Methods for Intelligent Systems

Lecture Notes on Clustering (V)
2009-2010

Davide Eynard

eynard@elet.polimi.it

Department of Electronics and Information

Politecnico di Milano

Course Schedule [*Tentative*]

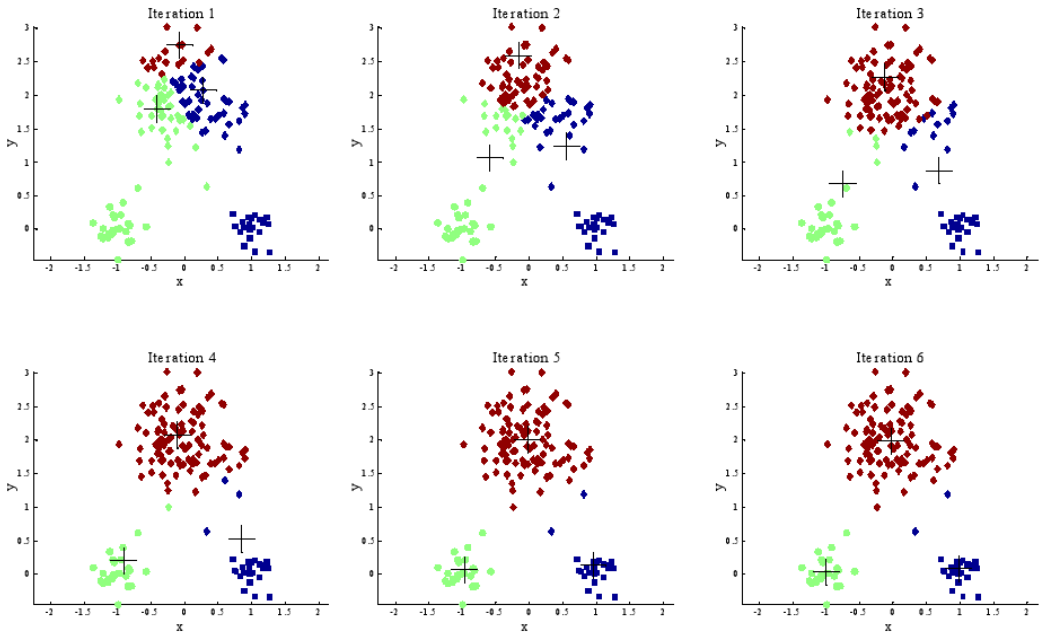
Date	Topic
11/03/2010	Clustering: Introduction
18/03/2010	Clustering: K-means & Hierarchical
25/03/2010	Clustering: Fuzzy, Gaussian & SOM
08/04/2010	Clustering: PDDP & Vector Space Model
15/04/2010	Clustering: Limits, DBSCAN & Jarvis-Patrick
29/04/2010	Clustering: Evaluation Measures

