Machine Learning

Lecture Notes on Clustering (III)

2018-2019

Davide Eynard

davide.eynard@usi.ch

Institute of Computational Science

Università della Svizzera italiana

Lecture outline

- Gaussian Mixtures
- DBSCAN
- Jarvis-Patrick

Mixture of Gaussians

Clustering as a Mixture of Gaussians

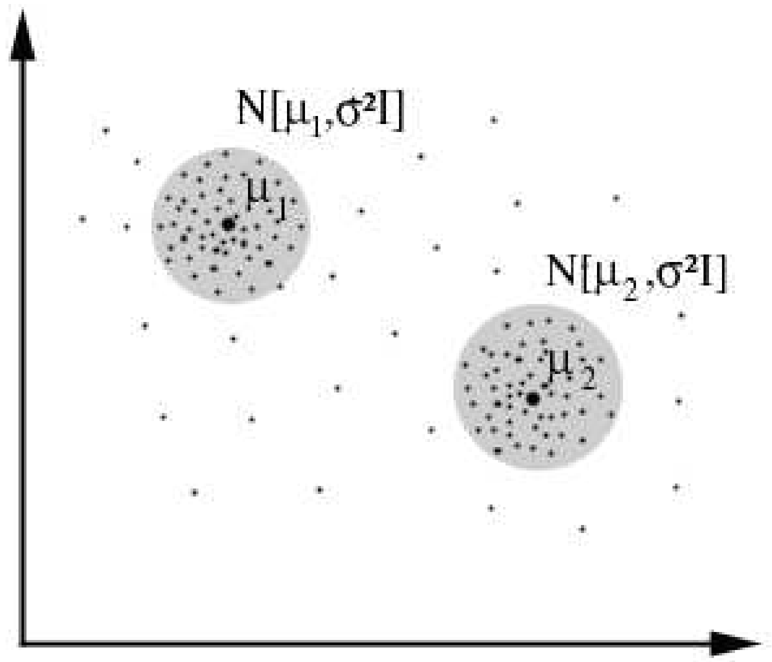
- Gaussians Mixture is a *model-based* clustering approach
    - It uses a statistical model for clusters and attempts to optimize the fit between the data and the model.
    - Each cluster can be mathematically represented by a parametric distribution, like a Gaussian (continuous) or a Poisson (discrete)
    - The entire data set is modelled by a *mixture* of these distributions
- A mixture model with high likelihood tends to have the following traits:
    - Component distributions have high "peaks" (data in one cluster are tight)
    - The mixture model "covers" the data well (dominant patterns in data are captured by component distributions)

Advantages of Model-Based Clustering

- well studied statistical inference techniques available
- flexibility in choosing the component distribution
- obtain a density estimation for each cluster
- a "soft" classification is available

Mixture of Gaussians

It is the most widely used model-based clustering method: we can actually consider clusters as Gaussian distributions centered on their barycentres (as we can see in the figure, where the gray circle represents the variance of the distribution).



How does it work?

- it chooses the component (the Gaussian) at random with probability $P(\omega_i)$
- it samples a point $N(\mu_i, \sigma^2 I)$
  - Let's suppose we have $x_1, x_2, \ldots, x_n$ and $P(\omega_1), \ldots, P(\omega_K), \sigma$
  - We can obtain the likelihood of the sample: $P(x|\omega_i, \mu_1, \mu_2, \ldots, \mu_K)$ (probability that an observation from class ω_i would have value x given class means $\mu_1, \ldots, \mu_K$)
  - What we really want is to maximize $P(x|\mu_1, \mu_2, \ldots, \mu_K)$

... Can we do it? How?

(let's first look at some examples on *Expectation Maximization*...)

The Algorithm

The algorithm is composed of the following steps:

1. Initialize parameters:

$$\lambda_0 = \{\mu_1^{(0)}, \mu_2^{(0)}, \ldots, \mu_k^{(0)}, p_1^{(0)}, p_2^{(0)}, \ldots, p_k^{(0)}\}$$

   where $p_i^{(t)}$ is shorthand for $P(\omega_i)$ at t-th iteration

2. E-step:

$$P(\omega_j|x_k, \lambda_t) = \frac{P(x_k|\omega_j, \lambda_t)P(\omega_j|\lambda_t)}{P(x_k|\lambda_t)} = \frac{P(x_k|\omega_i, \mu_i^{(t)}, \sigma^2)p_i(t)}{\sum_k P(x_k|\omega_j, \mu_j^{(t)}, \sigma^2)p_j^{(t)}}$$

