Clustering Aggregation

Boston University CS 506 - Lance Galletti

Clustering Aggregation

Some terminology:

Clustering: A group of clusters output by a clustering algorithm

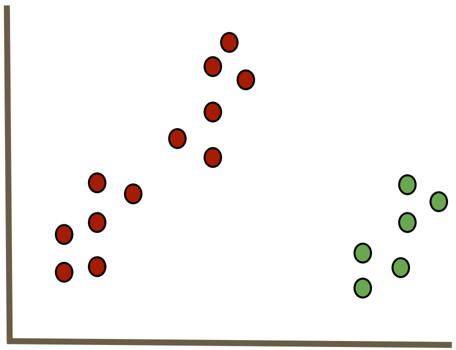
Cluster: A group of points

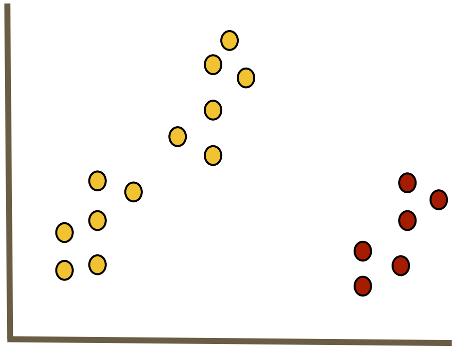
Clustering Aggregation

Goals:

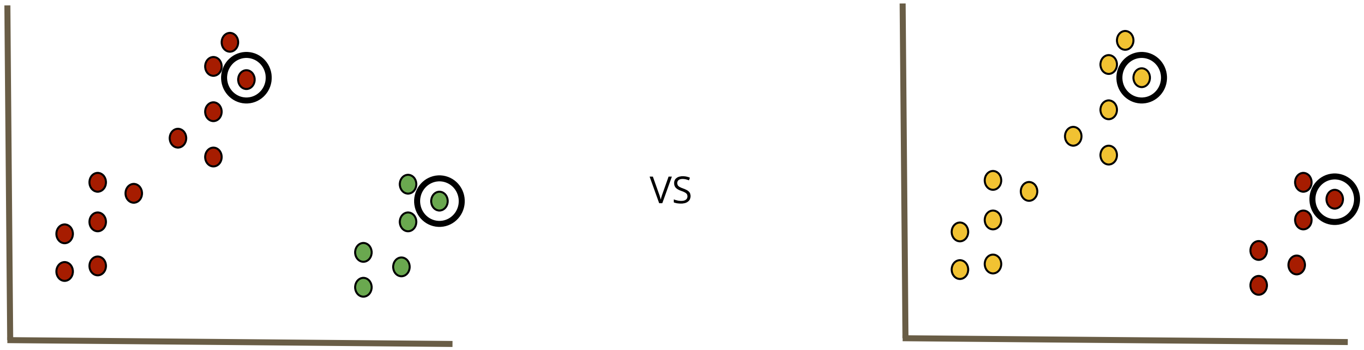
1. Compare clusterings
2. Combine the information from multiple clusterings to create a new clustering

Comparing Clusterings



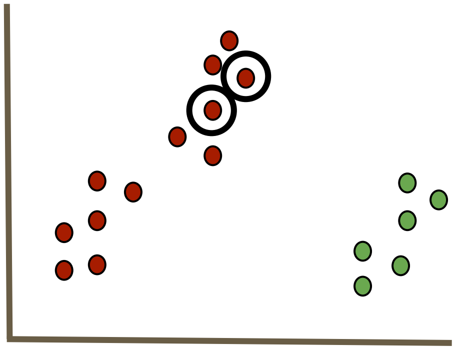


Comparing Clusterings

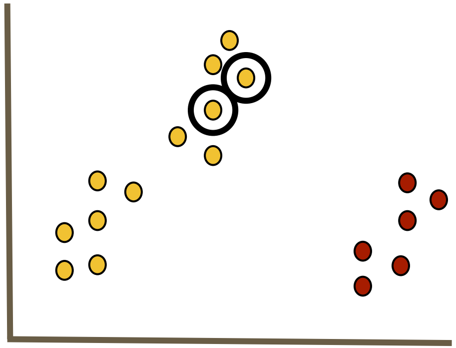


Are x and y clustered together in both P and C? Do P and C agree or disagree on whether x and y should be clustered together?

Comparing Clusterings



vs

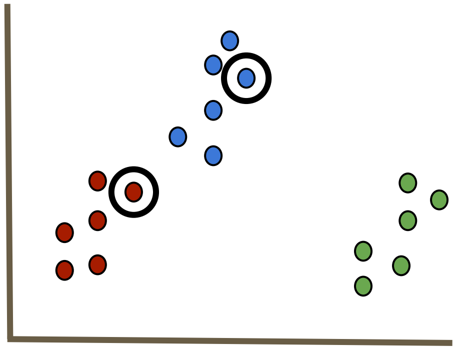


Are x and y clustered together in both P and C? Do P and C agree or disagree on whether x and y should be clustered together?

