Clustering scalable algorithms and density based methods

K. Gibert[1]

[1]Department of Statistics and Operation Research

Knowledge Engineering and Machine Learning group
Universitat Politècnica de Catalunya, Barcelona

Clustering very large data sets

Many strategies to accellerate the clustering process

One pass preprocessing

Sumarization

sampling

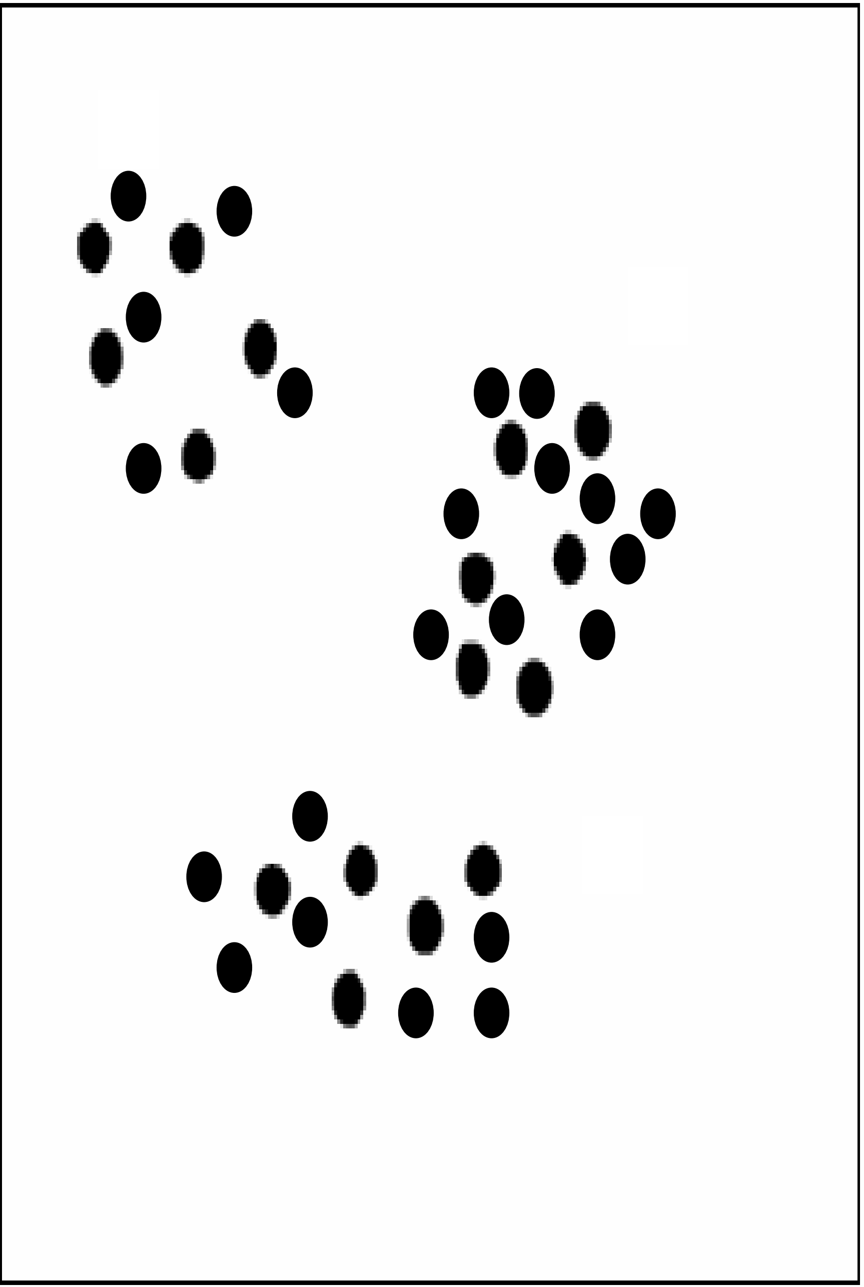
batches

Clustering very large data sets

combining k-means with hierarchical

1. Perform m (=2 or 3) times k-means (with k≈10)
2. Form the crosstable of m obtained partitions
3. Find centroids of the (non empty) cells of the crosstable
4. Weighed Hierarchical Clustering of centroids (cell-size
5. Cut the tree and find number of classes
6. Consolidate your clustering
    1. Find centroids of classes
    2. K-means using centroids as class seeds

CURE (Clustering Using REpresentatives) *[Guha Rastogi 2001]*

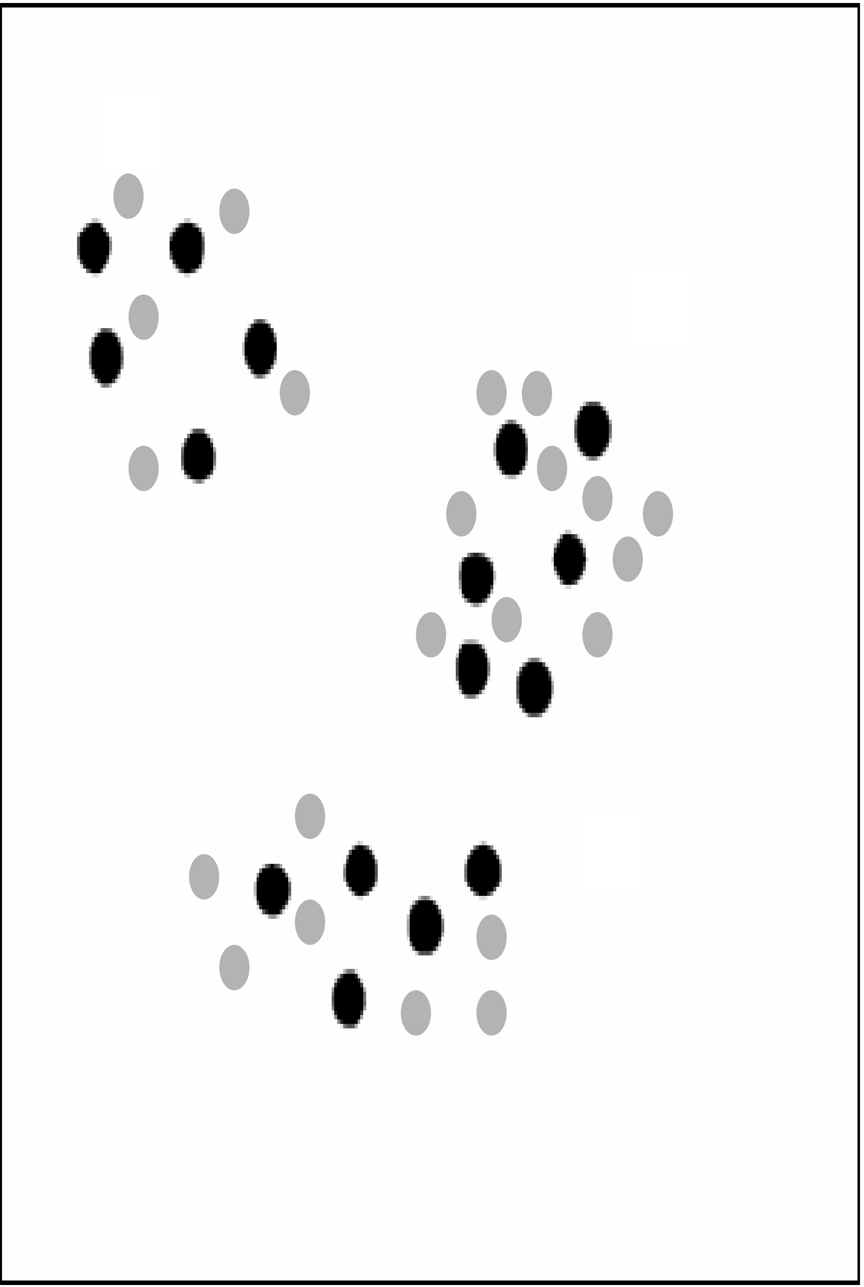


Accelerates hierarchical clustering

- Pick a reduced random sample

©K. Gibert

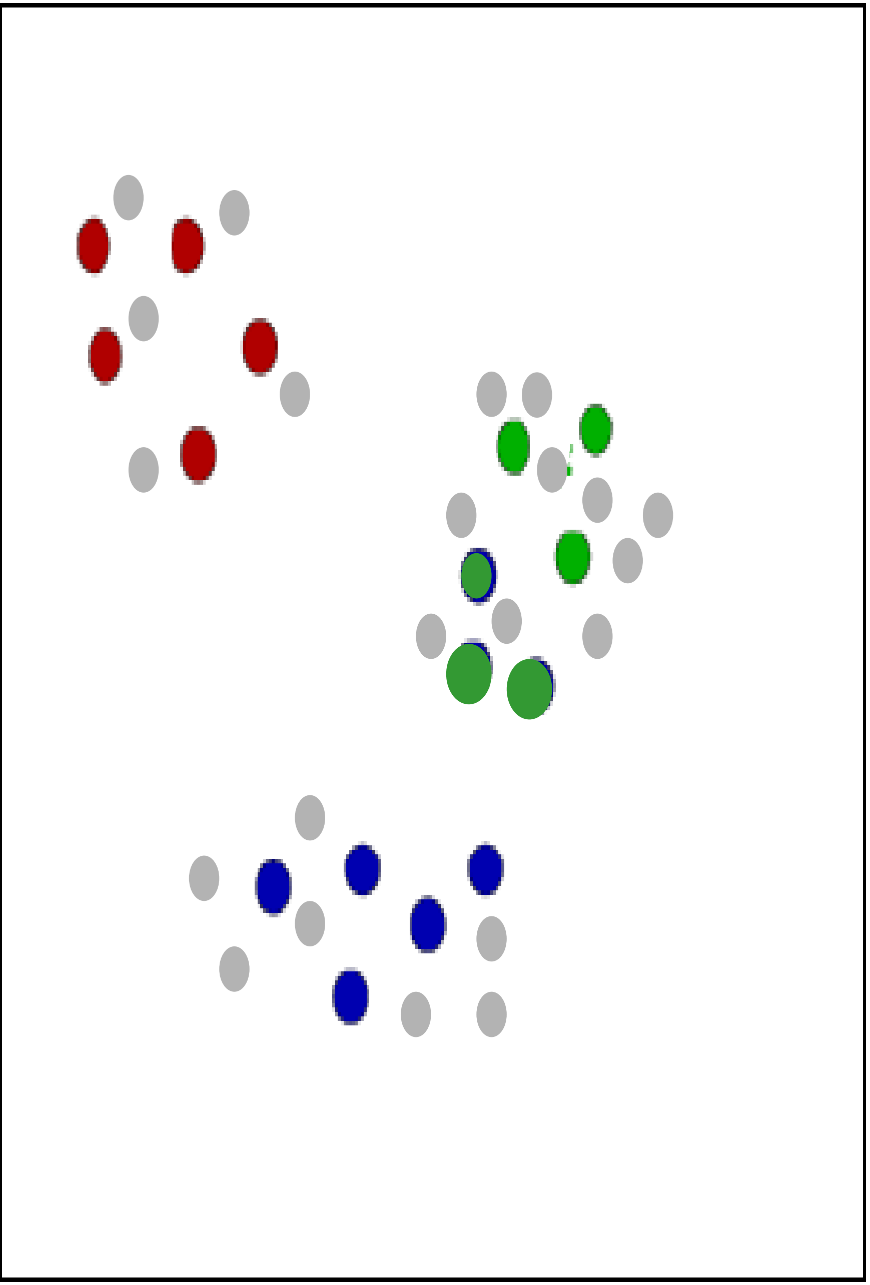
CURE (Clustering Using REpresentatives) *[Guha Rastogi 2001]*



