Clustering and outlier detection

WEEK 7

Outline

Unsupervised Learning

Clustering analysis (measurement and techniques)

Outlier and anomaly detection

Unsupervised Learning

No label to learn

(main) Key objectives:

1. Clustering analysis (detect aggregations/centralization behavior in data)
2. Outlier detection (detect abnormality behavior in data)

Clustering analysis is everywhere

Gain insights on product, provide data-driven decision making

Discovered Customer Prototypes	% of Customers	Departments: Fresh Meat	Packaged Foods	Dairy	Fish & Seafood	Gourmet	Fresh Produce	Bakery
Basic Shoppers	39%	3%	75%	2%	6%	1%	10%	3%
Meat Lovers	15%	59%	15%	4%	5%	3%	9%	5%
Produce Lovers	8%	9%	21%	5%	6%	2%	49%	7%
Gourmet Lovers	3%	1%	12%	0%	3%	73%	6%	4%
Variety Shoppers	35%	14%	39%	8%	12%	6%	19%	2%

Clustering

Definition on “cluster”

- Connectivity
- Centroid
- Distribution
- Density

Definition on “distance/similarity”

- Minkowski distance
- Cosine similarity
- Set related similarity

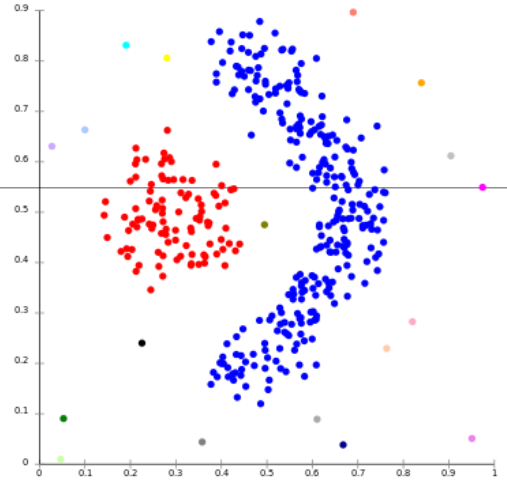
Major approach

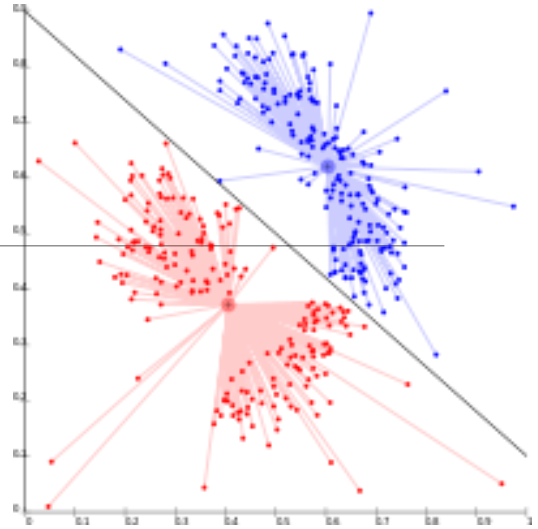
- Statistical
- Matrix factorization
- Machine learning

