Clustering Techniques for Large Data Sets

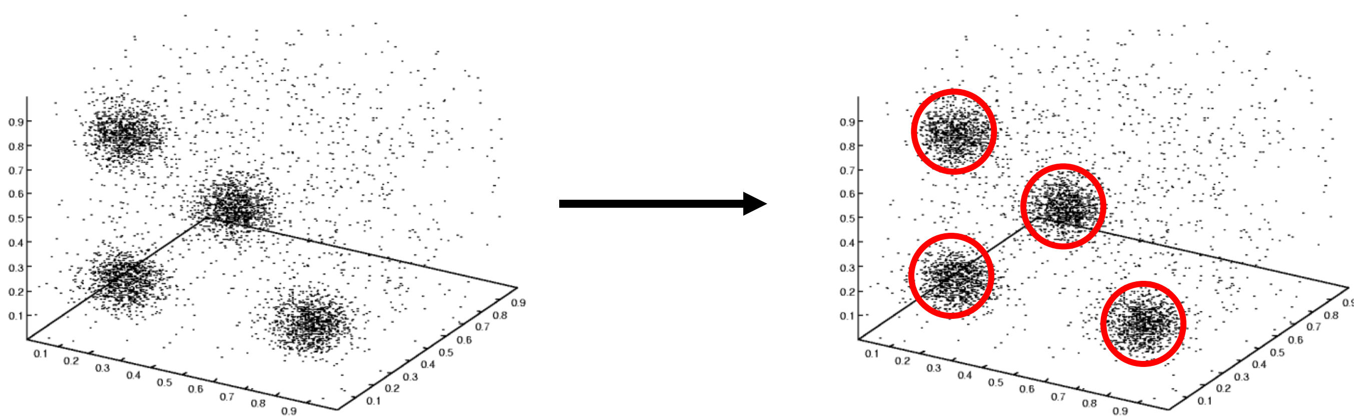
From the Past to the Future

Alexander Hinneburg, Daniel A. Keim

University of Halle

Introduction

- **Application Example:** Marketing
  - Given:
    - Large data base of customer data containing their properties and past buying records
  - Goal:
    - Find groups of customers with similar behavior
    - Find customers with unusual behavior



Introduction

- **Application Example:**
  Class Finding in CAD-Databases
  - Given:
    - Large data base of CAD data containing abstract feature vectors (Fourier, Wavelet, ...)
  - Goal:
    - Find homogeneous groups of similar CAD parts
    - Determine standard parts for each group
    - Use standard parts instead of special parts
      (→ reduction of the number of parts to be produced)

Introduction

Problem Description

- Given:

  A data set with *N* *d*-dimensional data items.

- Task:

  Determine a natural partitioning of the data set into a number of clusters (k) and noise.

Introduction

From the Past ...

- Clustering is a well-known problem in statistics [Sch 64, Wis 69]
- more recent research in
  - machine learning [Roj 96],
  - databases [CHY 96], and
  - visualization [Kei 96] ...

Introduction

... to the Future

- *Effective* and *efficient* clustering algorithms for *large high-dimensional* data sets with *high noise level*
- Requires ***Scalability*** with respect to
  - the *number of data points (**N**)*
  - the *number of dimensions (**d**)*
  - the *noise level*

Overview

1. Introduction
2. Basic Methods
   - 2.1 k-Means
   - 2.2 Linkage-based Methods
   - 2.3 Kernel-Density Estimation Methods
3. Methods Improving the Effectiveness and Efficiency
   - 2.1 Model- and Optimization-based Approaches
   - 2.2 Density-based Approaches
   - 2.3 Hybrid Approaches
4. Summary and Conclusions

From the Past ...

... to the Future

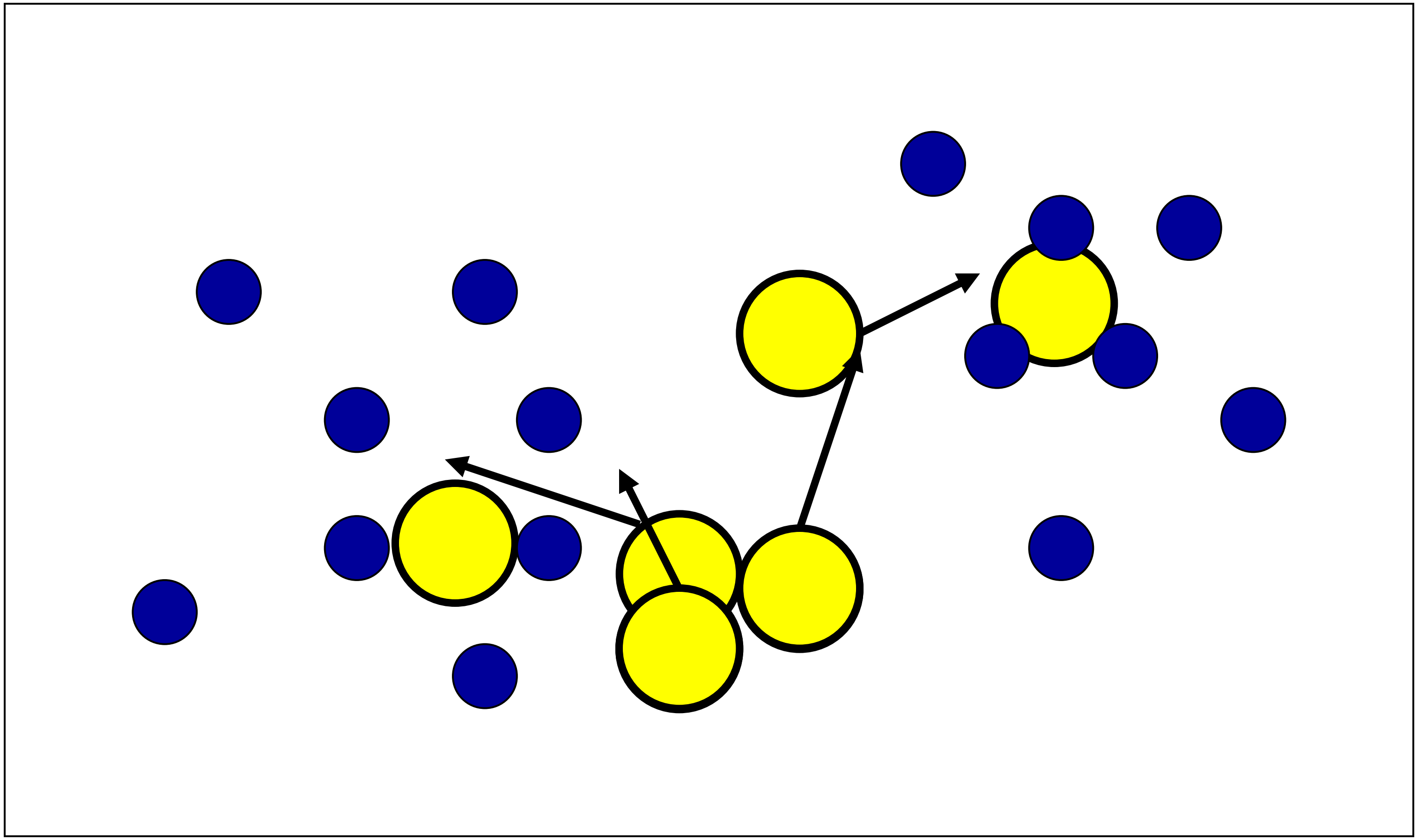
K-Means [Fuk 90]

- Determine *k* prototypes (*p*) of a given data set
- Assign data points to nearest prototype
- Minimize distance criterion:

$$\sum_{i=1}^{k} \sum_{j=1}^{N} d(p_i, x_j^i)$$

- Iterative Algorithm
  - – Shift the prototypes towards the mean of their point set
  - – Re-assign the data points to the nearest prototype

K-Means: Example

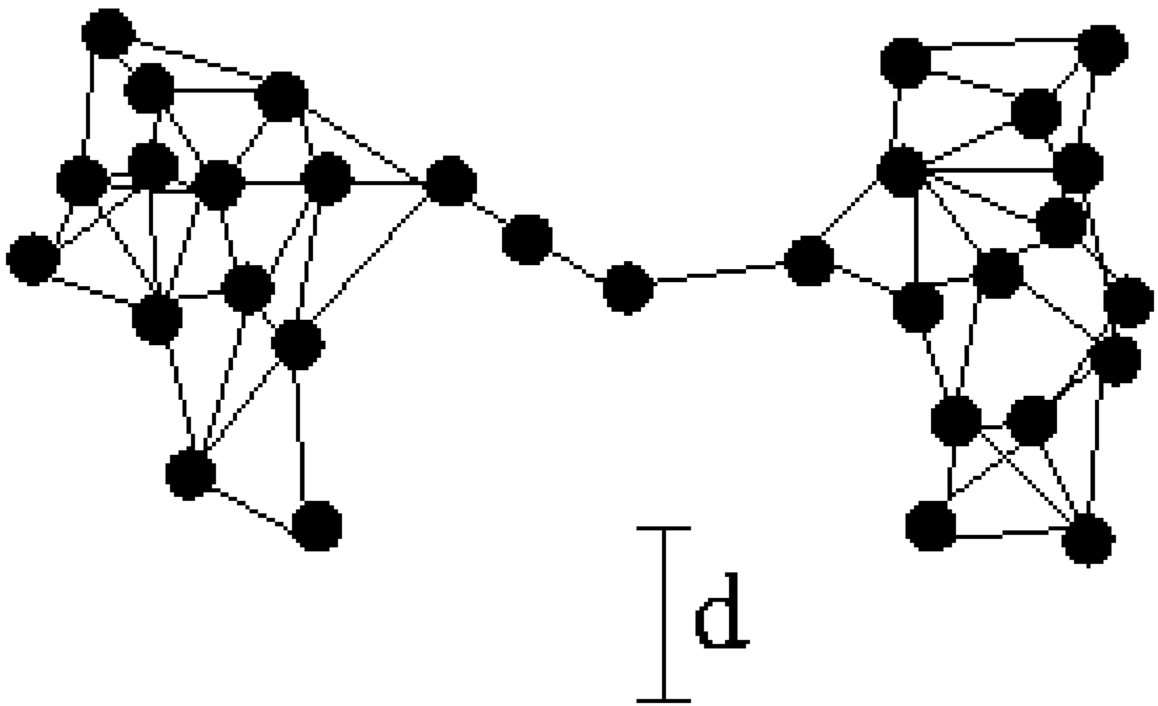


Expectation Maximization [Lau 95]

- Generalization of k-Means
  (➔ probabilistic assignment of points to clusters)
- Baisc Idea:
  - – Estimate parameters of k Gaussians
  - – Optimize the probability, that the mixture of parameterized Gaussians fits the data
  - – Iterative algorithm similar to k-Means

Linkage -based Methods
(from Statistics) [Boc 74]

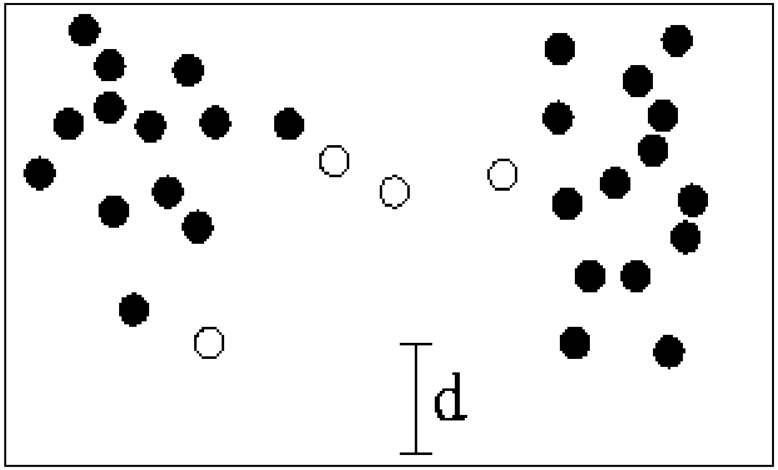
- Single Linkage (Connected components for distance d)



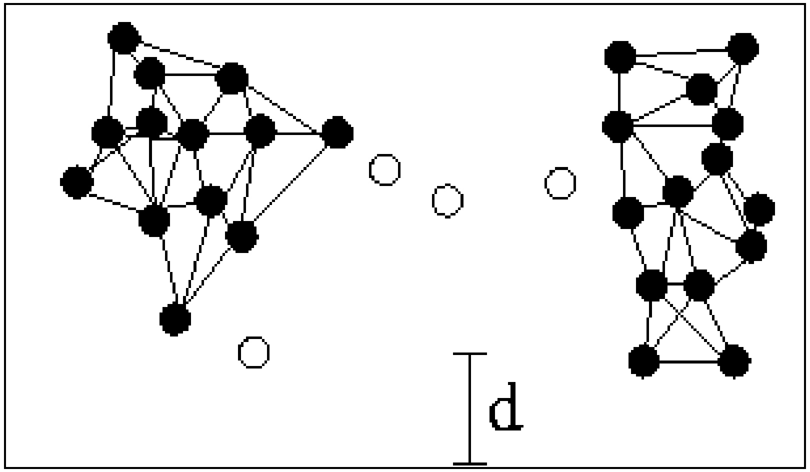
Linkage -based Methods [Boc 74]

- Method of Wishart [Wis 69] (Min. no. of points: c=4)

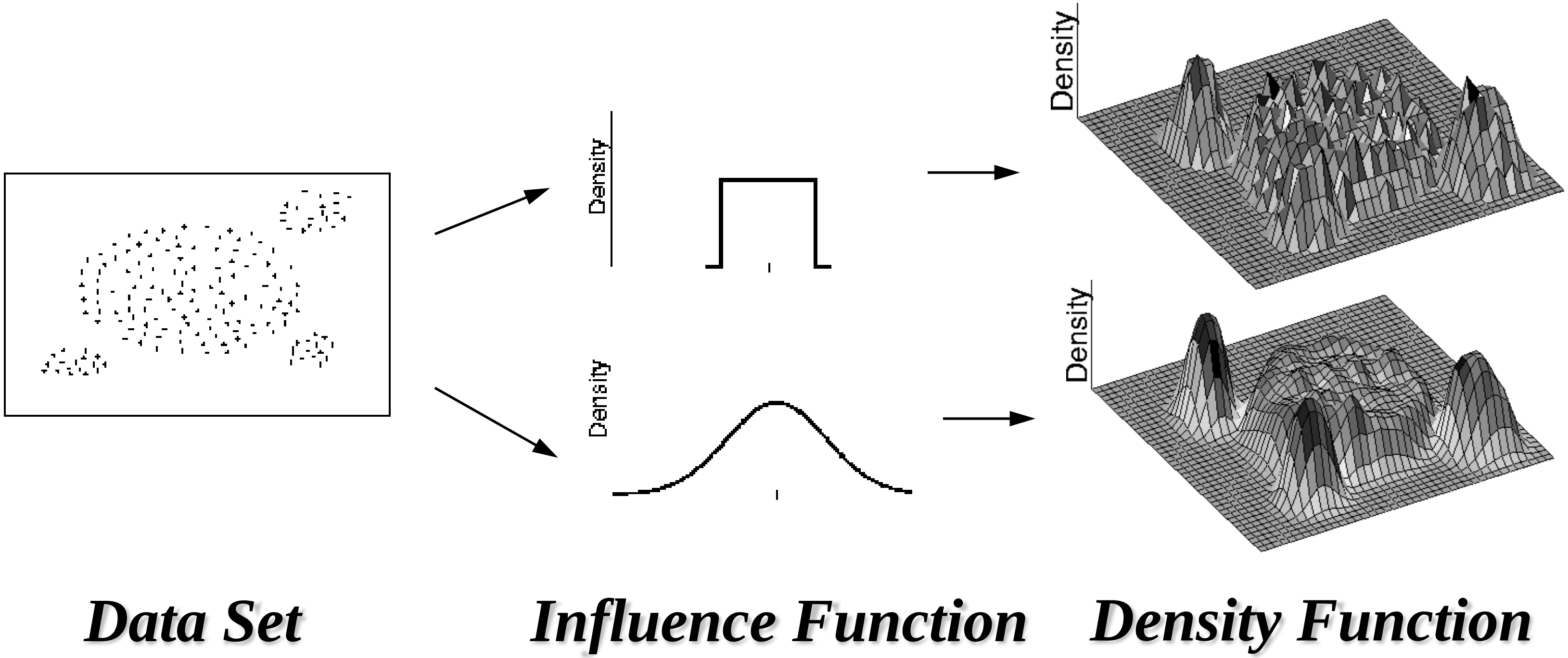
Reduce data set



Apply Single Linkage



Kernel Density Estimation



Influence Function: Influence of a data point in its neighborhood

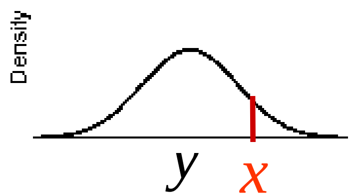
Density Function: Sum of the influences of all data points

