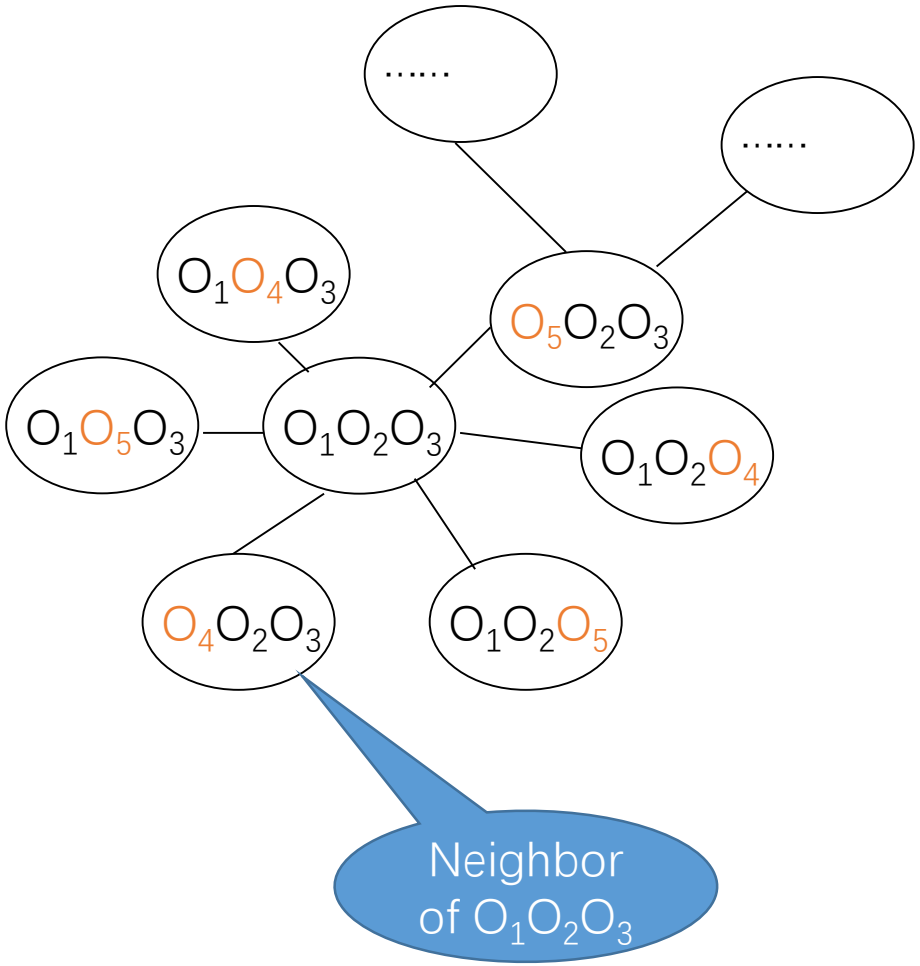
Density-based Method

CS385 – Machine Learning - Clustering

Partitioning – CLARANS



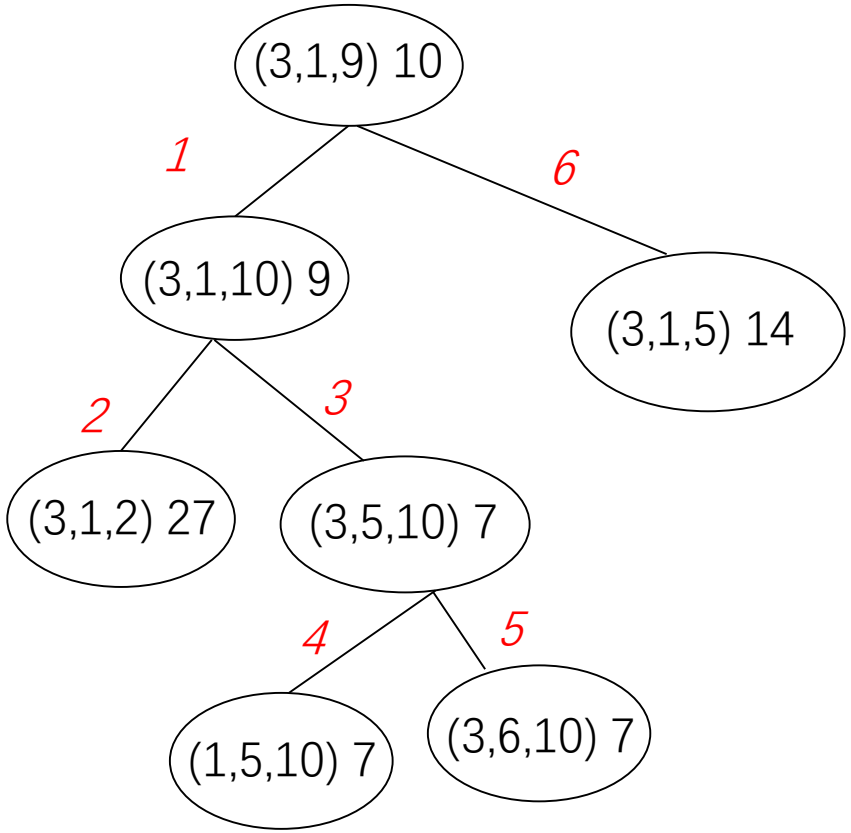
D ={3, 1, 9, 10, 2, 12, 5, 6 }, K=3

n=8, *k*=3, *Numberlocal, maxneighbor*

Graph search: backtracking

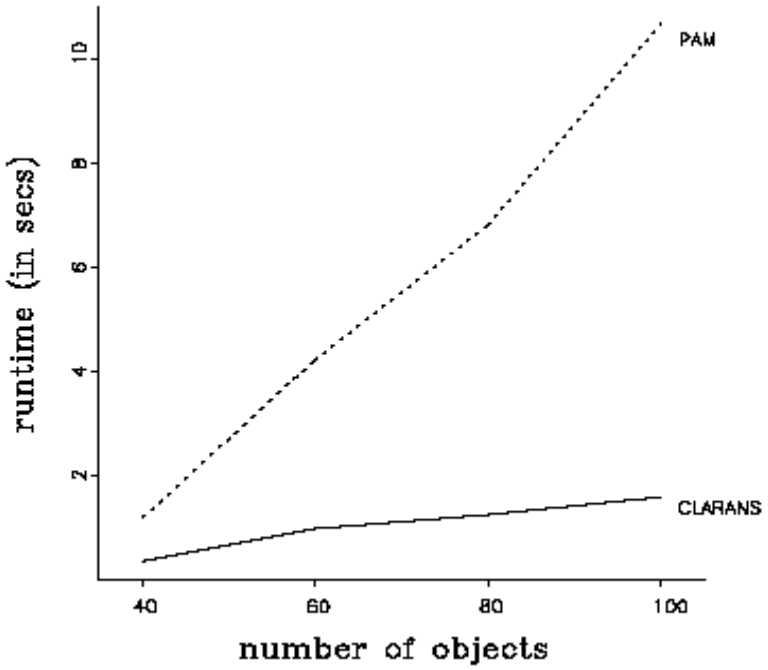
best-first with constraint

- D ={3, 1, 9, 10, 2, 12, 5, 6 }, K=3



Numberlocal=2 maxneighbor=2
Depth width

Partitioning – CLARANS

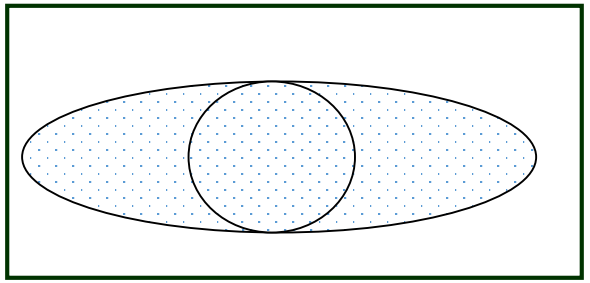


- PAM
  - O(k(n-k)) per iteration
- CLARANS
  - of the same quality

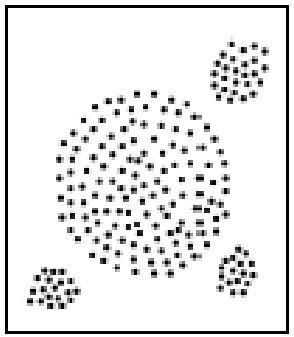
Local minimum

Clustering - Density

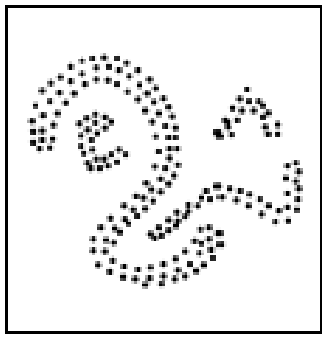
- Motivation



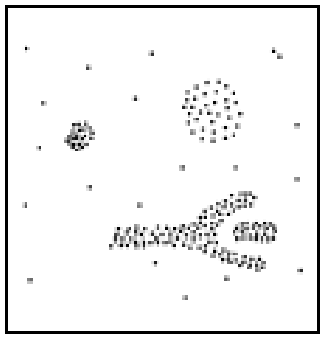
- Density



database 1

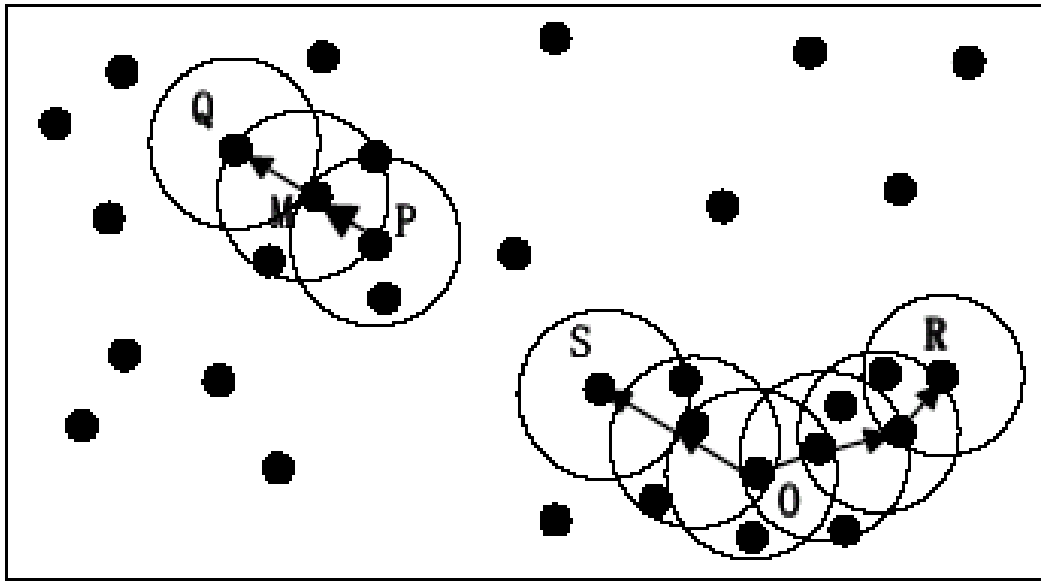


database 2



database 3

DensityBased - DBSCAN



