UCLA Samueli Computer Science

CS145 Discussion: Week 5

Classification evaluation, Clustering

Junheng Hao

Friday, 11/06/2020

UCLA
Engineer Change.

Roadmap

- Announcement
- Similarity Measurement
- Classification evaluation
- Clustering: K-means

Announcement

- Homework 3 due **Nov. 9 (Monday, Week 6) 11:59 PT**
  - Submit through **GradeScope** of 1 PDF (**2** python file and **2** jupyter notebooks into 1 PDF file)
  - Assign pages to the questions on GradeScope
- Midterm project due on **Nov. 11 (Wednesday, Week 6)**
  - 3-page midterm project report
  - At least one submission to Kaggle

- **Approximately 3 pages**
- Current progress about project, including
  - Data processing and transformation
  - Designed & tested models / methods
- Discussion and future project plan
  - Some conclusions and findings
  - Analysis of current models and techniques
  - Timeline of future project plan (around the next 4 weeks)

- Midterm exam on **Nov. 16 (Monday, Week 7)** on CCLE (with proper browser setting)

UCLA
Engineer Change.

Similarity Matrix

- Similarity and dissimilarity matrix
  - Pairwise measures how alike/different two data points are
- Example of numerical attribute
  - Three 2-dim input, x1,x2,x3
  - We write the dissimilarity matrix as a 3x3 lower-triangular matrix

$$\begin{bmatrix} 3 & 5 \\ 6 & 9 \\ 11 & 21 \end{bmatrix} \Rightarrow \begin{bmatrix} 0 & 0 & 0 \\ d(2,1) & 0 & 0 \\ d(3,1) & d(3,2) & 0 \end{bmatrix}$$

- Dissimilarity under Euclidean distance

$$\begin{bmatrix} 0 & 0 & 0 \\ \sqrt{(3-6)^2+(5-9)^2} & 0 & 0 \\ \sqrt{(3-11)^2+(5-21)^2} & \sqrt{(11-6)^2+(21-9)^2} & 0 \end{bmatrix}$$

UCLA
Engineer Change.

Nominal Attributes

- Student 1: likes Jazz, eats pizza, roots for the cubs, wears socks
- Student 2: likes Rock, eats pizza, roots for the cubs, goes barefoot
- d(Student 1, Student 2):
  - m: # of matches → 2
  - p: total # of variables → 4
  - d(Student 1, Student 2) = (4-2)/4 = 0.5

Binary Attributes

- Symmetric binary attributes:
  - Gender
- Asymmetric attributes:
  - Preference, diagnosis, etc.
- Can be manually defined

Binary Attributes

- Dissimilarity of Binary Attributes:
  - Define 0 and 1 and calculate q,s,r,t,p
- Symmetric binary variables:

$$d(i,j) = \frac{r+s}{q+r+s+t}$$

		Object j		
		1	0	sum
Object i	1	q	r	$q+r$
	0	s	t	$s+t$
	sum	$q+s$	$r+t$	p

- Asymmetric binary variables (0 is less important):

$$d(i,j) = \frac{r+s}{q+r+s}$$

- Jaccard coefficient (similarity measure for asymmetric binary variables):

$$sim_{Jaccard}(i,j) = \frac{q}{q+r+s}$$

Binary Attributes Example

Name	Gender	Fever	Cough	Test-1	Test-2	Test-3	Test-4
Jack	M	Y	N	P	N	N	N
Mary	F	Y	N	P	N	P	N
Jim	M	Y	P	N	N	N	N

Object *i* \ Object *j*	1	0	sum
1	q	r	$q+r$
0	s	t	$s+t$
sum	$q+s$	$r+t$	p

- Define M,Y,P as 1; Define F,N as 0
- Assume symmetric for Gender, asymmetric for other attributes
- i = Jack, j = Mary, what is d(i, j) in terms asymmetric attributes?
  - r = 0, s = 1, q = 2
  - d(i, j) = 0.32

Ordinal Attributes

- Order is important
  - Freshman, Sophomore, Junior, Senior
- Replace attribute by rank

$$r_{if} \in \{1,\ldots,M_f\}$$

- Convert rank to numeric values

$$z_{if} = \frac{r_{if} - 1}{M_f - 1}$$

UCLA
Engineer Change.