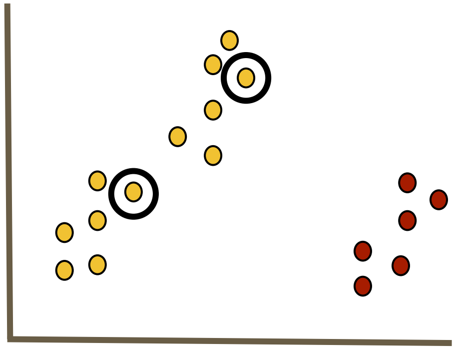
Comparing Clusterings

Are x and y clustered together in both P and C? Do P and C agree or disagree on whether x and y should be clustered together?

Comparing Clusterings



VS



Are x and y clustered together in both P and C? Do P and C agree or disagree on whether x and y should be clustered together?

Disagreement Distance

Given 2 clusterings P and C

$$D(P,C) = \sum_{x,y} \mathbb{I}_{P,C}(x,y)$$

where

$$\mathbb{I}_{P,C}(x,y) = \begin{cases} 1 & \text{if P \& C disagree on which clusters x \& y belong to} \\ 0 & \end{cases}$$

Disagreement occurs when:

One clustering groups two points together, while the other clustering separates them.

Disagreement Distance

	P	**C**
x_1	1	1
x_2	1	2
x_3	2	1
x_4	3	3
x_5	3	4

What is the disagreement distance between P and C?

Now, check each pair:

Pair	P (Same Cluster?)	C (Same Cluster?)	Disagreement?
x_1, x_2	Yes	No	**Yes**
x_1, x_3	No	Yes	**Yes**
x_1, x_4	No	No	No
x_1, x_5	No	No	No
x_2, x_3	No	No	No
x_2, x_4	No	No	No
x_2, x_5	No	No	No
x_3, x_4	No	No	No
x_3, x_5	No	No	No
x_4, x_5	Yes	No	**Yes**

Total **disagreements = 3 pairs**

Step 2: Compute Disagreement Distance

Total unique pairs:

$$\binom{5}{2} = \frac{5(5-1)}{2} = 10$$

$$\text{Disagreement Distance} = \frac{3}{10} = 0.3$$

Disagreement Distance

	P	C
x_1	1	a
x_2	1	b
x_3	2	a
x_4	3	c
x_5	3	d

x_2	x_1	1
x_3	x_1	1
x_4	x_1	0
x_5	x_1	0
x_3	x_2	0
x_4	x_2	0
x_5	x_2	0
x_4	x_3	0
x_5	x_3	0
x_4	x_5	1

Disagreement Distance

Is D(P, C) a distance function?

1. D(C, P) = 0 iff C = P
2. D(C, P) = D(P, C)
3. Triangle Inequality:

$$\mathbb{I}_{C_1,C_3}(x,y) \leq \mathbb{I}_{C_1,C_2}(x,y) + \mathbb{I}_{C_2,C_3}(x,y)$$

Since $\mathbf{I}_{C,P}$ can only be 0 or 1, the above can only be violated if

$\mathbf{I}_{x,y}(C_1,C_3) = 1$ **,** $\mathbf{I}_{x,y}(C_1,C_2) = 0$ **,** $\mathbf{I}_{x,y}(C_2,C_3) = 0$ is this possible?

Aggregate Clustering

Goal: From a set of clusterings $\mathbf{C_1, ..., C_m}$, generate a clustering $\mathbf{C^*}$ that minimizes:

$$\sum_{i=1}^{m} D(C^*, C_i)$$

The problem is equivalent to clustering categorical data

Aggregate Clustering

	City	Profession	Nationality
x_1	NY	Doctor	US
x_2	NY	Teacher	French
x_3	Boston	Lawyer	Canada
x_4	Boston	Doctor	US
x_5	LA	Lawyer	Canda
x_6	LA	Actor	French

Step 2: Compute Disagreement Distance

A **disagreeing pair** is one that is **clustered together in one scheme but not in another**.

Pair	City Clustering	Profession Clustering	Nationality Clustering	Disagreement Count
(x_1,x_2)	✅	❌	❌	2
(x_3,x_4)	✅	❌	❌	2
(x_5,x_6)	✅	❌	❌	2
(x_1,x_4)	❌	✅	✅	1
(x_3,x_5)	❌	✅	✅	1
(x_2,x_6)	❌	❌	✅	2

Step 3: Compute the Final Score

- **Total possible pairs**:

$$\binom{6}{2} = 15$$

- **Disagreeing pairs**: 10
- **Disagreement Distance**:

$$\frac{\text{Disagreeing Pairs}}{\text{Total Pa}} = \frac{10}{15} = 0.67$$

Aggregate Clustering

Benefits:

1. Can identify the best number of clusters (optimization function does not make any assumptions on the number of clusters)
2. Can handle / detect outliers (points where there is no consensus)
3. Improve robustness of the clustering algorithms - combining clusterings can produce a better result
4. Privacy preserving clustering (can compute aggregate clustering without sharing the data, need only share the assignments)

Aggregate Clustering

But… The problem is NP-Hard.