Core object: O M P R

Radius ε
MinPts =3

- ε-neighborhood of an object ***o*** is the space within a radius **ε** centered at ***o***
- Core objects
  - If its ε-neighborhood contains at least MinPts objects
- Density reachable
  - Q density reachable from P
  - P is not density reachable from Q

Density Based - DBSCAN

Algorithm: DBSCAN: a density-based clustering algorithm.

Input:

- D: a data set containing n objects,
- ϵ: the radius parameter, and
- *MinPts*: the neighborhood density threshold.

Output: A set of density-based clusters.

Method:

```
(1)  mark all objects as unvisited;
(2)  do
(3)      randomly select an unvisited object p;
(4)      mark p as visited;
(5)      if the ε-neighborhood of p has at least MinPts objects
(6)          create a new cluster C, and add p to C;
(7)          let N be the set of objects in the ε-neighborhood of p;
(8)          for each point p' in N
(9)              if p' is unvisited
(10)                 mark p' as visited;
(11)                 if the ε-neighborhood of p' has at least MinPts points,
                         add those points to N;
(12)             if p' is not yet a member of any cluster, add p' to C;
(13)         end for
(14)         output C;
(15)     else mark p as noise;
(16) until no object is unvisited;
```

$O(n^2)$ -
Identify neighbors

2 parameters related to each other

Text Extraction - DBSCAN

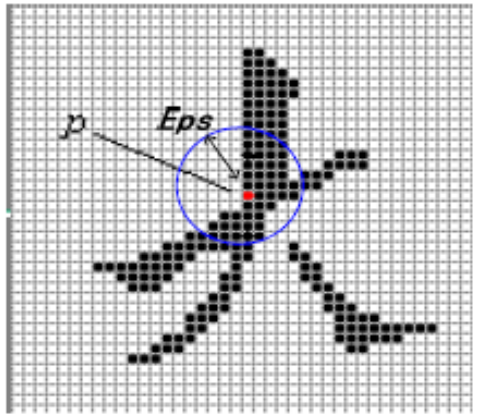


Figure 1. A magnified binary image (40*40) with one Chinese character. Pixel *p* is Labeled by ColorL='b', which is the same as the other black pixels. *p* is a core pixel with the current *Eps* if *Minpts* is 10.

Text Extraction - DBSCAN

(a)