Kernel Density Estimation

Influence Function

The influence of a data point *y* at a point *x* in the data space is modeled by a function $f_B^y : F^d \rightarrow \Re$,

e.g., $$f_{Gauss}^y(x) = e^{-\frac{d(x,y)^2}{2\sigma^2}}.$$

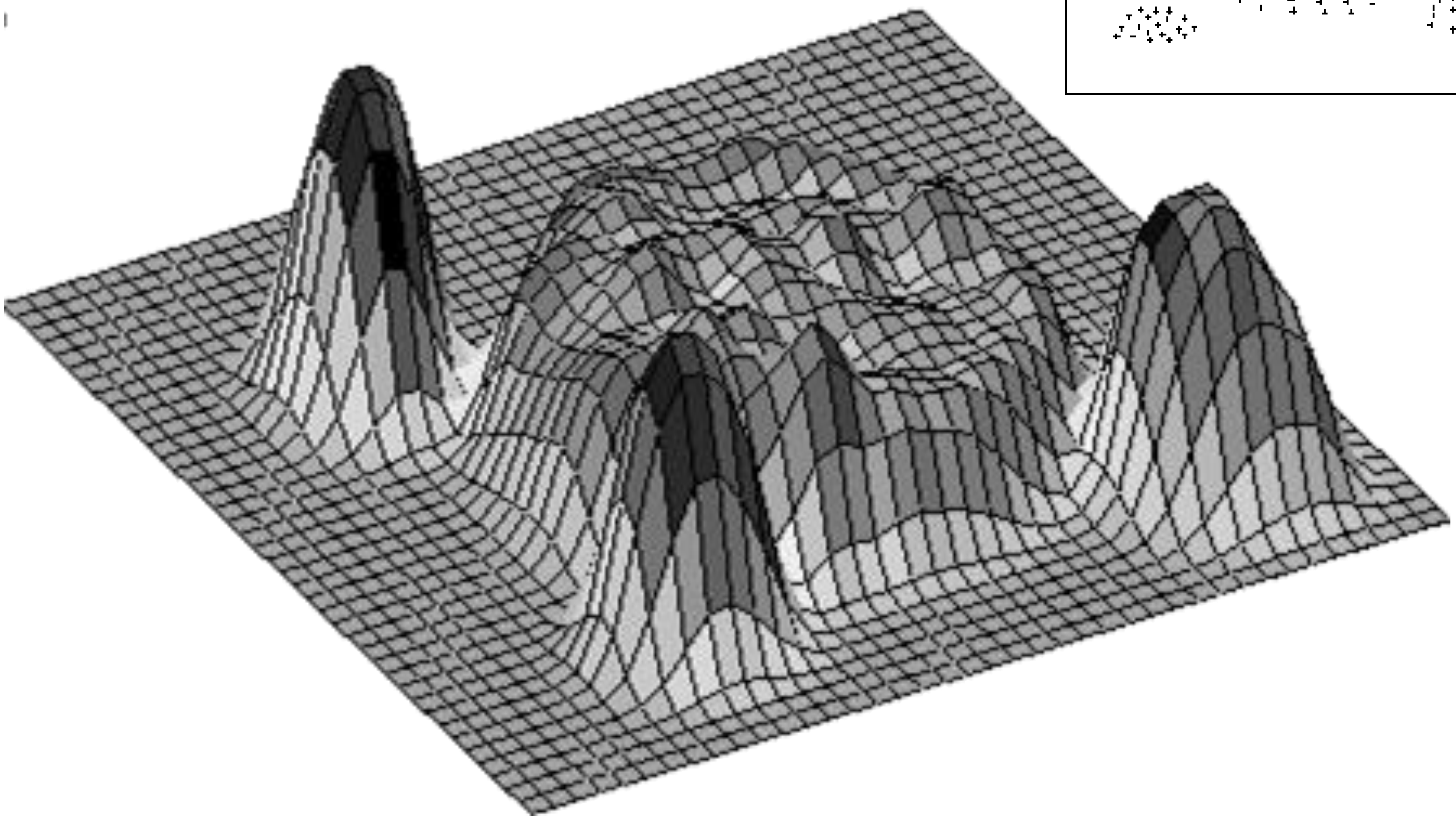


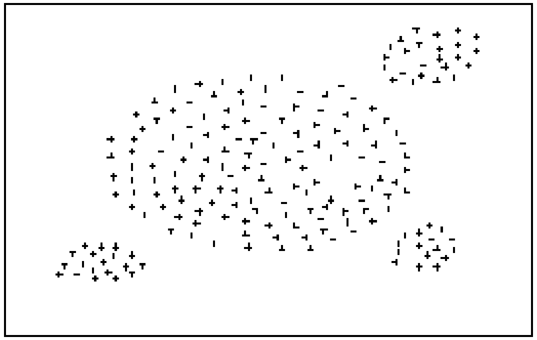
Density Function

The density at a point *x* in the data space is defined as the sum of influences of all data points x_i, i.e.

$$f_B^D(x) = \sum_{i=1}^{N} f_B^{x_i}(x)$$

Kernel Density Estimation





Hierarchical Methods

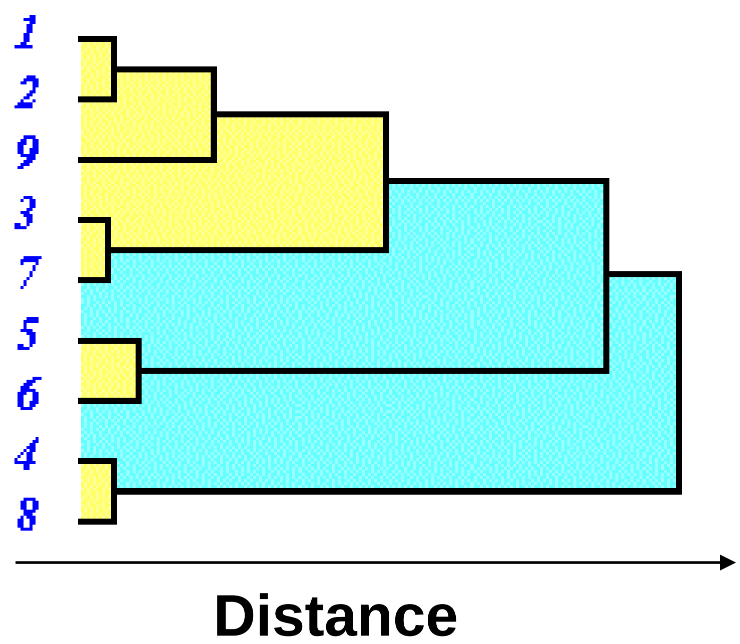
- Single Linkage
- Complete Linkage
- Average Linkage / Centroid Method
  (see also BIRCH)

Diversive: top-down

- Find the most inhomogenius cluster and split

Agglomerative: bottom-up

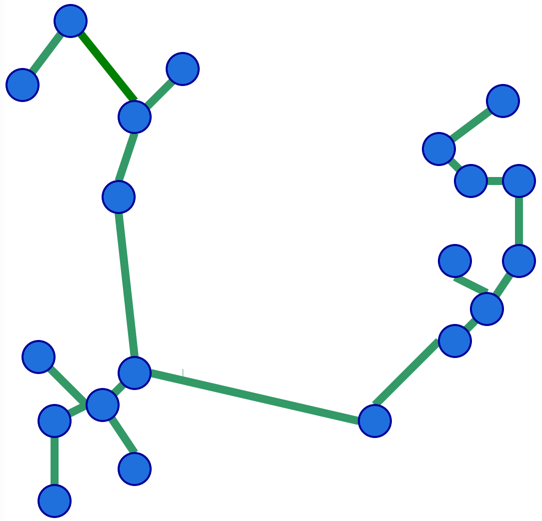
- Find the nearest pair of clusters and merge

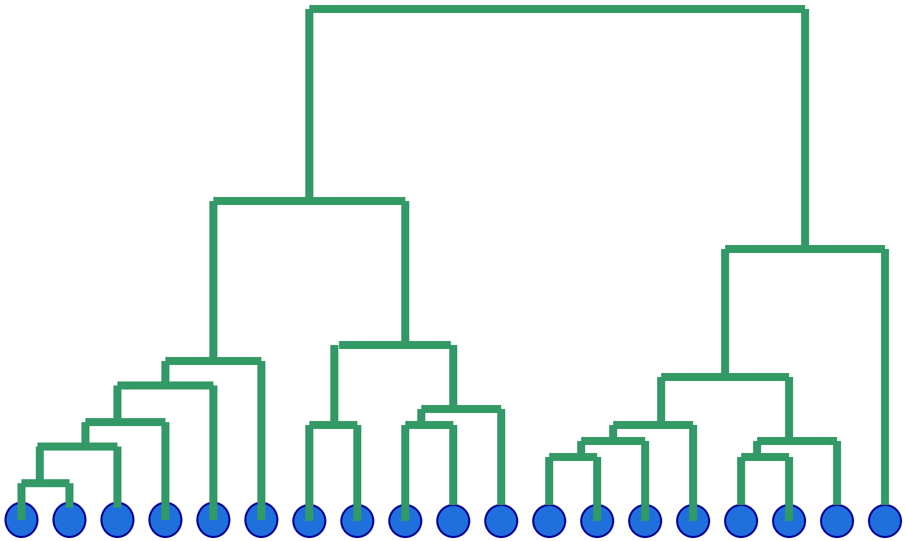


Single Linkage

- Distance between clusters (nodes):
  $$Dist(C_1, C_2) = \min_{p \in C_1, q \in C_2} \{dist(p, q)\}$$
- Merge Step: union the two subset of data points
- A single linkage hierarchy can be constructed using the minimal spanning tree

Example: Single Linkage

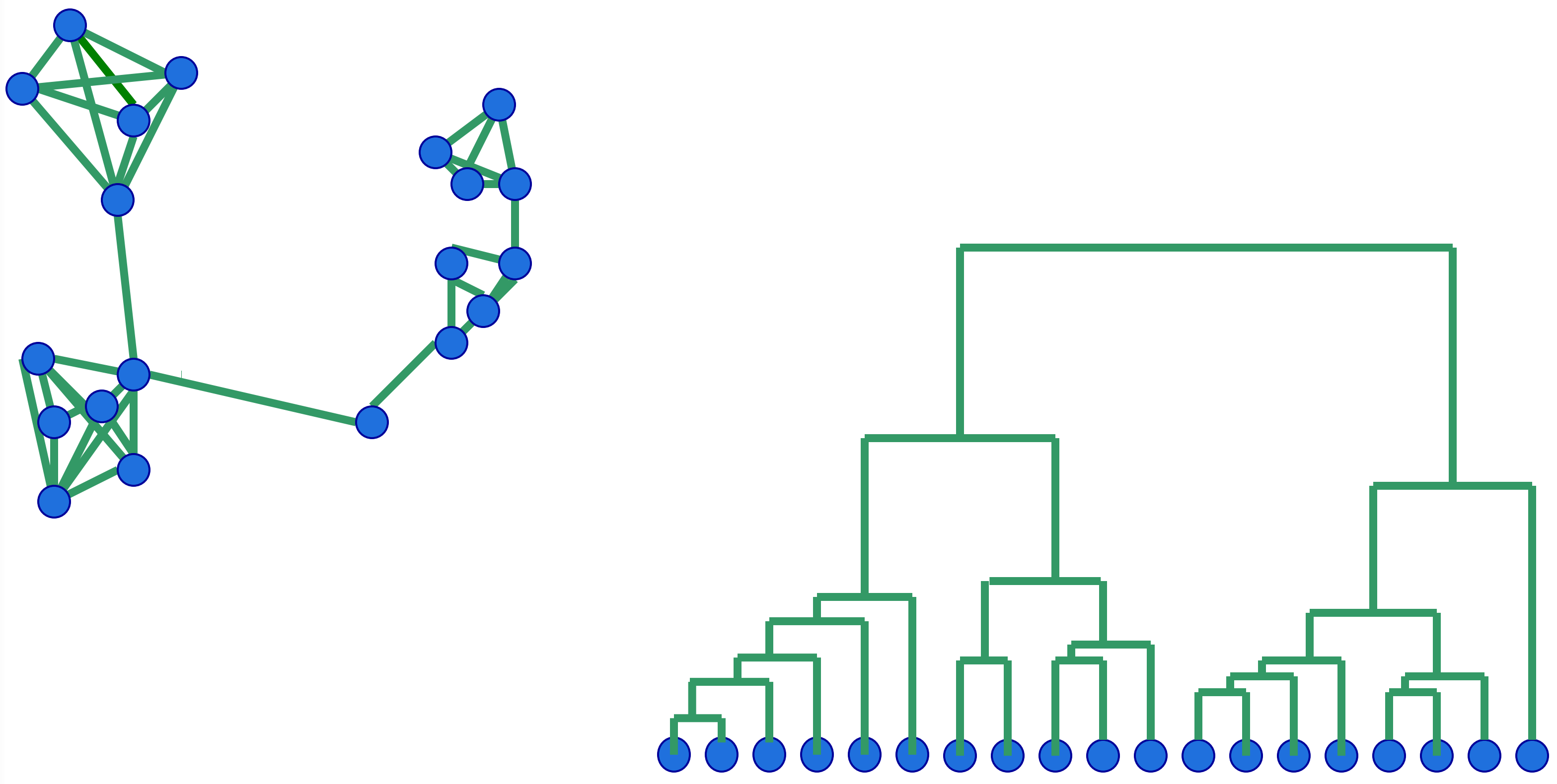




Complete Linkage

- Distance between clusters (nodes):
  $$Dist(C_1, C_2) = \max_{p \in C_1, q \in C_2} \{dist(p, q)\}$$
- Merge Step: union the two subset of data points
- Each cluster in a complete linkage hierarchy corresponds a complete subgraph

Example: Complete Linkage



Average Linkage / Centroid Method

- Distance between clusters (nodes):

$$Dist_{avg}(C_1, C_2) = \frac{1}{\#(C_1) \cdot \#(C_2)} \sum_{p \in C_1} \sum_{p \in C_2} dist(p, q)$$

$$Dist_{mean}(C_1, C_2) = dist[mean(C_1), mean(C_2)]$$

- Merge Step:
  – union the two subset of data points
  – construct the mean point of the two clusters

Scalability Problems

- Effectiveness degenerates
  – with dimensionality (d)
  – with noise level
- Efficiency degenerates
  – (at least) linearly with no of data points (N) and
  – exponentially with dimensionality (d)

Scaling Up Cluster Algorithms

- Sampling Techniques [EKX 95]
- Bounded Optimization Techniques [NH 94]
- Indexing Techniques [BK 98]
- Condensation Techniques [ZRL 96]
- Grid-based Techniques [SCZ 98, HK 98]

Indexing [BK 98]

- Cluster algorithms and their index structures
  - BIRCH: CF-Tree [ZRL 96]
  - DBSCAN: R*-Tree [Gut 84]
    X-Tree [BKK 96]
  - STING: Grid / Quadtree [WYM 97]
  - WaveCluster: Grid / Array [SCZ 98]
  - DENCLUE: B^+-Tree, Grid / Array [HK 98]

Methods for Improving the Effectiveness and Efficiency

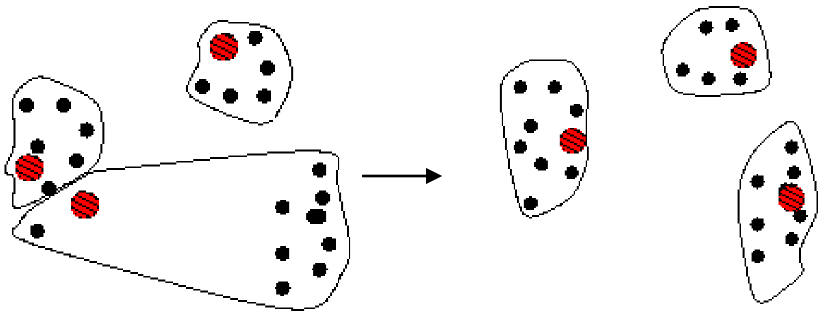
- Model- and Optimization-Based Approaches
- Density-Based Approaches
- Hybrid Approaches

Model- and Optimization-based Methods

- K-Means [Fuk 90]
- Expectation Maximization [Lau 95]
- CLARANS [NH 94]
- Foccused CLARANS [EKX 95]
- Self-Organizing Maps [KMS+ 91, Roj 96]
- Growing Networks [Fri 95]
- PROCLUS [APW+ 99]

CLARANS [NH 94]

- Medoid Method:
  - Medoids are special data points
  - All data points are assigned to the nearest medoid



- Optimization Criterion:

$$average\_distance(c) = \sum_{m_i \in M} \sum_{o \in cluster(m_i)} dist(o, m_i)$$

Bounded Optimization [NH 94]

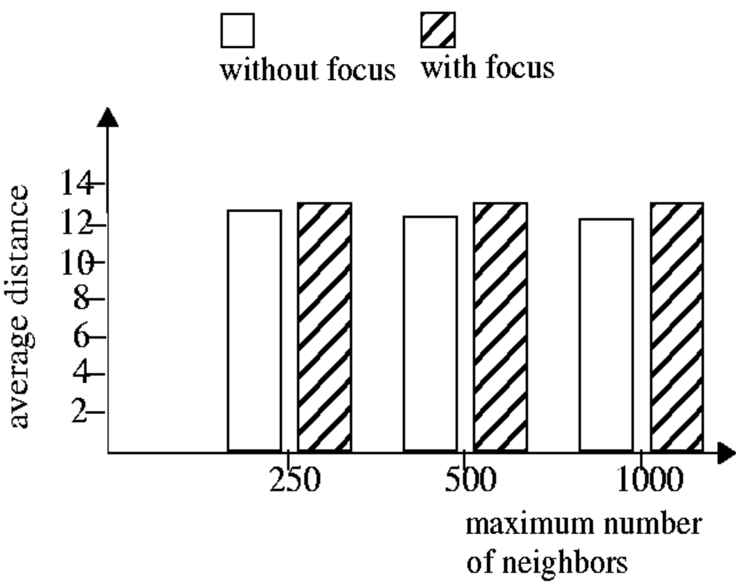
- CLARANS uses two bounds to restrict the optimization: *num_local, max_neighbor*
- Impact of the Parameters:
  - *num_local* → Number of iterations
  - *max_neighbors* → Number of tested neighbors per iteration

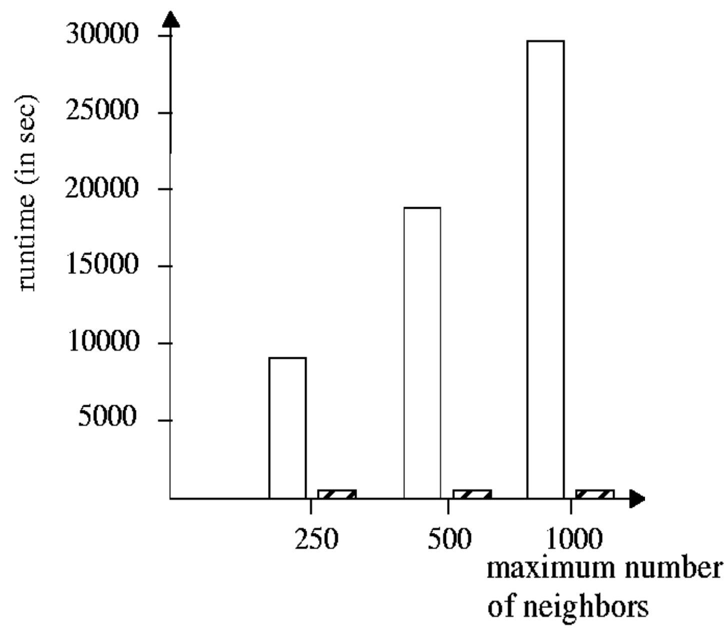
CLARANS

- Graph Interpretation:
  - Search process can be symbolized by a graph
  - Each node corresponds to a specific set of medoids
  - The change of one medoid corresponds to a jump to a neighboring node in the search graph
- Complexity Considerations:
  - The search graph has $\binom{N}{k}$ nodes and each node has *N*k* edges
  - The search is bound by a fixed number of jumps *(num_local)* in the search graph
  - Each jump is optimized by randomized search and costs *max_neighbor* scans over the data (to evaluate the cost function)