3. M-step:

$$\mu_i^{(t+1)} = \frac{\sum_k P(\omega_i|x_k, \lambda_t)x_k}{\sum_k P(\omega_i|x_k, \lambda_t)}$$

$$p_i^{(t+1)} = \frac{\sum_k P(\omega_i|x_k, \lambda_t)}{R}$$

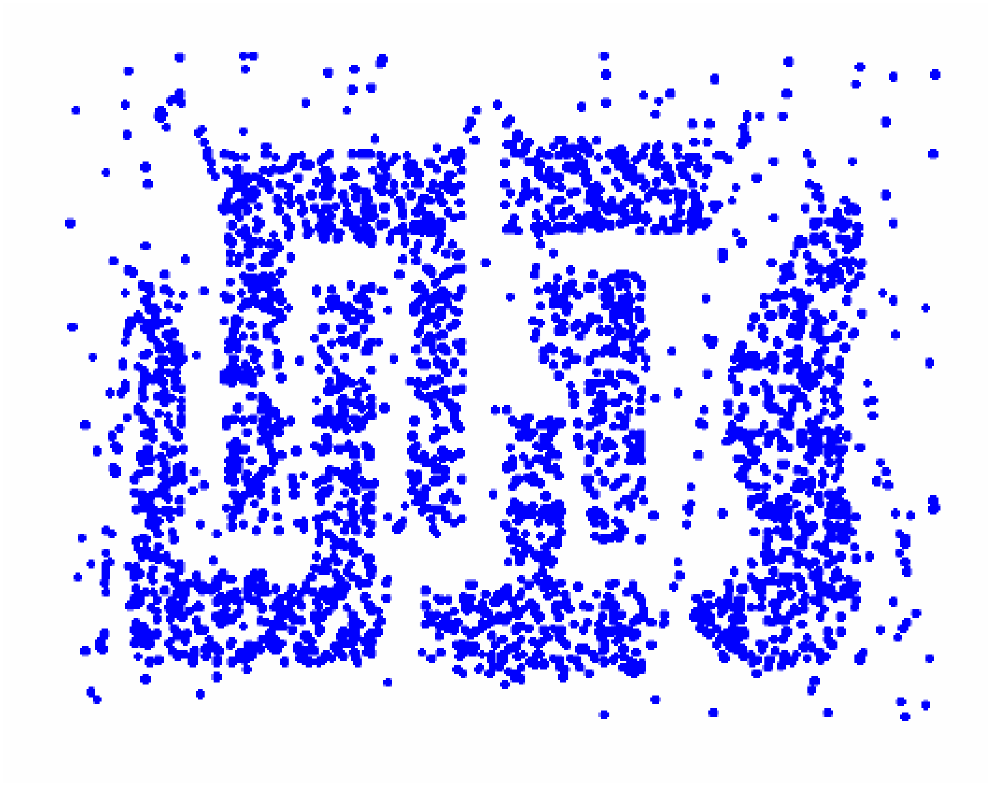
   where R is the number of records

Mixture of Gaussians Demo

Time for a demo!

Question

What if we had a dataset like this?



DBSCAN

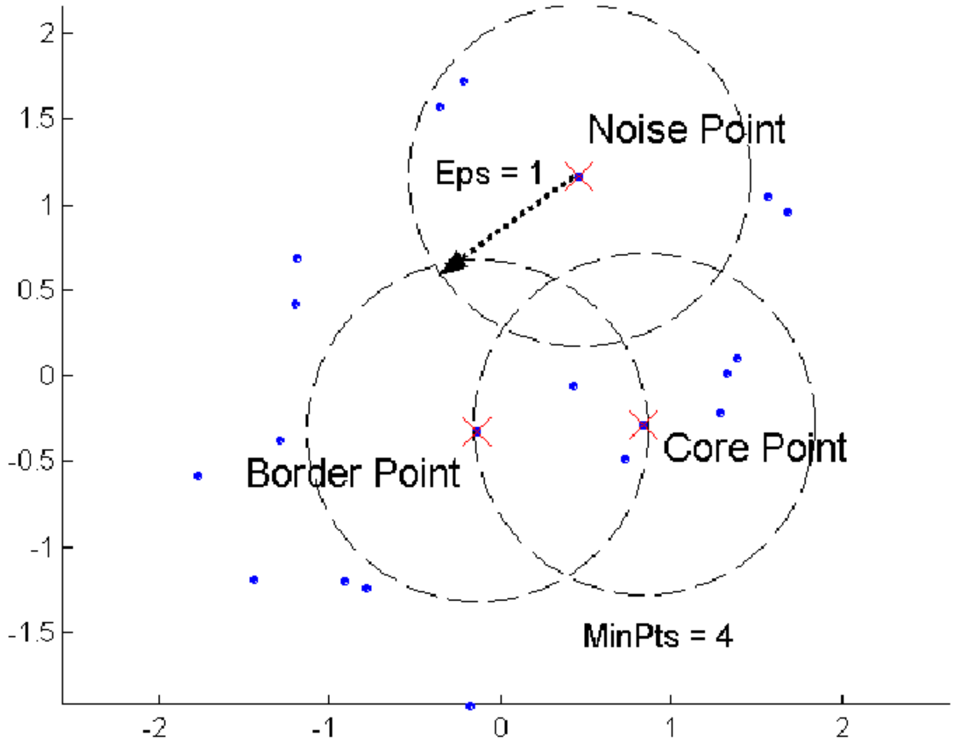
- Density Based Spatial Clustering of Applications with Noise
  - Data points are connected through *density*
- Finds clusters of arbitrary shapes
- Handles well noise in the dataset
- Single scan on all the elements of the dataset