Cosine Similarity

- For vector data
- d1: I like to go to the store
- d2: I like the cubs, go cubs go

$$\cos(d_1, d_2) = (d_1 \bullet d_2) / ||d_1|| \, ||d_2|| ,$$

Document	I	like	to	go	the	store	cubs
d1	1	1	2	1	1	1	0
d2	1	1	0	2	1	0	2

- cos(d1, d2)?
  - $\frac{1\cdot1+1\cdot1+2\cdot0+1\cdot2+1\cdot1+1\cdot0+0\cdot2}{\sqrt{1^2+1^2+2^2+1^2+1^2+1^2+0^2}\cdot\sqrt{1^2+1^2+0^2+2^2+1^2+0^2+2^2}}$

Mixed Attributes

All Together

$$\begin{bmatrix} 0 & & & \\ 1 & 0 & & \\ 1 & 1 & 0 & \\ 0 & 1 & 1 & 0 \end{bmatrix}$$

$$\begin{bmatrix} 0 & & & \\ 1.0 & 0 & & \\ 0.5 & 0.5 & 0 & \\ 0 & 1.0 & 0.5 & 0 \end{bmatrix}$$

Object Identifier	*test-1 (nominal)*	*test-2 (ordinal)*	*test-3 (numeric)*
1	code A	excellent	45
2	code B	fair	22
3	code C	good	64
4	code A	excellent	28

$$\begin{bmatrix} 0 & & & \\ 0.55 & 0 & & \\ 0.45 & 1.00 & 0 & \\ 0.40 & 0.14 & 0.86 & 0 \end{bmatrix}$$

- d(3,1)?
  - $\frac{1(1)+1(0.5)+1(0.45)}{3}$

UCLA
Engineer Change.

Confusion Matrix

Confusion Matrix: How your model got confused

Actual class\Predicted class	C_1	$\neg C_1$
C_1	**True Positives (TP)**	**False Negatives (FN)**
$\neg C_1$	**False Positives (FP)**	**True Negatives (TN)**

UCLA
Engineer Change.

Evaluation Metric

A\P	C	¬C	
C	TP	FN	P
¬C	FP	TN	N
	P'	N'	All

Accuracy = (TP + TN)/All

Error rate = (FP + FN)/All

Sensitivity = TP/P

Specificity = TN/N

$$precision = \frac{TP}{TP + FP}$$

$$recall = \frac{TP}{TP + FN}$$

UCLA

Evaluation Metric

A\P	C	¬C	
C	TP	FN	P
¬C	FP	TN	N
	P'	N'	All

Why do we need all these measures? Isn't accuracy telling us how good the model is?

- Imbalanced data
- Imbalanced importance of positive and negative

Accuracy = (TP + TN)/All

Error rate = (FP + FN)/All

Sensitivity = TP/P

Specificity = TN/N

$$precision = \frac{TP}{TP + FP}$$

$$recall = \frac{TP}{TP + FN}$$

UCLA
Engineer Change.

Evaluation Metric

Single number metric

- F measure (F1 score)

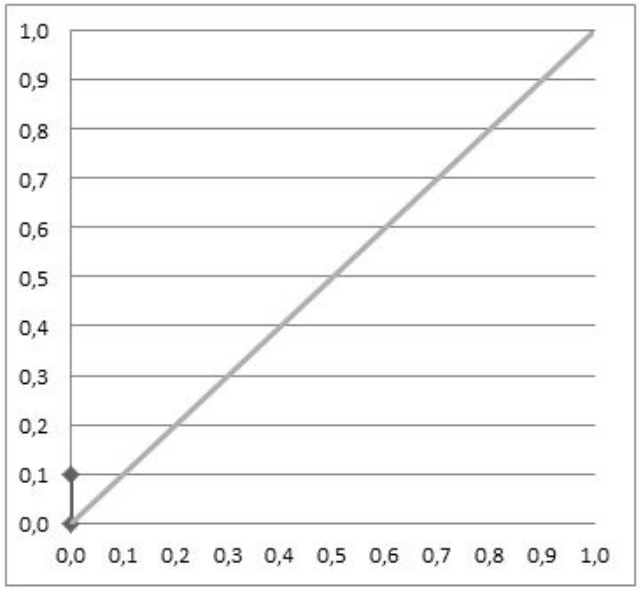
$$F = \frac{2 \times precision \times recall}{precision + recall}$$