Sampling [EKX 95]

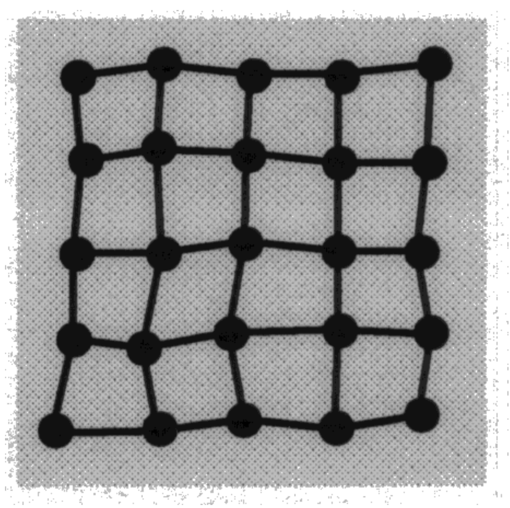
- R*-Tree Sampling
- Comparison of Effectiveness versus Efficiency (example CLARANS)

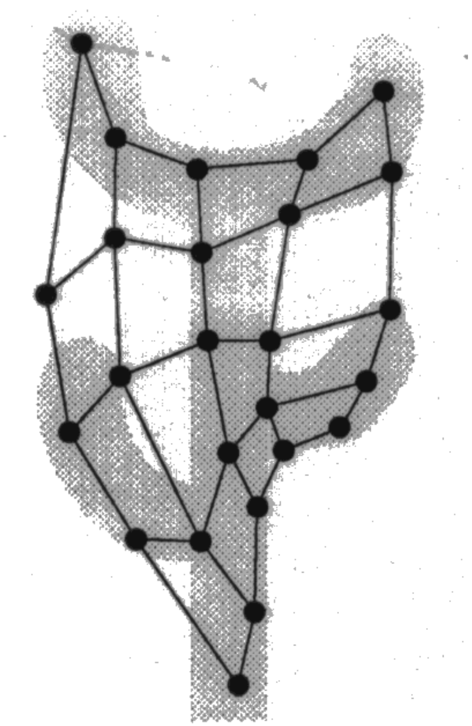




AI Methods

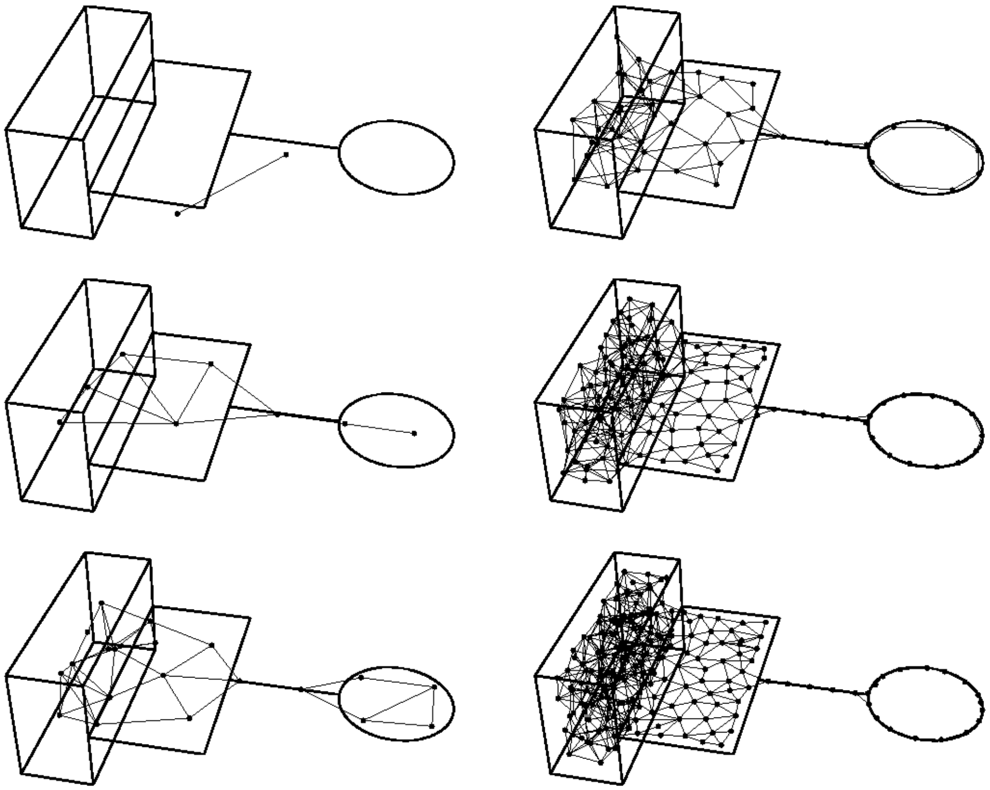
- Self-Organizing Maps [Roj 96, KMS 91]
  - – Fixed map topology (grid, line)





AI Methods

- Growing Networks [Fri 95]
  - Iterative insertion of nodes
  - Adaptive map topology

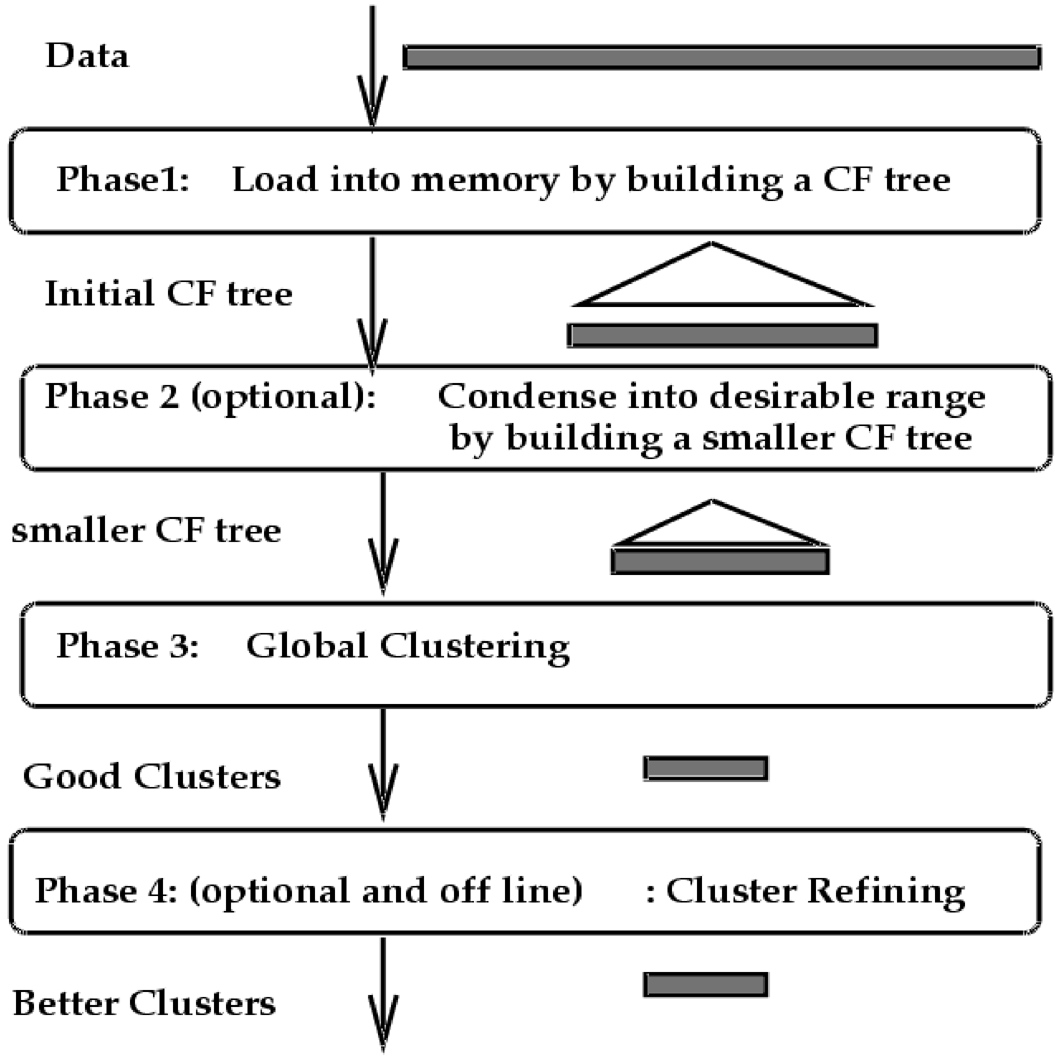


Density-based Methods

- Linkage -based Methods [Boc 74]
- Kernel-Density Estimation [Sil 86]
- BIRCH [ZRL 96]
- DBSCAN [EKS+ 96]
- DBCLASD [XEK+ 98]
- STING [WYM 97]
- Hierarchical Grid Clustering [Sch 96]
- WaveCluster [SCZ 98]
- DENCLUE [HK 98]
- OPTICS [ABKS 99]

BIRCH [ZRL 96]

Clustering in BIRCH



BIRCH

Basic Idea of the CF-Tree

- Condensation of the data $\{\vec{X}_i\}$ using CF-Vect $\mathbf{CF} = (N, \vec{LS}, SS)$

$$\vec{LS} = \sum_{i=1}^{N} \vec{X}_i \,,\; SS = \sum_{i=1}^{N} \vec{X}_i^{\,2}$$

- CF-tree uses sum of CF-vectors to build higher levels of the CF-tree

BIRCH

Insertion algorithm for a point x:

(1) Find the closest leaf b

(2) If x fits in b, insert x in b;

otherwise split b

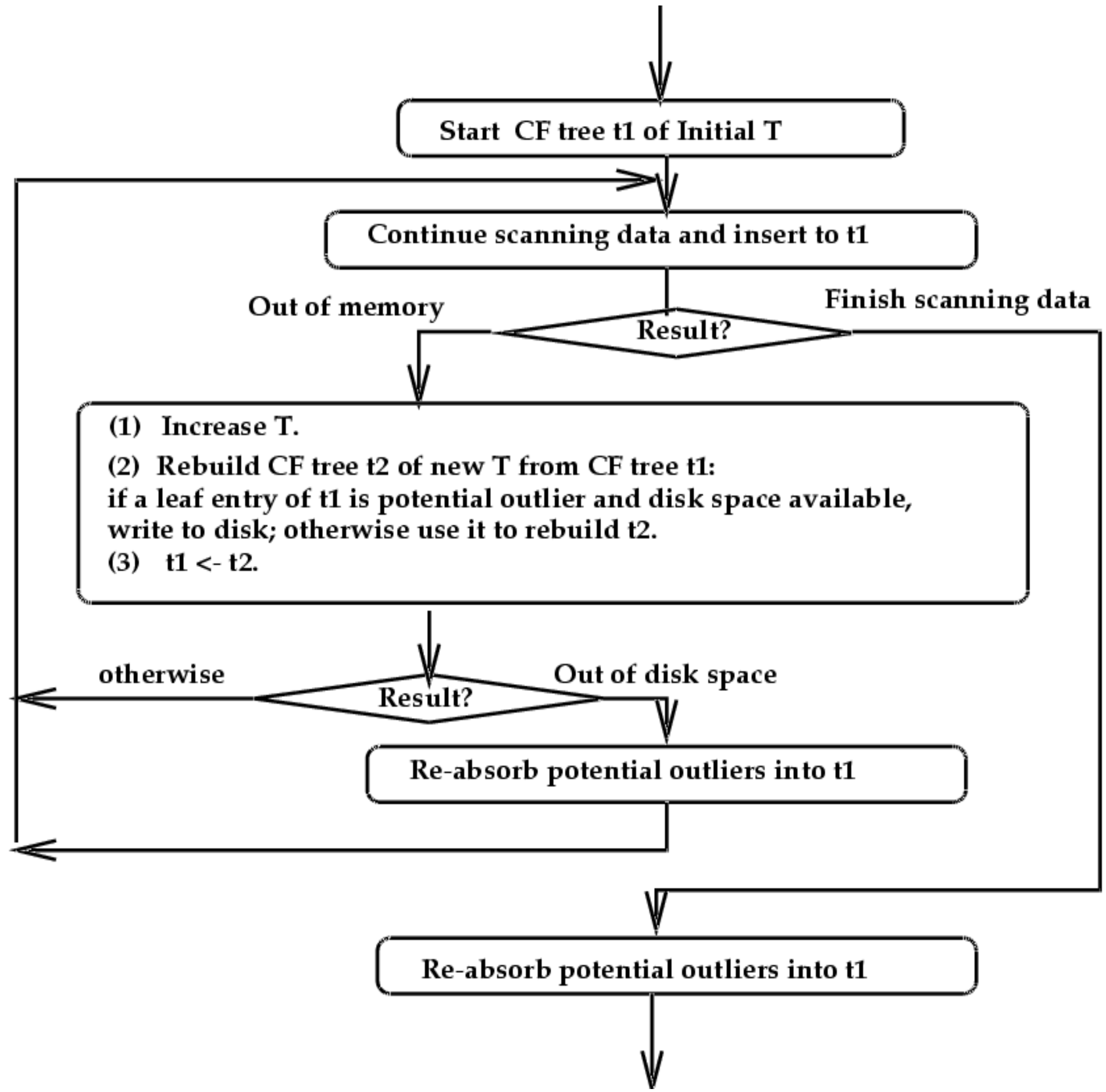
(3) Modify the path for b

(4) If tree is to large, condense the tree

by merging the closest leaves

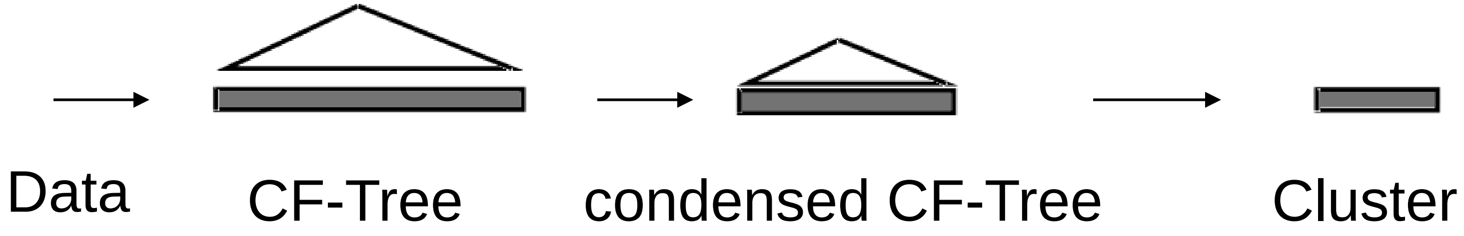
BIRCH

CF-Tree Construction



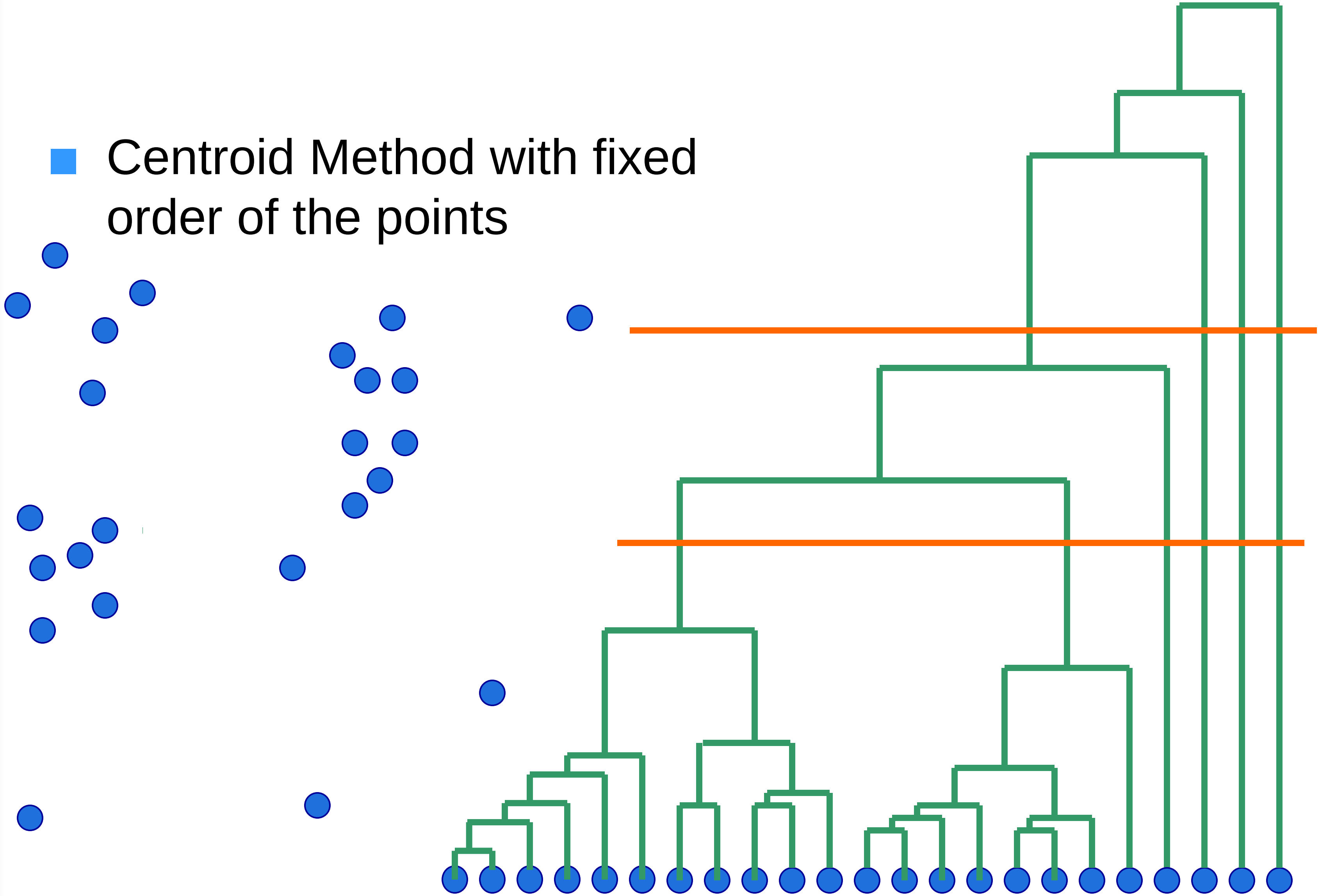
Condensing Data

- BIRCH [ZRL 96]:
  - Phase 1-2 produces a condensed representation of the data (CF-tree)
  - Phase 3-4 applies a separate cluster algorithm to the leafs of the CF-tree
- Condensing data is crucial for efficiency