DBSCAN: background

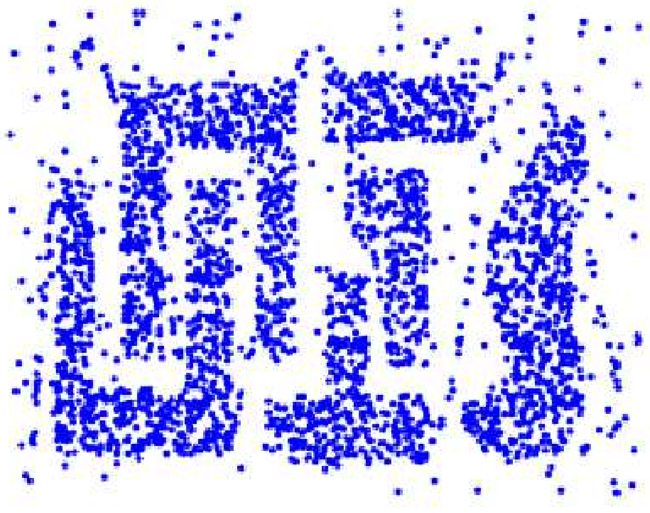
- Two parameters to define density:
  - Eps: radius
  - $MinPts$: minimum number of points within the specified radius
- Number of points within a specified radius:
  - $N_{Eps}(p) : \{q \in D | dist(p, q) \leq Eps\}$

DBSCAN: background

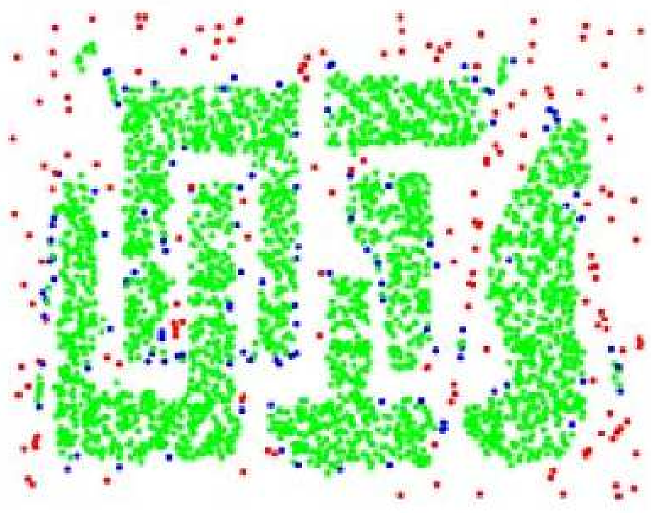
- A point is a **core point** if it has more than $MinPts$ points within Eps
- A **border point** has fewer than $MinPts$ within Eps, but is in the neighborhood of a core point
- A **noise point** is any point that is not a core point or a border point.