Lecture outline

- K-Means limits
- Hierarchical algorithms limits
- DBSCAN
- Jarvis-Patrick

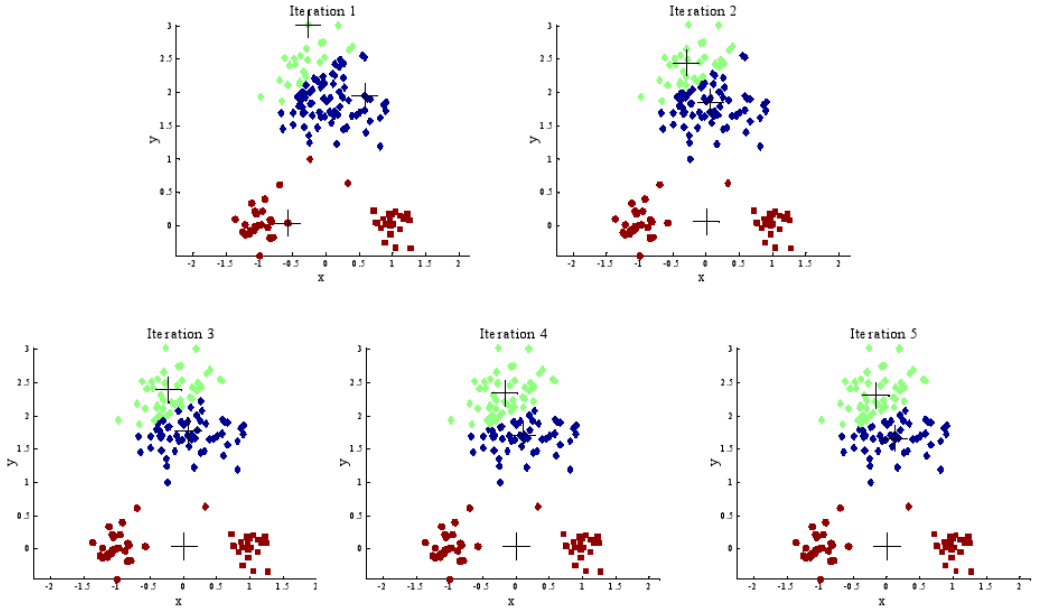
K-Means limits

Importance of choosing initial centroids



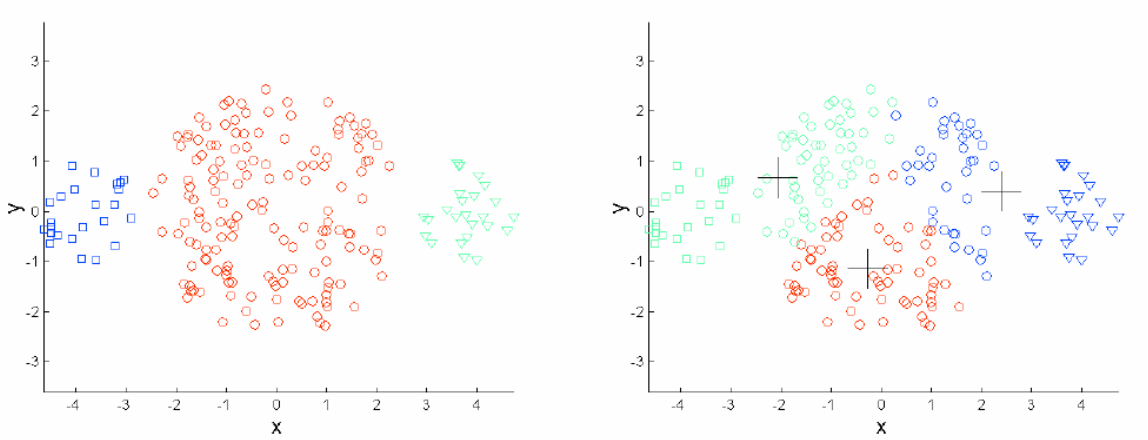
K-Means limits

Importance of choosing initial centroids



K-Means limits

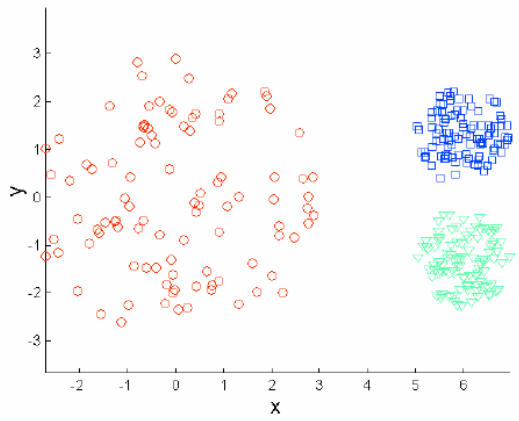
Differing sizes



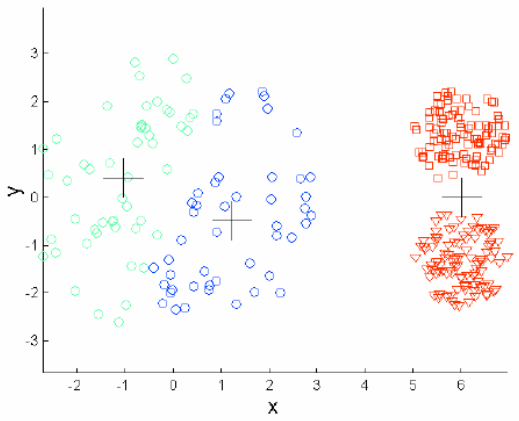
Original Points **K-means Clusters**

K-Means limits

Differing density



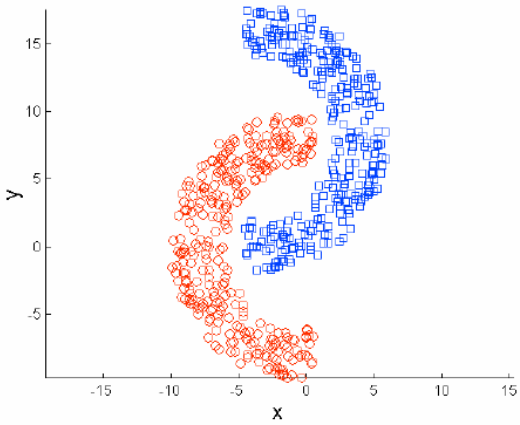