Accelerates hierarchical clustering

- Pick a reduced random sample

CURE (Clustering Using REpresentatives) *[Guha Rastogi 2001]*



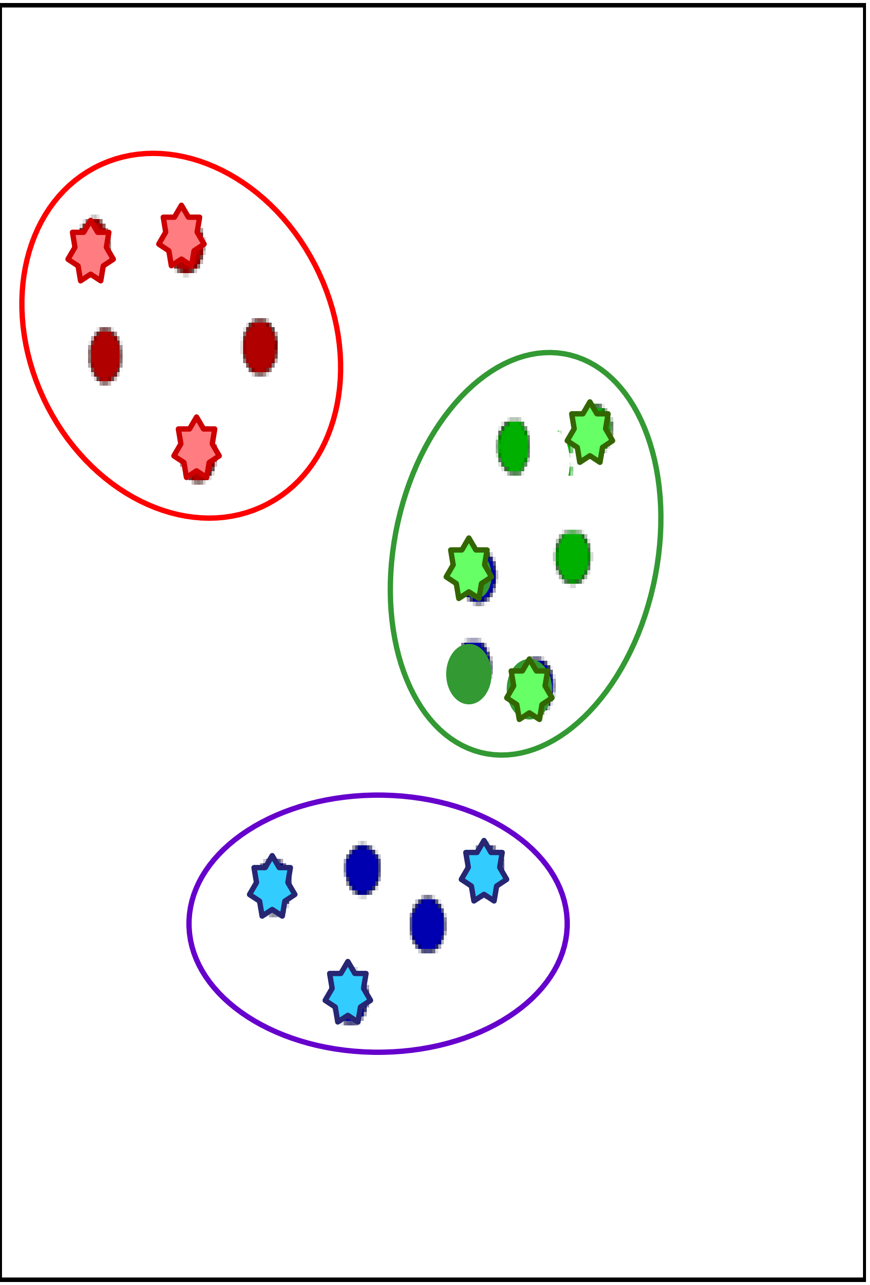
Accelerates hierarchical clustering

- Pick a reduced random sample
- Hierarchical clustering of the sample *(euclidean distance and single linkage)*

Cut the tree and find classes

©K. Gibert

CURE (Clustering Using REpresentatives) *[Guha Rastogi 2001]*



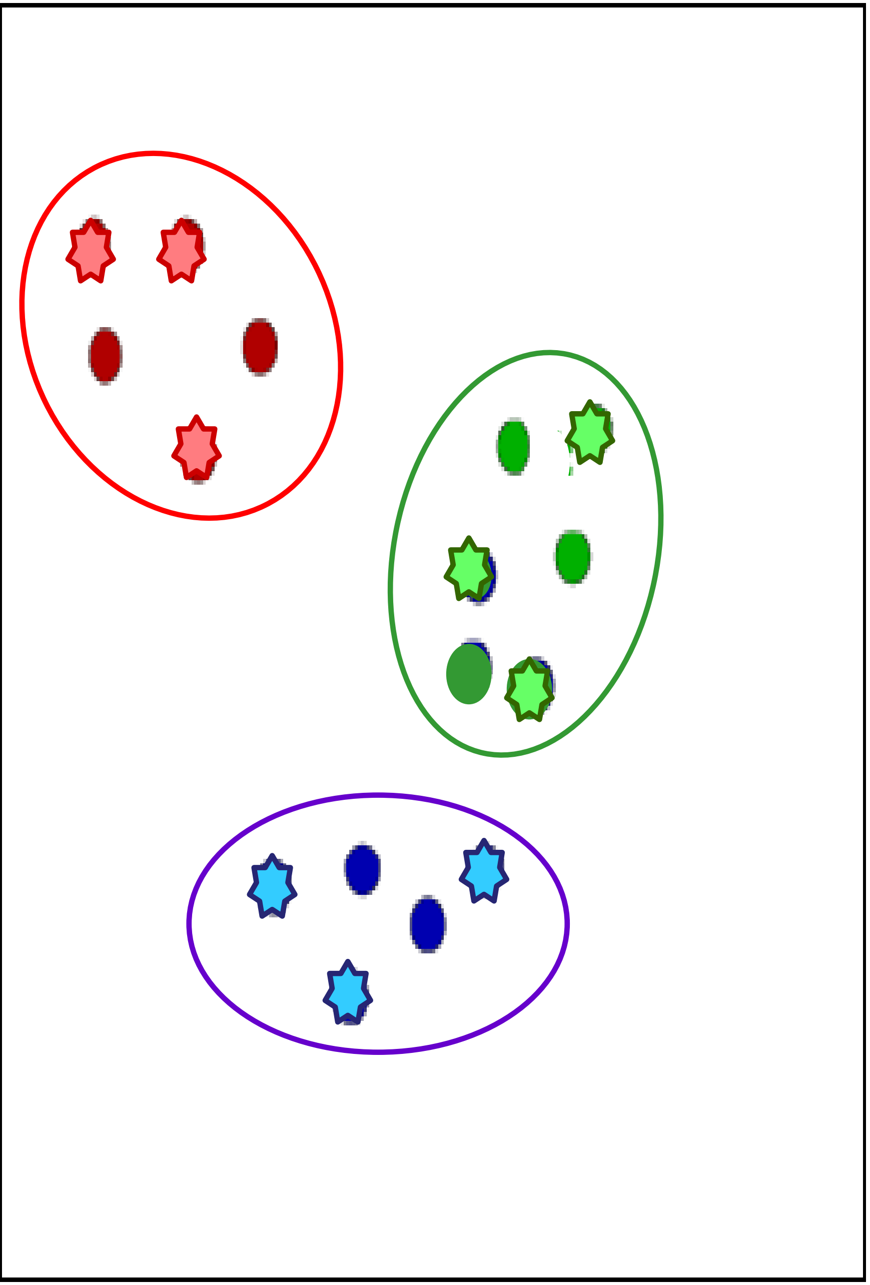
Accelerates hierarchical clustering

- Pick a reduced random sample
- Hierarchical clustering of the sample
  (decide metrics and aggregation criterion)

  Cut the tree and find classes
- Select a set of disperse points/class

©K. Gibert

CURE (Clustering Using REpresentatives) *[Guha Rastogi 2001]*



Accelerates hierarchical clustering

- Pick a reduced random sample

- Hierarchical clustering of the sample
  (decide metrics and aggregation criterion)

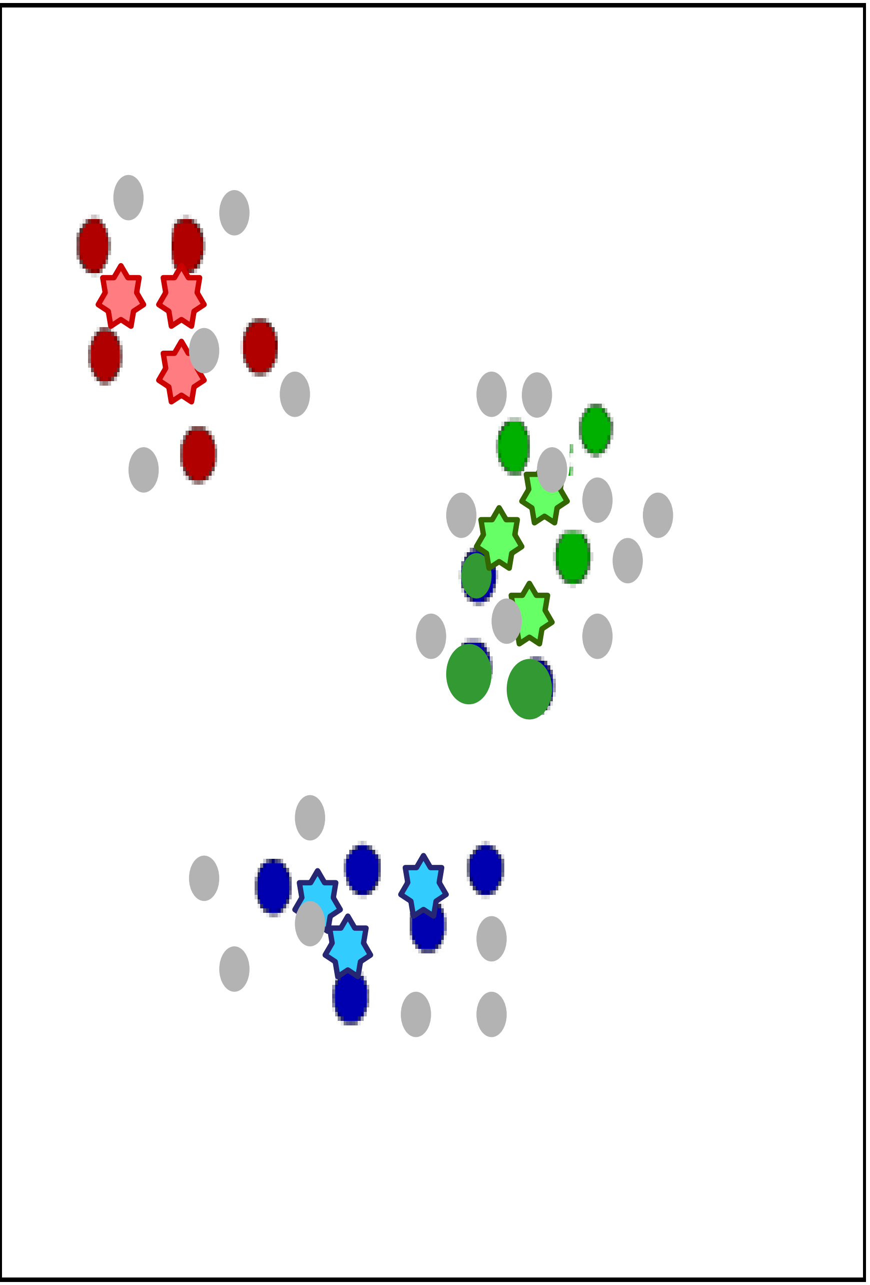
  Cut the tree and find classes

- Select a set of disperse points/class

- Move representative h% vers centroid (h% = 20%,.....)