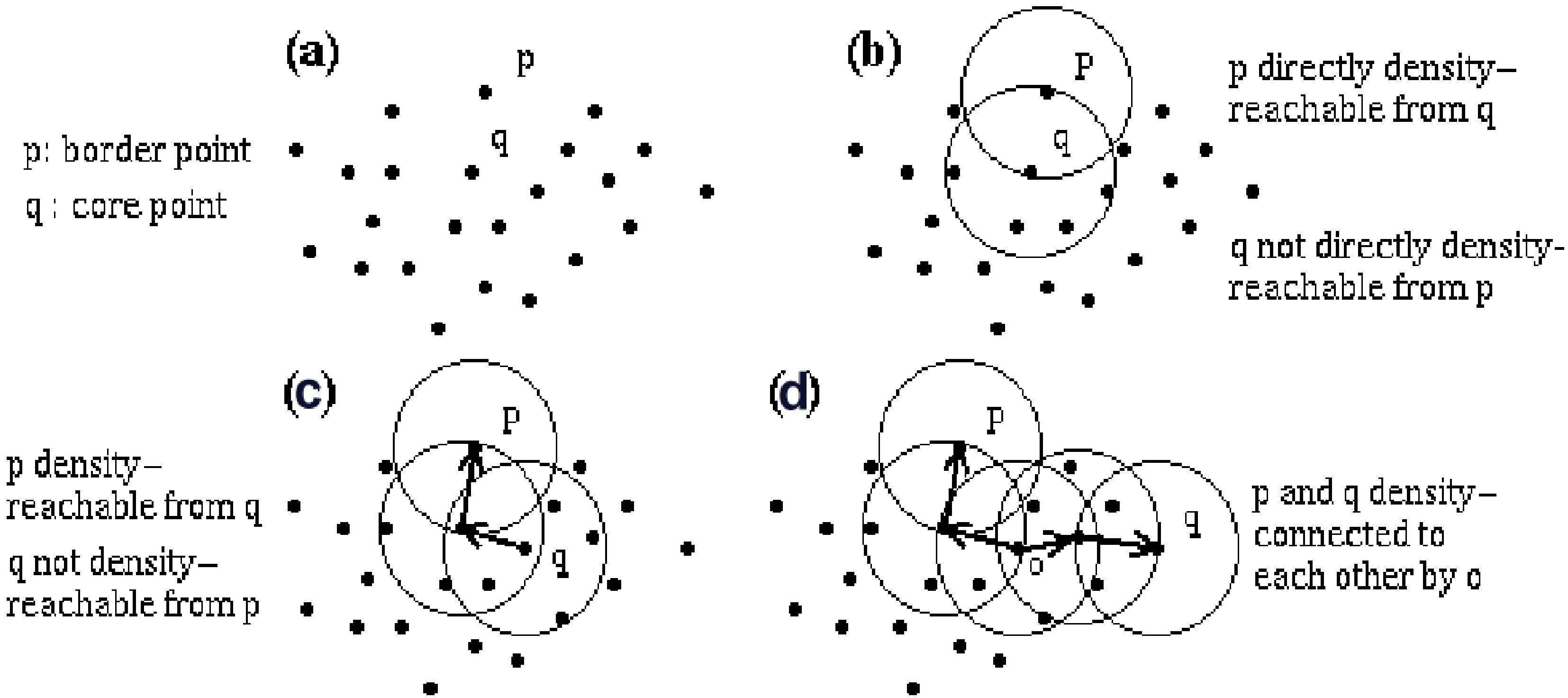
Problems of BIRCH

- Centroid Method with fixed order of the points



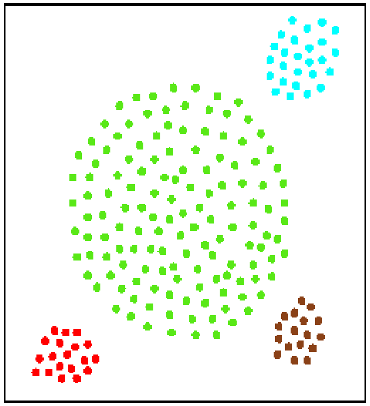
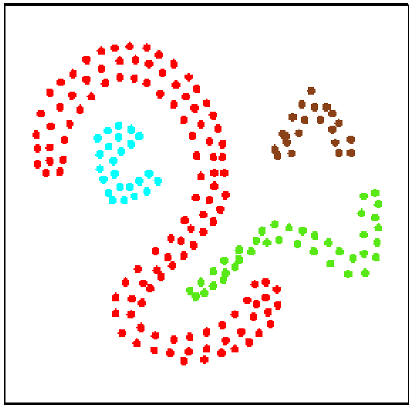
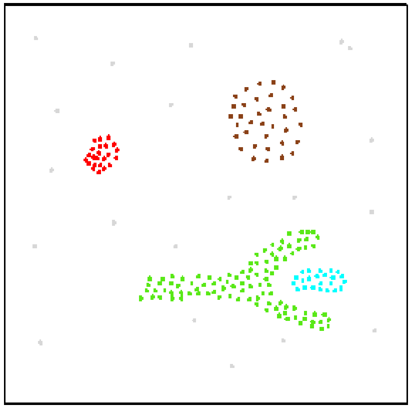
DBSCAN [EKS+ 96]

- Clusters are defined as Density-Connected Sets (wrt. MinPts, ε)



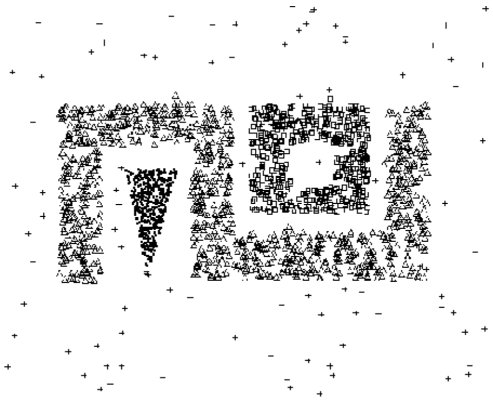
DBSCAN

- For each point, DBSCAN determines the ε-environment and checks, whether it contains more than MinPts data points
- DBSCAN uses index structures for determining the ε-environment
- Arbitrary shape clusters found by DBSCAN

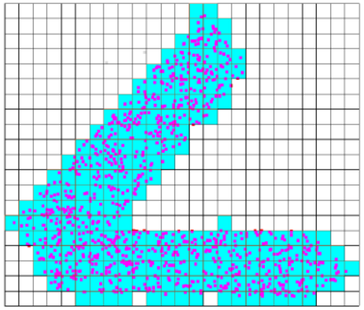




DBCLASD [XEK+ 98]

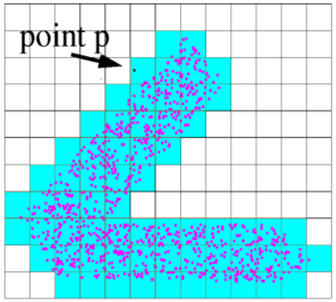
- Distribution-based method
- Assumes arbitrary-shape clusters of uniform distribution
- Requires no parameters
- Provides grid-based approximation of clusters



Before the insertion of point p



After the insertion of point p



DBCLASD

- Definition of a cluster *C* based on the distribution of the NN-distance *(NNDistSet)*:

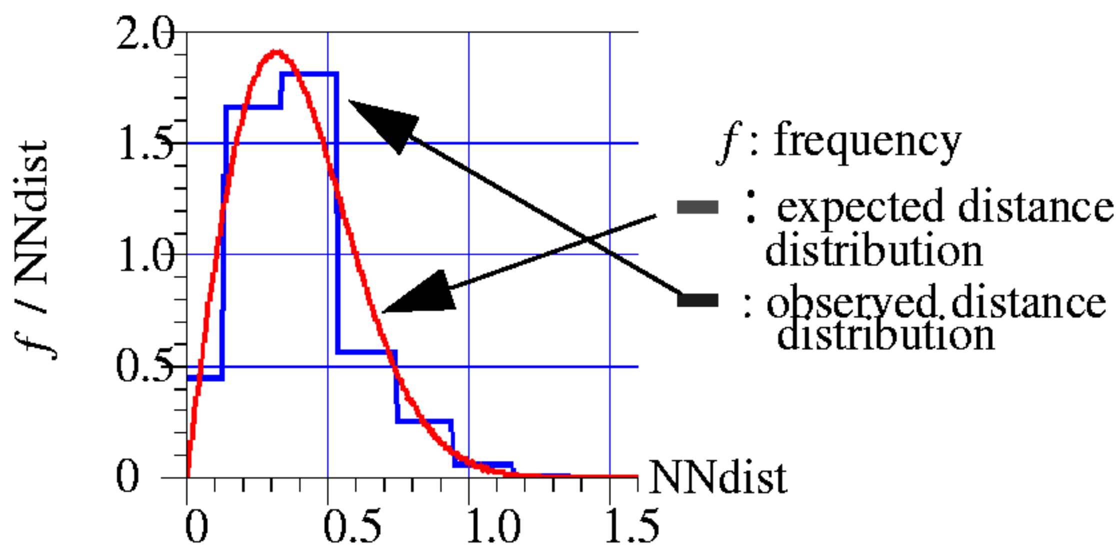
(1) a

(2) oring

).

(3) of the

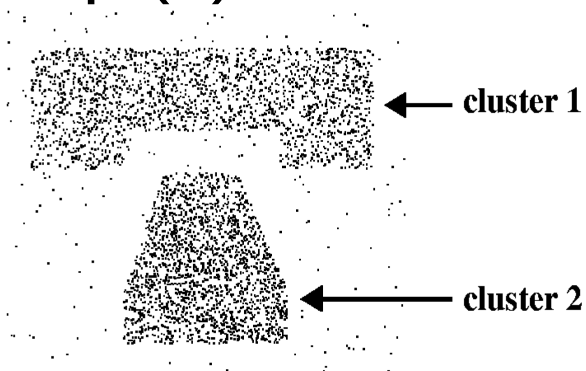
ecting

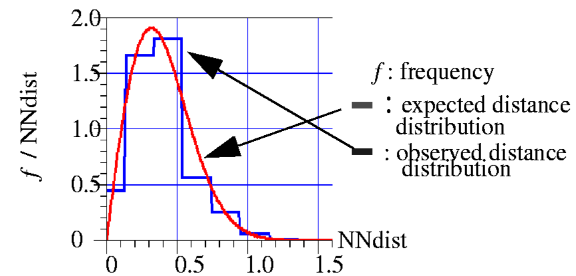


The expected and the observed distance distributions for cluster 1

DBCLASD

- Step (1) uses the concept of the χ^2-test





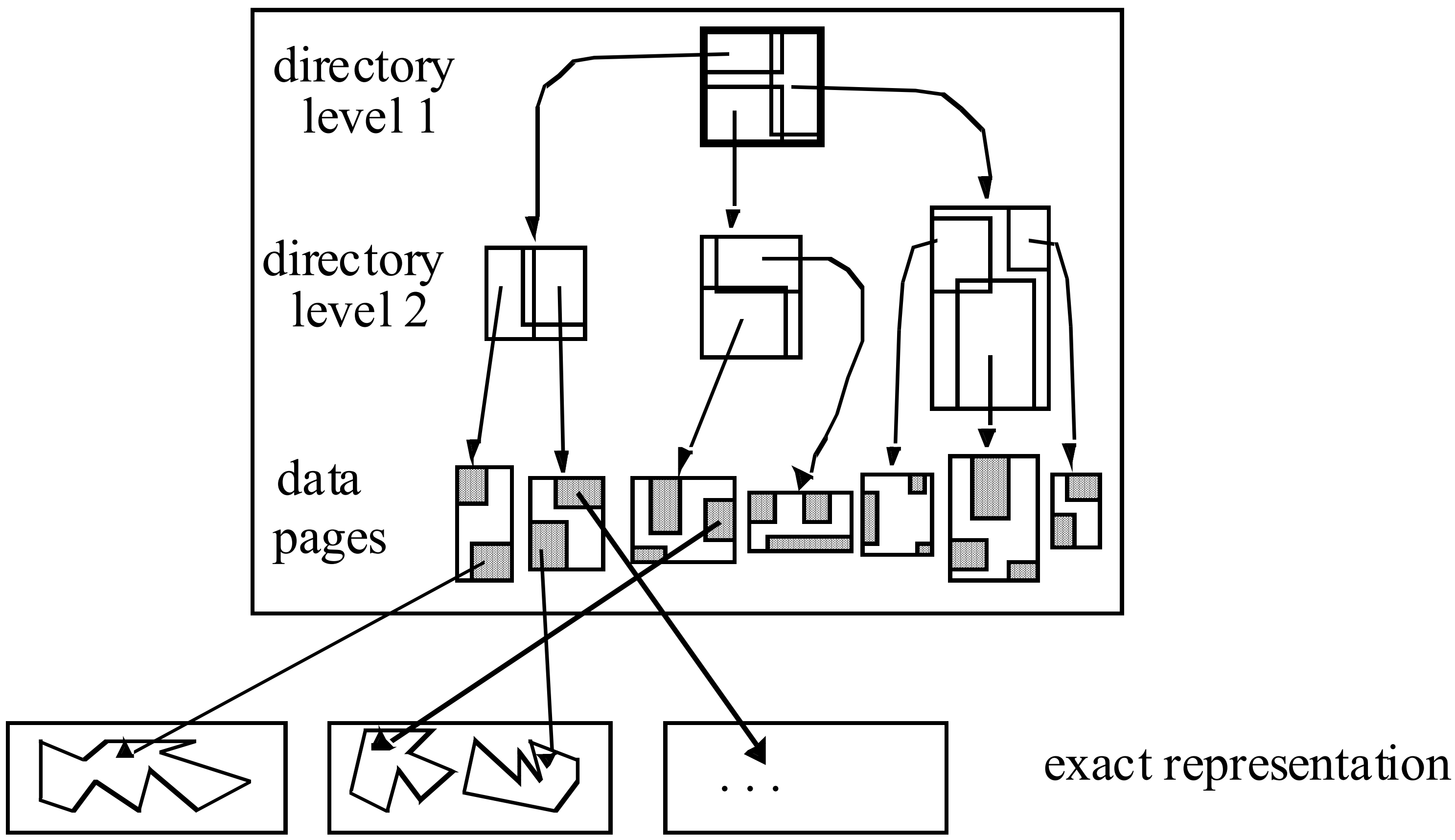
The expected and the observed distance distributions for cluster 1

- Incremental augmentation of clusters by neighboring points (order-depended)
  - unsuccessful candidates are tried again later
  - points already assigned to some cluster may switch to another cluster

DBSCAN / DBCLASD

- DBSCAN and DBCLASD use index structures to speed-up the ε-environment or nearest-neighbor search
- the index structures used are mainly the R-tree and variants

R-Tree: [Gut 84]
The Concept of Overlapping Regions



Variants of the R-Tree

Low-dimensional

- R^+-Tree [SRF 87]
- R^*-Tree [BKSS 90]
- Hilbert R-Tree [KF94]

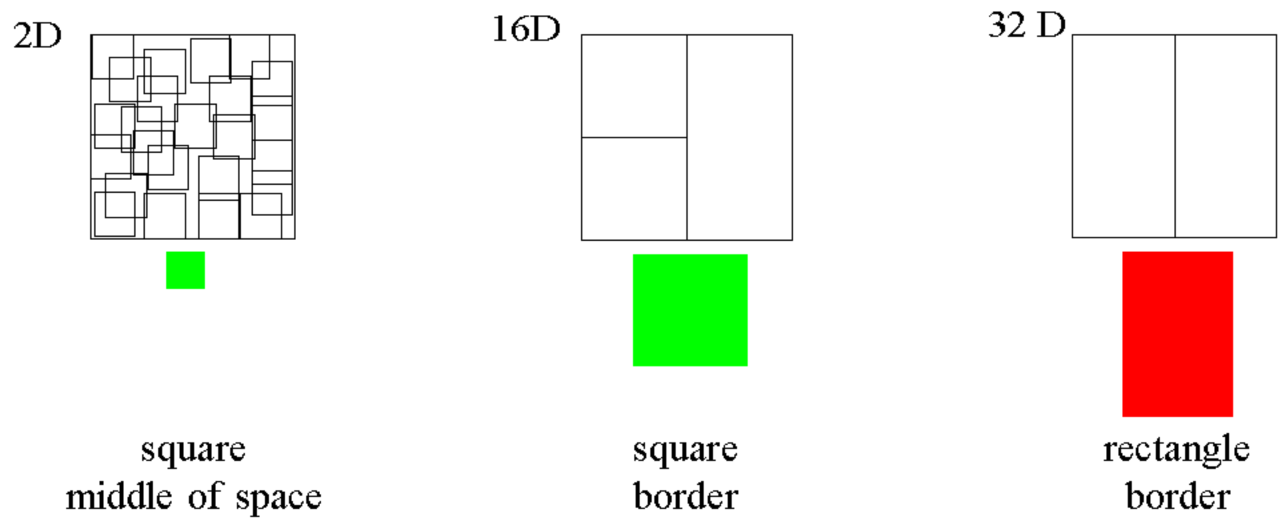
High-dimensional

- TV-Tree [LJF 94]
- X-Tree [BKK 96]
- SS-Tree [WJ 96]
- SR-Tree [KS 97]