Original Points



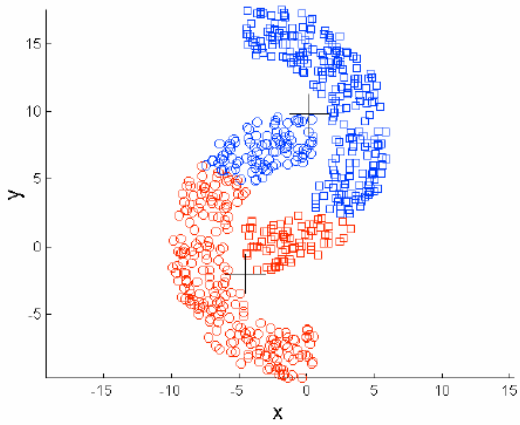
K-means Clusters

K-Means limits

Non-globular shapes



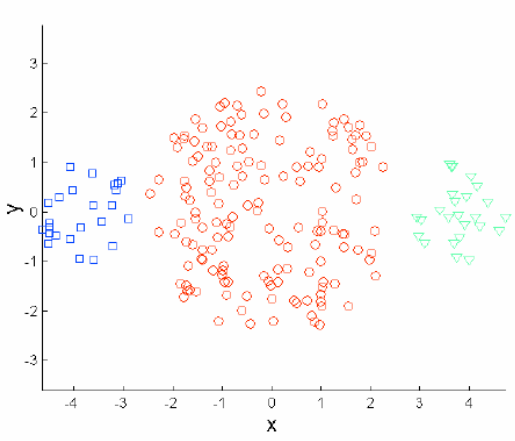
Original Points



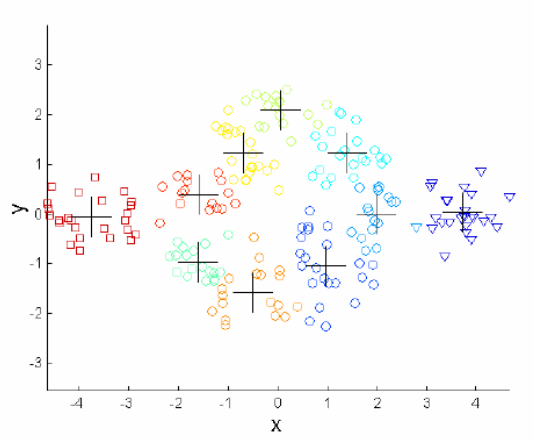
K-means Clusters

K-Means: higher K

What if we tried to increase K to solve K-Means problems?



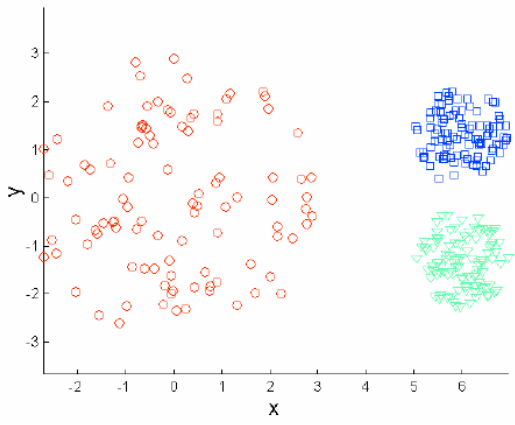
Original Points



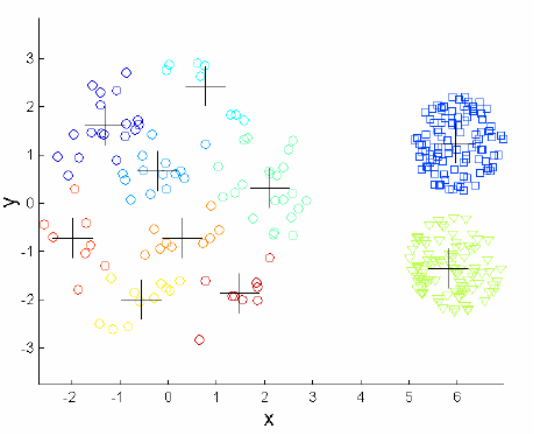
K-means Clusters

K-Means: higher K

What if we tried to increase K to solve K-Means problems?