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

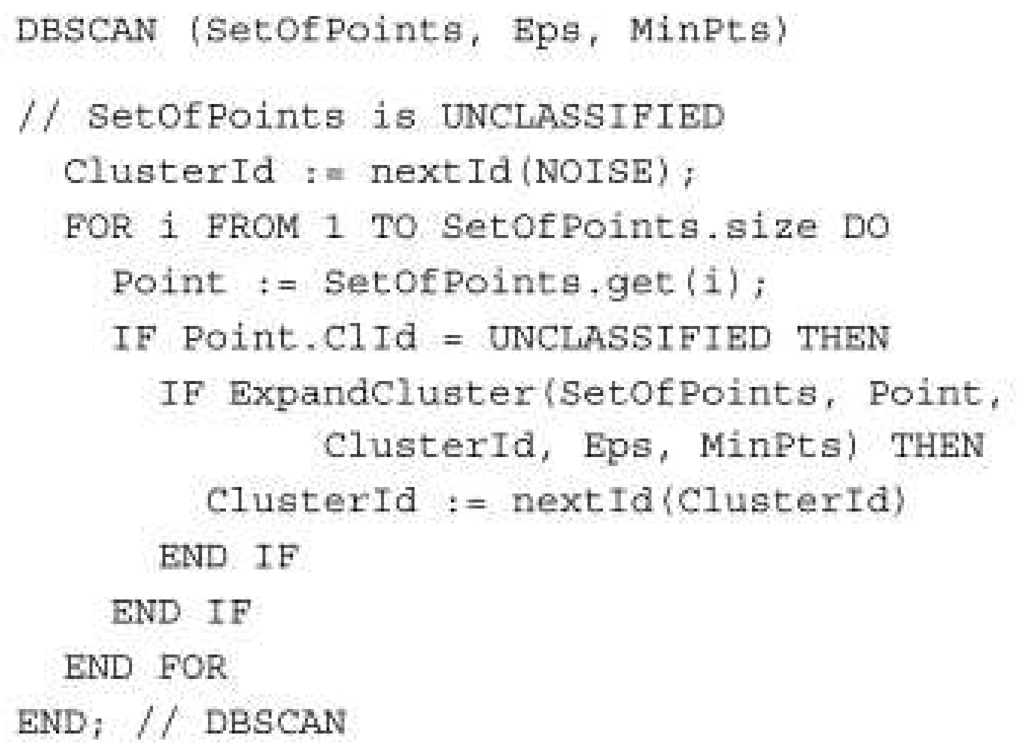
  (the relation is symmetric only for pairs of core points)
- A point p is **density-reachable** from q if there is a chain of points $p_1, \ldots, p_n$ (where $p_1 = q$ and $p_n = p$) such that p_{i+1} is *directly density-reachable* from p_i for every i
  - (the relation is transitive, but symmetric only for core points)
- 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



```
DBSCAN (SetOfPoints, Eps, MinPts)

// SetOfPoints is UNCLASSIFIED
  ClusterId := nextId(NOISE);
  FOR i FROM 1 TO SetOfPoints.size DO
    Point := SetOfPoints.get(i);
    IF Point.ClId = UNCLASSIFIED THEN
      IF ExpandCluster(SetOfPoints, Point,
                ClusterId, Eps, MinPts) THEN
        ClusterId := nextId(ClusterId)
      END IF
    END IF
  END FOR
END; // DBSCAN
```