Effects of High Dimensionality

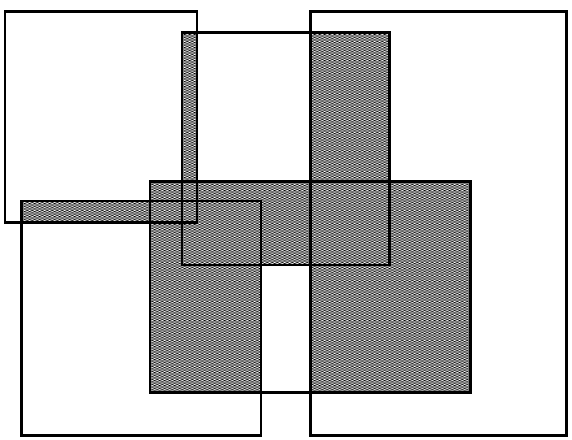
Location and Shape of Data Pages

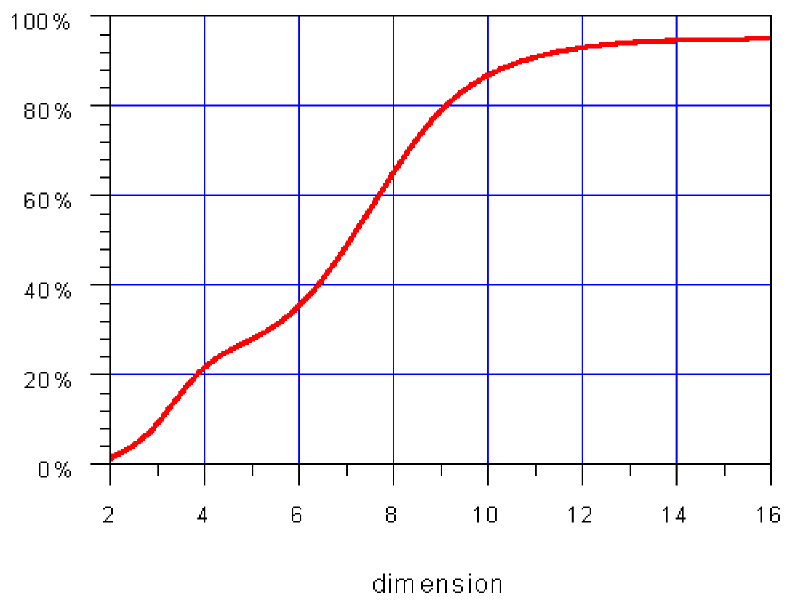
- Data pages have large extensions
- Most data pages touch the surface of the data space on most sides



The X-Tree [BKK 96]
(eXtended-Node Tree)

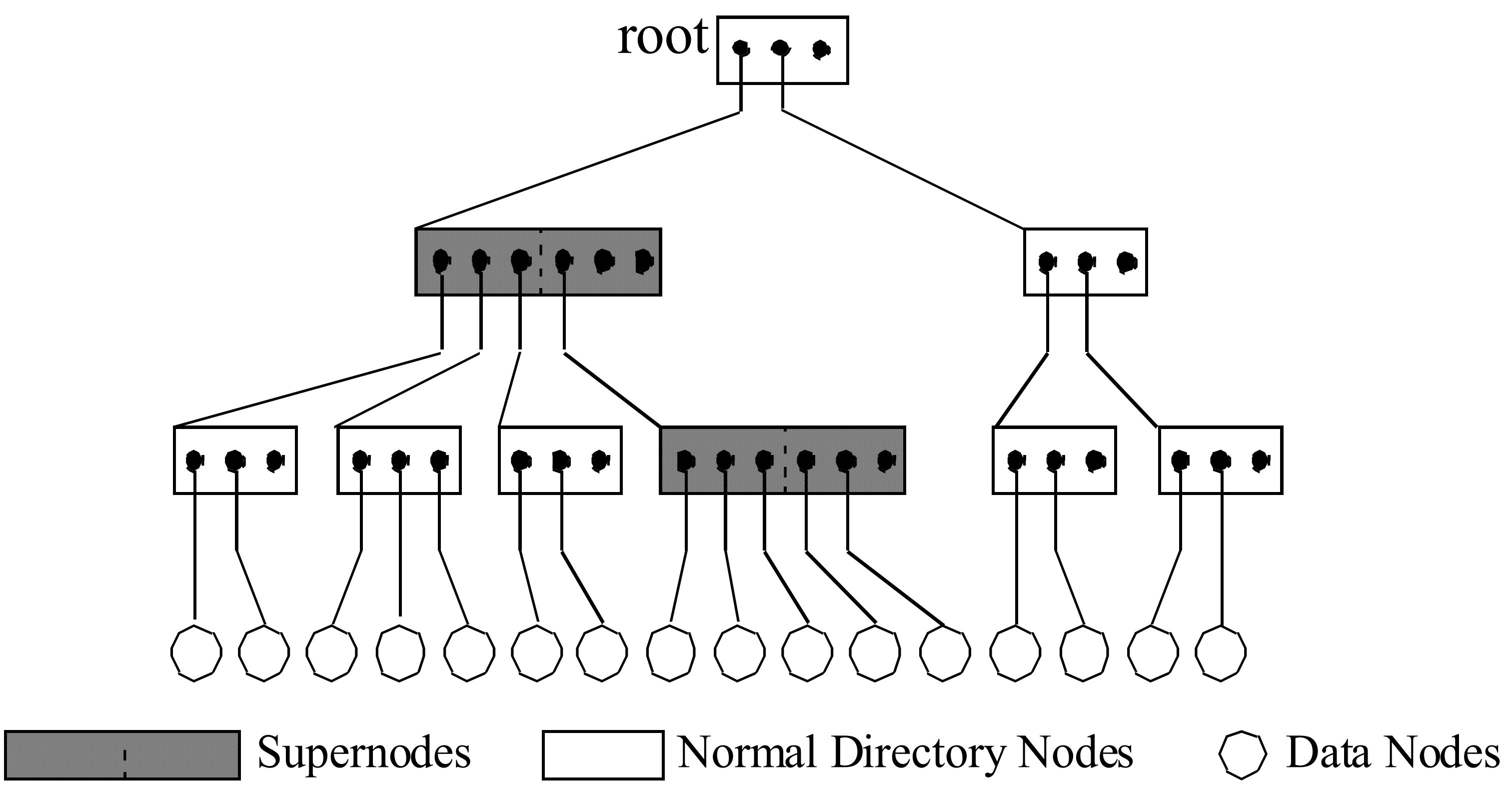
- Motivation:
  Performance of the R-Tree degenerates in high dimensions
- Reason: overlap in the directory



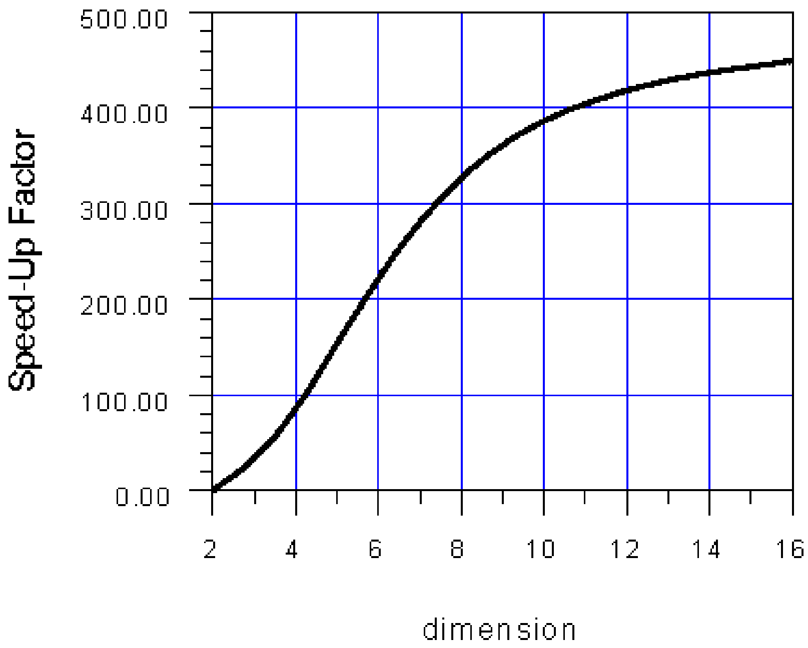


The X-Tree

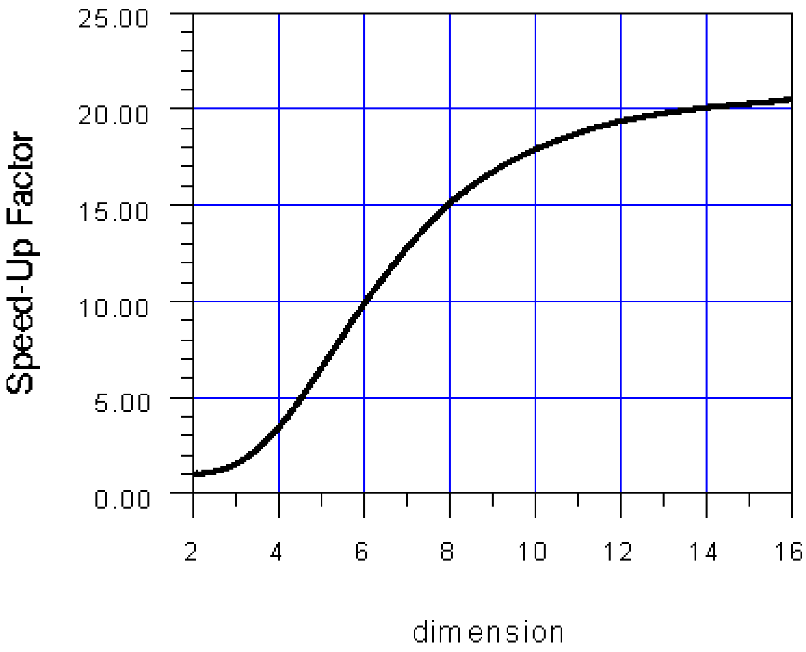
- X-tree avoids overlap in the directory by using
  - an overlap-free split
  - the concept of supernodes



Speed-Up of X-Tree over the R*-Tree



Point Query



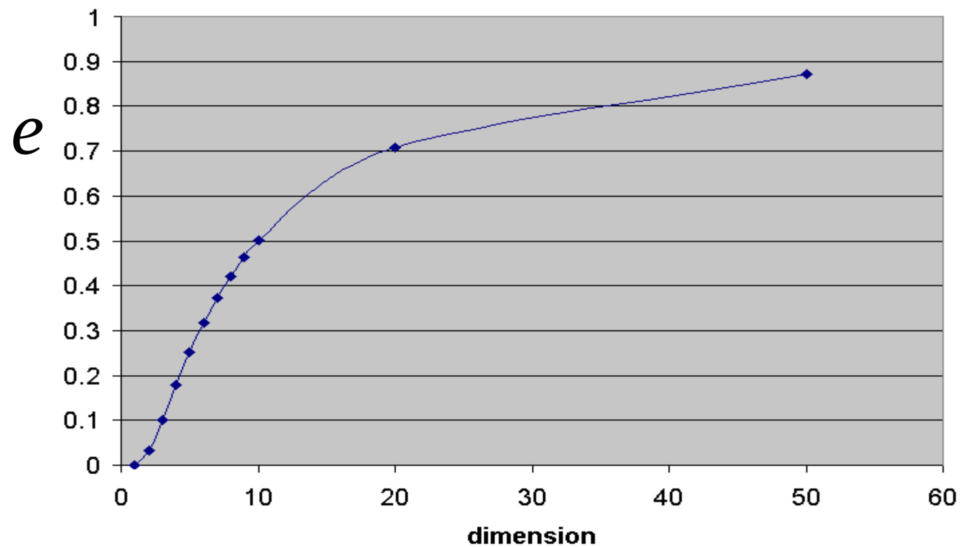
10 NN Query

Effects of High Dimensionality

Selectivity of Range Queries

- The selectivity depends on the volume of the query

$$e = \sqrt[d]{Vol_{cube}}$$



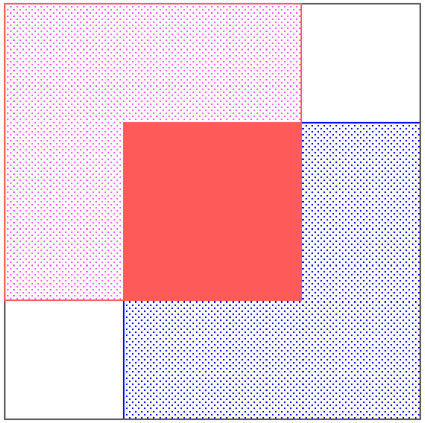
selectivity = 0.1 %

$\Rightarrow$ **no fixed ε-environment (as in DBSCAN)**

Effects of High Dimensionality

Selectivity of Range Queries

- In high-dimensional data spaces, there exists a region in the data space which is affected by ANY range query (assuming uniformly distributed data)

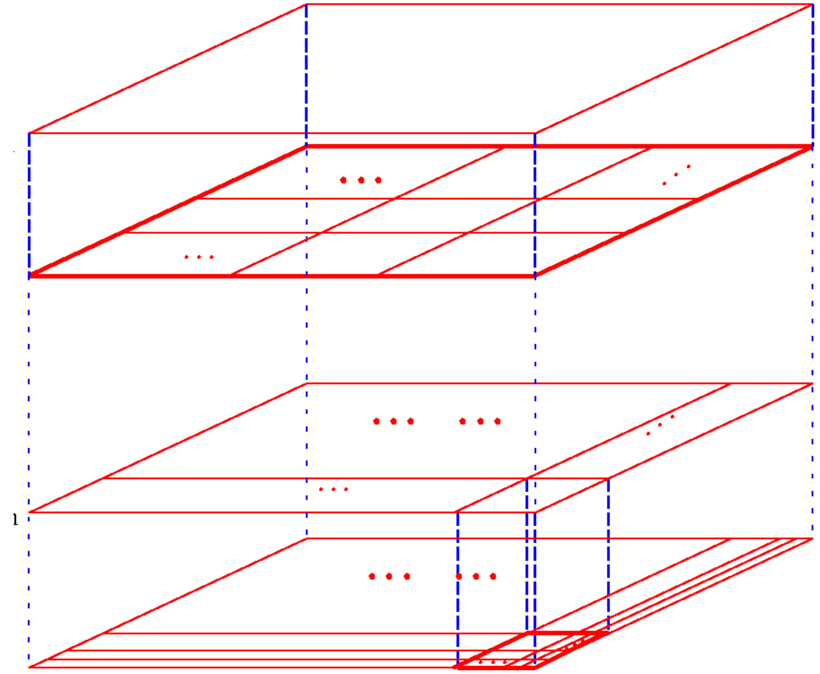


⇒ **difficult to build an efficient index structure**

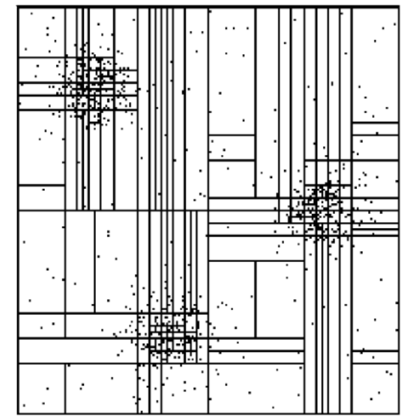
⇒ **no efficient support of range queries (as in DBSCAN)**

STING [WYM 97]

- Uses a quadtree-like structure for condensing the data into grid cells
- The nodes of the quadtree contain statistical information about the data in the corresponding cells
- STING determines clusters as the density-connected components of the grid
- STING approximates the clusters found by DBSCAN

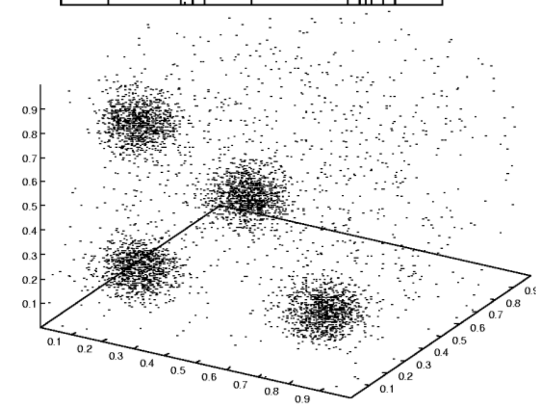


Hierarchical Grid Clustering [Sch 96]

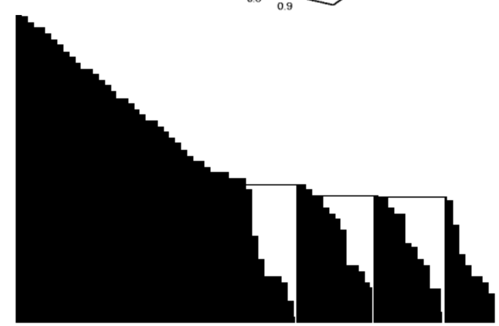


- Organize the data space as a grid-file
- Sort the blocks by their density

$$DB = \frac{p_B}{V_B} \longrightarrow \langle B_{1'}, B_{2'}, \ldots B_{b'} \rangle$$



- Scan the blocks iteratively and merge blocks, which are adjacent over a (d-1)-dim. hyperplane.
- The order of the merges forms a hierarchy



WaveCluster [SCZ 98]

- Clustering from a signal processing perspective using wavelets

Input: Multidimensional data objects' feature vectors
Output: clustered objects

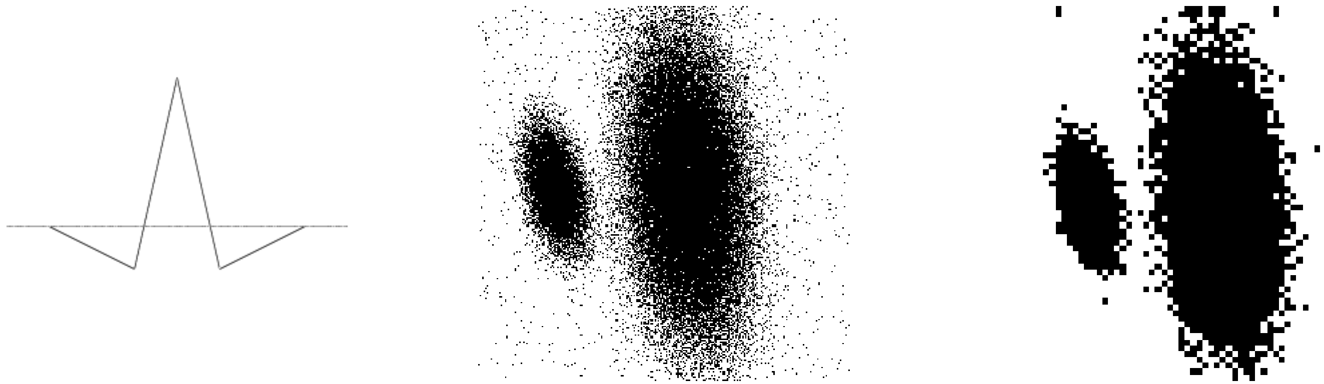
1. Quantize feature space, then assign objects to the units.
2. Apply wavelet transform on the feature space.
3. Find the connected components (clusters) in the subbands of transformed feature space, at different levels.
4. Assign label to the units.
5. Make the lookup table.
6. Map the objects to the clusters.