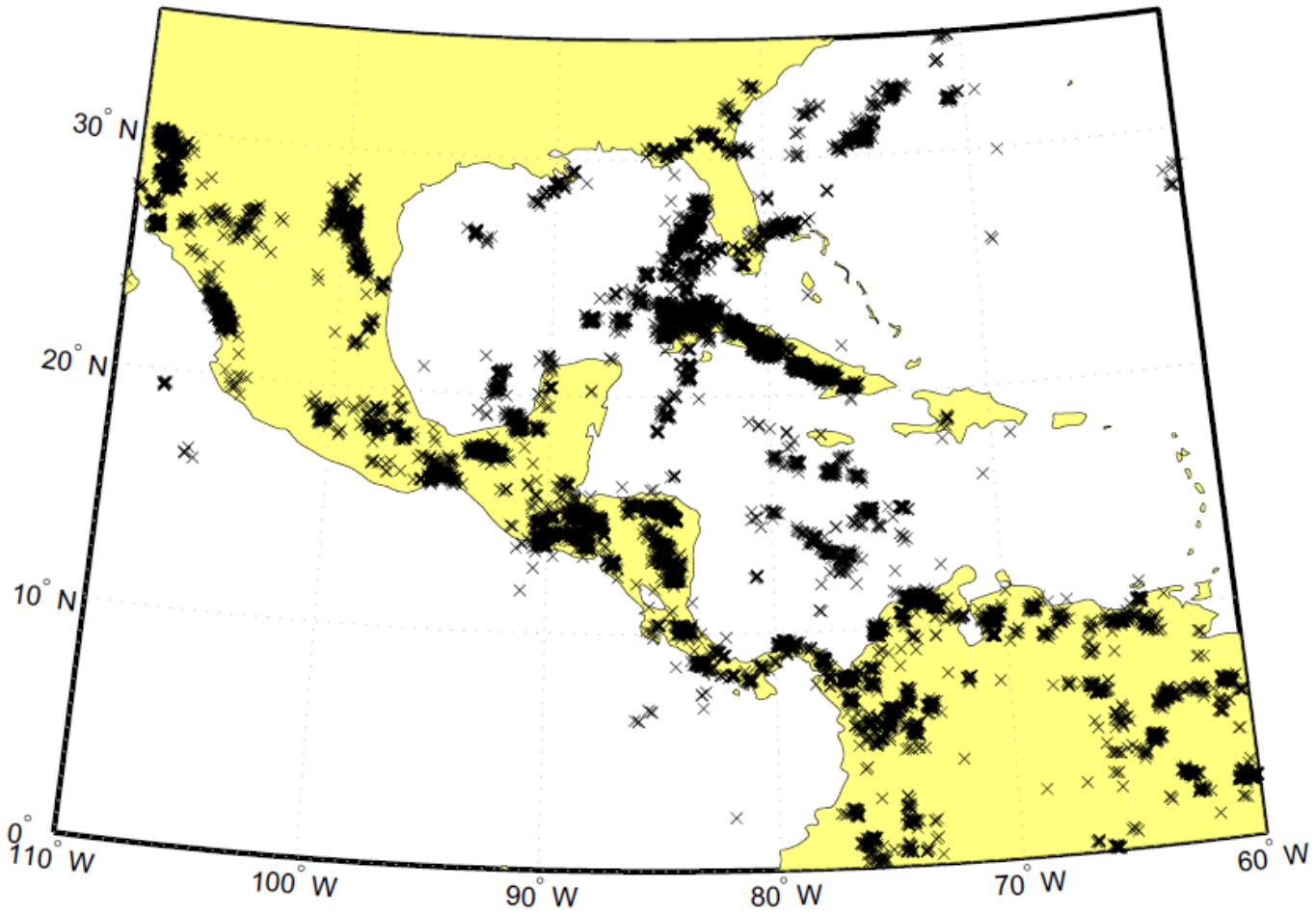
(b)

(c)

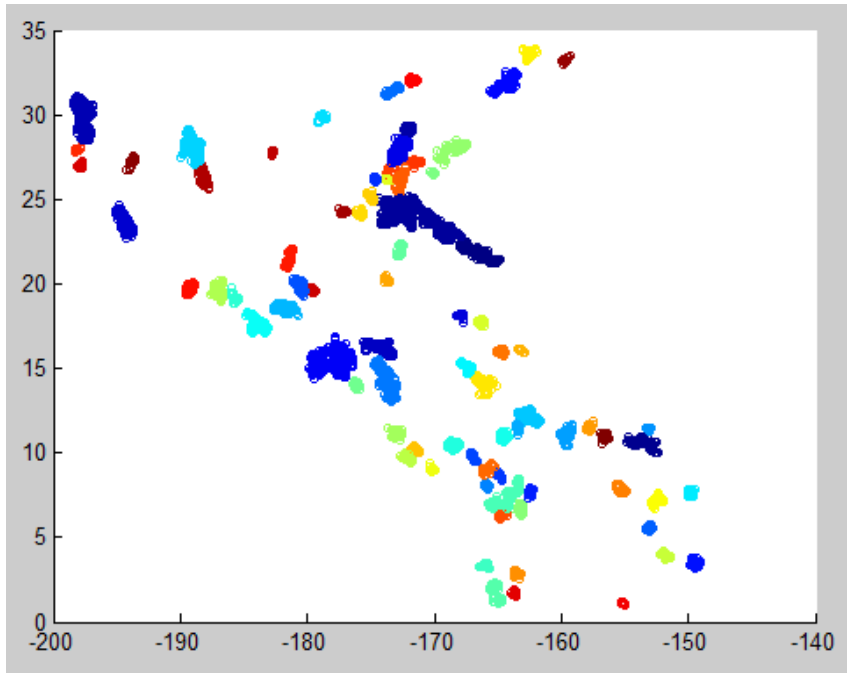
(b) result with *Eps* = min(image width, image height)/20
(c) result with *Eps* = min(image width, image height)/50.