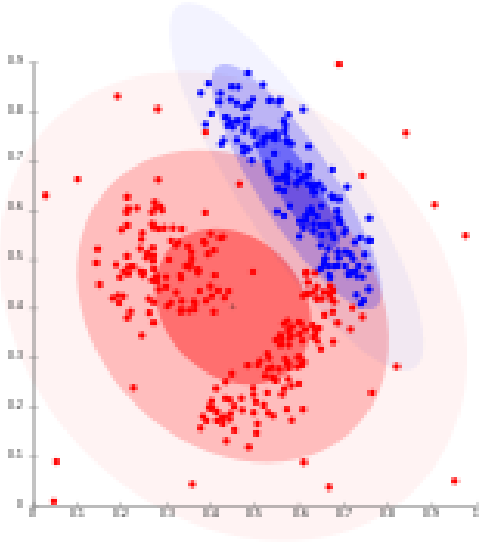
Clustering

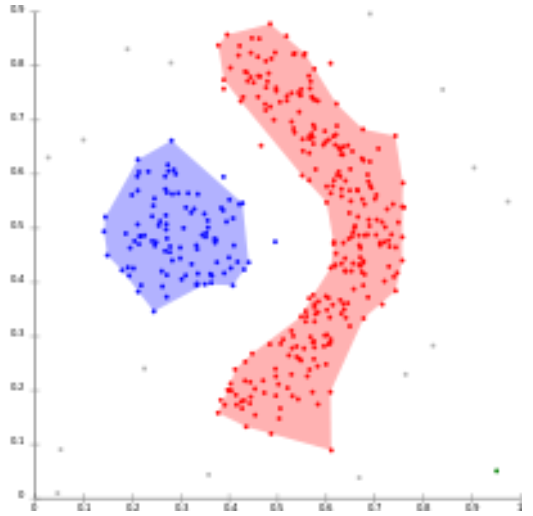
Different ways to define clusters

- Connectivity based
- Centroid based
- Distribution based
- Density based









K-mean algorithm

given N points, on d dimension space, with k clusters.

$x_1, x_2, \cdots x_N$ $\quad$ $t_1, t_2, \cdots t_k$

① initialize $t_1, \cdots t_k$ randomly (or use kmean++)

② loop until $t_1, \cdots t_k$ converge:

2-a: assign $X \to t$ corresponding distance.

2-b: update $t \to t_{new}$ based on X's mean for each t_i

time complexity:

assume total loop is p times

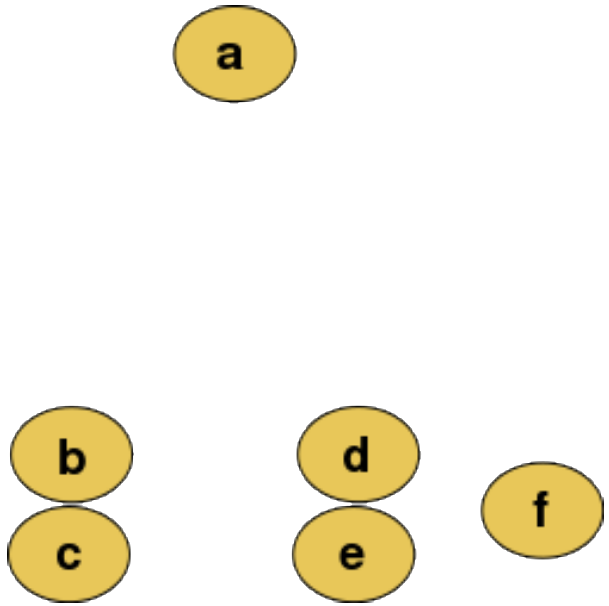
① $\to O(0)$

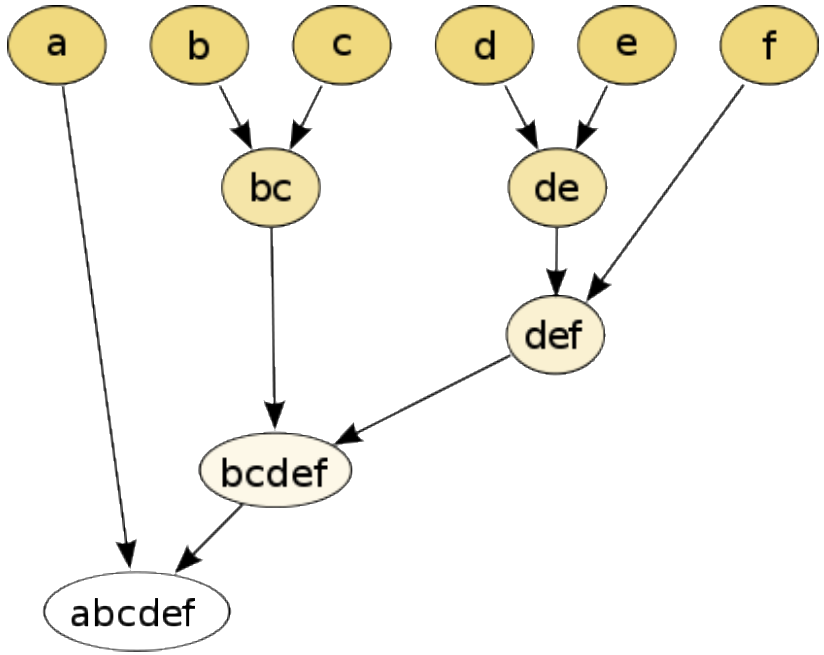
2-a $\to N*k$ distance calculation $\Rightarrow N*k*d$