WaveCluster

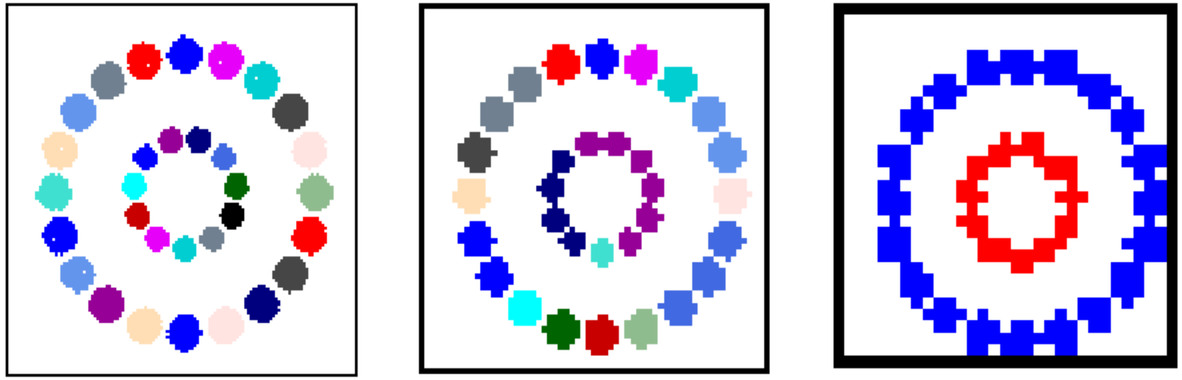
- Grid Approach
  - Partition the data space by a grid $\rightarrow$ reduce the number of data objects by making a small error
  - Apply the wavelet-transformation to the reduced feature space
  - Find the connected components as clusters
- Compression of the grid is crucial for the efficiency
- Does not work in high dimensional space!

WaveCluster

- Signal transformation using wavelets

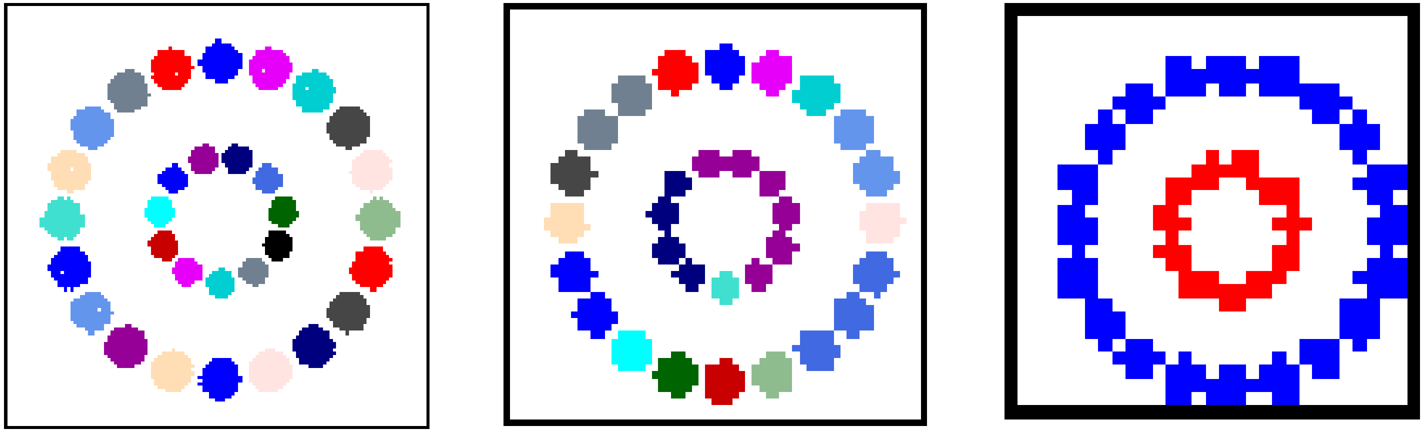


- Arbitrary shape clusters found by WaveCluster

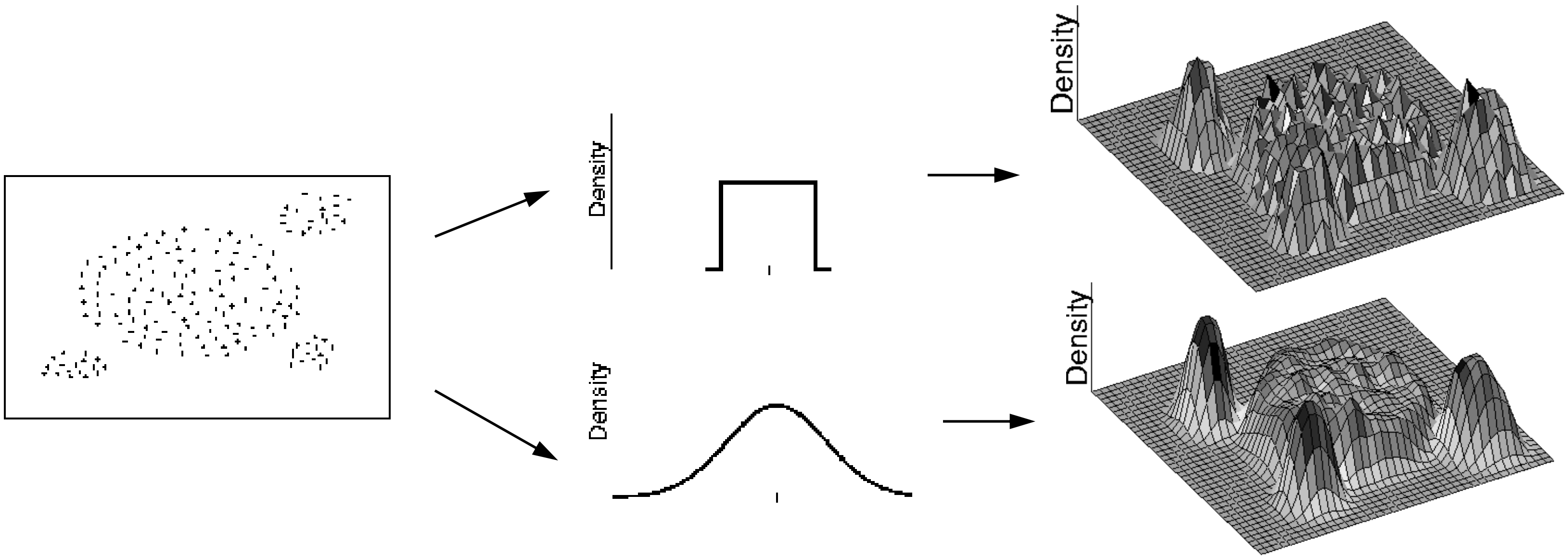


Hierarchical Variant of WaveCluster [SCZ 98]

- WaveCluster can be used to perform multiresolution clustering
- Using coarser grids, cluster start to merge



DENCLUE



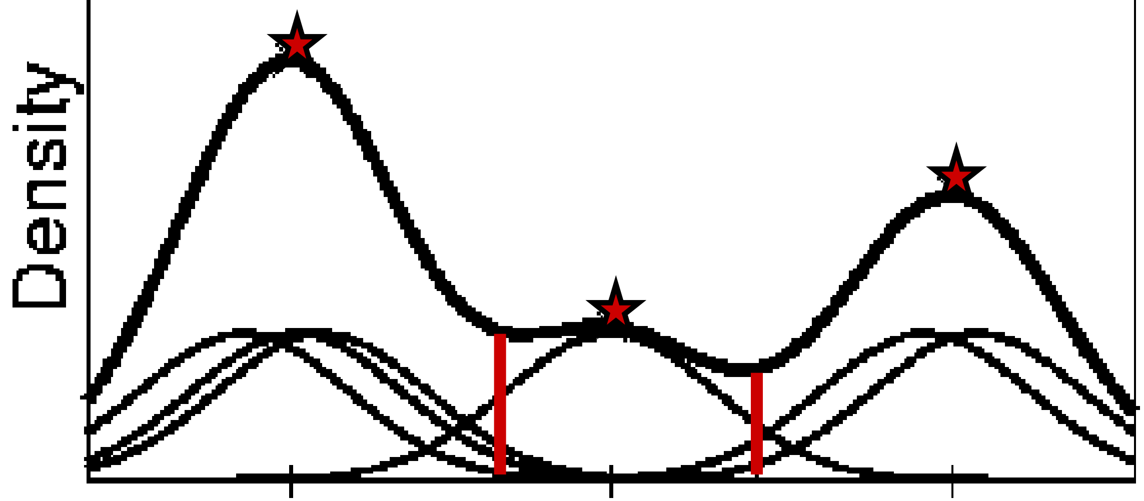
Data Set ***Influence Function*** ***Density Function***

Influence Function: Influence of a data point in its neighborhood

Density Function: Sum of the influences of all data points

DENCLUE

Definitions of Clusters



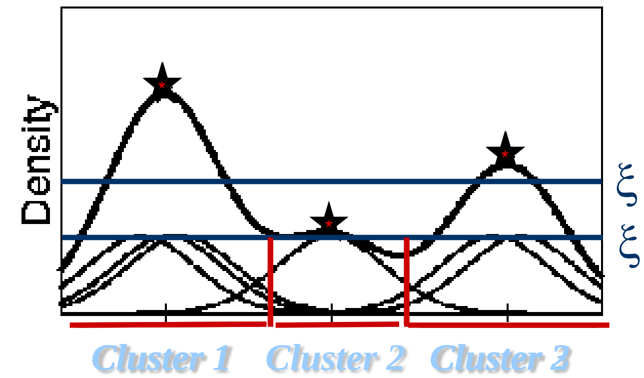
Density Attractor/Density-Attracted Points (★)

- local maximum of the density function
- density-attracted points are determined by a gradient-based hill-climbing method

DENCLUE

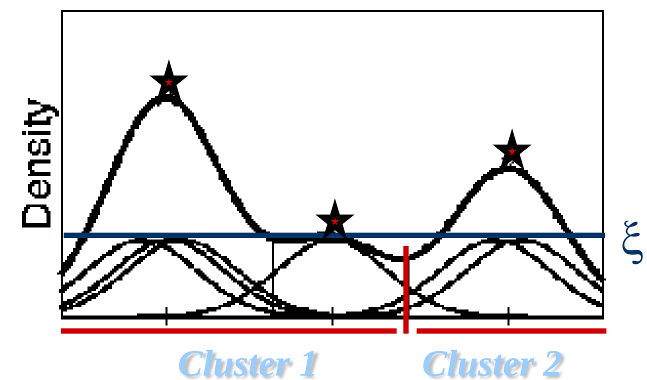
Center-Defined Cluster

A center-defined cluster with density-attractor x* ($f_B^D(x^*) > \xi$) is the subset of the database which is density-attracted by x*.



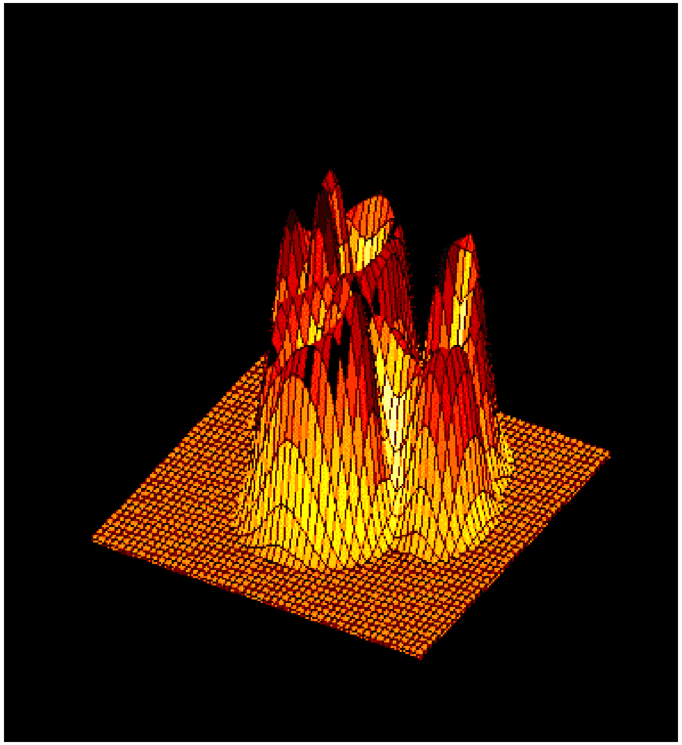
Multi-Center-Defined Cluster

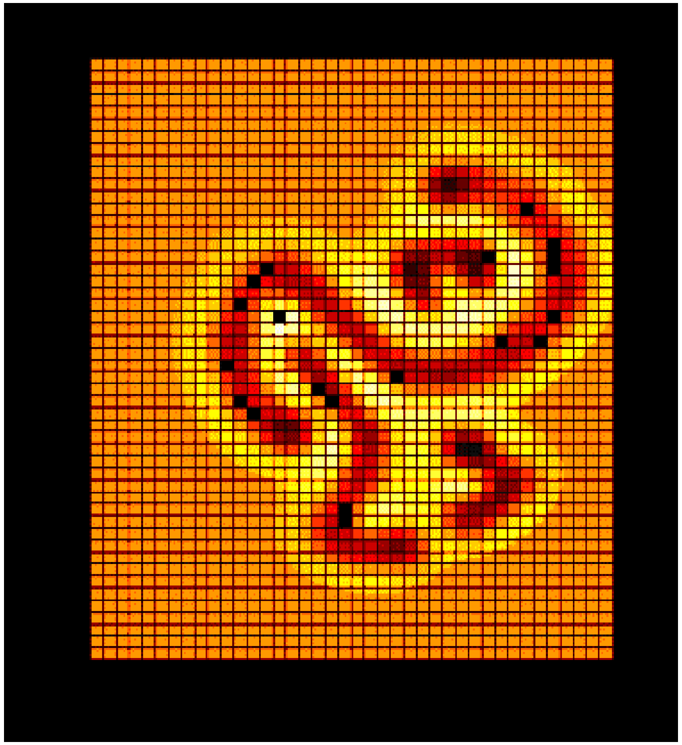
A multi-center-defined cluster consists of a set of center-defined clusters which are linked by a path with significance ξ.



DENCLUE

Impact of different Significance Levels (ξ)