Text Extraction - DBSCAN

WWLLN

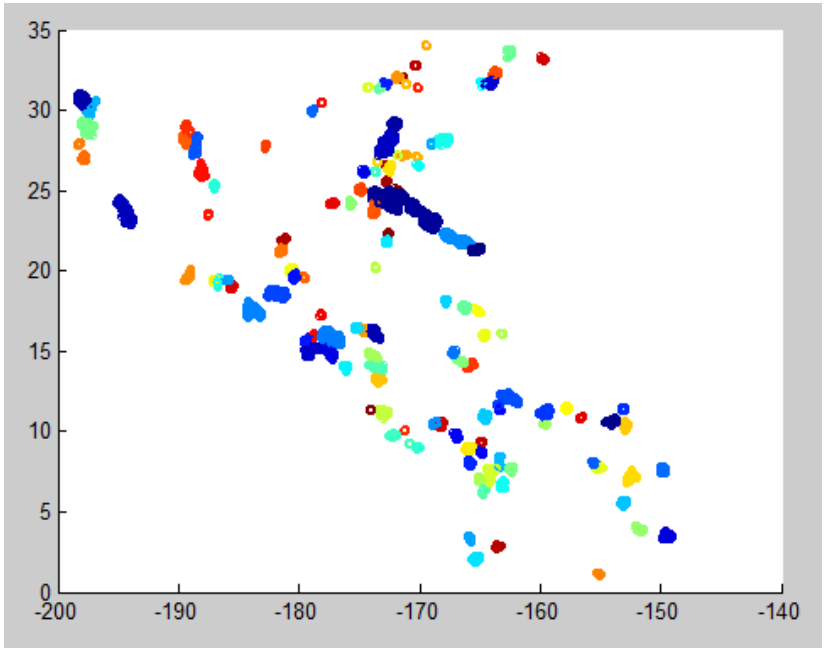


WWLLN - DBSCAN



Clusters: 80
Minpts: 20
Eps: 0.25

Clusters: 129
Minpts: 10
Eps: 0.125



Density Based - DENCLUE

- Influence Function

$$f_B^y(x) = f_B(x, y)$$

$$f_{Square}(x, y) = \begin{cases} 0 & \text{if } d(x, y) > \sigma \\ 1 & \text{otherwise} \end{cases}$$

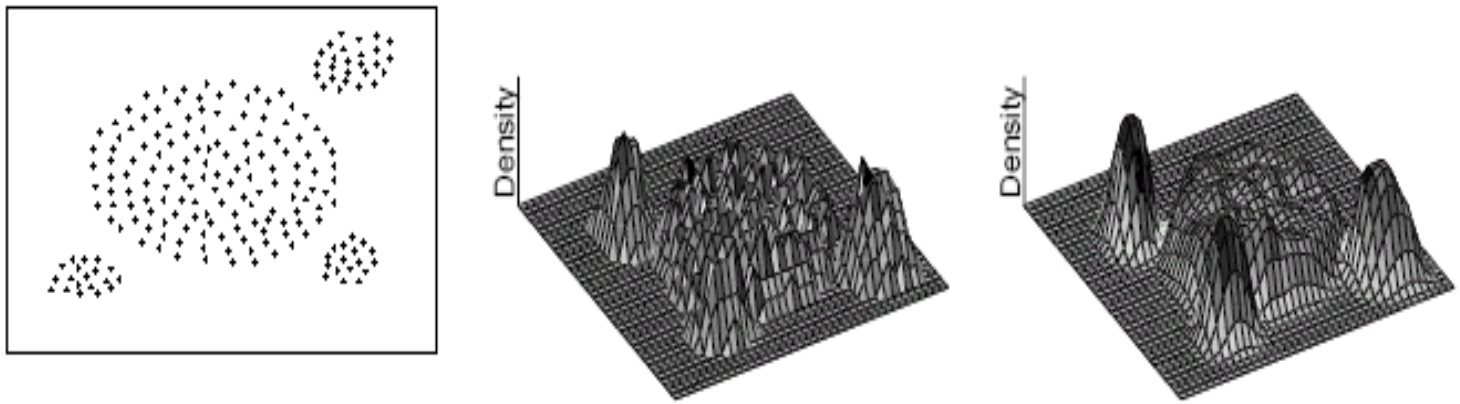
$$f_{Gauss}(x, y) = e^{-\frac{d(x,y)^2}{2\sigma^2}}$$