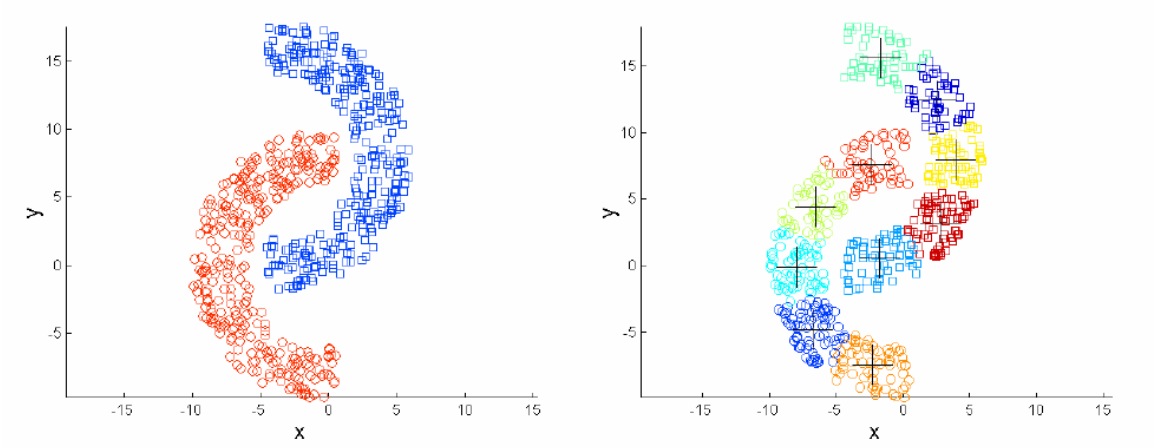
Original Points



K-means Clusters

K-Means: higher K

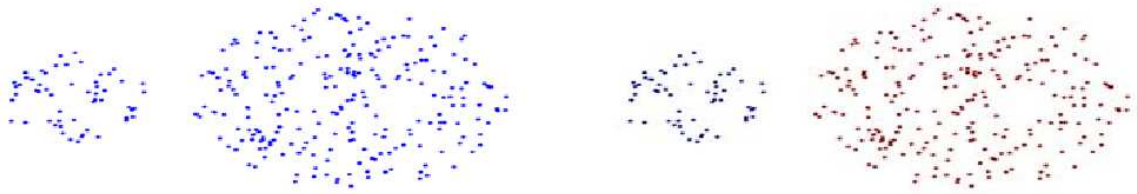
What if we tried to increase K to solve K-Means problems?



Original Points **K-means Clusters**

Hierarchical algorithms limits

Strength of MIN

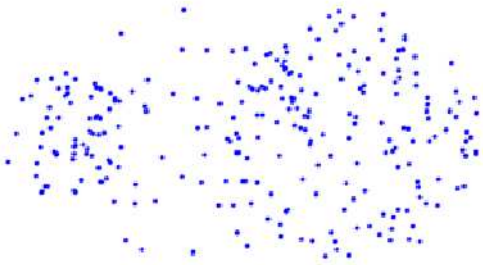


Original Points **Two Clusters**

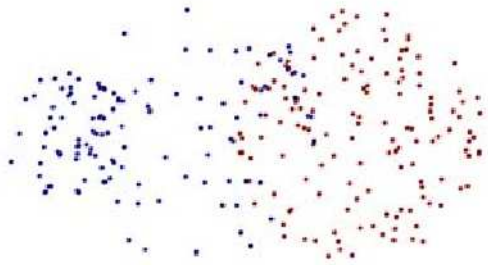
- Easily handles clusters of different sizes
- Can handle non elliptical shapes