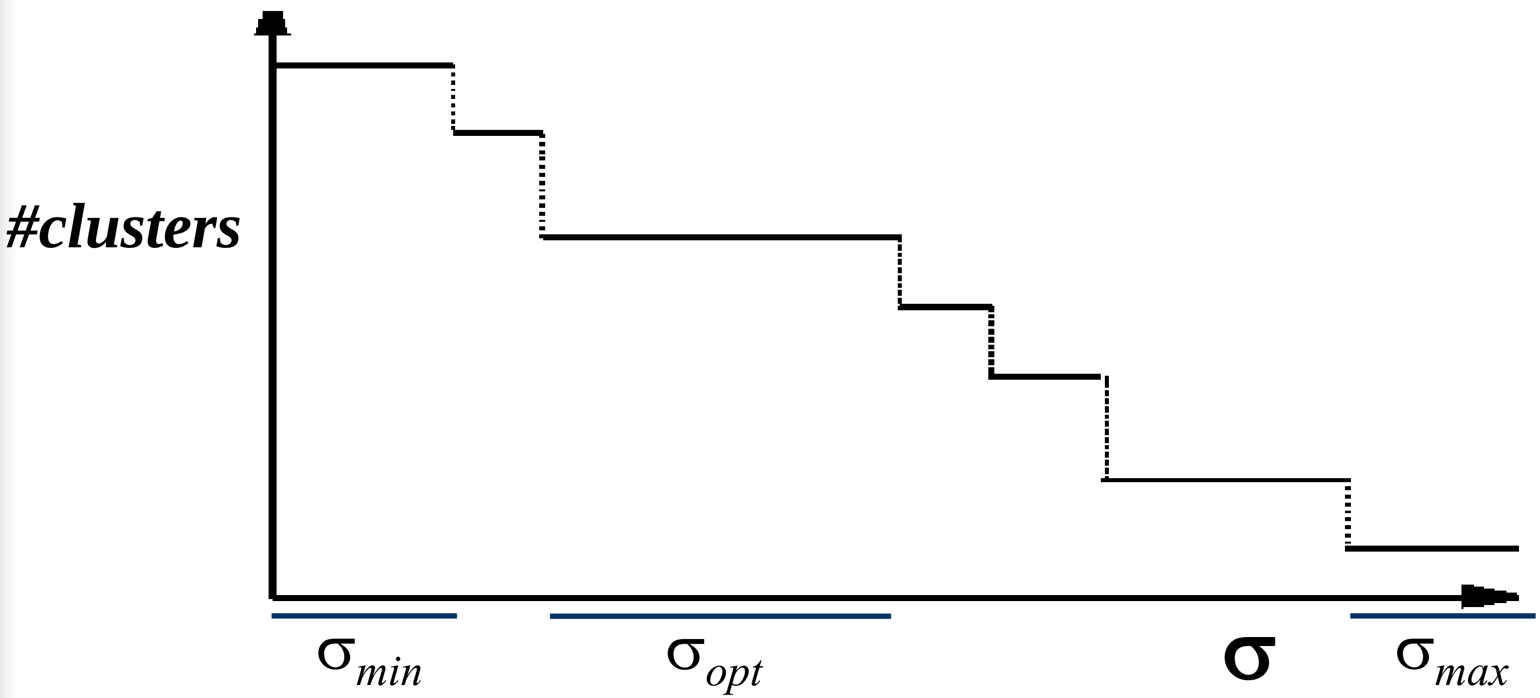




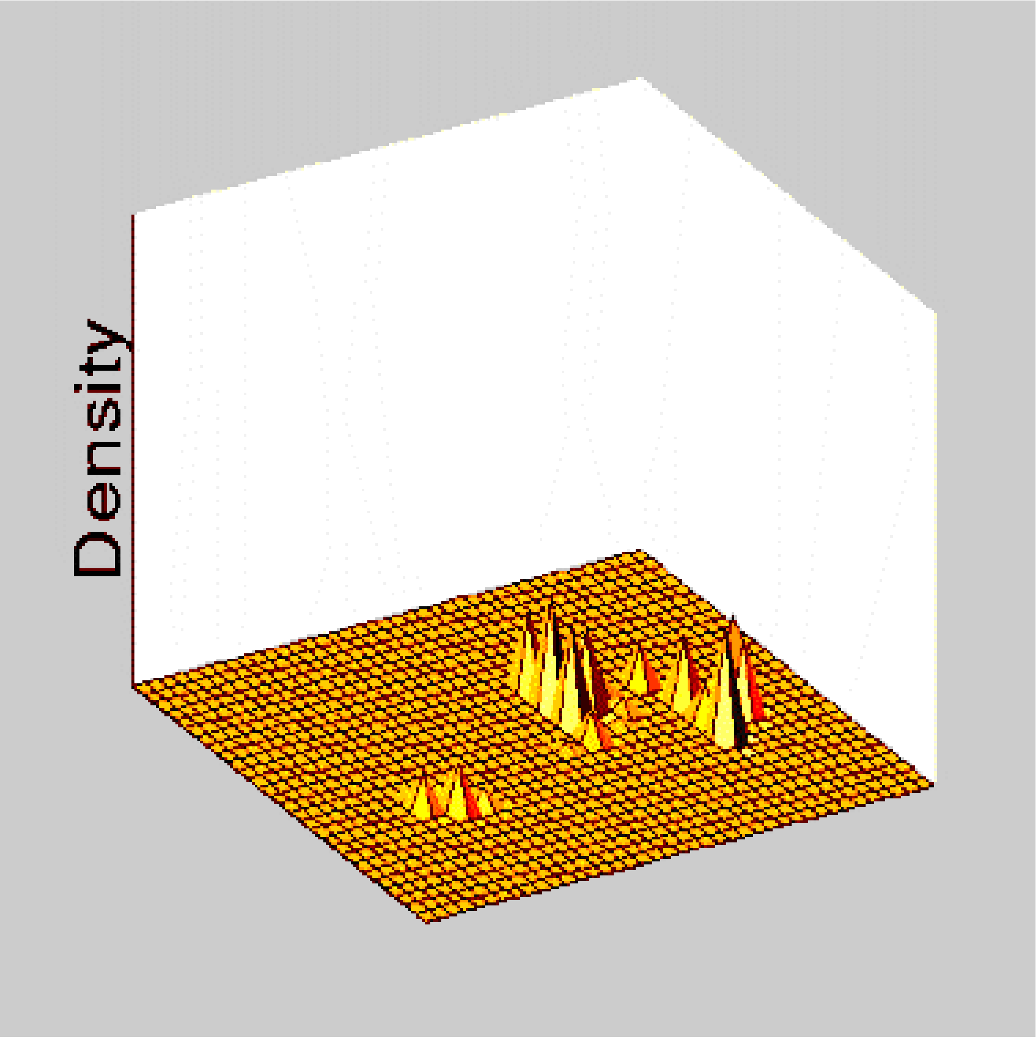
DENCLUE

Choice of the Smoothness Level (σ)

Choose σ such that *number of density attractors* is constant for a long interval of σ!



Building Hierarchies (σ)



DENCLUE

Noise Invariance

Assumption: Noise is uniformly distributed in the data space

Lemma:

The density-attractors do not change when increasing the noise level.

Idea of the Proof:

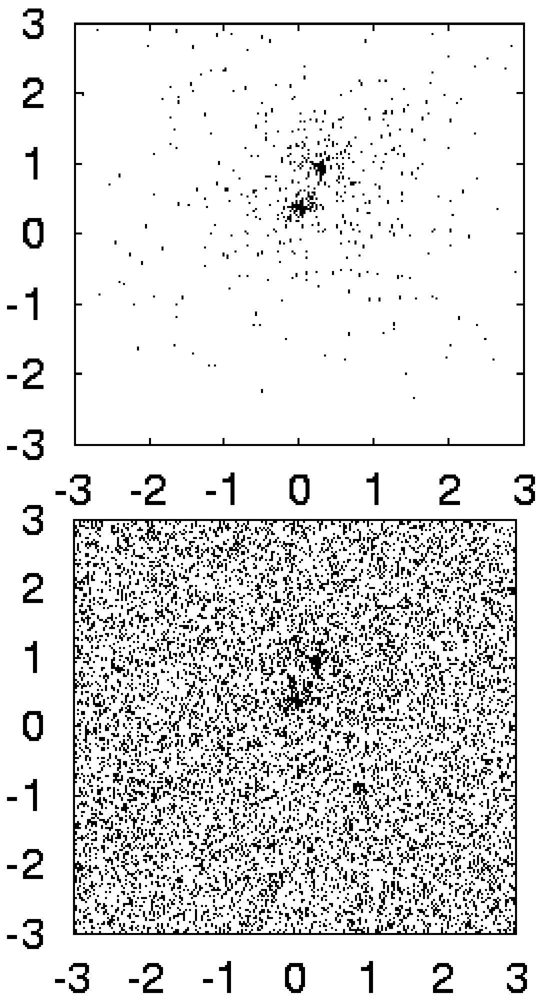
- partition density function into signal and noise

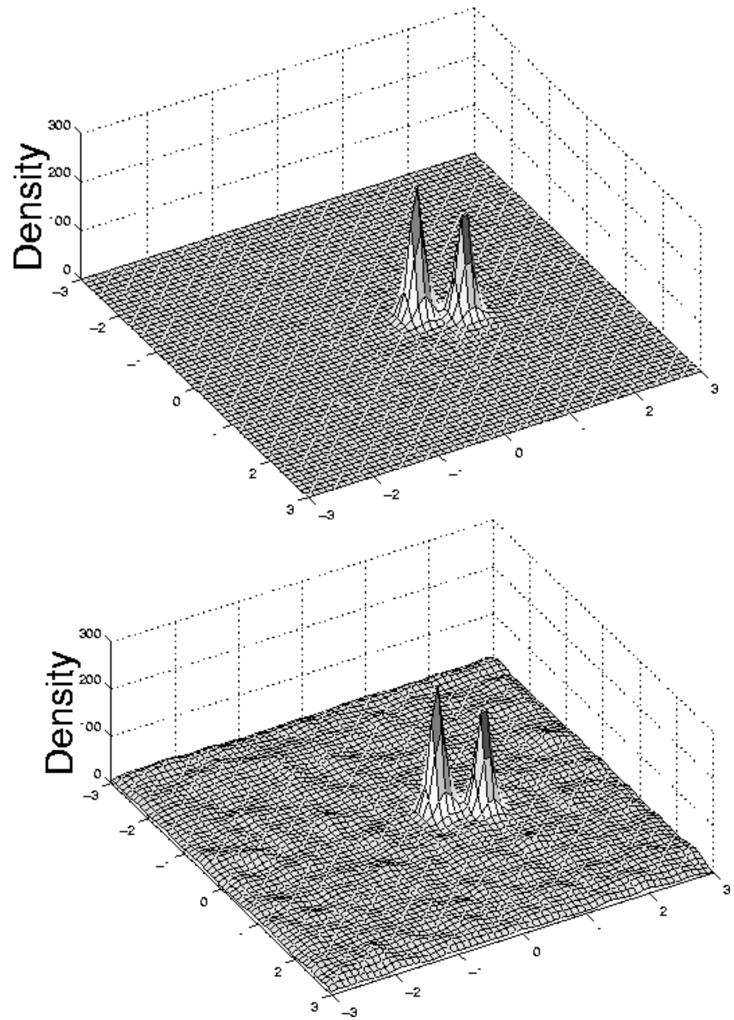
$$f^D(x) = f^{D_C}(x) + f^N(x)$$

- density function of noise approximates a constant $(f^N(x) \approx const.)$

DENCLUE

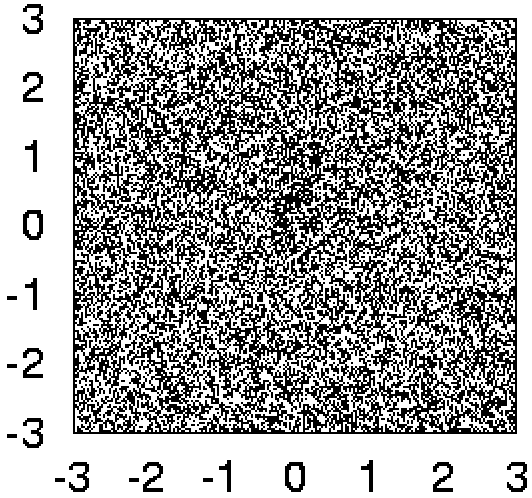
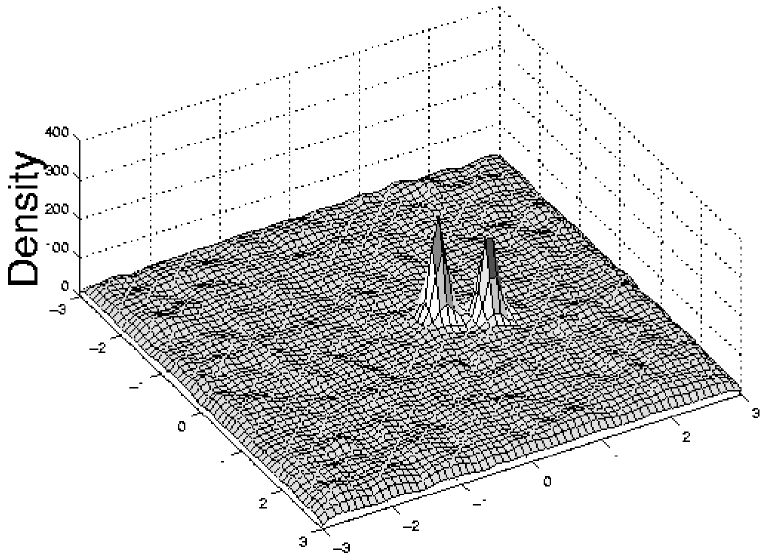
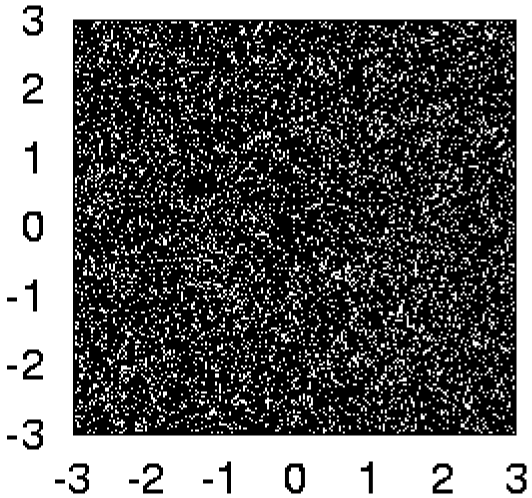
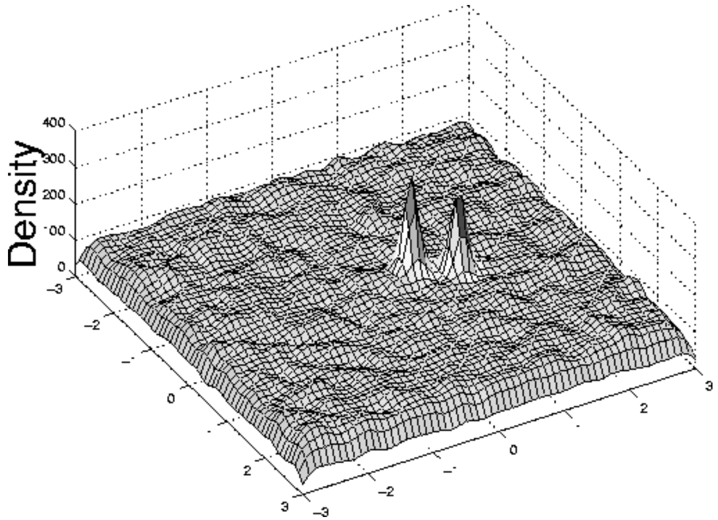
Noise Invariance





DENCLUE

Noise Invariance

DENCLUE Algorithm [HK 98]

Basic Idea

- Use ***Local Density Function*** which approximates the Global Density Function
- Use ***CubeMap Data Structure*** for efficiently locating the relevant points

DENCLUE

Local Density Function

Definition

The local density $\hat{f}_B^D(x)$ is defined as

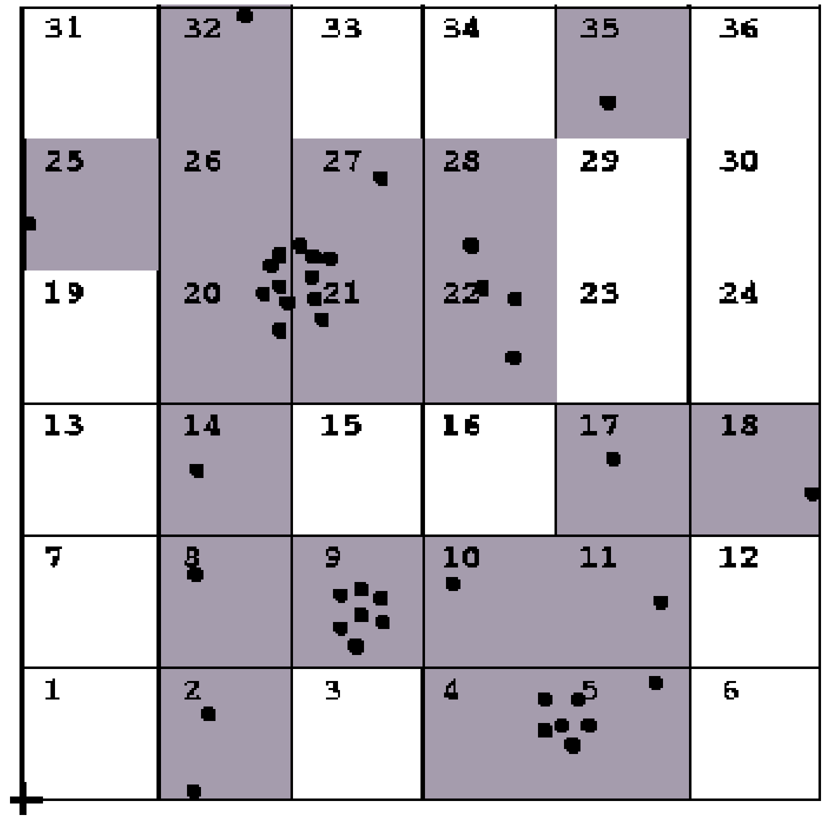
$$\hat{f}_B^D(x) = \sum_{x_i \in near(x)} f_B^{x_i}(x) \; .$$

Lemma (Error Bound)

If $near(x) = \{x_i \in D \mid d(x, x_i) \leq k\sigma\}$, the error is bound by:

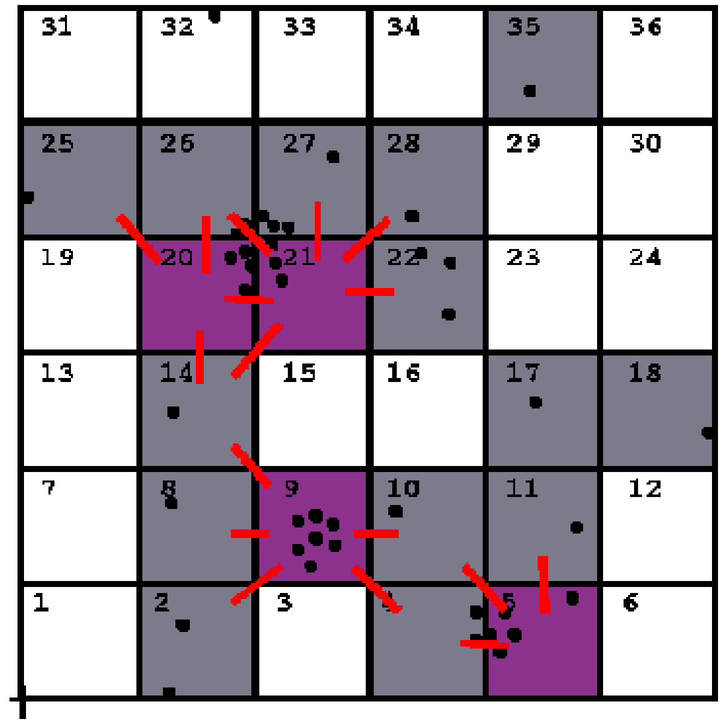
$$Error = \sum_{x_i \in D,\, d(x_i, x) > k\sigma} e^{-\frac{d(x, x_i)^2}{2\sigma^2}} \leq \| \{x_i \in D \mid d(x, x_i) > k\sigma\} \| \cdot e^{-\frac{k^2}{2}}$$

CubeMap



Data Structure based on regular cubes for storing the data and efficiently determining the density function

DENCLUE Algorithm



DENCLUE (D, σ, ξ)

(a) $MBR \leftarrow DetermineMBR(D)$

(b) $C_p \leftarrow DetPopCubes(D, MBR, \sigma)$

$C_{sp} \leftarrow DetHighlyPopCubes(C_p, \xi_c)$

(c) $map, C_r \leftarrow ConnectMap(C_p, C_{sp}, \sigma)$

(d) $clusters \leftarrow DetDensAttractors(map, C_r, \sigma, \xi)$

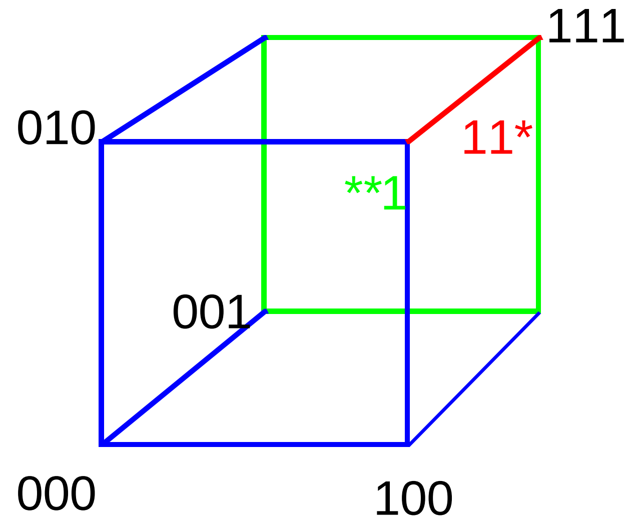
Effects of High Dimensionality

Number of Surfaces and Grid Cells

- Number of *k*-dimensional surfaces in a *d*-dimensional hypercube?