- Density Function

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

$$f_{Gauss}^D(x) = \sum_{i=1}^{n} e^{-\frac{d(x,x_i)^2}{2\sigma^2}}$$

DensityBased - DENCLUE



(a) Data Set (b) Square Wave (c) Gaussian

Figure 1: Example for Density Functions

DensityBased - DENCLUE

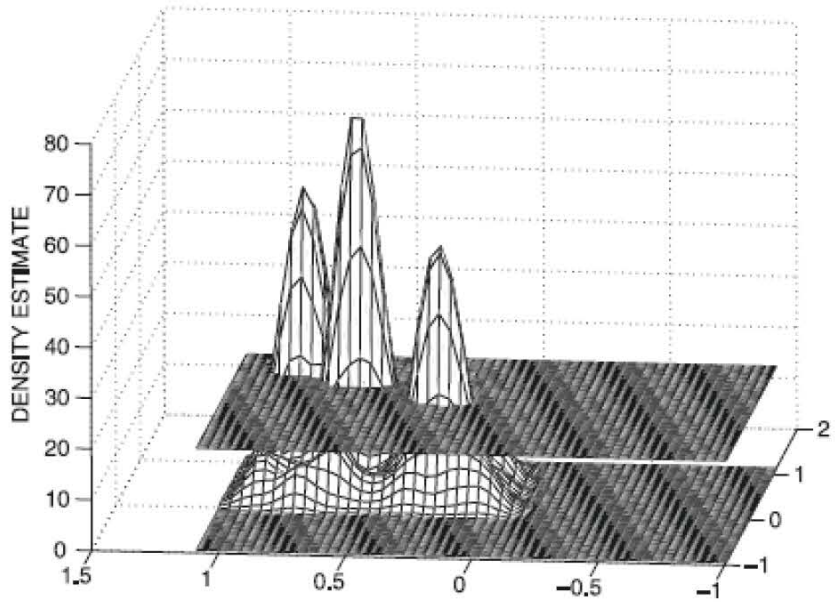
- Density-Attractor
  - A point x* is called a density-attractor for a given influence function, if x* is a local maximum of the density-function
- Center-Defined Cluster

  $\forall x \in C, \exists x^* \in X, f_B^D(x^*) \geq \xi$

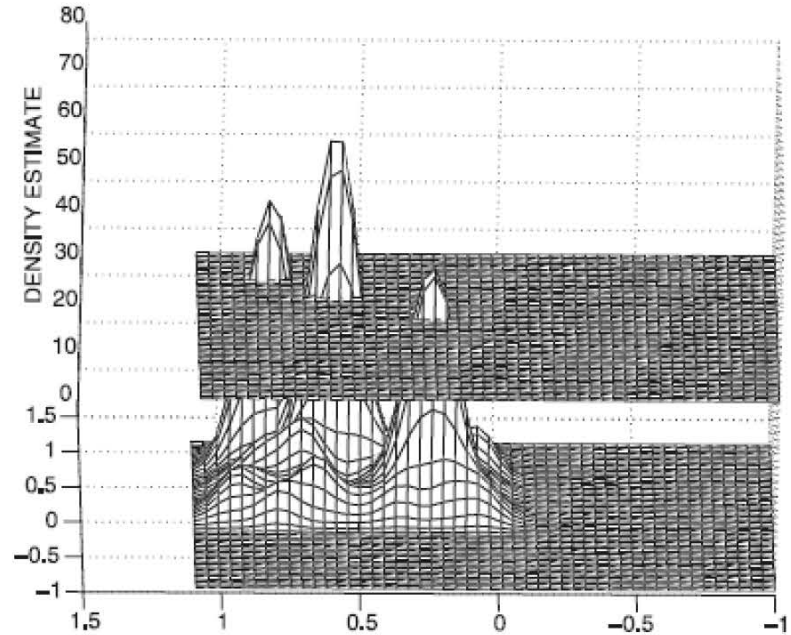
  - x is density attracted by x*
- Arbitrary-Shape Cluster

  $\forall x \in C, \exists x^* \in X, f_B^D(x^*) \geq \xi$

  $\forall x_1^*, x_2^* \in X, \exists P \subset F^d \; from \; x_1^* \; to \; x_2^*, \forall p \in P, f_B^D(p) \geq \xi$