Hierarchical algorithms limits

Limitations of MIN



Original Points

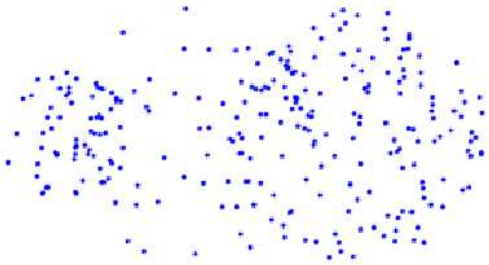


Two Clusters

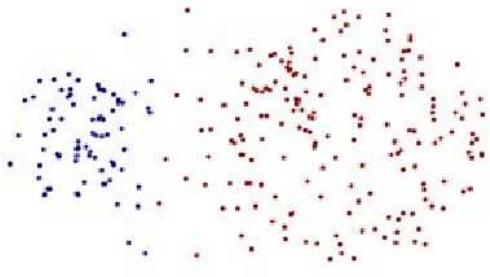
- Sensitive to noise and outliers

Hierarchical algorithms limits

Strength of MAX



Original Points

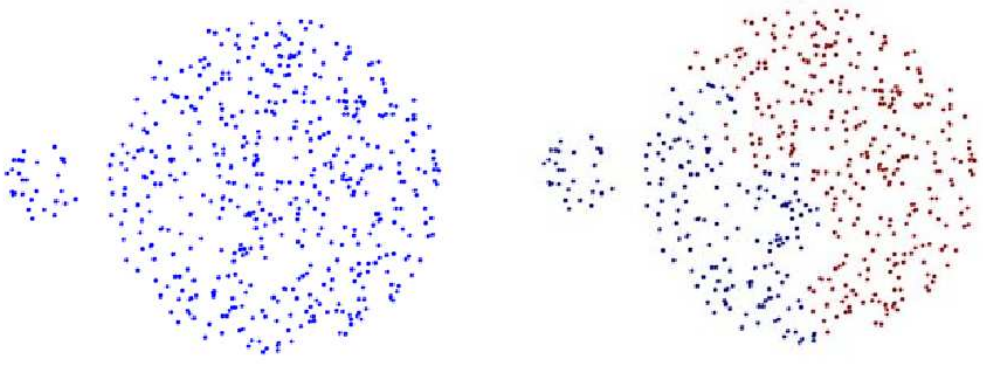


Two Clusters

- Less sensible to noise and outliers

Hierarchical algorithms limits

Limitations of MAX

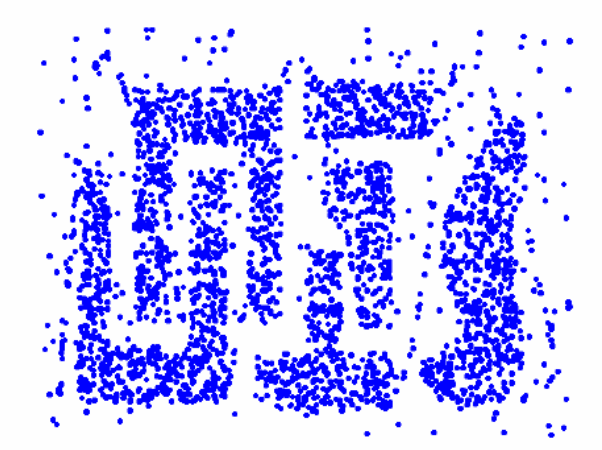


Original Points Two Clusters

- Tends to break large clusters
- Biased toward globular clusters

Question

What if we had a dataset like this?



DBSCAN

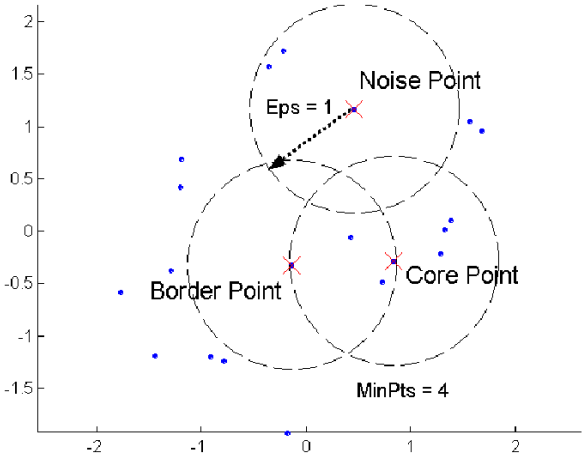
- Density Based Spatial Clustering of Applications with Noise
  - Data points are connected through *density*
- Finds clusters of arbitrary shapes
- Handles well noise in the dataset
- Single scan on all the elements of the dataset