2-b $\to N*d$ calculate mean.

$[O(0) + O(N*k*d) + O(N*d)] \times p = O(Nkdp)$

Hierarchical clustering algorithm





Hierarchical clustering algorithm

① compute distance matrix for $X_1, \cdots X_N$,

② repeat until get 1 cluster:

2-a: merge the closest two clusters

2-b: update distance matrix for the change.

X_1, X_2, X_3, X_4 $d=1$, $X_1=0, X_2=1, X_3=4, X_4=6$.

① distance matrix.

$$\begin{array}{c} X_0 \\ X_1 \\ X_2 \\ X_3 \end{array}\begin{pmatrix} 0 & 1 & 4 & 6 \\ 1 & 0 & 3 & 5 \\ 4 & 3 & 0 & 2 \\ 6 & 5 & 2 & 0 \end{pmatrix}$$

use minimun distance between clusters.

② cycle 1: merge X_0, X_1 ⓐ, update matrix ⓑ $\Rightarrow$

$$\begin{array}{c} X_0+X_1 \\ X_2 \\ X_3 \end{array}\begin{pmatrix} 0 & 3 & 5 \\ 3 & 0 & 2 \\ 5 & 2 & 0 \end{pmatrix}$$

cycle-2: merge X_2, X_3 ⓐ, merge matrix ⓑ $\Rightarrow$

$$\begin{array}{c} X_0+X_1 \\ X_2+X_3 \end{array}\begin{pmatrix} 0 & 3 \\ 3 & 0 \end{pmatrix}$$

cycle-3: merge X_0+X_1, X_2+X_3 ⓐ, update metrix ⓑ $\Rightarrow (0)$ Done!

time complexity:

① $\Rightarrow O(N^2 * d)$ (ignored for now)

② cycle $N-1$ times. for cycle i, remain matrix is $N-i$ to find the min distance (closest merge) search.