DBSCAN evaluation

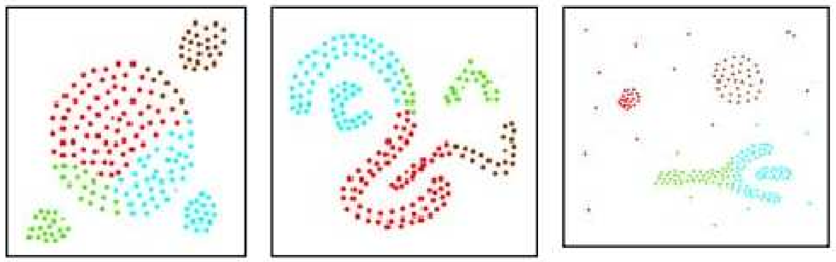


figure 5: Clusterings discovered by CLARANS

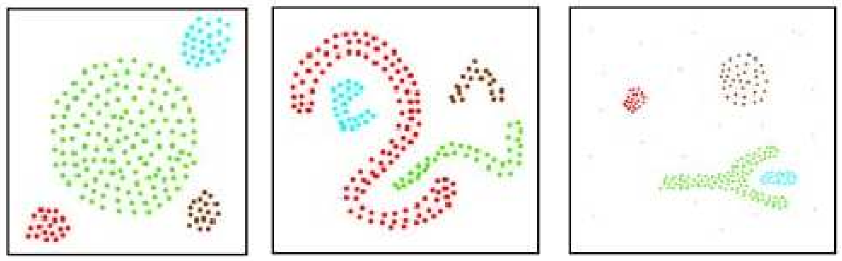
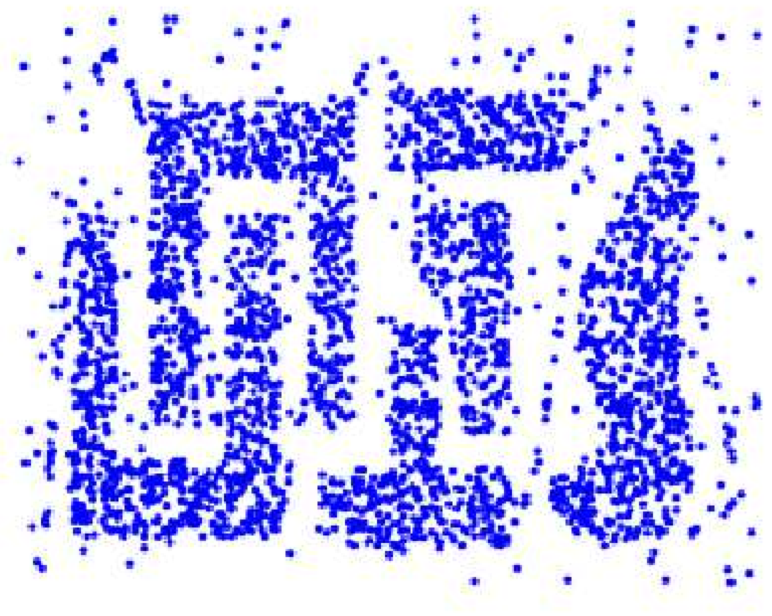
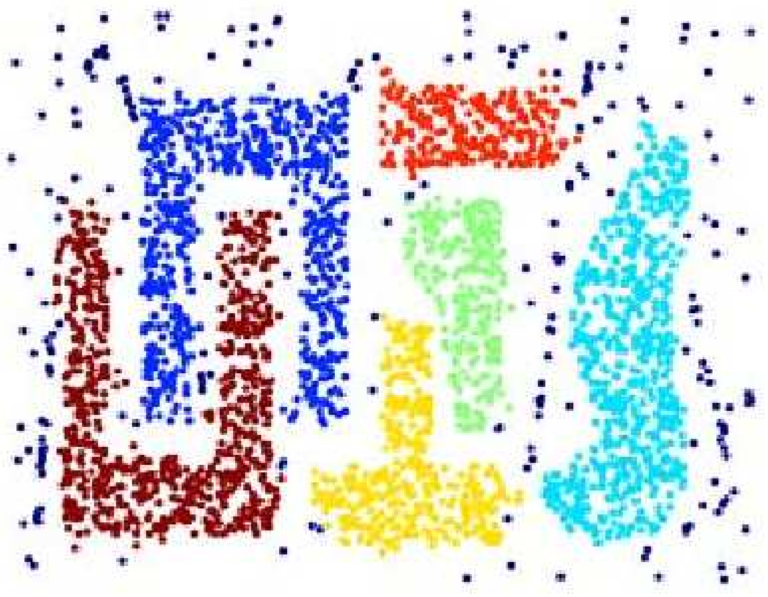


figure 6: Clusterings discovered by DBSCAN

- 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

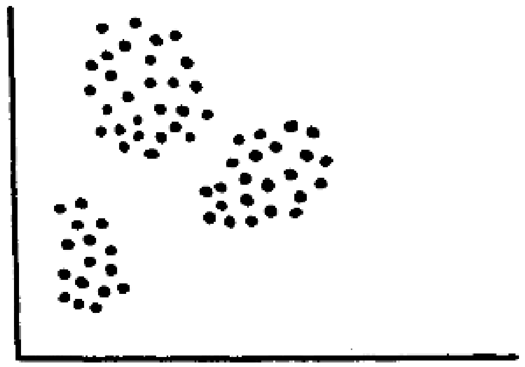


Fig. 1. Globular clusters.

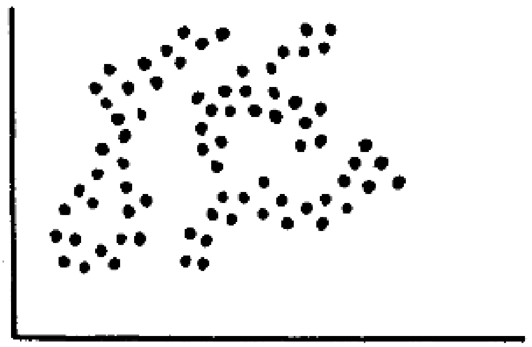
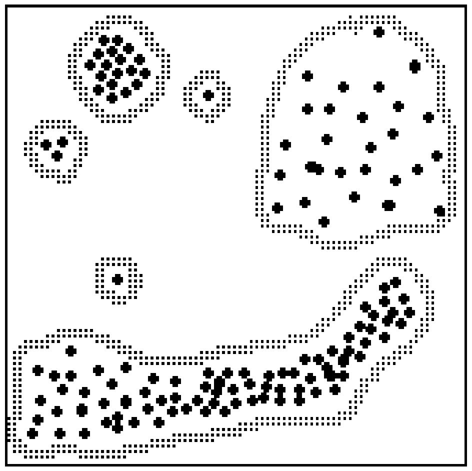


Fig. 2. Nonglobular clusters.

