


Generalized Process

- based on Shneidermann information seeking mantra
- Visualization, Interaction,
- algorithmic platform based on multiple clustering solutions





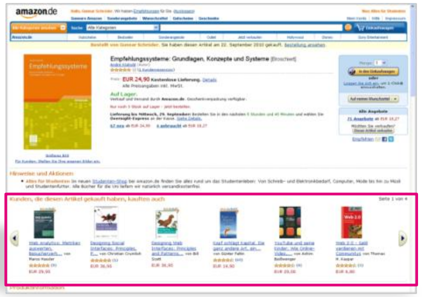


A complex use-case scenario

- self-optimising recommender systems

Dipl.-Inf. Gunnar Schröder
gunnar.schroeder@tu-dresden.de

amazon.com



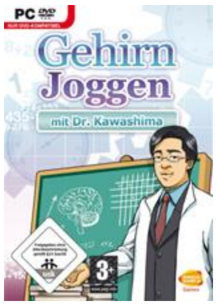
last.fm

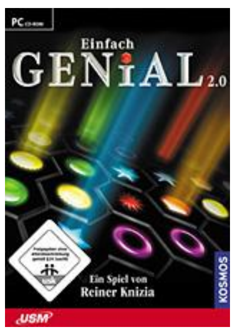
facebook

Predict user ratings:

Products for cross selling & product classification:

Similar items & similar users





> Challenges

Huge amounts of data

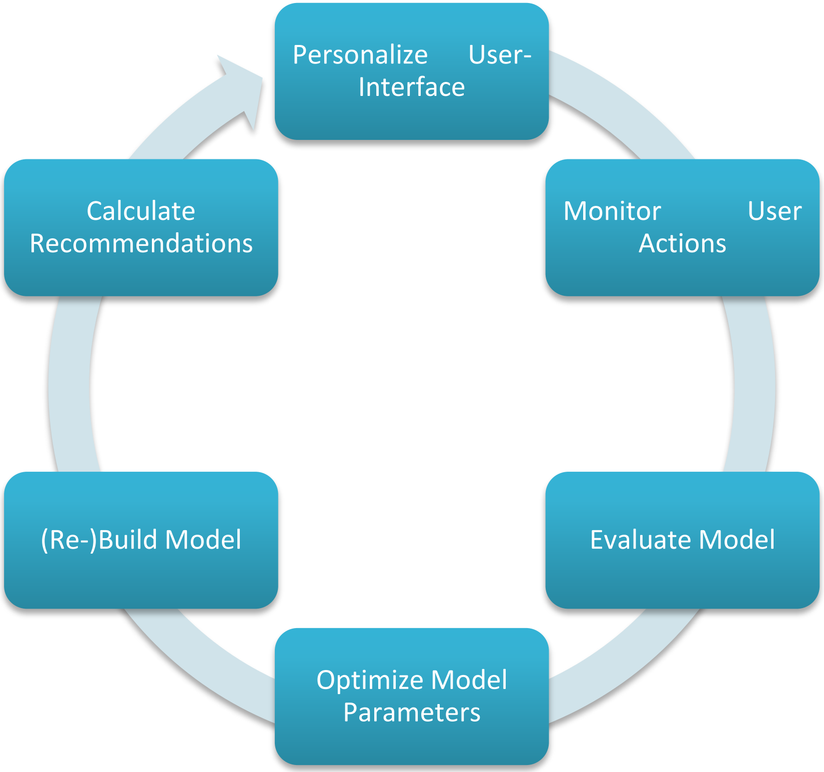
- building recommender models is **expensive!**

Dynamic

- models age
- user preferences change
- new users & items

Parameterization

- data and domain dependent
- support
- automatic model maintenance



Questions?