$\frac{(N-i)(N-i+1)}{2} \sim O((N-i+1)^2)$

update $\Rightarrow O(N-i+1)$

$\sum_{i=1}^{N} [O(N-i+1)^2 + O(N-i+1)] = O(N^3)$

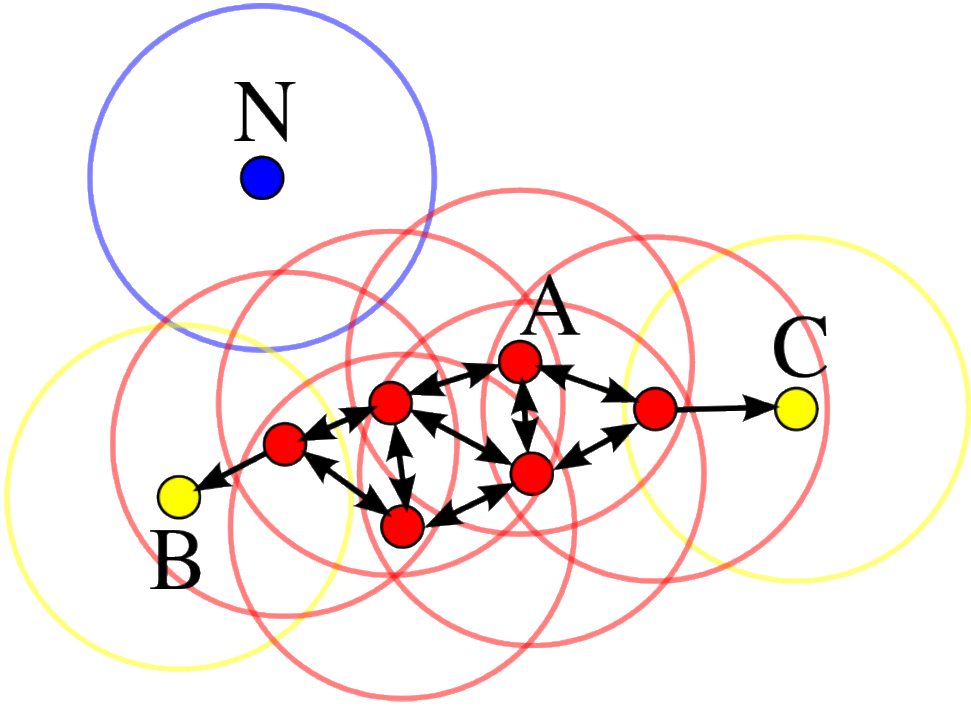
naive implementation $\Rightarrow O(N^3)$

DBSCAN algorithm

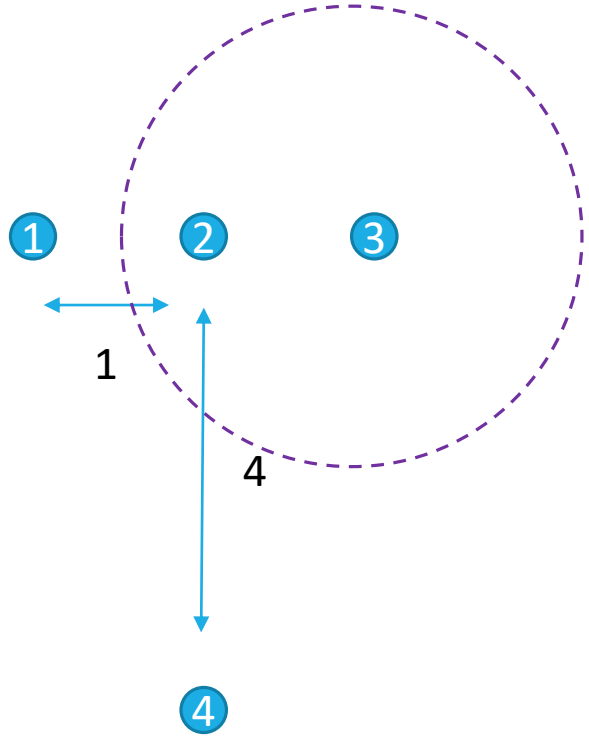
Density-based spatial clustering of applications with noise (DBSCAN)

Three type of points:

1. core
2. reachable (edge)
3. outlier



DBSCAN algorithm



Circle size = 1.5, min points = 3

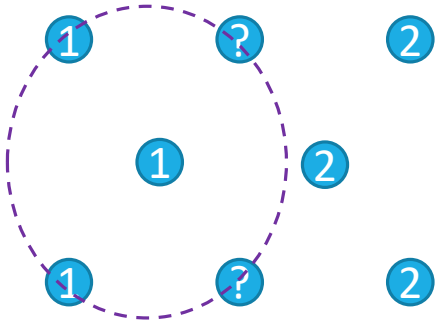
Cycle 1. P1 is picked, np = 2 < 3; P1 => outlier

Cycle 2. P4 is picked, np = 1 < 3; P4 => outlier

Cycle 3. P2 is picked, np = 3 >= 3; P2 => core (assign cluster-A)

- For all points P2 is connected with,
- Cycle 3-1. P1 is picked, np = 2 < 3; P1 => edge (cluster - A)
- Cycle 3-2. P3 is picked, np = 2 < 3; P3 => edge (cluster - A)

Is DBSCAN fully deterministic?