$$\binom{d}{k} \cdot 2^{(d-k)}$$

***



- Number of grid cells resulting from a binary partitioning?

$$2^d$$

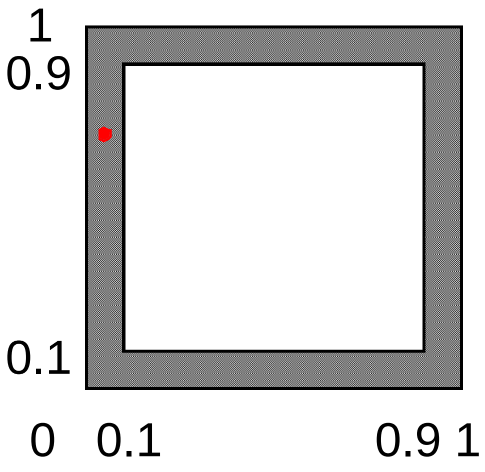
$\Rightarrow$ **grid cells can not be stored explicitly**

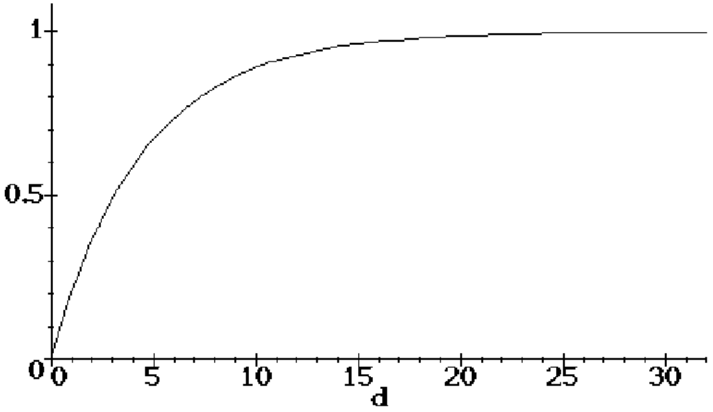
$\Rightarrow$ **most grid cells do not contain any data points**

Effects of High Dimensionality

The Surface is Everything

- Probability that a point is closer than 0.1 to a (*d*-1)-dimensional surface



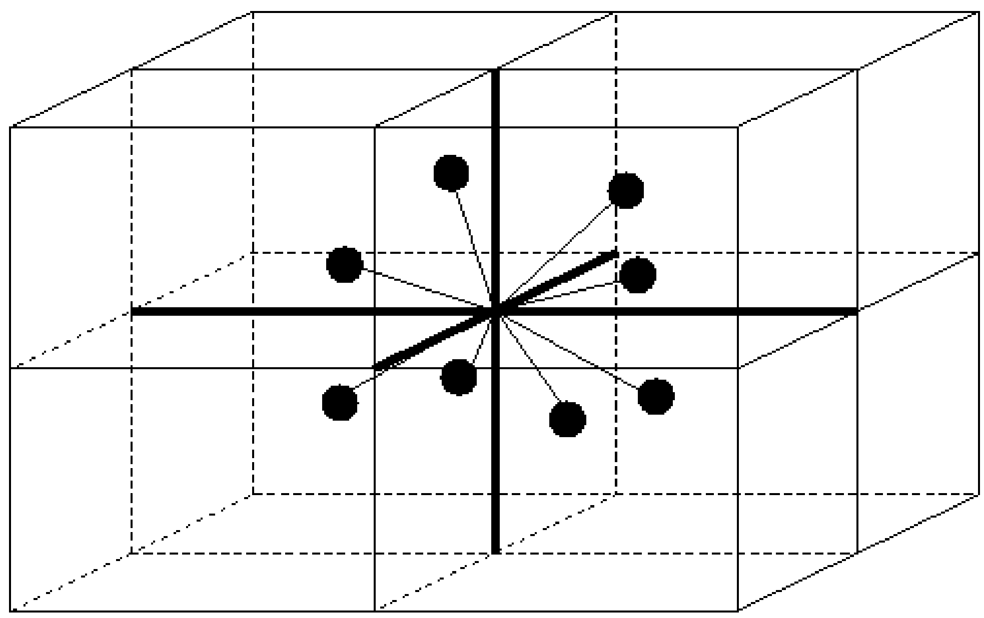


⇒ **no of directions (from center) increases exponentially**

Effects of High Dimensionality

Number of Neighboring cells

- Probability that Cutting Planes partition clusters increases



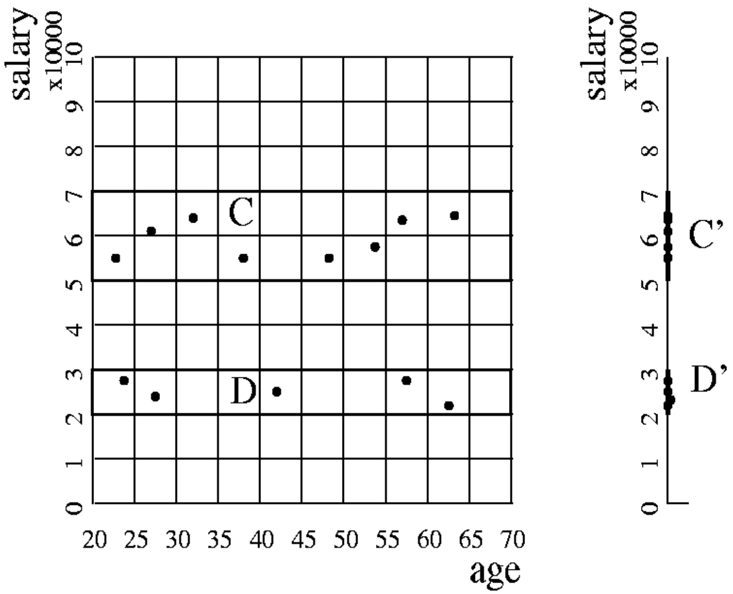
⇒ **cluster can not be identified using the grid**

Hybrid Methods

- CLIQUE [AGG+ 98]
- OptiGrid [HK 99]
- ...

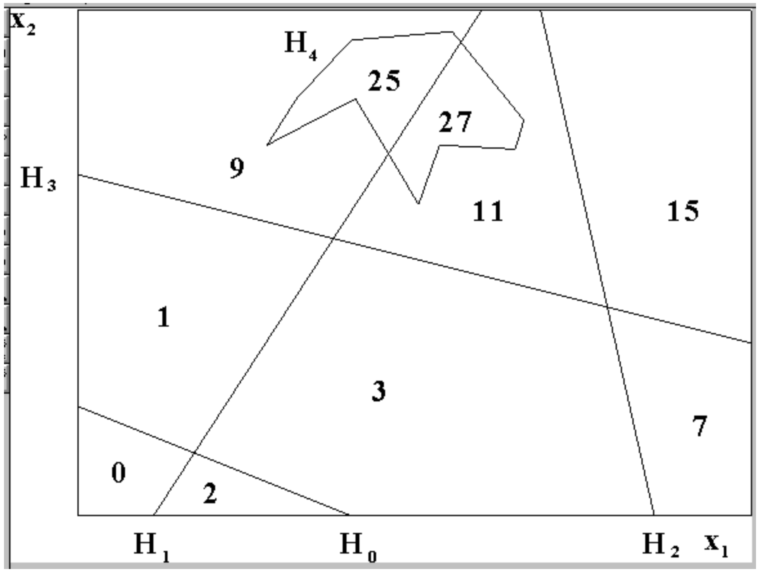
CLIQUE [AGG+ 98]

- Subspace Clustering
- Monotonicity Lemma:
  If a collection of points *S* is a cluster in a *k*-dimensional space, then S is also part of a cluster in any (k-1)-dimensional projection of this space.
- Bottom-up Algorithm for determining the projections

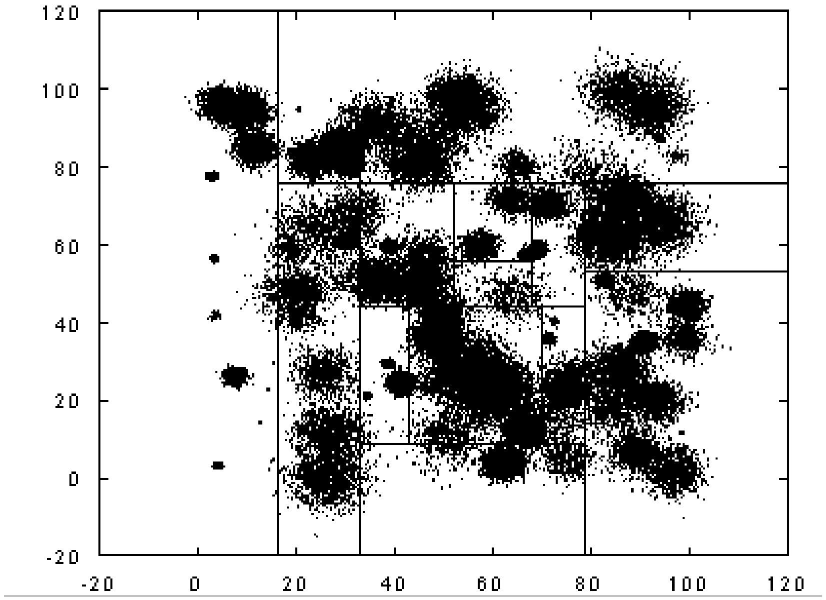


OptiGrid [HK 99]

- Optimal Grid Partitioning:



Generalized Grid



Recursive Partitioning

Summary and Conclusions

- A number of *effective* and *efficient* Clustering Algorithms is available for *small to medium size* data sets and *small dimensionality*
- **Efficiency** suffers severely for large dimensionality *(d)*
- **Effectiveness** suffers severely for large dimensionality *(d)*, especially in combination with a high *noise level*

Open Research Issues

- *Efficient Data Structures* for large ***N*** and large ***d***
- *Clustering Algorithms* which work *effectively* for large ***N***, large ***d*** and large *Noise Levels*
- *Integrated Tools* for an Effective Clustering of High-Dimensional Data
  (combination of automatic, visual and interactive clustering techniques)

References

- [AGG+ 98] R. Aggrawal, J. Gehrke, D. Gunopulos, P. Raghavan, *Automatic Subspace Clustering of High Dimensional Data for Data Mining Applications*, Proc. ACM SIGMOD Int. Conf. on Managment of Data, pp. 94-105, 1998
- [Boc 74] H.H. Bock, *Autmatic Classification*, Vandenhoeck and Ruprecht, Göttingen, 1974
- [BK 98] S. Berchtold, D.A. Keim, *High-Dimensional Index Structures, Database Support for Next Decade's Applications*, ACM SIGMOD Int. Conf. on Management of Data, 1998.
- [BBK 98] S. Berchtold, C. Böhm, H-P. Kriegel, *The Pyramid-Technique: Towards Breaking the Curse of Dimensionality*, Proc. ACM SIGMOD Int. Conf. on Management of Data, pp. 142-153, 1998.
- [BKK 96] S. Berchtold, D.A. Keim, H-P. Kriegel, *The X-Tree: An Index Structure for High-Dimensional Data*, Proc. 22th Int. Conf. on Very Large Data Bases, pp. 28-39, 1996.
- [BKK 97] S. Berchtold, D. Keim, H-P. Kriegel, *Using Extended Feature Objects for Partial Similarity Retrieval*, VLDB Journal, Vol.4, 1997.
- [BKSS 90] N. Beckmann., h-P. Kriegel, R. Schneider, B. Seeger, *The R*-tree: An Efficient and Robust Access Method for Points and Rectangles*, Proc. ACM SIGMOD Int. Conf. on Management of Data, pp. 322-331, 1990.

- [CHY 96] Ming-Syan Chen, Jiawei Han, Philip S. Yu: *Data Mining: An Overview from a Database Perspective*. TKDE 8(6), pp. 866-883, 1996.
- [EKS+ 96] M. Ester, H-P. Kriegel, J. Sander, X. Xu, *A Density-Based Algorithm for Discovering Clusters in Large Spatial Databases with Noise*, Proc. 2nd Int. Conf. on Knowledge Discovery and Data Mining, 1996.
- [EKSX 98] M. Ester, H-P. Kriegel, J. Sander, X. Xu, *Clustering for Mining in Large Spatial Databases*, Special Issue on Data Mining, KI-Journal, ScienTec Publishing, No. 1, 1998.

- [EKSX 98] M. Ester, H-P. Kriegel, J. Sander, X. Xu, *Clustering for Mining in Large Spatial Databases*, Special Issue on Data Mining, KI-Journal, ScienTec Publishing, No. 1, 1998.

- [EKX 95] M. Ester, H-P. Kriegel, X. Xu, *Knowlege Discovery in Large Spatial Databases: Focusing Techniques for Efficient Class Identification*, Lecture Notes in Computer Science, Springer 1995.
- [EKX 95b] M. Ester, H-P. Kriegel, X. Xu, *A Database Interface for Clustering in Large Spatial Databases*, Proc. 1st Int. Conf. on Knowledge Discovery and Data Mining, 1995.
- [EW 98] M. Ester, R. Wittmann, *Incremental Generalization for Mining in a Data Warehousing Environment*, Proc. Int. Conf. on Extending Database Technology, pp. 135-149, 1998.
- [DE 84] W.H. Day and H. Edelsbrunner, *Efficient algorithms for agglomerative hierachical clustering methods*, Journal of Classification, 1(1):7-24, 1984.
- [DH 73] R.O. Duda and P.E. Hart, *Pattern Classifaication and Scene Analysis*, New York; Wiley and Sons , 1973.
- [Fuk 90] K. Fukunaga, *Introduction to Statistical Pattern Recognition*, San Diego, CA, Academic Press 1990.

- [Fri 95] B. Fritzke, *A Growing Neural Gas Network Learns Topologies*, in G. Tesauro, D.S. Touretzky and T.K. Leen (eds.) *Advances in Neural Information Processing Systems 7*, MIT Press, Cambridge MA, 1995.
- [FH 75] K. Fukunaga and L.D. Hosteler, *The Estimation of the Gradient of a density function with Applications in Pattern Recognition*, IEEE Trans. Info. Thy., IT-21, 32-40, 1975.
- [HK 98] A. Hinneburg, D.A. Keim, *An Efficient Approach to Clustering in Large Multimedia Databases with Noise*, Proc. 4th Int. Conf. on Knowledge Discovery and Data Mining, 1998.
- [HK 99] A. Hinneburg, D.A. Keim, *The Muti-Grid: The Curse of Dimensionality in High-Dimensional Clustering* , submitted for publication
- [Jag 91] J. Jagadish, *A Retrieval Technique for Similar Shapes*, Proc. ACM SIGMOD Int. Conf. on Management of Data, pp. 208-217, 1991.
- [Kei 96] D.A. Keim, *Databases and Visualization*, Tutorial on ACM SIGMOD Int. Conf. on Management of Data, 1996.
- [KMN 97] M.Kearns, Y. Mansour and A. Ng, *An Information-Theoretic Analysis of Hard and Soft Assignment Methods for Clustering*, Proc. 13th Conf. on Uncertainty in Artificial Intelligence, pp. 282-293, 1997, Morgan Kaufmann.
- [KMS+ 98] T. Kohonen, K. Mäkisara, O.Simula and J. Kangas, *Artificaial Networks*, Amsterdam 1991.
- [Lau 95] S.L. Lauritzen, *The EM algorithm for graphical association models with missing data*, Computational Statistics and Data Analysis, 19:191-201, 1995.
- [Mur 84] F. Murtagh, *Complexities of hierarchic clustering algorithms: State of the art*, Computational Statistics Quarterly, 1:101-113, 1984.

- [MG 93] R. Mehrotra, J. Gary, *Feature-Based Retrieval of Similar Shapes*, Proc. 9th Int. Conf. on Data Engeneering, April 1993.
- [NH 94] R.T. Ng, J. Han, *Efficient and Effective Clustering Methods for Spatial Data Mining*, Proc. 20th Int. Conf. on Very Large Data Bases, pp. 144-155, 1994.
- [Roj 96] R. Rojas, *Neural Networks - A Systematic Introduction*, Springer Berlin, 1996.
- [Sch 64] P. Schnell, *A Method for Discovering Data-Groups*, Biometrica 6, 47-48, 1964.
- [Sil 86] B.W. Silverman, *Density Estimation for Statistics and Data Analysis*, Chapman and Hall, 1986.
- [Sco 92] D.W. Scott, *Multivariate Density Estimation*, Wiley and Sons, 1992.
- [Sch 96] E. Schikuta, *Grid clustering: An efficient hierarchical method for very large data sets*, Proc. 13th Conf. on Patter Recognition, Vol. 2 IEEE Computer Society Press, pp. 101-105, 1996.
- [SCZ 98] G.Sheikholeslami, S. Chatterjee and A. Zhang, *WaveCluster: A Multi-Resolution Clustering Approach for Very Large Spatial Databases*, Proc. 24th Int. Conf. on Very Large Data Bases, 1998.
- [Wis 69] D. Wishart, *Mode Analysis: A Generalisation of Nearest Neighbor, which reducing Chaining Effects*, in A. J. Cole (Hrsg.), 282-312, 1969.
- [WYM 97] W. Wang, J. Yang, R. Muntz, *STING: A Statistical Information Grid Approach to Spatial Data Mining*, Proc. 23rd Int. Conf. on Very Large Data Bases 1997.
- [XEK+ 98] X. Xu, M. Ester, H-P. Kriegel and J. Sander., *A Distribution-Based Clustering Algorithm for Mining in Large Spatial Databases*, Proc. 14th Int. Conf. on Data Engineering (ICDE'98), Orlando, FL, 1998, pp. 324-331.
- [ZRL 96] T. Zhang, R. Ramakrishnan and M. Livny, *An Efficient Data Clustering Method for Very Large Databases*. Proc. ACM SIGMOD Int. Conf. on Managment of Data, pp. 103-114, 1996