©K. Gibert

CURE (Clustering Using REpresentatives) *[Guha Rastogi 2001]*



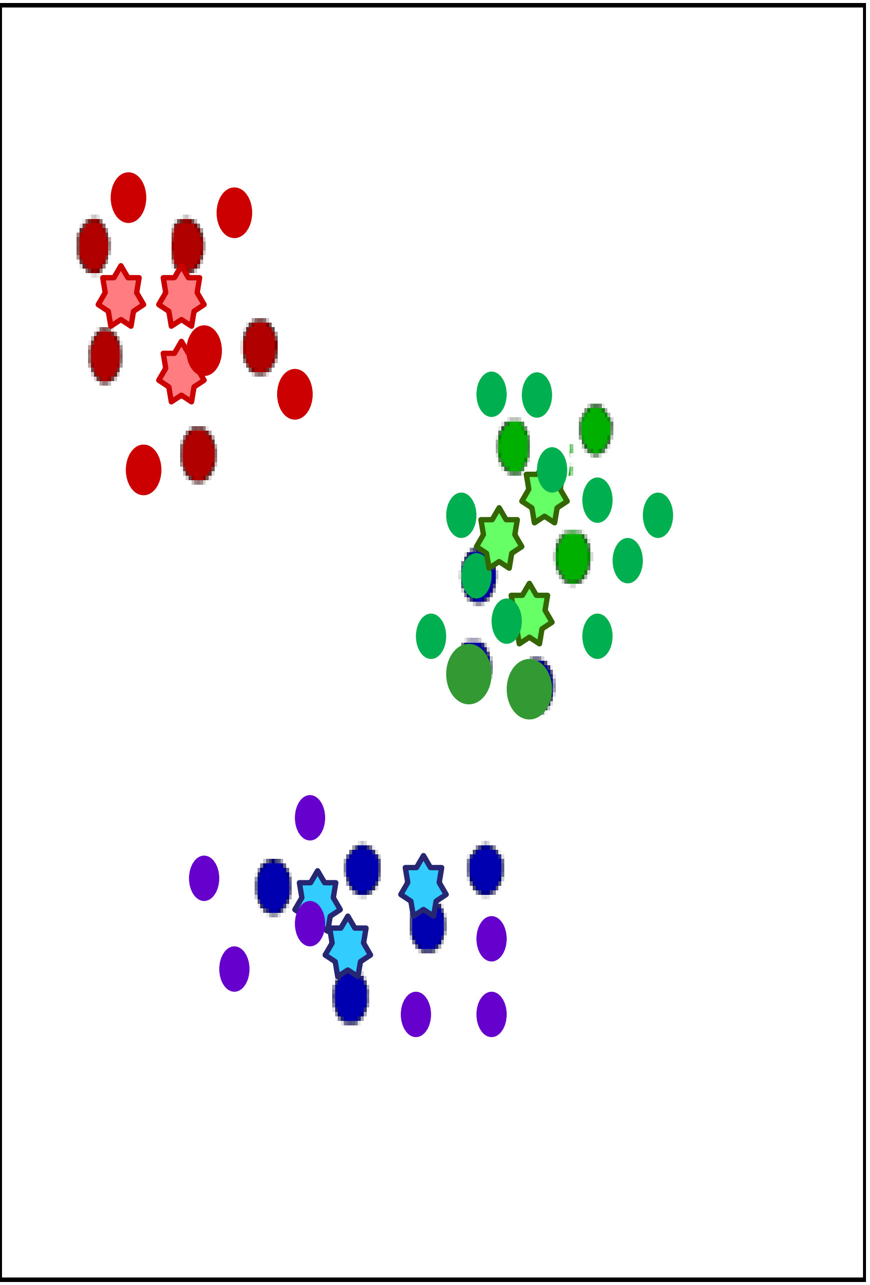
Accelerates hierarchical clustering

- Pick a reduced random sample
- Hierarchical clustering of the sample *(decide metrics and aggregation criterion)*

  Cut the tree and find classes
- Select a set of disperse points/class
- Move representative h% vers centroid (h% = 20%,.....)
- Assign remaining objects to closest representative

©K. Gibert

CURE (Clustering Using REpresentatives) *[Guha Rastogi 2001]*



Accelerates hierarchical clustering

- Pick a reduced random sample
- Hierarchical clustering of the sample *(decide metrics and aggregation criterion)*

  Cut the tree and find classes
- Select a set of disperse points/class
- Move representative h% vers centroid (h% = 20%,.....)
- Assign remaining objects to closest representative

©K. Gibert

CHAMELEON (Karypis, Han et al. 1999)

Hierarchical Clustering Using Dynamic Modeling

- Measures the similarity based on a dynamic model
  - Aggregation:

    interconnectivity and *closeness (proximity)* between high *relative to* interconnectivity and closeness of items within the clusters

  - Merging scheme preserves self-similarity

- Graph-based
- Clusters: densely populated regions
- Free-shaped clusters
- Robust to Unbalanced sizes, noise and outliers

CHAMELEON

Two-phase algorithm

0. Represent data by a graph:
   1. K-NN graph (usually sparse)
   2. Catch neighborhood dynamically (by density of region)
   3. Outliers taken appart

1. (multi-level) Graph-partitioning algorithm *(hMeTiS library)*:
   1. Minimize the edge cut *(edge meand similarity between points)*

   large number of relatively small and well-connected sub-clusters

2. Agglomerative hierarchical clustering to merge sub-clusters

   agregation criterion: RI(C,C') RC(C,C')$^{\alpha}$

   --- Relative interconnectivity $RIC = \dfrac{\underline{AbsoluteIC(C,C')}}{(internal\ IC(C)+internal\ IC(C'))\ /\ 2}$

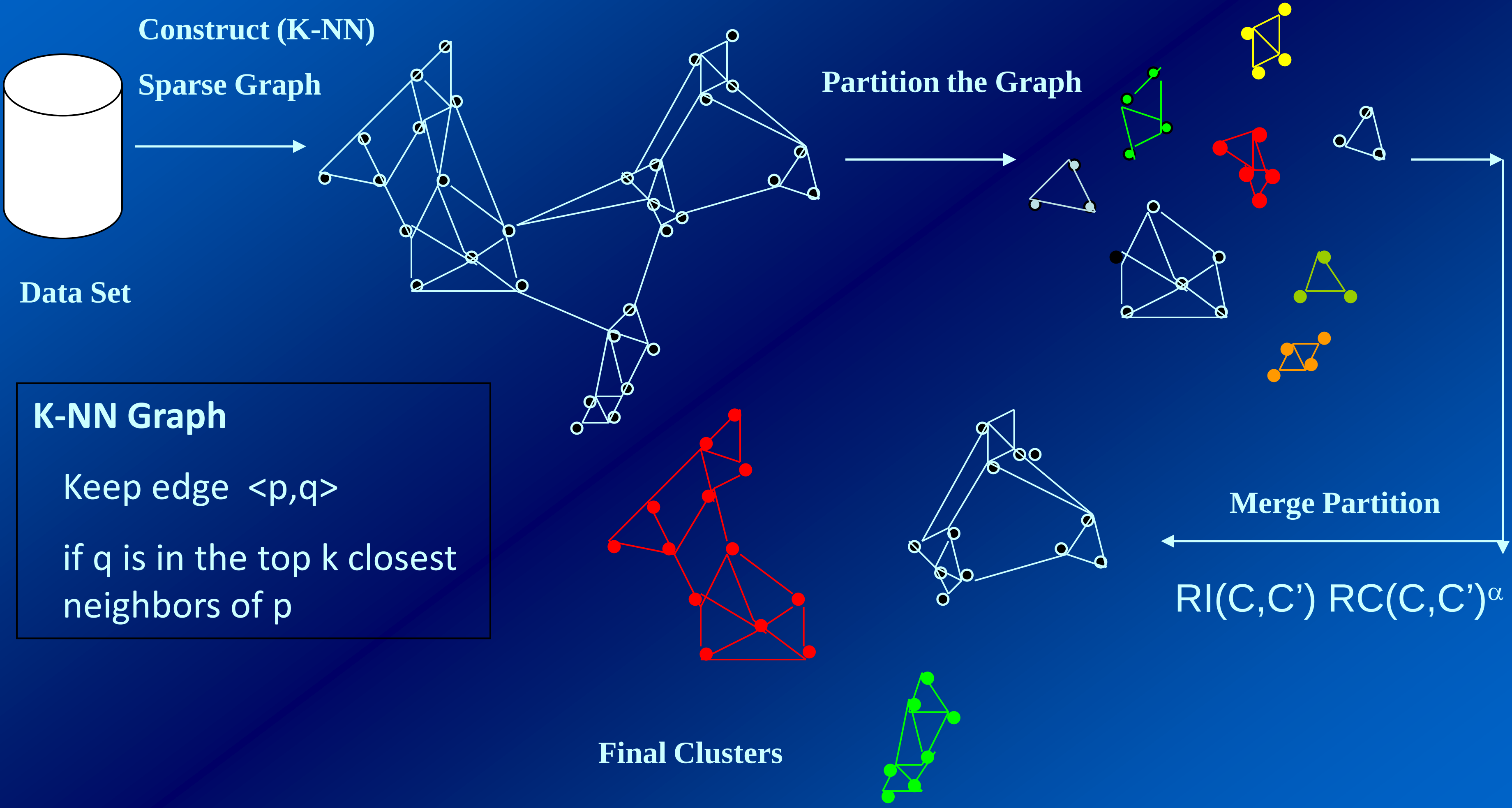
   AbsoluteIC(C,C')= sum of weights of edges connecting C and C'.
   internalIC(C) = weighted sum of edges partitioning cluster in equal parts.

   --- Relative closeness:

   absolute cl normalized w.r.t. internal cl of both clusters
   absolute cl: average weight (simil) of points C(i) -> C'(j)

©K. Gibert

CHAMELEON



Construct (K-NN)