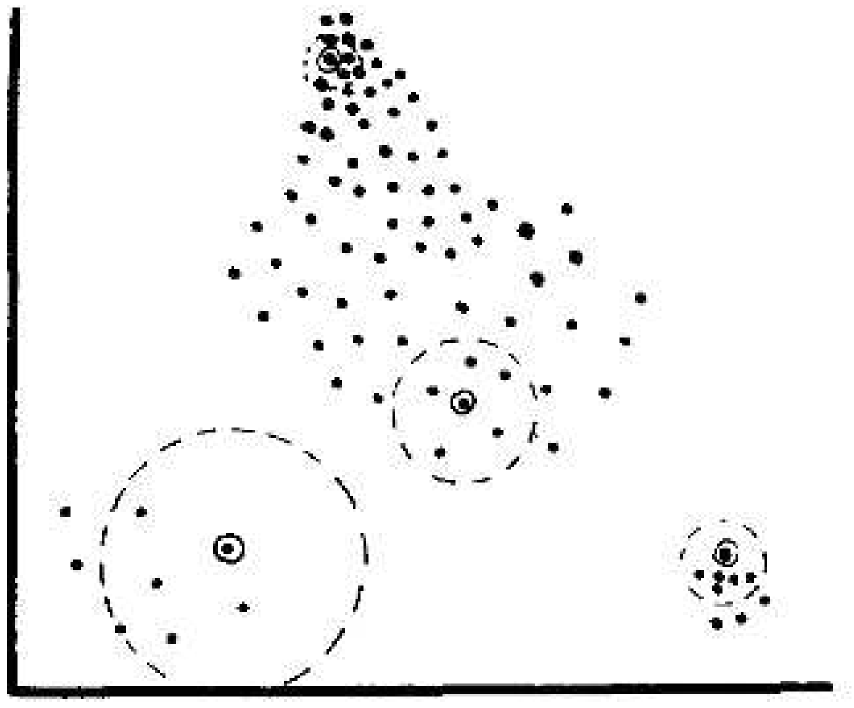


Jarvis-Patrick

- 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

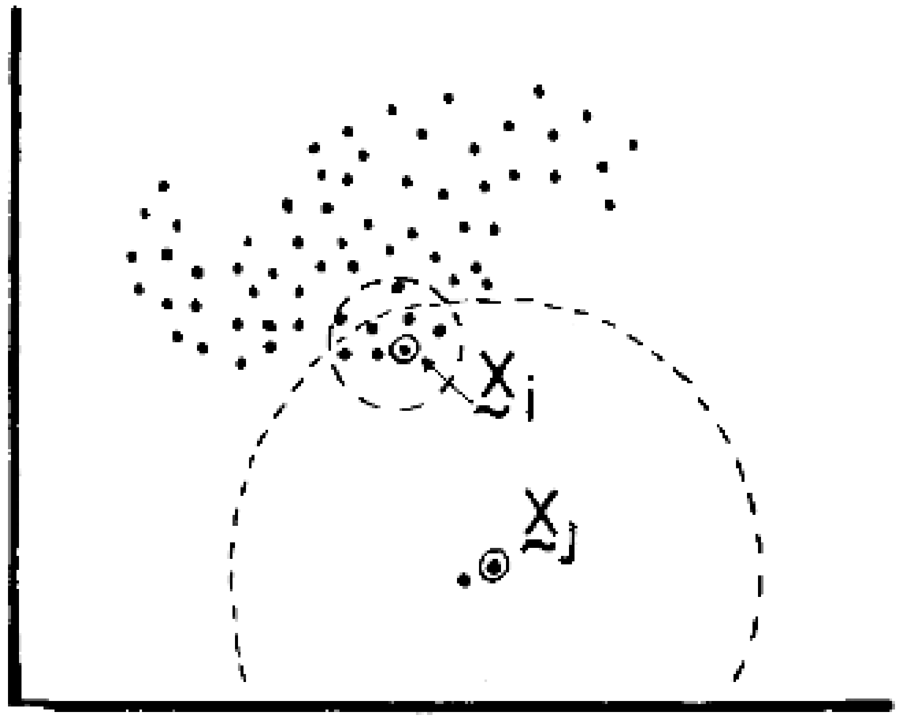
Jarvis-Patrick

Automatic scaling of neighborhoods (k=5)