- Area under the curve (AUC)
  - ROC curve (true positive against false positive)
  - PRC (precision against recall)

Receiver Operating Characteristics(ROC)

TPR

#	C	Score
1	P	0,9
2	P	0,8
3	N	0,7
4	P	0,6
5	P	0,55
6	P	0,54
7	N	0,53
8	N	0,52
9	P	0,51
10	N	0,505
11	P	0,4
12	N	0,39
13	P	0,38
14	N	0,37
15	N	0,36
16	N	0,35
17	P	0,34
18	N	0,33
19	P	0,3
20	N	0,1



FPR

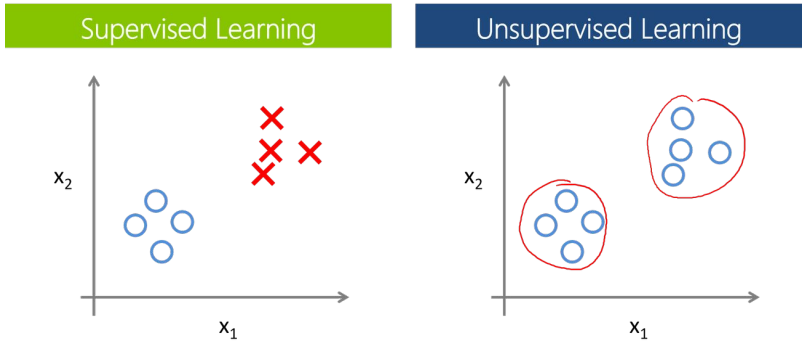
True positive rate: $TPR = TP/P$ (sensitivity)

False positive rate: $FPR = FP/N$ (1-specificity)

http://mlwiki.org/index.php/ROC_Analysis

Clustering

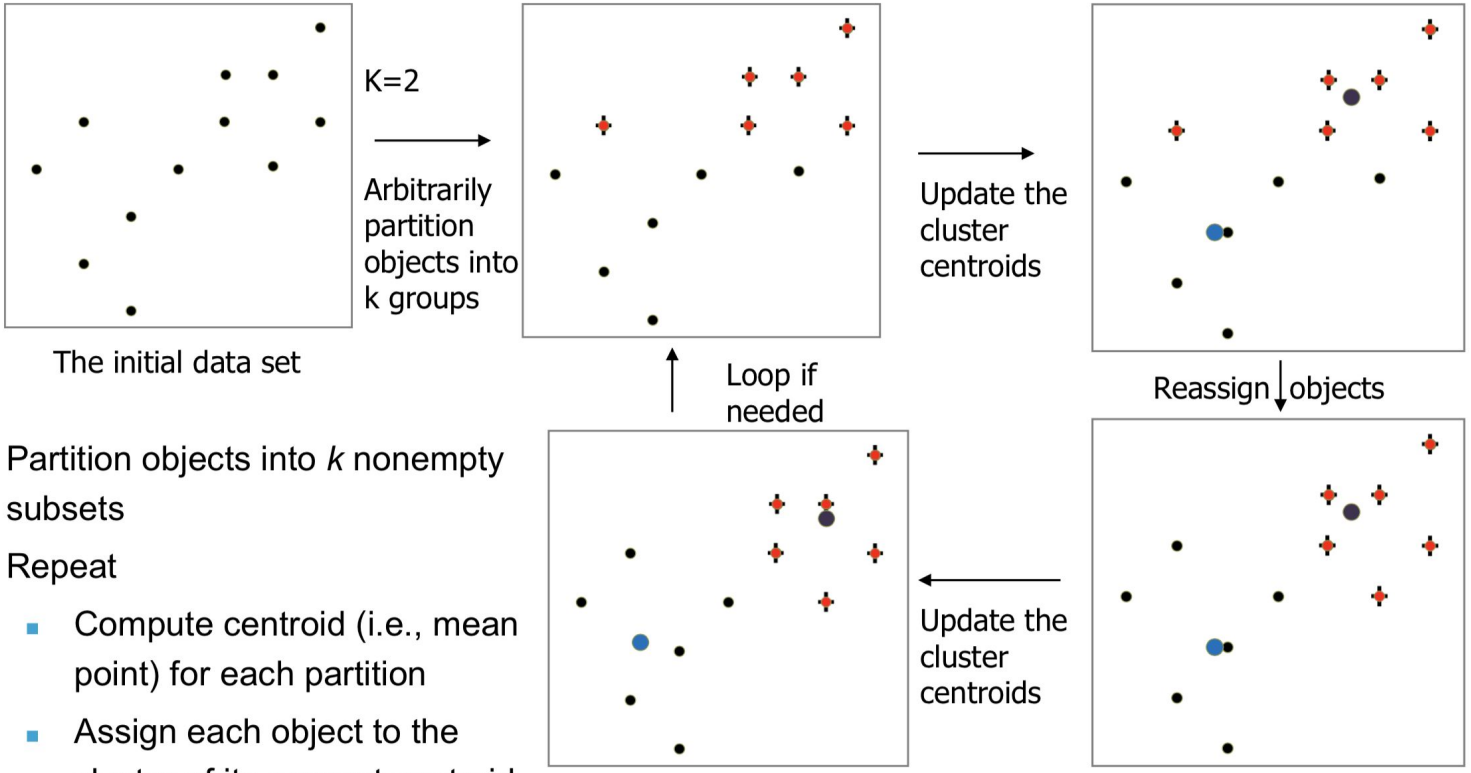
- Input / Output / Goal of clustering analysis
  - Large amount of unlabeled data in real life
- Supervised learning v.s. unsupervised learning
- Typical clustering algorithm examples:
  - K-means
  - Hierarchical clustering
  - DB-SCAN
  - Mixture models