Sparse Graph

Data Set

Partition the Graph

Merge Partition

$RI(C,C')\ RC(C,C')^{\alpha}$

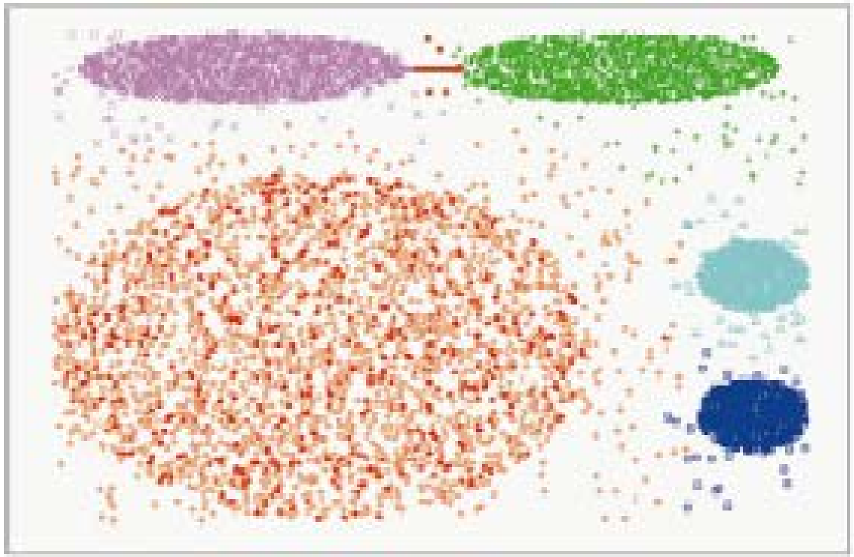
Final Clusters

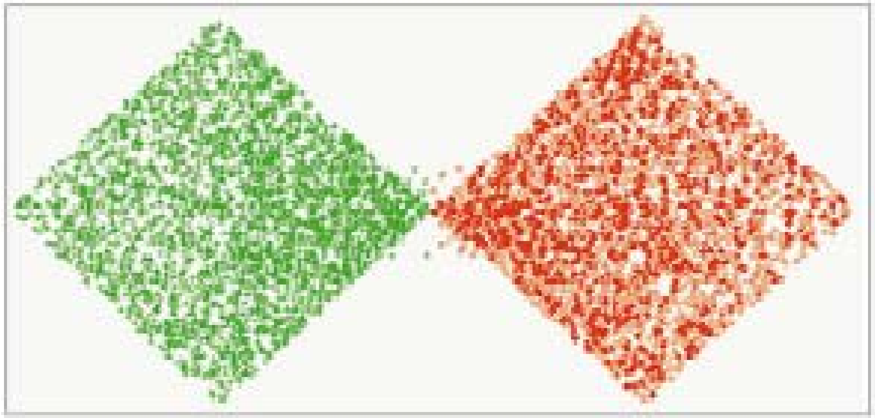
K-NN Graph

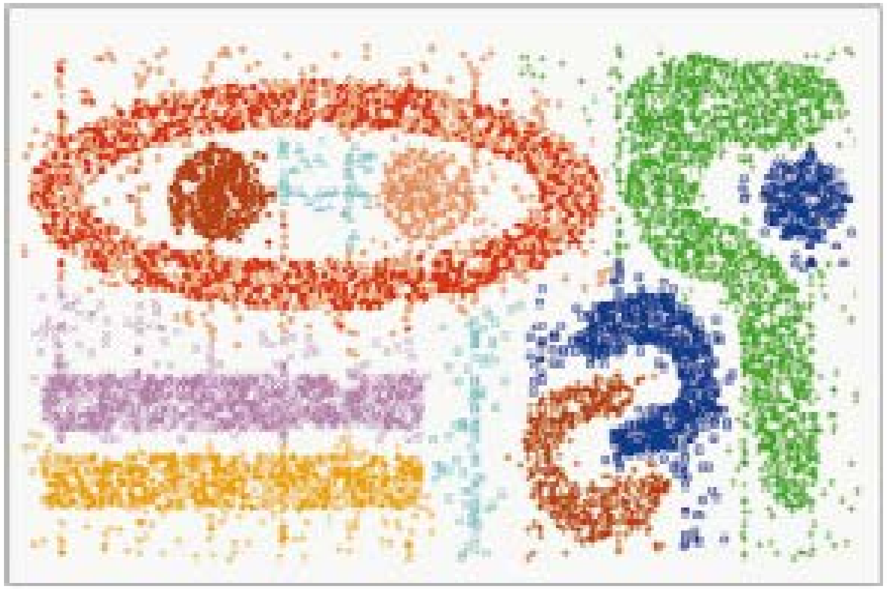
Keep edge <p,q>

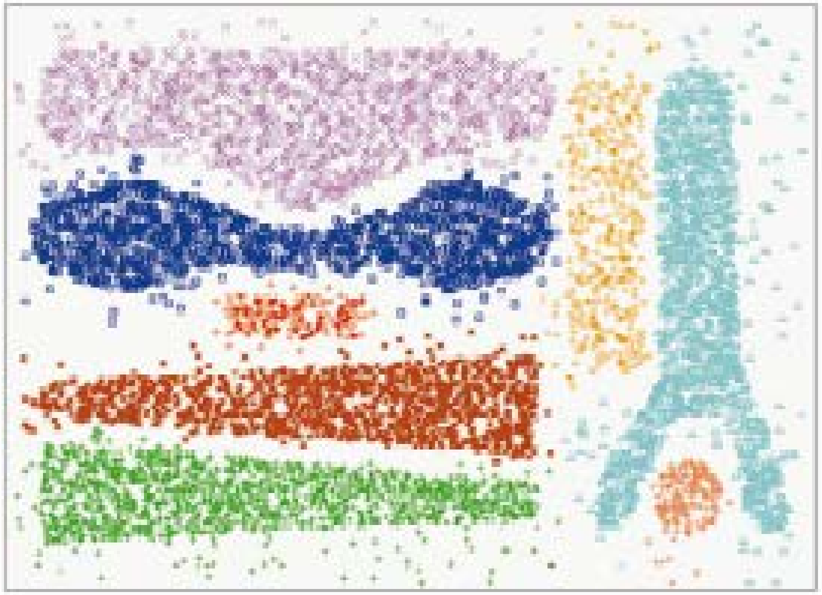
if q is in the top k closest neighbors of p

CHAMELEON (Clustering Complex Objects)









Density-Based Clustering Methods

- Clustering based on density (local cluster criterion), such as density-connected points
- Major features:
  - Discover clusters of arbitrary shape
  - Handle noise
  - One scan
  - Need density parameters as termination condition
- Several interesting studies:
  - DBSCAN: Ester, et al. (KDD'96)
  - OPTICS: Ankerst, et al (SIGMOD'99).
  - DENCLUE: Hinneburg & D. Keim (KDD'98)
  - CLIQUE: Agrawal, et al. (SIGMOD'98)

DBSCAN (Density-based Spatial Clustering of Applications with Noise) *[Ester96]*

- Density-estimation based clustering method. Scalable

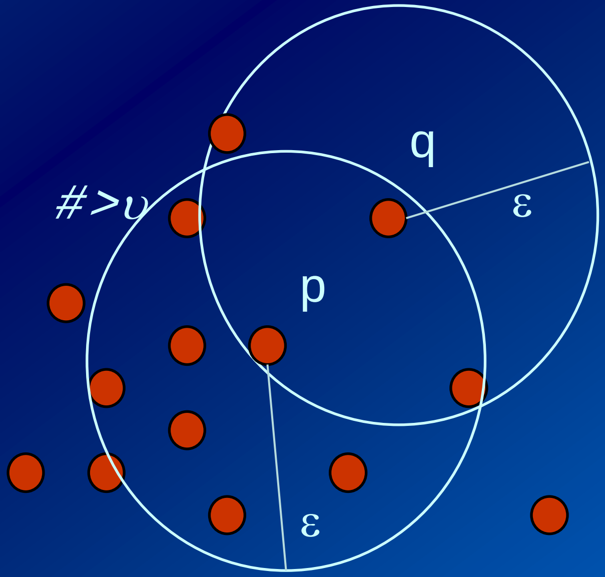
  q directly density-reachable (ddr) from p iff

  Given ε, υ

  i) d(p,q) < ε and

  ii) p is in a *dense* zone

  card{o:d(o,p)<ε}>υ



©K. Gibert

DBSCAN (Density-based Spatial Clustering of Applications with Noise) *[Ester96]*

- Density-estimation based clustering method. Scalable

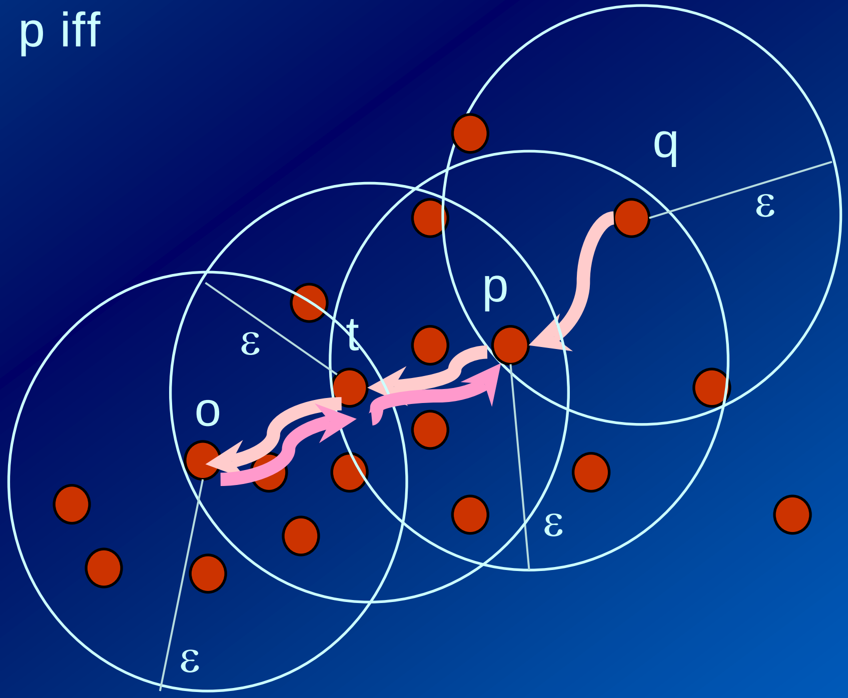
  q directly density-reachable (ddr) from p iff

  Given ε, υ

  i) d(p,q) < ε and

  ii) p is in a *dense* zone

  card{o:d(o,p)<ε}>υ



o density-reachable from q through a path of ddr points (assimetric)

o, p density-connected if both density-reachable by an intermediate object t

©K. Gibert

DBSCAN (Density-based Spatial Clustering of Applications with Noise) *[Ester96]*

- Density-estimation based clustering method. Scalable

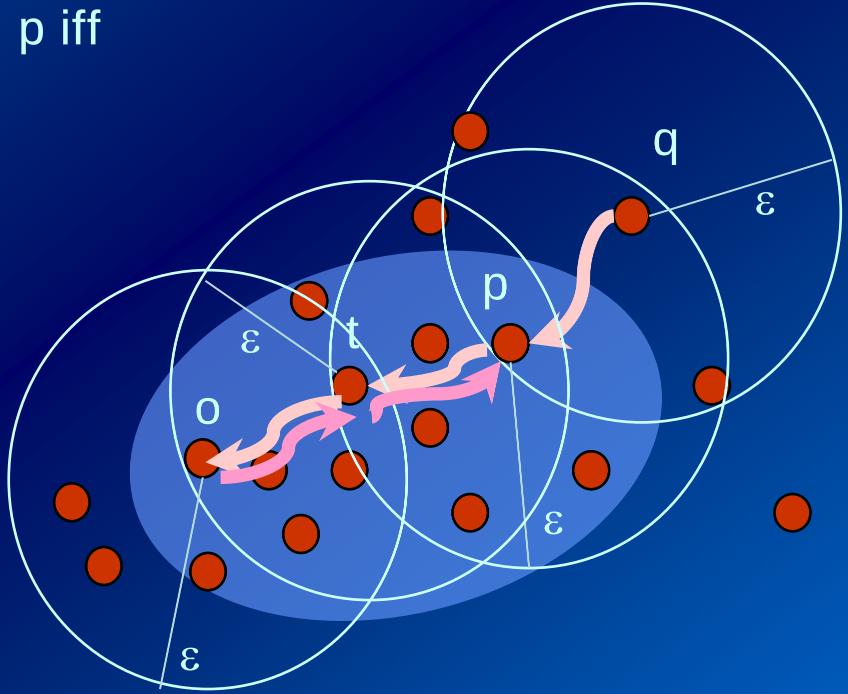
  q directly density-reachable (ddr) from p iff

  Given ε, υ

  i) d(p,q) < ε and

  ii) p is in a *dense* zone

  card{o:d(o,p)<ε}>υ



o density-reachable from q through a path of ddr points (assimetric)

o, p density-connected if both density-reachable by an intermediate object t

All points in a cluster mutually density-connected.

Cluster includes all points density-connected with the whole cluster points

©K. Gibert

DBSCAN (Density-based Spatial Clustering of Applications with Noise)

- Algorithm

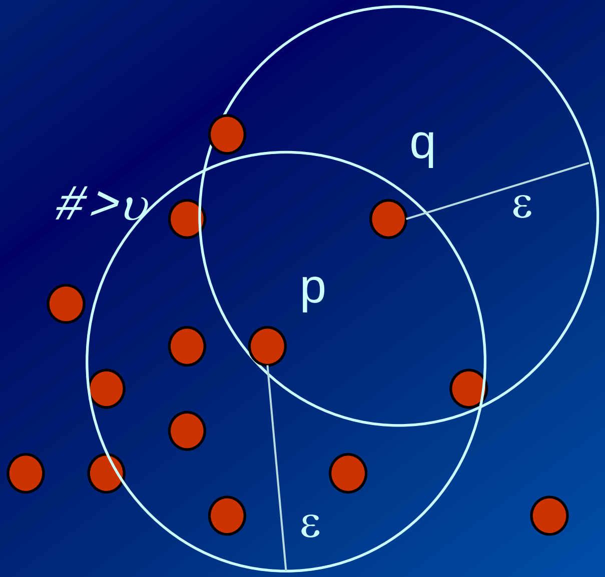
Visit a random point

Retrieve the ε-neigborhoud density-reachable

if card > υ form a cluster, else label point as noise

Growth the cluster with all free dense points in the ε-neigborhoud recursively

Repeat with another unvisited point



Number of clusters as an output.

Free shape of clusters

Order depending

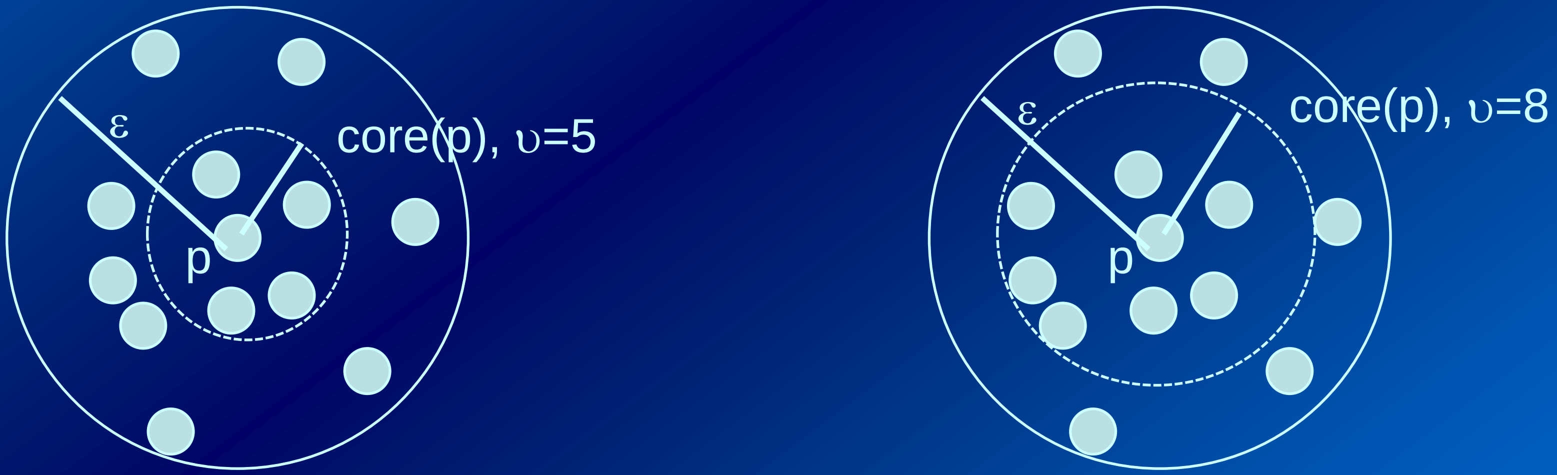
Euclidean distance useless in high-dimensional spaces (curse of dimensionality)