Jarvis-Patrick

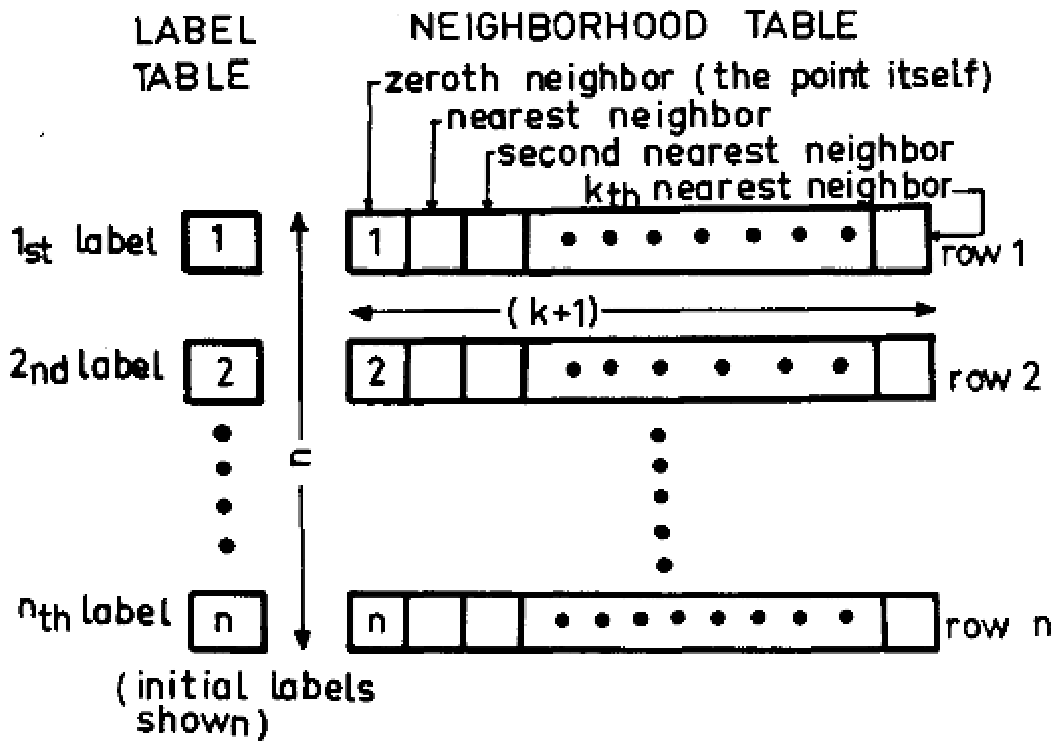
"Trap condition" for k=7: X_i belongs to X_j's neighborhood, but not vice versa.



JP algorithm

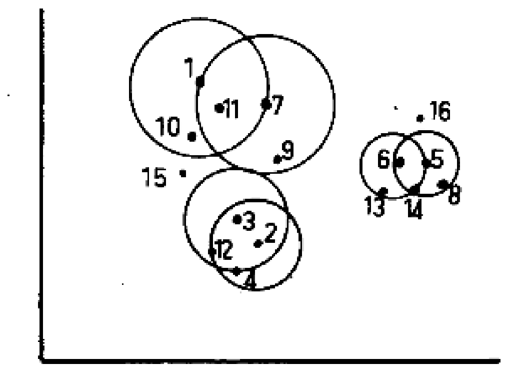
1. for each point in the dataset, list the k nearest neighbors by order number. Regard each point as its own zeroth neighbor. Once the neighborhood lists have been tabulated, the raw data can be discarded.
2. Set up an integer label table of length n, with each entry initially set to the first entry 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

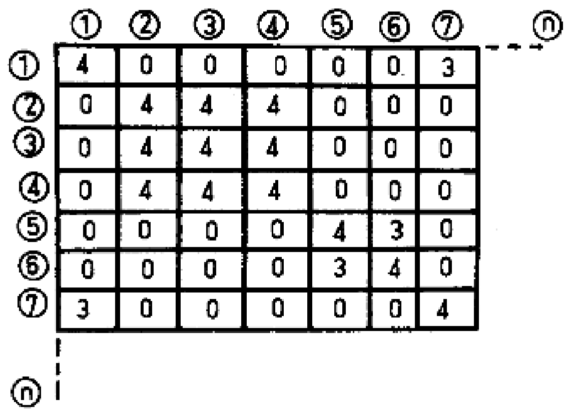


JP: alternative approaches

Similarity matrix



(a)