K-means

- Demo 1: http://stanford.edu/class/ee103/visualizations/kmeans/kmeans.html
- Demo 2: https://www.naftaliharris.com/blog/visualizing-k-means-clustering/

K-means



- Partition objects into *k* nonempty subsets
- Repeat
  - Compute centroid (i.e., mean point) for each partition
  - Assign each object to the cluster of its nearest centroid
- Until no change

UCLA
Engineer Change.

K-means

- Key idea of K-means algorithms:
  - Step 1: Partition into k non-empty subsets (select K points as initial centroids)
  - Step 2: Iteration: Update mean point and assign object to cluster again
  - Step 3: Stop when converge
- Partition-based clustering methods
- Can be considered as a special case of GMM

K-means

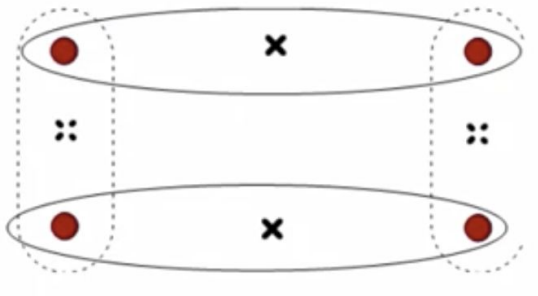
- Q1: Will K-means converge?
- Q2: Will different initialization of K-means generate different clustering results?

K-means

- Q1: Will K-means converge?
- A1: Yes.

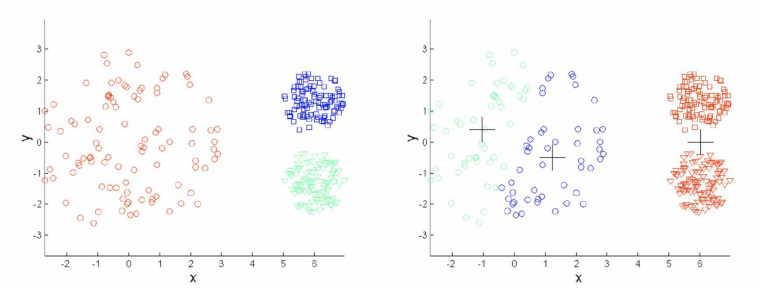
$$J = \sum_{j=1}^{k} \sum_{C(i)=j} d(x_i, c_j)^2$$

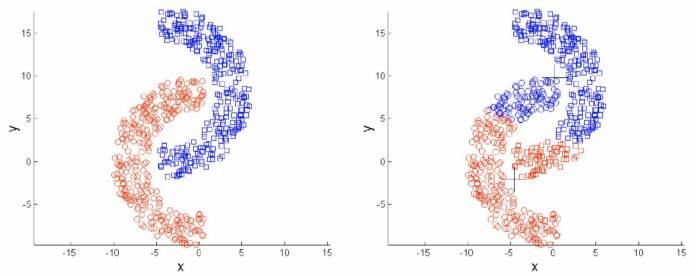
- Q2: Will different initialization of K-means generate different clustering results?
- A2: Yes. Initialization matters!