Bad performance if densities of clusters are very different (global ε υ)

OPTICS *[Ankerst 99]*

- *OPTICS:* generalizes DBSCAN to multiple ranges
  - The ε becomes an upperbound and υ becomes minimum cluster size
  - Hierarchical
  - Output: Reachability plot

  - p is a core object if it has a dense neighbourhood: $\mathit{card}\{o{:}d(o,p)<\varepsilon\}\geq\upsilon$
  - Core_distance: core(p)= Minimum ε such that p is a core object



  - Robust to noise and outliers
  - Free-shaped clusters

OPTICS

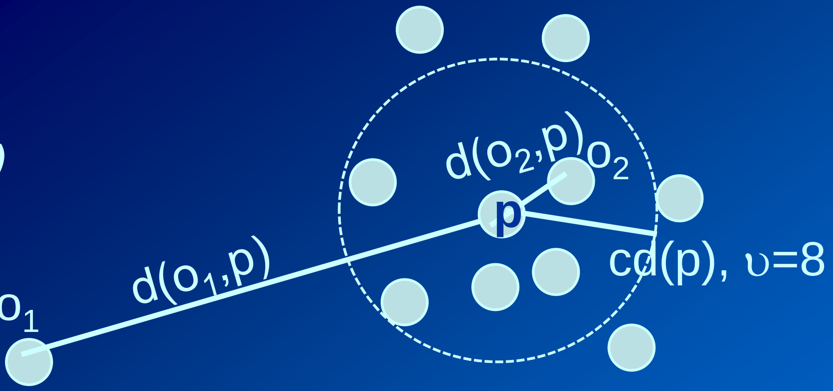
- *OPTICS:* generalizes DBSCAN to multiple ranges
  - The ε becomes an upperbound and υ becomes minimum cluster size
  - Hierarchical
  - Output: Reachability plot

  - p is a core object if it has a dense neighbourhood: *card{o:d(o,p)<ε}≥υ*
  - Core_distance: core(p)= Minimum ε such that p is a core object
  - Reachability_distance:

rp(p,o): max(core_distance(p), d(p,o))

$rd(p, o_1) = d(p, o_1)$

$rd(p, o_2) = cd(p)$



©K. Gibert

OPTICS

- *OPTICS:* generalizes DBSCAN to multiple ranges
  - The ε becomes an upperbound and υ becomes minimum cluster size
  - Hierarchical
  - Output: Reachability plot

  - p is a core object if it has a dense neighbourhood: *card{o:d(o,p)<ε}$\geq\upsilon$*
  - Core_distance: core(p)= Minimum ε such that p is a core object
  - Reachability_distance: *rp(p,o): max(core_distance(p), d(p,o))*

Visit a random p0.

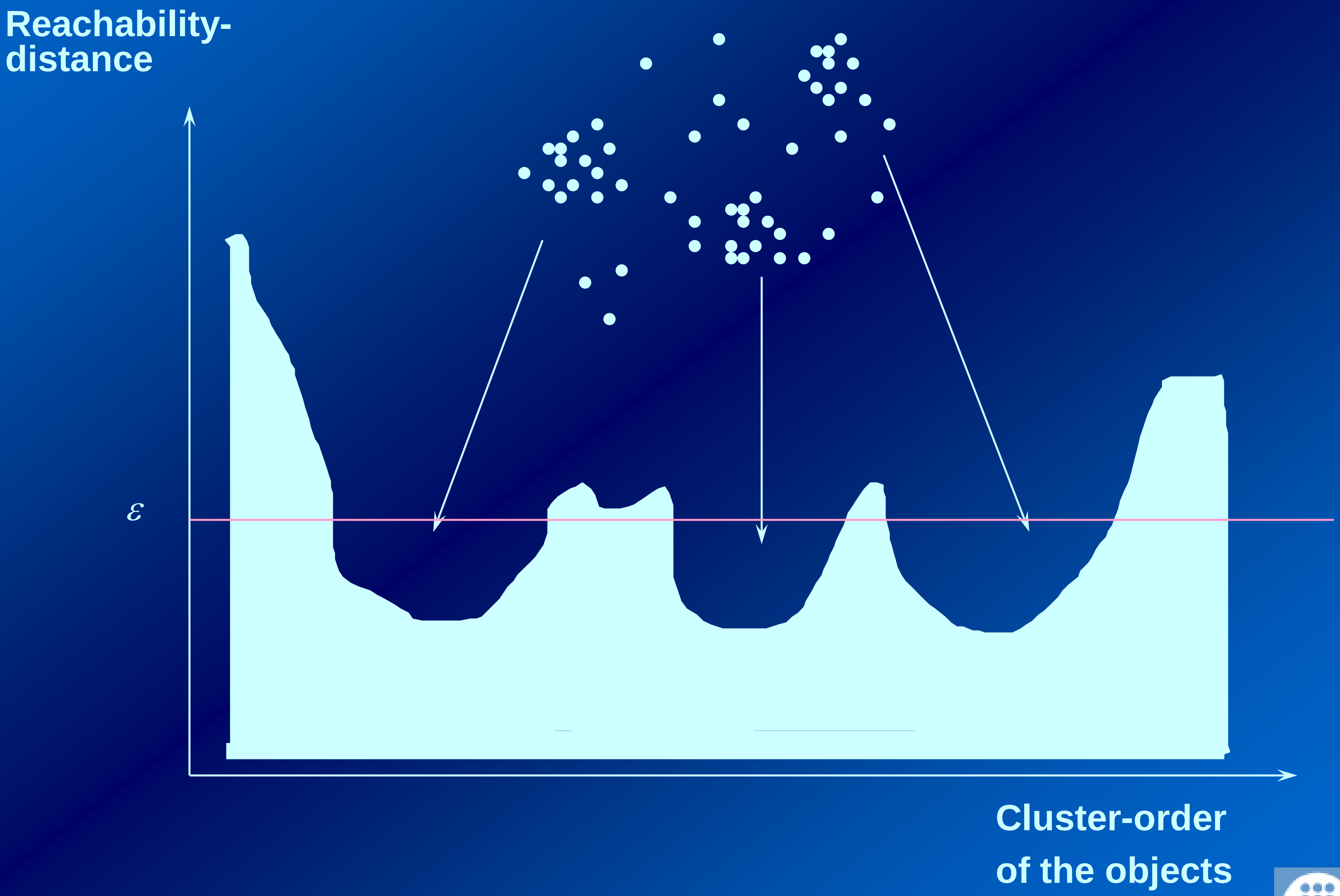
Find the neighbor with min reachability-distance to visited core_object

Proceed till the whole data set is visited

Defines an ordering of points

- Output: Reachability plot: x= visit-ordering X y= reachability distance

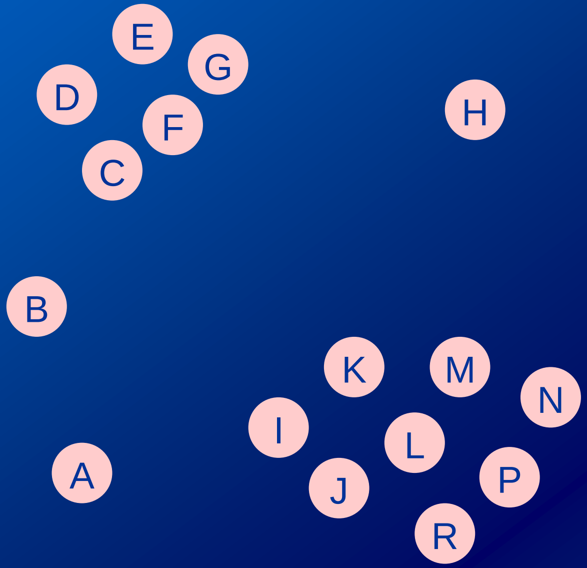
Reachability plot

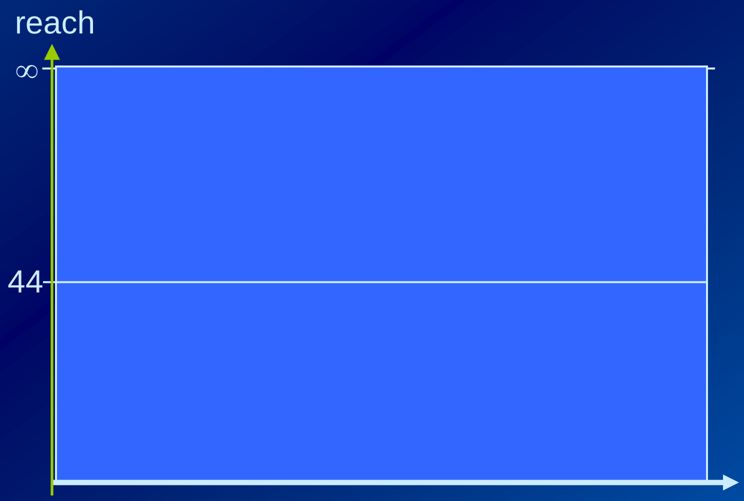


©K. Gibert

Example *[Pfeifle2004]*

- Example Database (2-dimensional, 16 points)
- $\varepsilon = 44$, $\upsilon = 3$



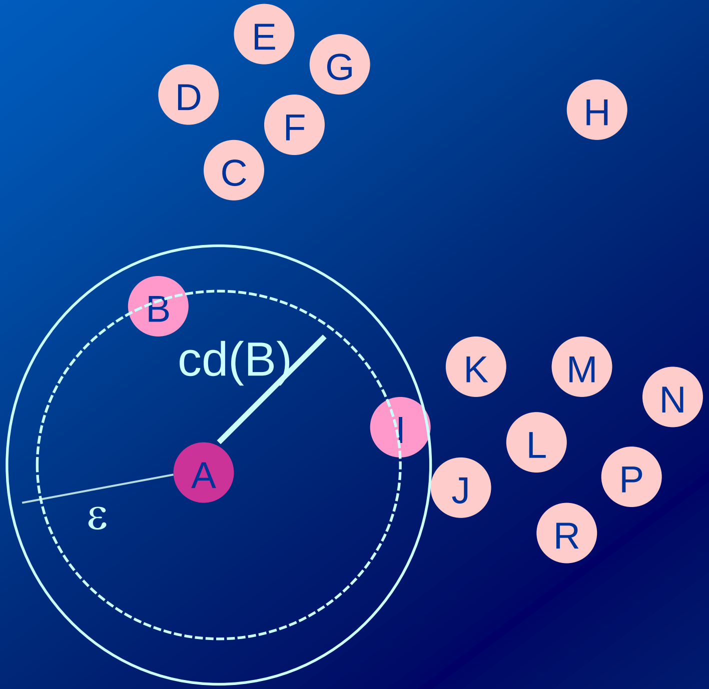


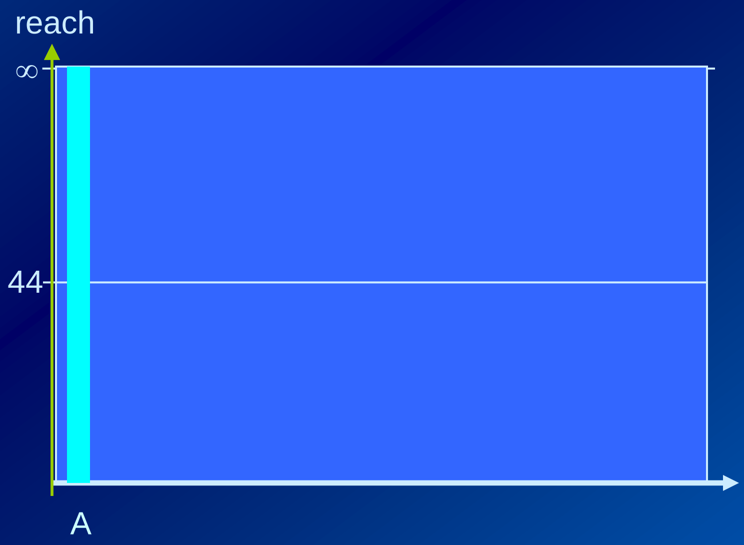
seedlist:

©*K. Gibert*

Example *[Pfeifle2004]*

- Example Database (2-dimensional, 16 points)
- $\varepsilon = 44$, $\upsilon = 3$