DBSCAN: background

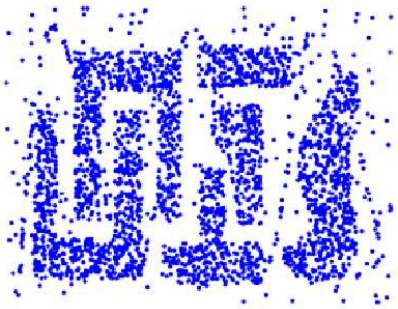
- Two parameters to define density:
  - Eps: radius
  - $MinPts$: minimum number of points within the specified radius
- Number of points within a specified radius:
  - $N_{Eps}(p) : \{q \in D | dist(p,q) \leq Eps\}$

DBSCAN: background

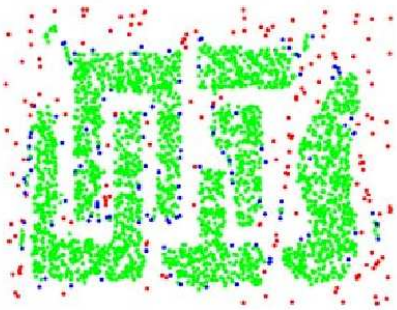
- A point is a **core point** if it has more than $MinPts$ points within Eps
- A **border point** has fewer than $MinPts$ within Eps, but is in the neighborhood of a core point
- A **noise point** is any point that is not a core point or a border point.



DBSCAN: core, border and noise points



Original Points



Point types: core, border and noise

$Eps = 10, MinPts = 4$

DBSCAN: background

- A point p is **directly density-reachable** from q with respect to $(Eps, MinPts)$ if:
  1. $p \in N_{Eps}(q)$
  2. q is a *Core* point

  (the relation is symmetric for pairs of core points)
- A point p is **density-reachable** from q if there is a chain of points $p_1, \ldots, p_n$ (where $p_1 = q$ and $p_n = p$) such that p_{i+1} is *directly density-reachable* from p_i for every i
  - (two border points might not be density-reachable)
- A point p is **density-connected** to q if there's a point o such that both p and q are *density-reachable* from o
  - (given two border points in the same cluster C, there must be a core point in C from which both border points are density-reachable)

DBSCAN: background

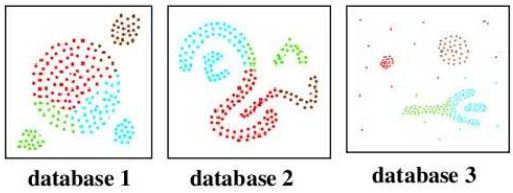
- Density-based notion of a cluster:
  - a cluster is defined to be a set of density-connected points which is maximal wrt. density-reachability
  - Noise is simply the set of points in the dataset D not belonging to any of its clusters

DBSCAN algorithm

- Eliminate noise points
- Perform clustering on the remaining points

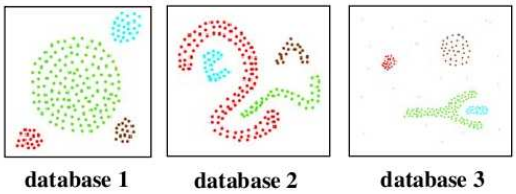
```
current_cluster_label ← 1
for all core points do
  if the core point has no cluster label then
    current_cluster_label ← current_cluster_label + 1
    Label the current core point with cluster label current_cluster_label
  end if
  for all points in the Eps-neighborhood, except i^th the point itself do
    if the point does not have a cluster label then
      Label the point with cluster label current_cluster_label
    end if
  end for
end for
```


DBSCAN evaluation



database 1 database 2 database 3

figure 5: Clusterings discovered by CLARANS

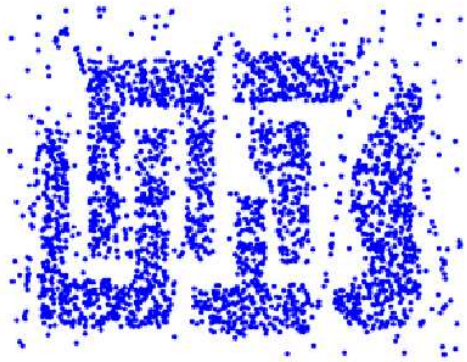


database 1 database 2 database 3

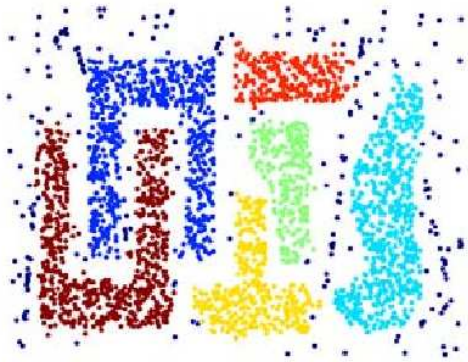
figure 6: Clusterings discovered by DBSCAN