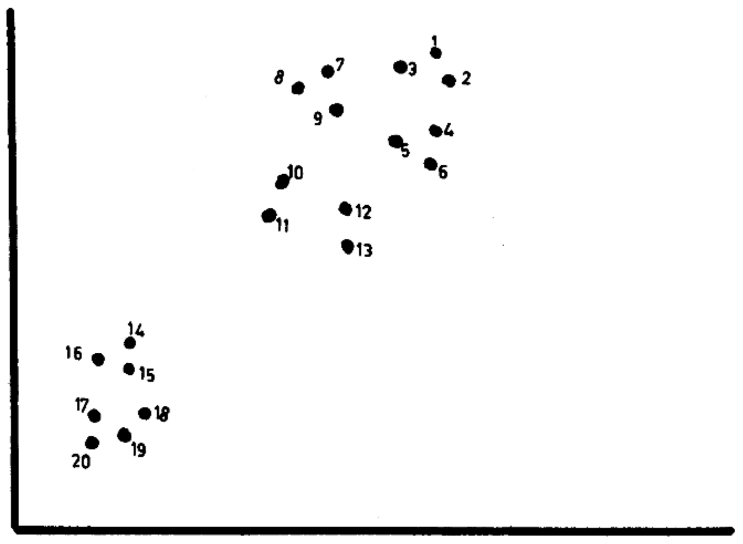
	1	2	3	4	5	6	7
1	4	0	0	0	0	0	3
2	0	4	4	4	0	0	0
3	0	4	4	4	0	0	0
4	0	4	4	4	0	0	0
5	0	0	0	0	4	3	0
6	0	0	0	0	3	4	0
7	3	0	0	0	0	0	4

(b)

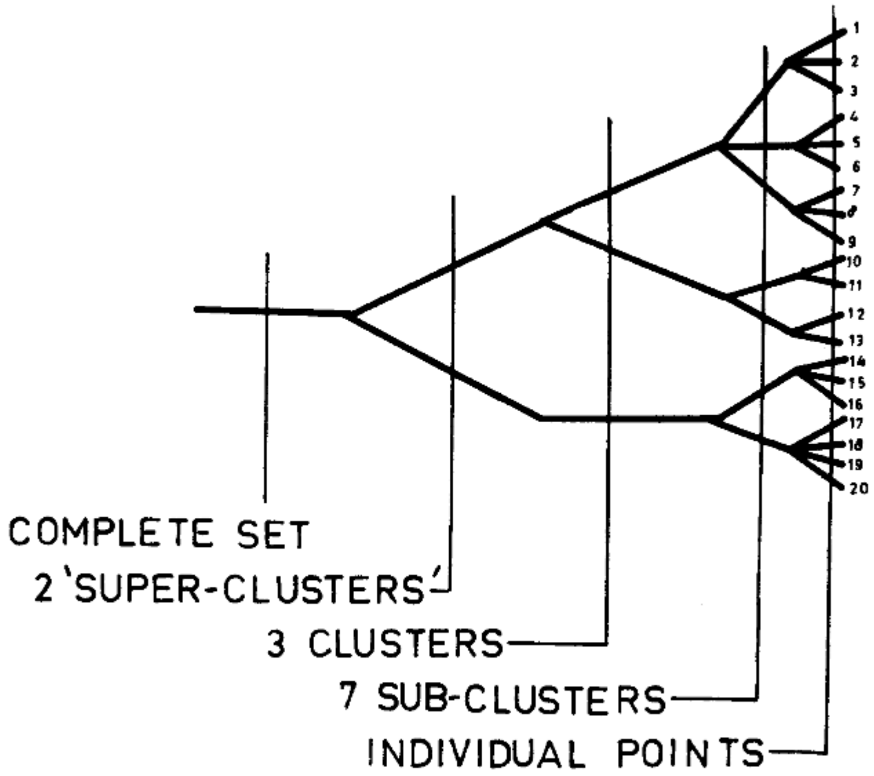
Fig. 8. Example of similarity matrix using the number of shared near neighbors as point pair similarity measure with equally weighted votes. (a) Sample points. (b) Similarity matrix for $k = 3$.

JP: alternative approaches

Hierarchical clustering - dendrogram



(a)



(b)

JP: conclusions

Pros:

- The same results are produced regardless of input order
- The number of clusters is not required in advance
- 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

- Clustering with gaussian mixtures
  Andrew W. Moore
- Clustering Using a Similarity Measure Based on Shared Near Neighbors (paper here)
- A Density-Based Algorithm for Discovering Clusters in Large Spatial Databases with Noise (paper here)

- The end