K-means: Discussion

- Efficiency: **O(tkn)** normally k,t are much smaller than n → efficient
- Can terminate at a local optimum
- Need to specify k (or take time to find best k)
- Sensitive to noisy data and outliers
- Different sizes and variances
- Not suitable to discover clusters with non-convex shapes
  - Can K-medoids help?
- Many variants of K-means:
  - K-means++, Genetics K-means, etc.

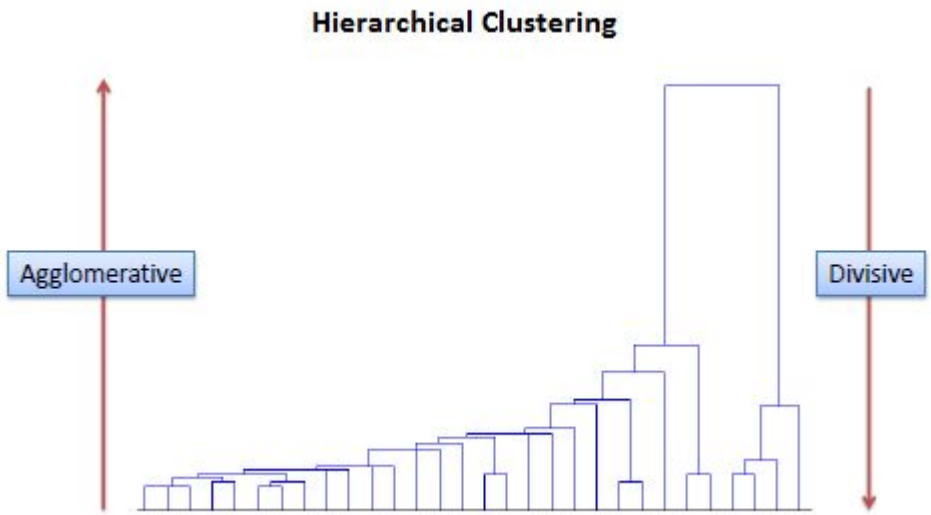




Hierarchical Clustering

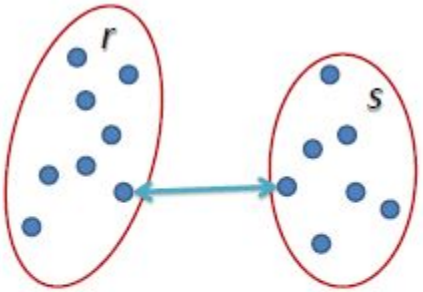
- Method
  - Divisive (Top-down)
  - Agglomerative (Bottom-up)

- Distance metrics
  - Single linkage
  - Complete linkage
  - Average linkage
  - Centroid
  - Medoid