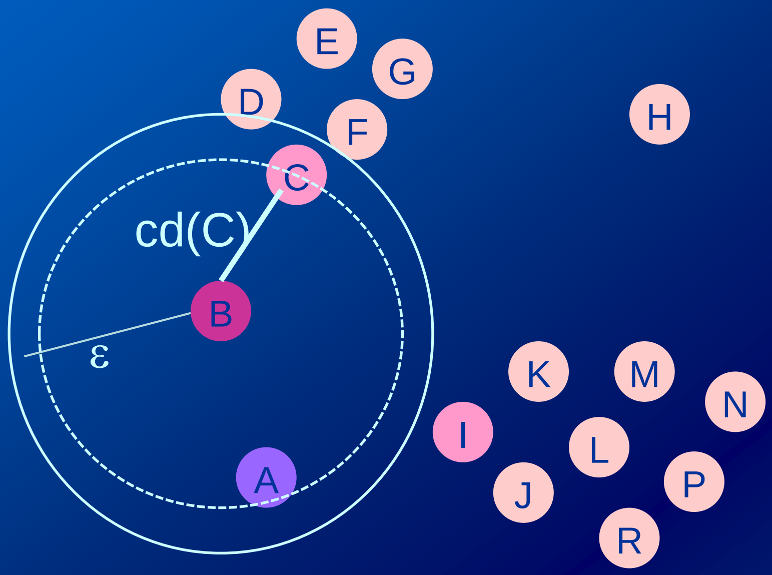


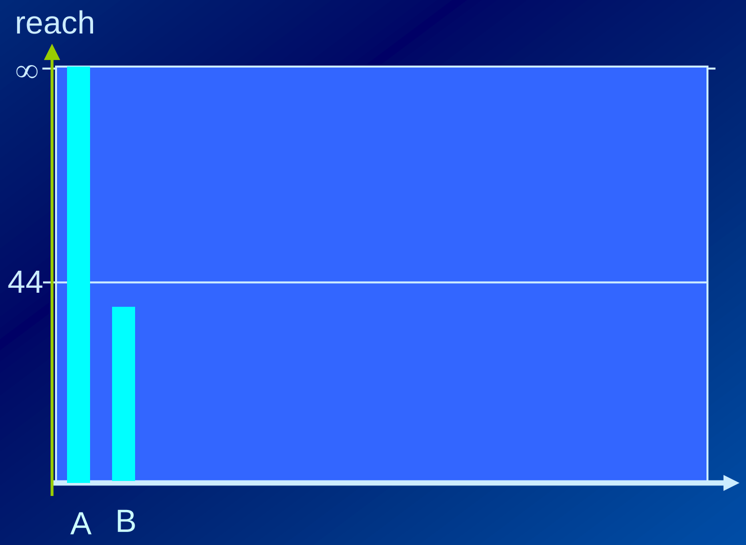


seedlist: (B,40) (I, 40)

Example *[Pfeifle2004]*

- Example Database (2-dimensional, 16 points)
- $\varepsilon = 44$, $\upsilon = 3$





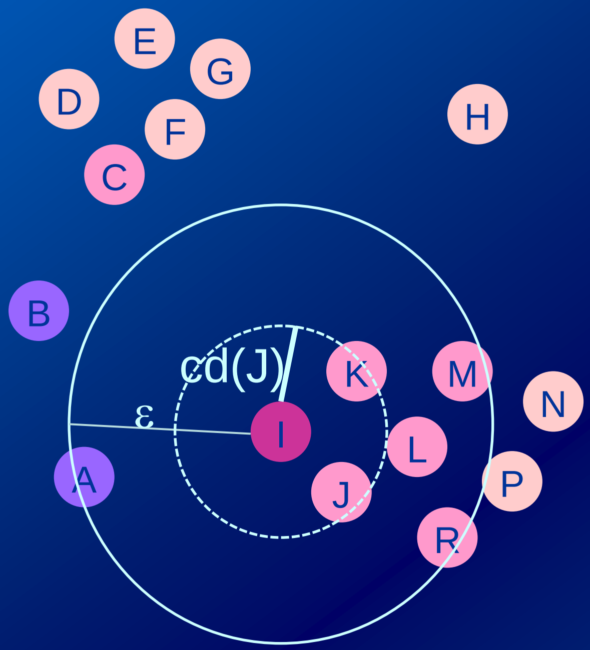
seedlist: (B,40) (I, 40)

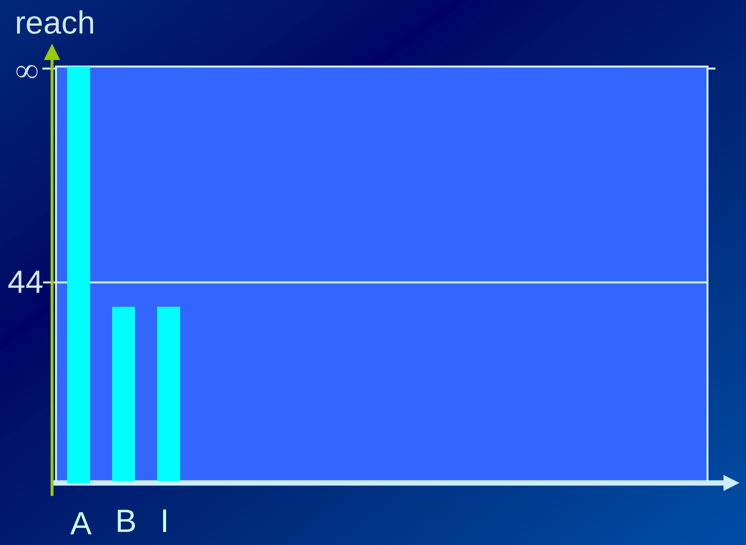
seedlist: (I, 40) (C,40)

©K. Gibert

Example *[Pfeifle2004]*

- Example Database (2-dimensional, 16 points)
- $\varepsilon = 44$, $\upsilon = 3$





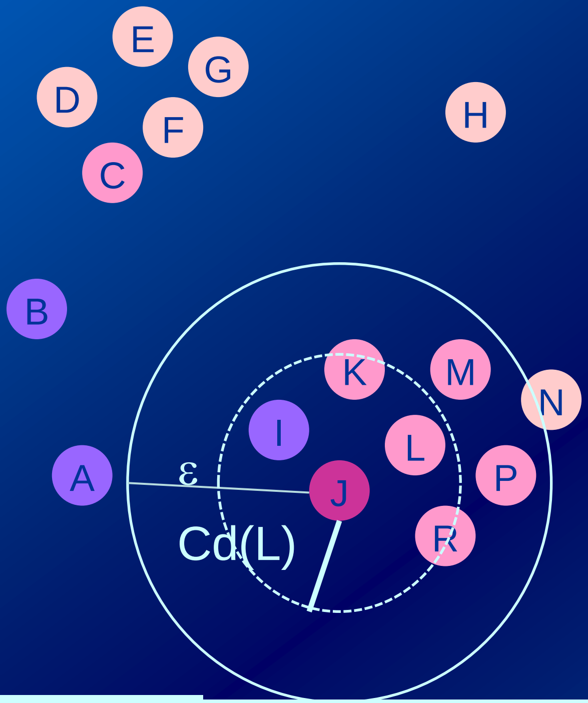
seedlist: (I, 40) (C,40)

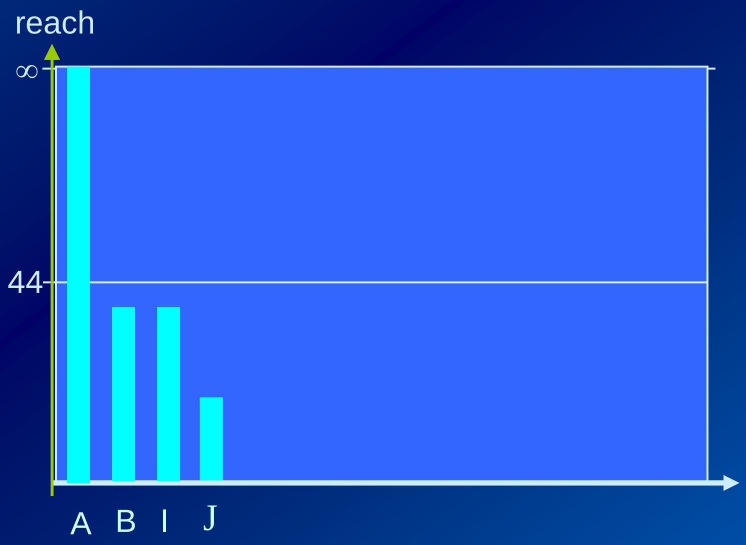
seedlist: (J, 20) (K, 20) (L,31)(C,40) (M, 40) (R, 43)

©K. Gibert

Example *[Pfeifle2004]*

- Example Database (2-dimensional, 16 points)
- $\varepsilon = 44$, $\upsilon = 3$





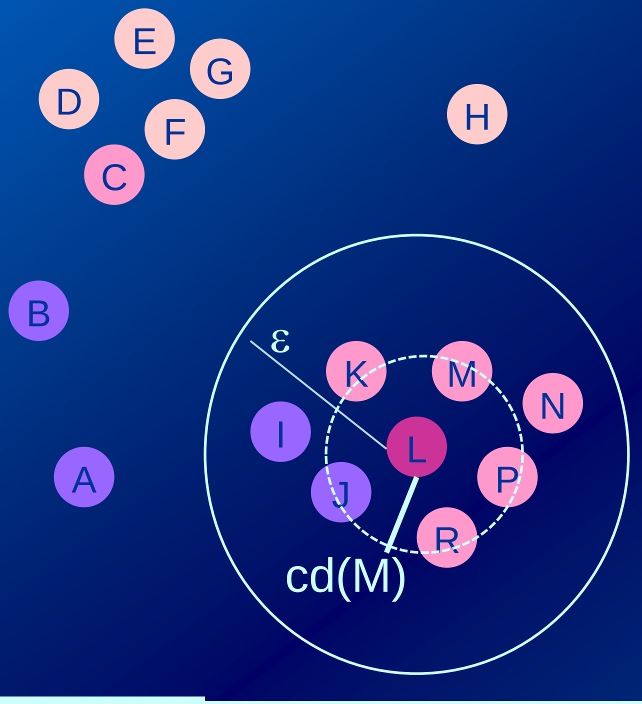
seedlist: (J, 20) (K, 20) (L,31)(C,40) (M, 40) (R, 43)

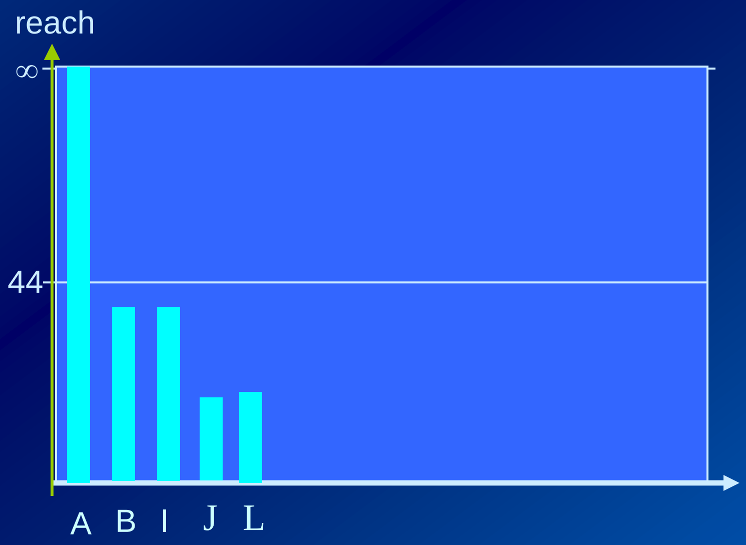
seedlist: (L, 19) (K, 20) (R, 21) (M, 30) (P, 31) (C, 40)

©K. Gibert

Example *[Pfeifle2004]*

- Example Database (2-dimensional, 16 points)
- $\varepsilon = 44$, $\upsilon = 3$