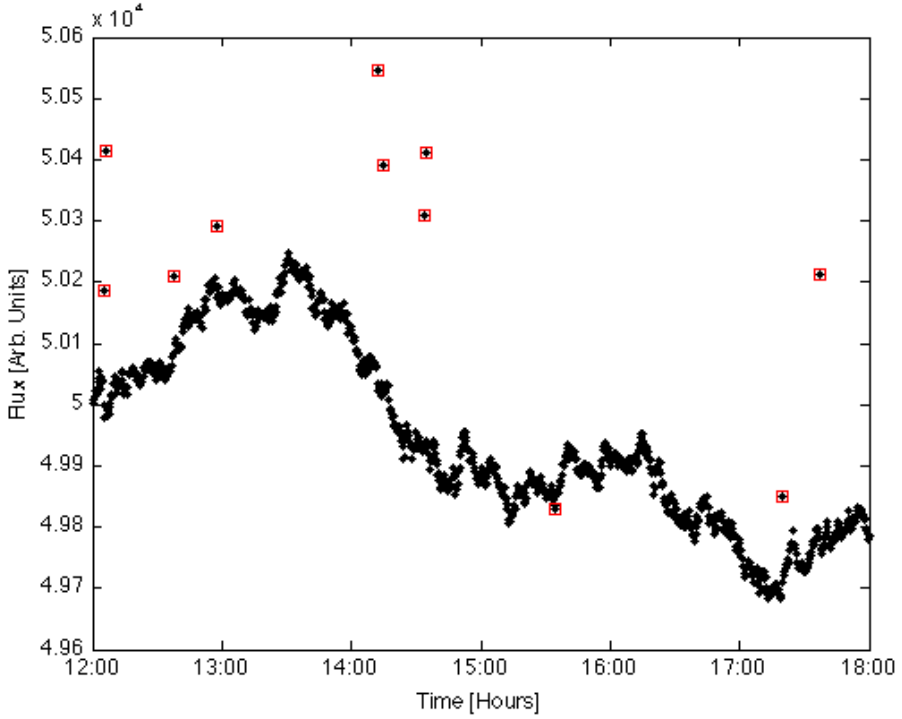
Think about how the scenario:

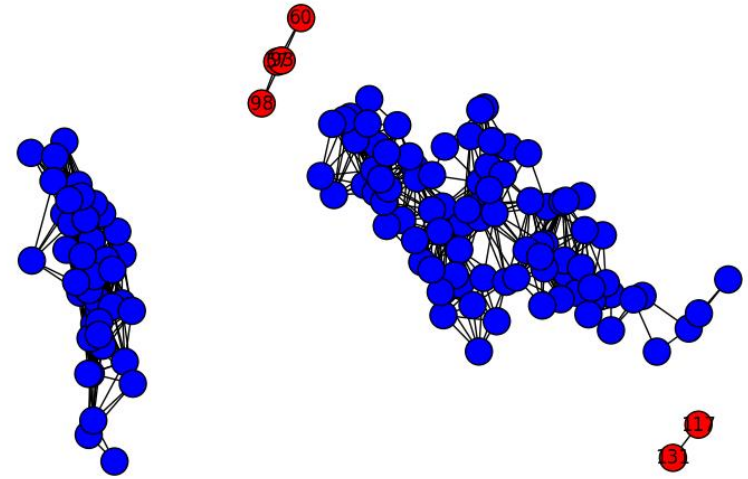
Circle radius = 0.75, min point = 4

Q. how to make it fully deterministic?

Outlier detection is also widely used

Fraud detection, data processing, etc.





Novelty & outlier detection

Two types

Novelty detection: training data contain NO outliers

Outlier detection: training data contain outlier

One-class SVM approach