- CLARANS, a K-Medoid algorithm, compared with DBSCAN

When DBSCAN works well



Original Points



Clusters

- Resistant to noise
- Can handle clusters of different shapes and sizes

Clustering using a similarity measure

- R.A. Jarvis and E.A. Patrick, 1973
- Many clustering algorithms are biased towards finding globular clusters. Such algorithms are not suitable for chemical clustering, where long "stringy" clusters are the rule, not the exception.
- To be effective for clustering chemical structures, a clustering algorithm must be self-scaling, since it is expected to find both straggly, diverse clusters and tight ones
- => Cluster data in a nonparametric way, when the globular concept of a cluster is not acceptable

Jarvis-Patrick

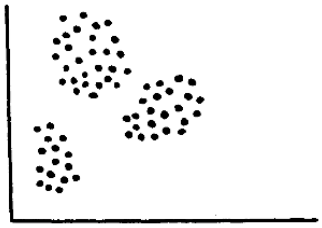


Fig. 1. Globular clusters.

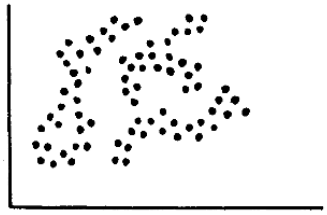
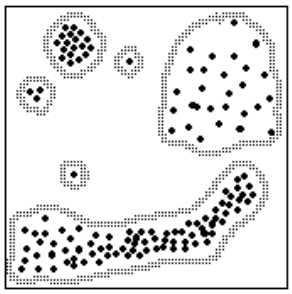


Fig. 2. Nonglobular clusters.

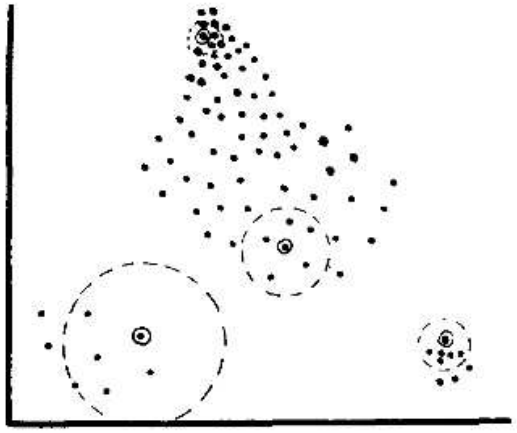


Jarvis-Patrick

- Let $x_1, x_2, \ldots, x_n$ be a set of data vectors in an L-dimensional Euclidean vector space
- Data points are similar to the extent that they share the same near neighbors
  - In particular, they are similar to the extent that their respective k nearest neighbor lists match
  - In addition, for this similarity measure to be valid, it is required that the tested points themselves belong to the common neighborhood

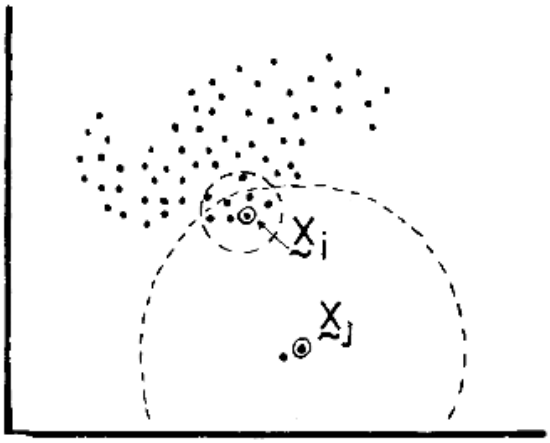
Jarvis-Patrick

Automatic scaling of neighborhoods (k=5)



Jarvis-Patrick

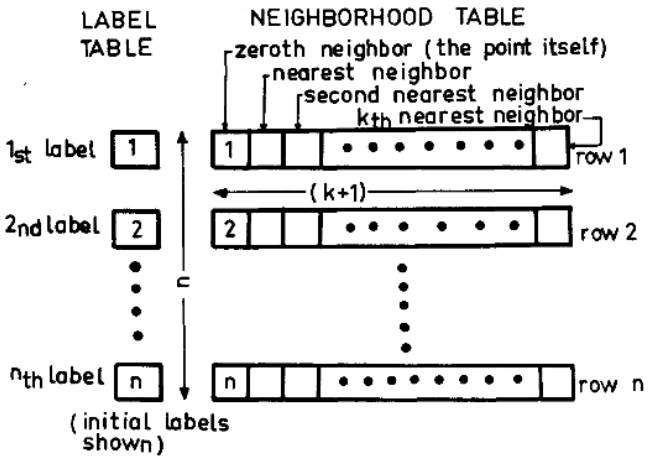
"Trap condition" for k=7: X_i belongs to X_j's neighborhood, but not vice versa.