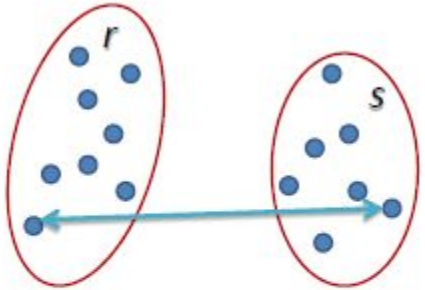
Hierarchical Clustering

- Single Linkage



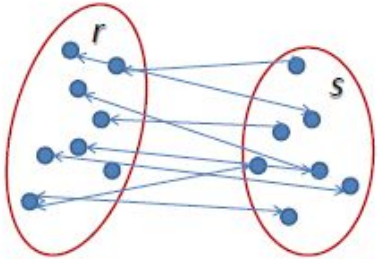
$$L(r,s) = \min(D(x_{ri}, x_{sj}))$$

- Complete Linkage



$$L(r,s) = \max(D(x_{ri}, x_{sj}))$$

- Average Linkage

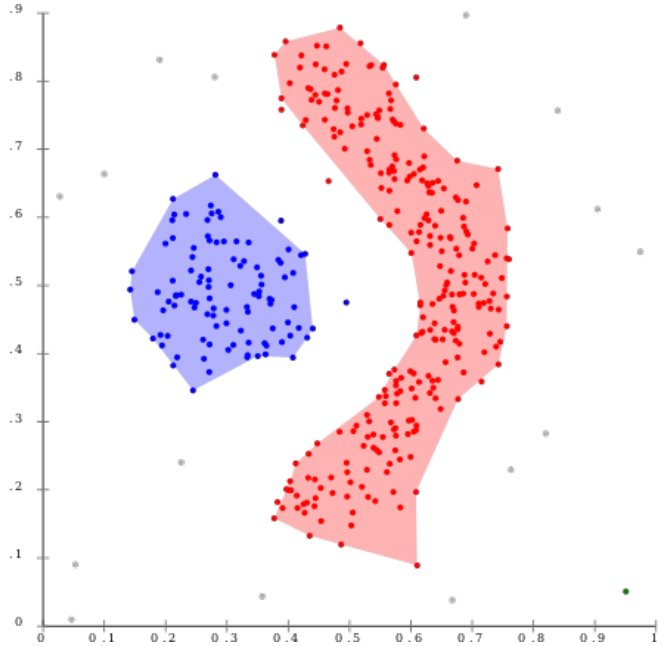


$$L(r,s) = \frac{1}{n_r n_s} \sum_{i=1}^{n_r} \sum_{j=1}^{n_s} D(x_{ri}, x_{sj})$$

UCLA
Engineer Change.

DB-SCAN

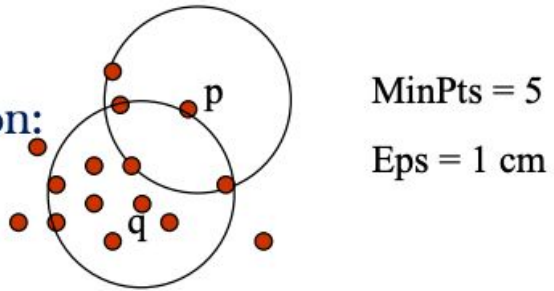
- Density-based clustering method
- Discover clusters of arbitrary shape
- Handle noise
- Demo



DB-SCAN

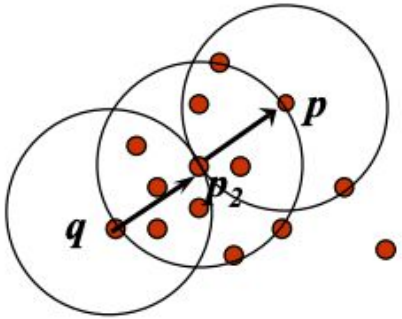
- Two parameters:
  - *Eps*: Maximum radius of the neighborhood
  - *MinPts*: Minimum number of points in an Eps-neighborhood of that point
- $N_{Eps}(q)$: {p belongs to D | dist(p,q) ≤ Eps}
- Directly density-reachable: A point *p* is directly density-reachable from a point *q* w.r.t. *Eps*, *MinPts* if
  - *p* belongs to $N_{Eps}(q)$
  - q is a core point, core point condition:

$$|N_{Eps}(q)| \geq MinPts$$

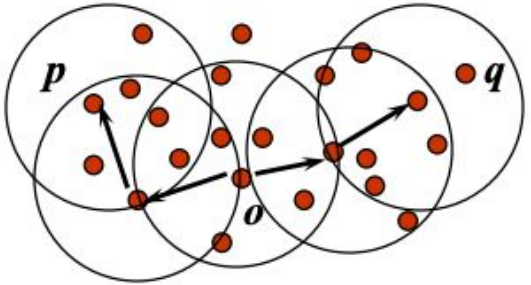


DB-SCAN

- Density-reachable:
  - A point p is density-reachable from a point q w.r.t. *Eps*, *MinPts* if there is a chain of points $p_1, \ldots, p_n$, $p_1 = q$, $p_n = p$ such that p_{i+1} is directly density-reachable from p_i