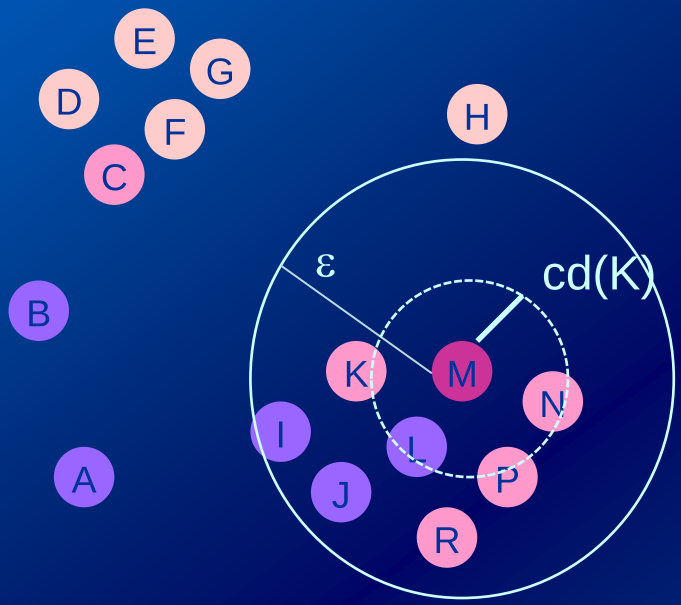


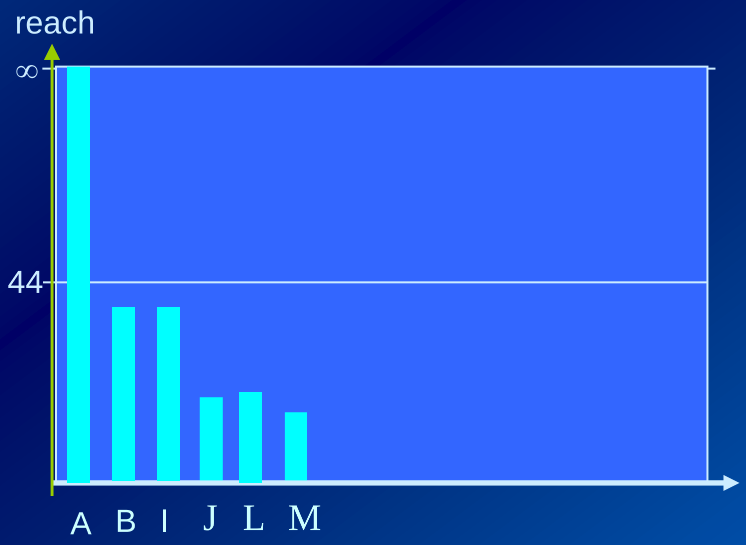
seedlist: (L, 19) (K, 20) (R, 21) (M, 30) (P, 31) (C, 40)

seedlist: (M, 18) (K, 18) (R, 20) (P, 21) (N, 35) (C, 40)

Example *[Pfeifle2004]*

- Example Database (2-dimensional, 16 points)
- $\varepsilon = 44$, $\upsilon = 3$



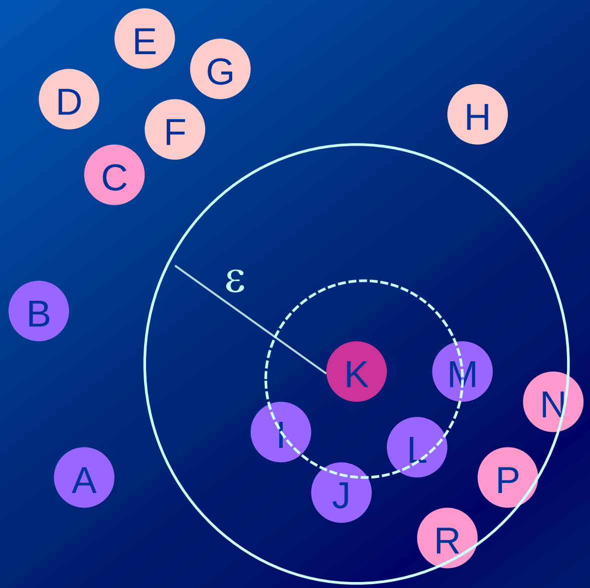


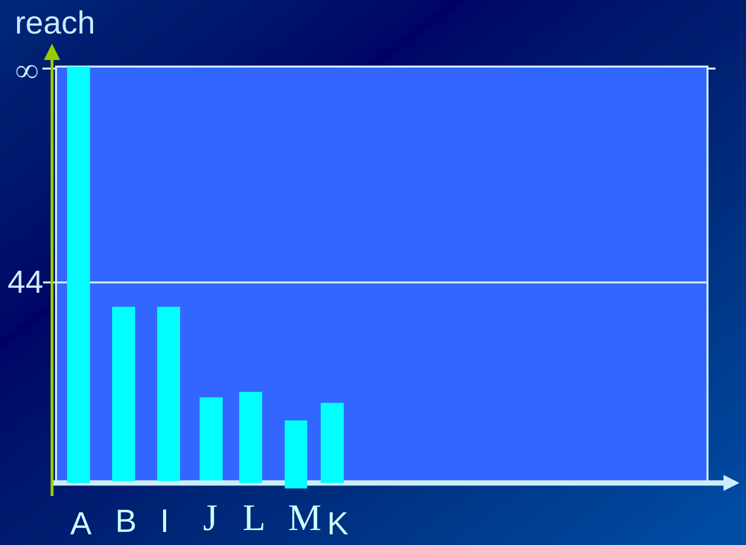
seedlist: (M, 18) (K, 18) (R, 20) (P, 21) (N, 35) (C, 40)

seedlist: (K, 18) (N, 19) (P, 20) (R, 40) (C, 40)

Example *[Pfeifle2004]*

- Example Database (2-dimensional, 16 points)
- $\varepsilon = 44$, $\upsilon = 3$



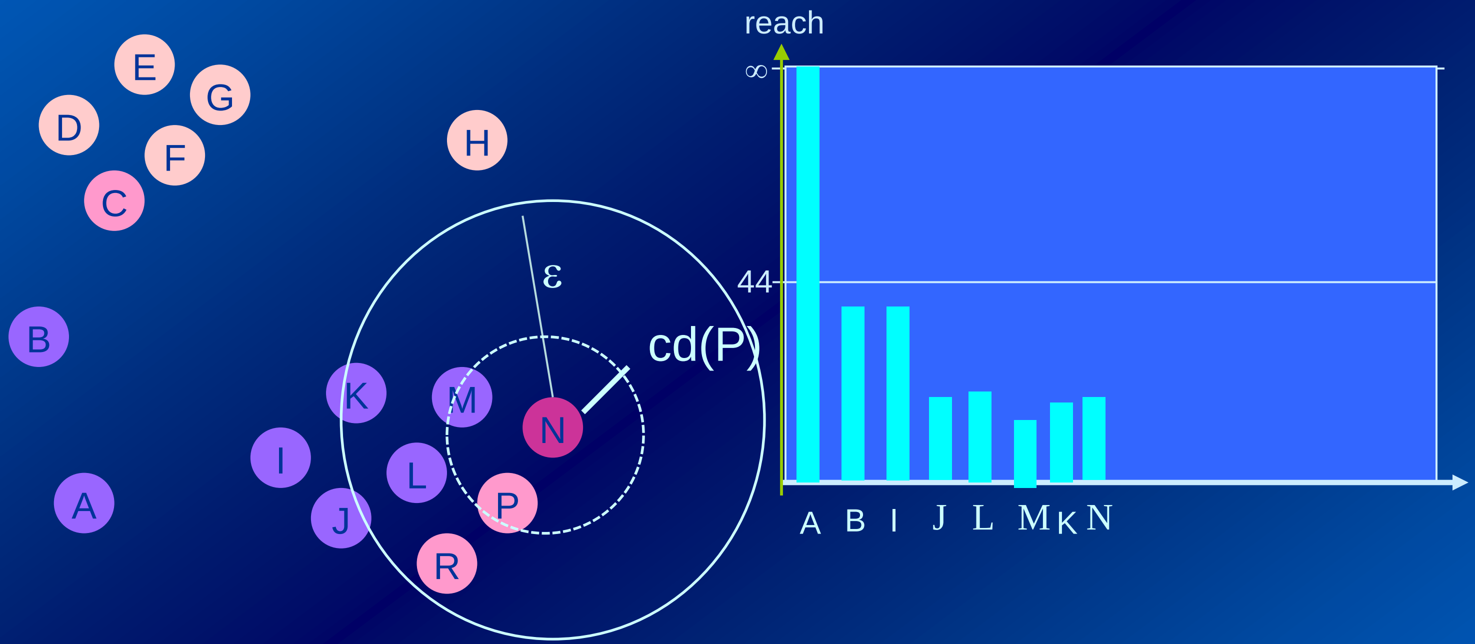


seedlist: (K, 18) (N, 19) (P, 20) (R, 40) (C, 40)

seedlist: (N, 19) (P, 20) (R, 20) (C, 40)

Example *[Pfeifle2004]*

- Example Database (2-dimensional, 16 points)
- $\varepsilon = 44$, $\upsilon = 3$



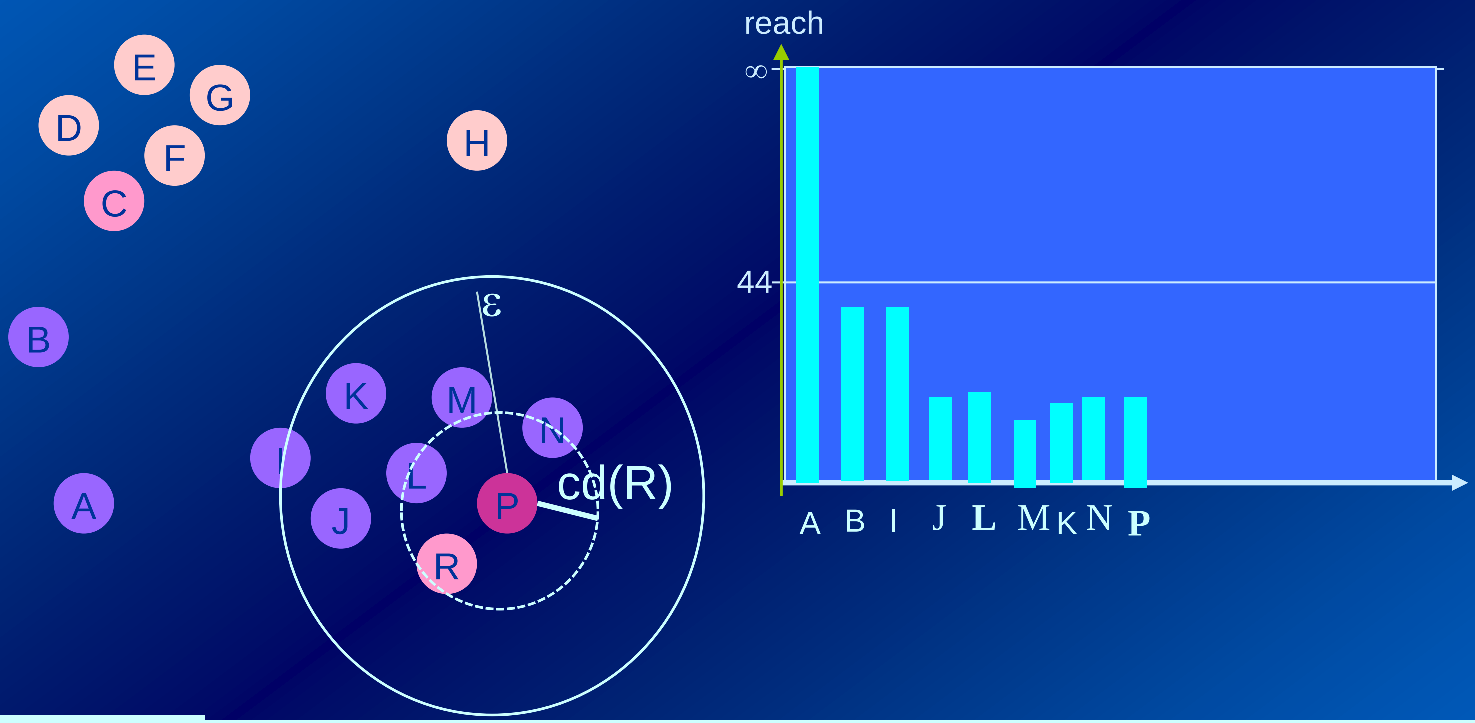
seedlist: (N, 19) (P, 20) (R, 20) (C, 40)

seedlist: (P, 20) (R, 20) (C, 40)

©K. Gibert

Example *[Pfeifle2004]*

- Example Database (2-dimensional, 16 points)
- $\varepsilon = 44$, $\upsilon = 3$