JP algorithm

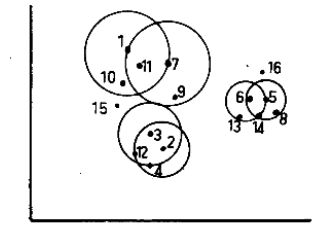
1. for each point in the dataset, list the k nearest neighbors by order number. Regard each point as its own zeroth neighbor. Once the neighborhood lists have been tabulated, the raw data can be discarded.
2. Set up an integer label table of length n, with each entry initially set to the first entry of the corresponding neighborhood row.
3. All possible pairs of neighborhood rows are tested as follows: replace both label entries by the smaller of the two existing entries if both 0th neighbors are found in both neighborhood rows and at least k_t neighbor matches exist between the two rows. Also, replace all appearances of the higher label (throughout the entire label table) with the lower label if the above test is successful.
4. The clusters under the k, k_t selections are now indicated by identical labeling of the points belonging to the clusters.

JP algorithm



JP: alternative approaches

Similarity matrix



(a)

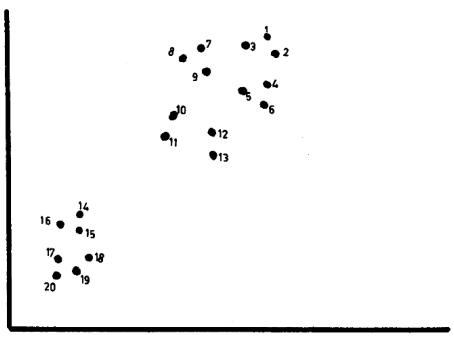
	①	②	③	④	⑤	⑥	⑦	ⓝ
①	4	0	0	0	0	0	3	
②	0	4	4	4	0	0	0	
③	0	4	4	4	0	0	0	
④	0	4	4	4	0	0	0	
⑤	0	0	0	0	4	3	0	
⑥	0	0	0	0	3	4	0	
⑦	3	0	0	0	0	0	4	
ⓝ								

(b)

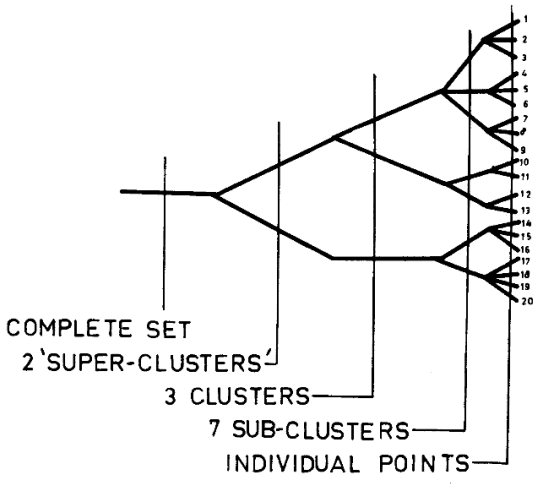
Fig. 8. Example of similarity matrix using the number of shared near neighbors as point pair similarity measure with equally weighted votes. (a) Sample points. (b) Similarity matrix for $k = 3$.

JP: alternative approaches

Hierarchical clustering - dendogram



(a)



(b)

JP: conclusions

Pros:

- The same results are produced regardless of input order
- It's a non-parametric method
- Parameters k, k_t can be adjusted to match a particular need
- Auto scaling is built into the method
- It will find tight clusters embedded in loose ones
- It is not biased towards globular clusters
- The clustering step is very fast
- Overhead requirements are relatively low

Cons:

- it requires a list of near neighbors which is computationally expensive to generate

Bibliography

- Slides about clustering for the Data Mining course prof. Salvatore Orlando (link)
- Tan, Steinbach, Kumar: "Introduction to Data Mining", Ch. 8 (link)
- Jarvis, Patrick: "Clustering Using a Similarity Measure Based on Shared Near Neighbors" (link)
- As usual, more info on del.icio.us