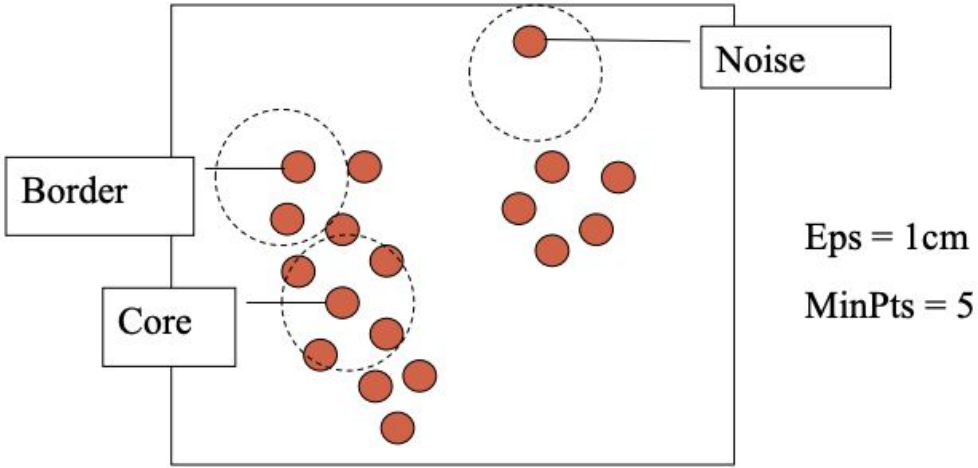


- Density-connected
  - A point p is density-connected to a point q w.r.t. *Eps*, *MinPts* if there is a point o such that both, p and q are density-reachable from o w.r.t. *Eps* and *MinPts*



DB-SCAN

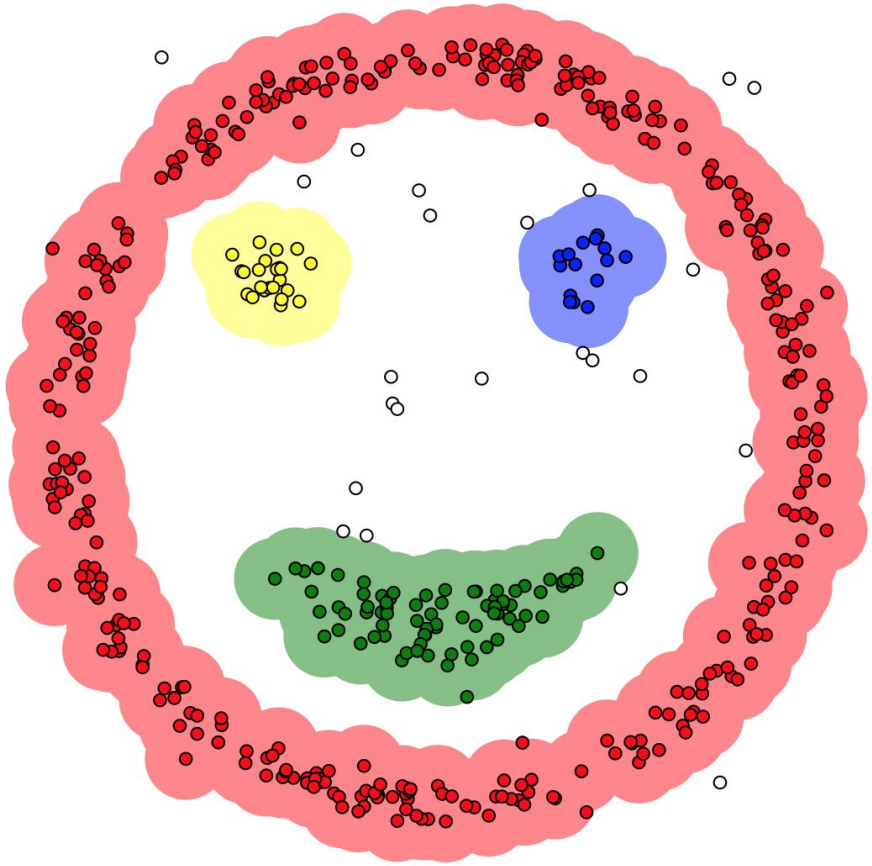
- Relies on a *density-based* notion of cluster: A *cluster* is defined as a maximal set of density-connected points
- Noise: object not contained in any cluster is noise
- Discovers clusters of arbitrary shape in spatial databases with noise



UCLA
Engineer Change.

DB-SCAN

- Density-based clustering method
- Demo
- Pros and Cons
  - It allows noise, so it is robust to outliers
  - It can figure out number of clusters automatically, as opposed to k-means
  - It can find arbitrarily shaped clusters
  - It is not deterministic, depending on the data processing order
  - It is not a good choice for clustering data sets with large differences in densities



epsilon = 1.00
minPoints = 4

Restart

UCLA Samueli Computer Science

Thank you!

Q & A