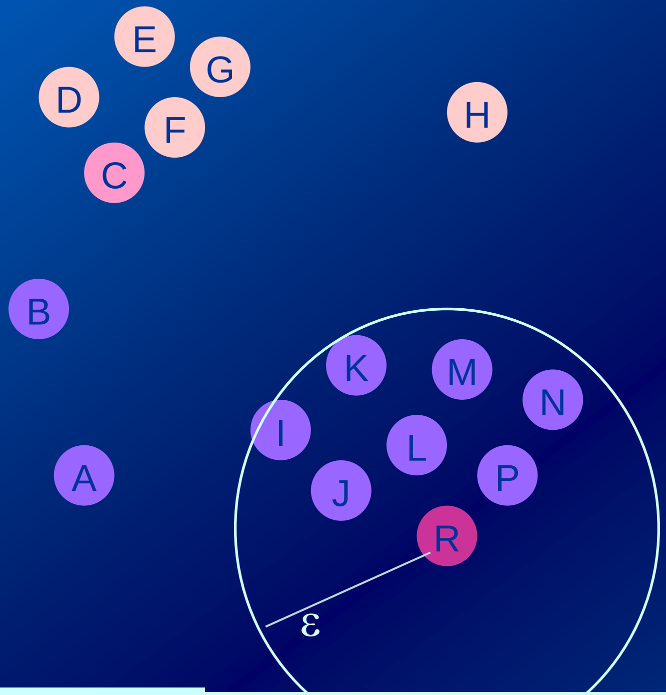
seedlist: (P, 20) (R, 20) (C, 40)

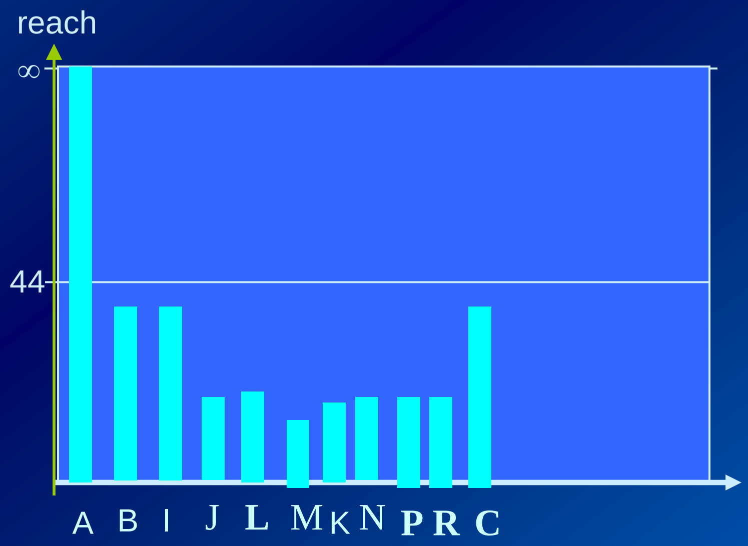
seedlist: (R, 20) (C, 40)

©K. Gibert

Example *[Pfeifle2004]*

- Example Database (2-dimensional, 16 points)
- $\varepsilon = 44$, $\upsilon = 3$



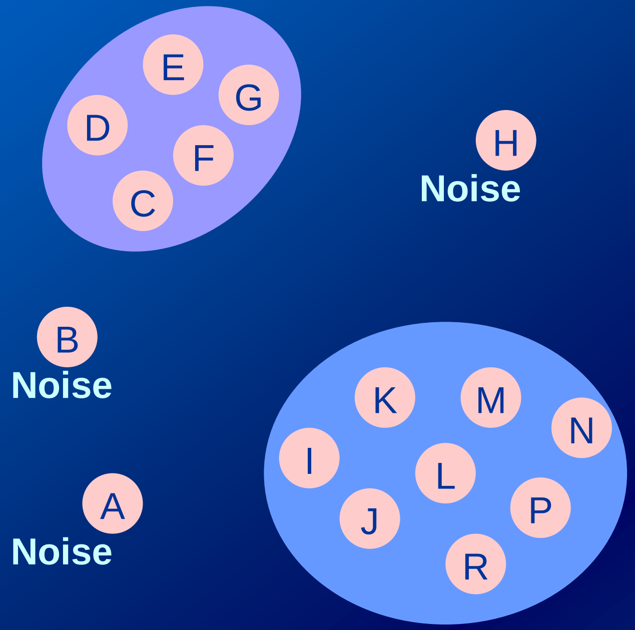


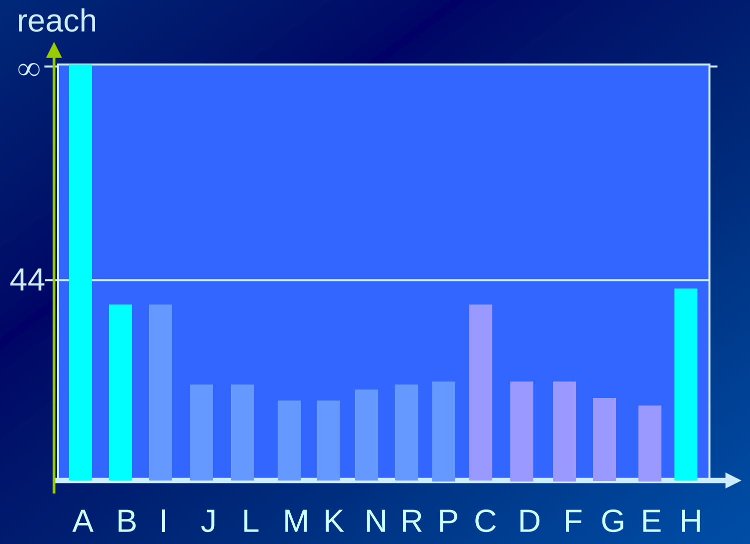
seedlist: (R, 20) (C, 40)

seedlist: (C, 40)

Example *[Pfeifle2004]*

- Example Database (2-dimensional, 16 points)
- $\varepsilon = 44$, $\upsilon = 3$





seedlist:

Other Scalable algorithms

- *BIRCH* (Zhang, Ramakrishnan et al. 1996) (Balanced Iterative Reducing and Clustering using Hierarchies) is an incremental and hierarchical clustering algorithm for very large databases. The two main building components in the Birch algorithm are a hierarchical clustering component, and a main memory structure component. Birch uses a **main memory** (of limited size) data structure called *CF tree*. The tree is organized in such a way that (i) the leafs contain actual clusters, and (ii) the size of any cluster in a leaf is not larger than *R*. Initially, the data points are in one cluster. As the data arrives, a check is made whether the size of the cluster does not exceed *R*. If the cluster size grows too big, the cluster is split into two clusters, and the points are redistributed. The points are then continuously inserted to the cluster which enlarges less. At each node of the tree the CF tree keeps information about the mean of the cluster, and the mean of the sum of squares to compute the size of the clusters efficiently. The tree structure also depends on the branching parameter *T*, which determines the maximum number of children each node can have.
- *ROCK (Guha, Rastogi et al. 2000)* (The RObust Clustering using linKs) clustering algorithm is based on links between data points, instead of distances when it merges clusters. These links represent the relation between a pair of objects and their common neighbours. The notion of links between data helps to overcome the problems with distance based coefficients. For this reason, this method is extended to non-metric similarity measures that are relevant in situations where a domain expert/similarity table is the only source of knowledge. ROCK works with categorical features.
- *RCH (* Relative hierarchical clustering) considers both the internal distance (distance between a pair of clusters which may be merged to yield a new cluster) and the external distance (distance from the two clusters to the rest), and uses their ratio to decide the proximities (Mollineda and Vidal 2000).
- SBAC (similarity-based agglomerative clustering) which was developed by Li and Biswas (Li and Biswas 1999) extends agglomerative clustering techniques to deal with both numeric and nominal data. It employs a mixed data measure scheme that pays extra attention to less common matches of feature values (Li and Biswas 2002).
-

References

[Ankerst 99] Mihael Ankerst, Markus M. Breunig, Hans-Peter Kriegel, Jörg Sander (1999). "OPTICS: Ordering Points To Identify the Clustering Structure". ACM SIGMOD international conference on Management of data. ACM Press. pp. 49–60.

[Ester 96]Martin Ester, Hans-Peter Kriegel, Jörg Sander, Xiaowei Xu (1996-). "A density-based algorithm for discovering clusters in large spatial databases with noise". In Evangelos Simoudis, Jiawei Han, Usama M. Fayyad. Proceedings of the Second International Conference on Knowledge Discovery and Data Mining (KDD-96). AAAI Press. pp. 226–231.

[Fayyad et al., 1996] U.M. Fayyad, G. Piatetsky-Shapiro, P. Smyth & R. Uthurusamy. Advances in Knowledge Discovery and Data Mining. AAAI Press / MIT Press, 1996.

[Gibe 1996] K. Gibert. Inferència. Centre de Publicacions d'Abast, BCN, 1996

[Gibe96b] Gibert, K. and Cort\'es, U. Weighing quantitative and qualitative variables in clustering methods Mathware and Soft Computin 4(3) 251-266 [Gibert 1999] K. Gibert, Z. Sonicki Clustering based on rules and medical research. Journal on Applied Stochastic Models in Business and Industry, form. JASMDA 15(4):319—324, Wiley, 1999

[Gibert 2008] Gibert K, J. Spate, M. Sànchez-Marrè, I. Athanasiadis, J. Comas (Data Mining for Environmental Systems. In Environmental Modeling, Software and Decision Support. State of the art and New Perspectives. IDEA Series v3 (Jackeman, A. J., Voinov, A., Rizzoli, A., and Chen, S. eds), pp 205-228. Elsevier NL.

[Gibert 2008b] K. Gibert, A. García-Rudolph, G. Rodríguez-Silva: The role of KDD Support-Interpretation tools in the conceptualization of medical profiles: An application to neurorehabilitation. Acta Informatica Medica 16(4):178-182

\revi{Knowledge Discovery about Quality of Life changes of Spinal Cord Injury patients: Clustering based on rules by States}{ 579--583}{Studies in Health Technology and Informatics}{150}{Aug 2009}{IOSSPress}{K. Gibert, A. García-Rudolph, L. Curcoll, D. Soler, L. Pla, J. M. Tormos}{09

[Gibert 2010] K. Gibert, M. Sànchez-Marrè, V. Codina. Choosing the right data mining technique: classification of methods and intelligent recommenders. Proc. of the IEMSs'10, 5th biennial meeting (III DMTES Workshop), S23.03.1-S23.03.9, 2010.

[Gibert2010b] Gibert K, Rodríguez-Silva G, Rodríguez-Roda I, 2010: Knowledge Discovery with Clustering based on rules by States: A water treatment application. *Environmental Modelling&Software* 25:712-723

[Gibert 2012] K. Gibert. Mixed Intelligent-Multivariate Missing Inputation. Int'l Journal of Computer Mathematics (in press)

[Gibert 2012b] K. Gibert, D. Conti, D. Vrecko : Assisting the end-user in the interpretation of profiles for decision support. An application to wastewater treatment plants. Environmental Engineering and Management Journal 11(5): 931-944

[GowdaDiday92] Diday, E. and Gowda, K.C. Symbolic clustering using a new similaritiy measure IEEE Trans. on systems, mans,and cybernetics 22(2) 368-378, 1992

[Gower 1971] Gower, J.C 1971: A General coefficient of similarity and some of its properties. Biometrics 27, 857--874

Guha, Rastogi et al. 2001

[Gleick 86] Gleick J: Hole in ozone over south pole worries scientists, The NY Times, July 29th, 1986

[Hammond 2004] Hammond M. The Fact Gap: The disconnect between data and decisions. Business objects 2004

[Ichino94] Ichino, M. and Yaguchi, H Generalized Minkowski Metrics for Mixed feature-type data analysis IEEE Tr. on SMC 22(2) 146-153, 1994

[Kanevski12] Kanevski, M. Multitask learning of Environmental Spatial Data. In procs iEMSs 2012, Leipzig 2012.

[Kanevski et al 09] Kanevski, Poudzunov, Timorin: Machine learning for spatial environmental data, EPFL Press, 2009.

[Karypis Han 99] A hierarchical clustering algorithm using dynamic modelling. IEEE computer 32(8):68-75, 1999

[Lebart90] Lebart 1990: Traitement statistique des donnèes. DUNOD, Paris

[Little88] Little RJA, A test of missing completely at random form multivariate data with missing values. Journal of American Statistical Association 88; 83:1198-1202

[McKinsey 2011] Big data: The next frontier for innovation, competition and productivity. Executive report, MaKinsey global institute, May 2011.

[Pfeifle2004] ICDM 2004, Brighton

[Nanni, 2006] Nanni, Pedreschi: Time-focused clustering of trajectories of moving objects. JIIS, 27(3) 267-289, 2006

[Ralambondrainy 1995] Ralambondrainy, H A conceptual version of the K-means algorithm Lifetime Learning Publications, Belmont, California, 1995

[Simon, 1960] H. A. Simon. *The New Science of Management Decision*. New York: Harper & Row, 1960.

[Tukey77] Tukey JW : Exploratory Data Analysis, Addison-Wesley, 1977