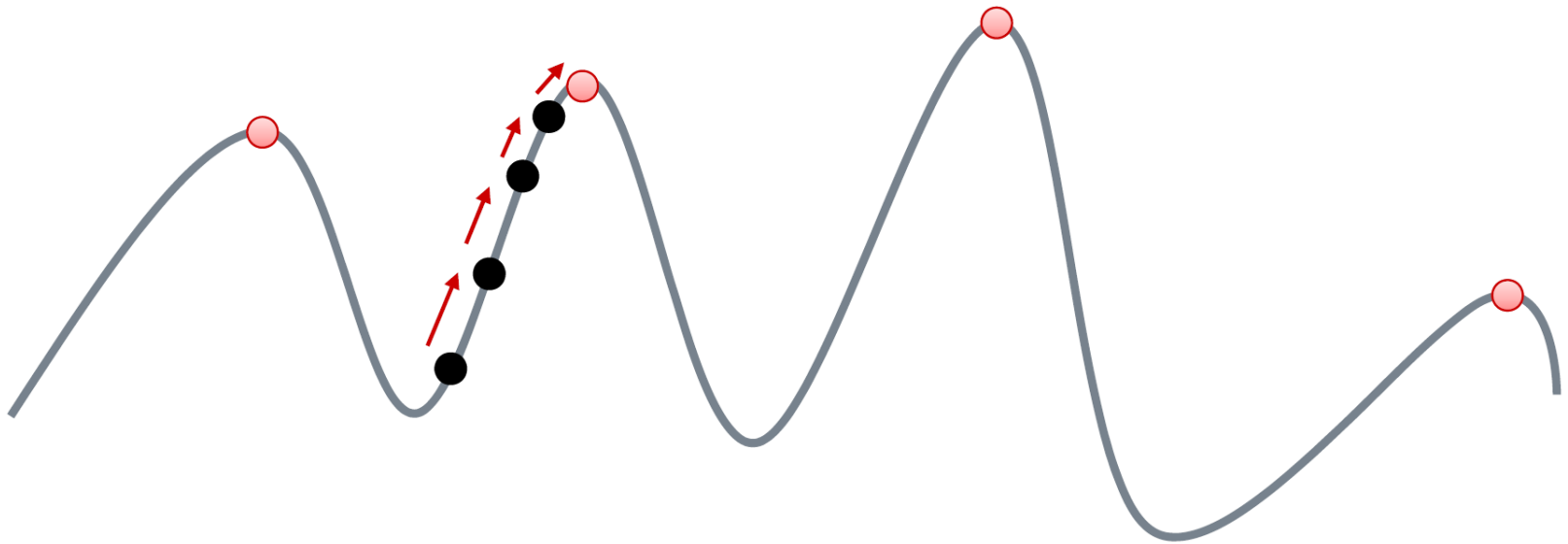
2 clusters



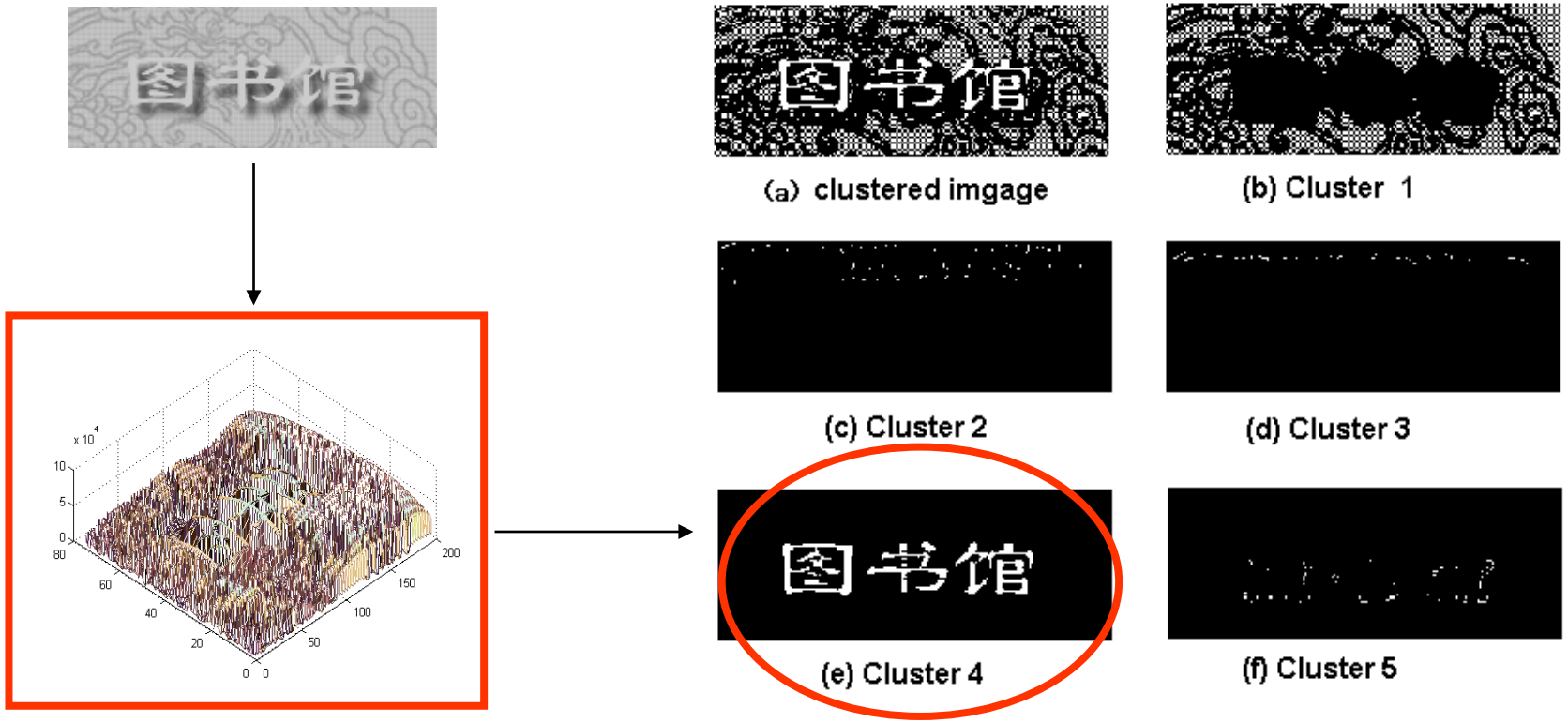
3 clusters

Density attracted

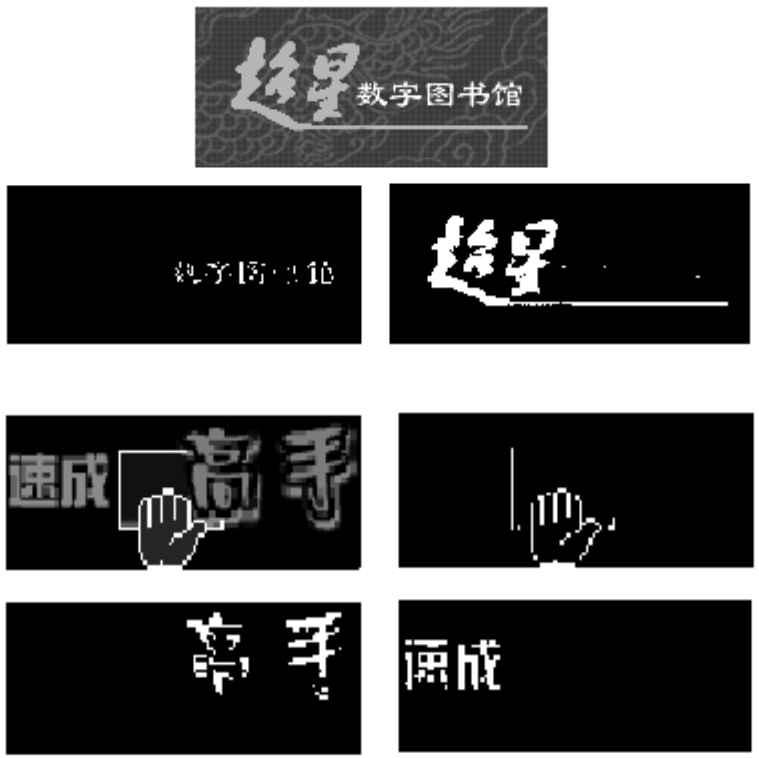
$$x = x + \alpha \nabla f_B^D(x)$$

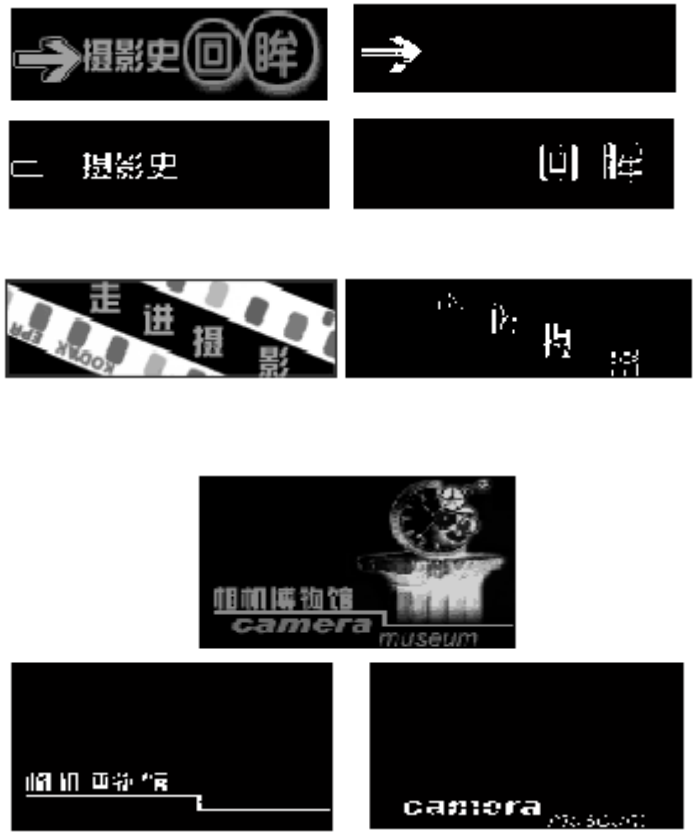


Text Extraction - DENCLUE



Text Extraction - DENCLUE



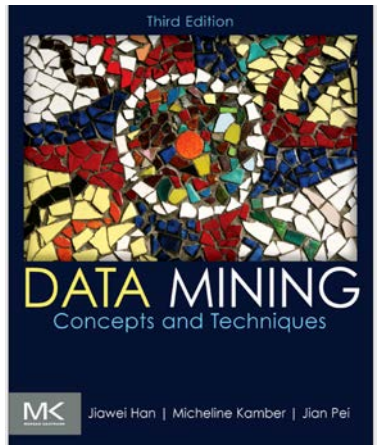


References

Section 10.4 *Density-based methods*

from

Data Mining: Concepts and Techniques by Jiawei Han etc.



The e-book can be found via DigiPen Resource Library – Online Safari Books