Often use approximations and heuristics to solve this problem.

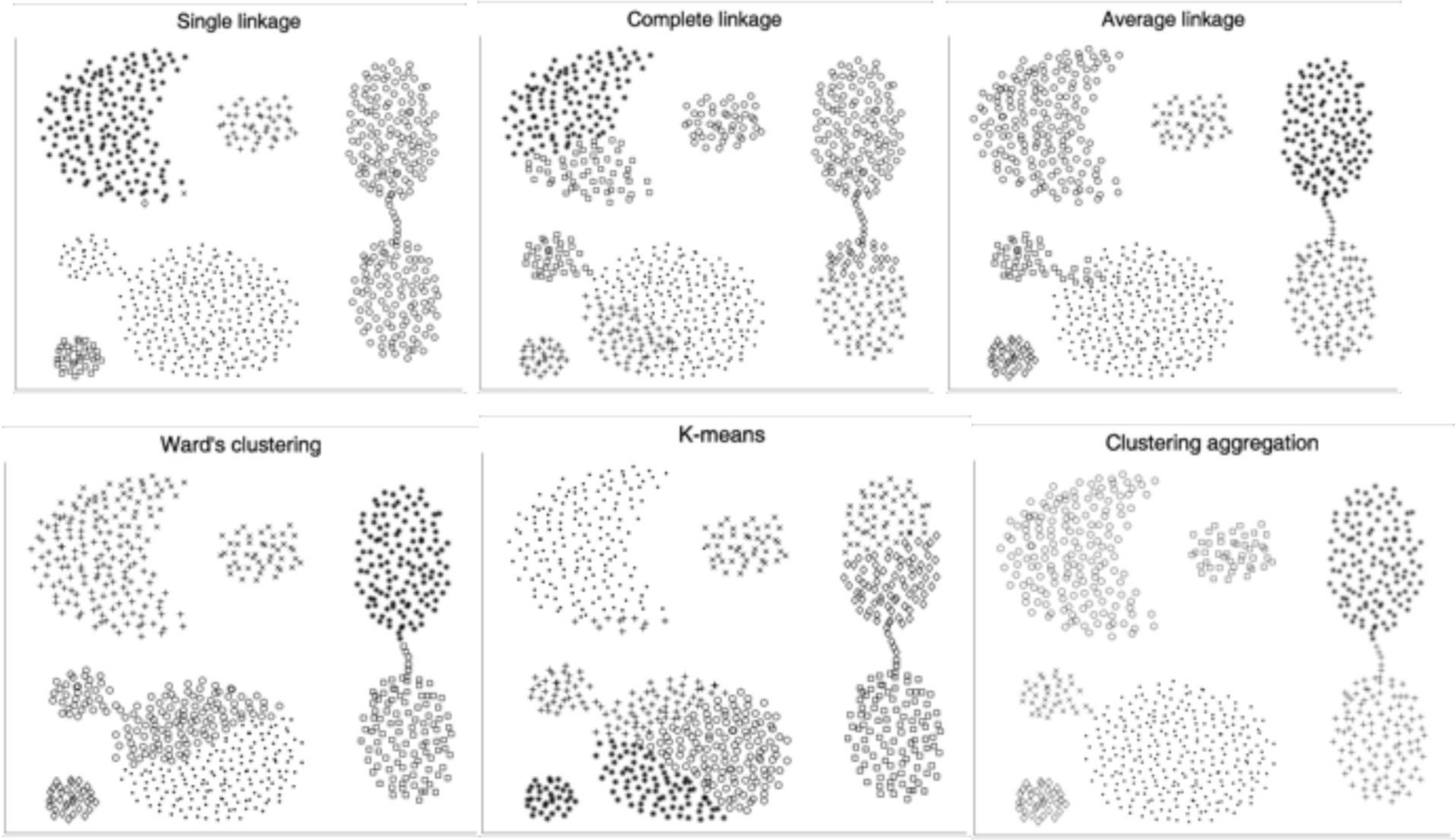
What about the majority rule?

This only works **if** it produces a clustering

Possible to have a majority saying:

1. x_1 & x_2 together
2. x_2 & x_3 together
3. x_1 & x_3 separate

elongated


Single linkage
Complete linkage
Average linkage
Ward's clustering
K-means
Clustering aggregation