Clustering

Lecture 7

Today: Learning Objectives

1. Discuss Unsupervised Learning methods using the Simpsons!
2. Represent “group” with distance of similarity
3. Learn some clustering algorithms: partitional and hierarchical
4. See how k-means algorithm works

1. Unsupervised Learning

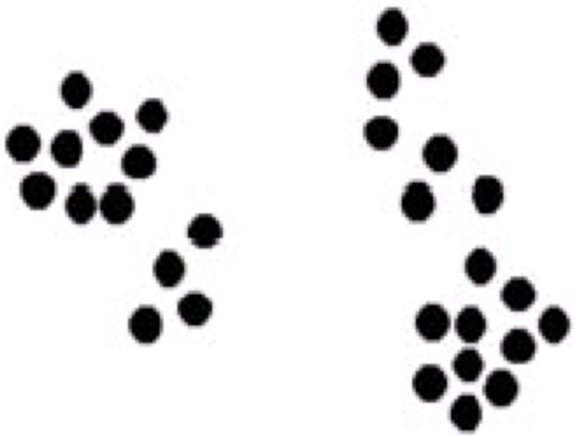
Unlabeled Dataset

	$\mathbf{x}_1$	$\mathbf{x}_2$	$\mathbf{x}_3$		$\mathbf{x}_n$
	total_bedrooms	population	households		median_income
$\mathbf{x}^{(1)}$	129.0	322.0	126.0	. . .	8.3252
$\mathbf{x}^{(2)}$	1106.0	2401.0	1138.0	. . .	8.3014
$\mathbf{x}^{(3)}$	190.0	496.0	177.0	. . .	7.2574
	⋮	⋮	⋮		⋮
$\mathbf{x}^{(m)}$	280.0	565.0	259.0	. . .	3.8462

$\mathbf{X}$

No $\mathbf{y}$, no label, no annotation is given!

An unlabeled cluster?



Are there any clusters or groups? How many?

What is each group? How to identify it?

What is clustering?

Clustering is the process of grouping a set of objects into classes of similar objects

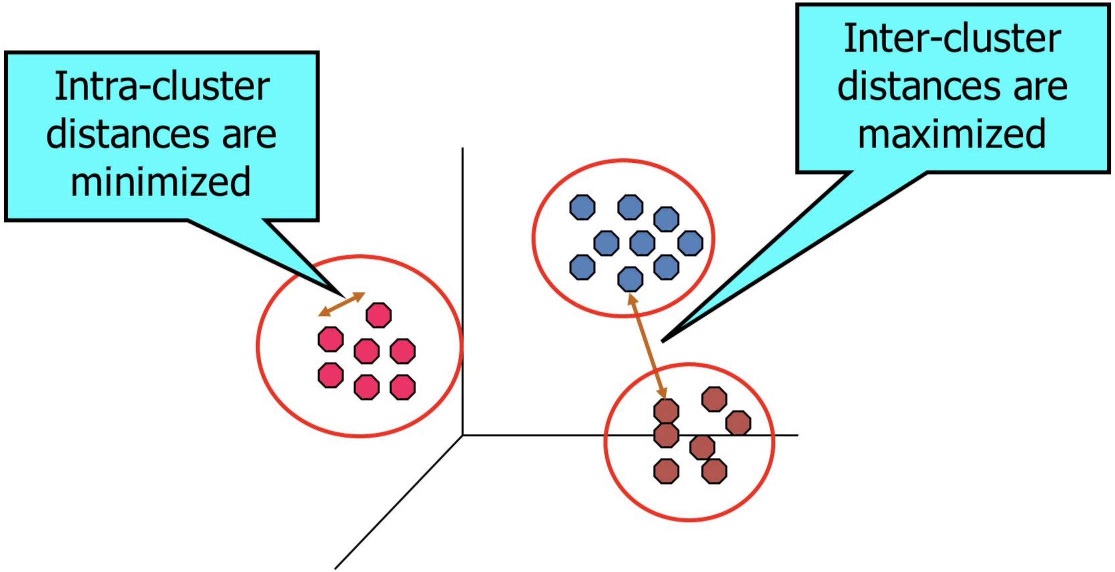
- High intra-class similarity
- Low inter-class similarity

As the **most common** form of unsupervised learning, it has many applications in Science, Engineering, Health,...

- Group genes that perform the same function
- Group individuals based on their activity on your website
- Detect any anomaly/ defect in a product
- Segment image into parts according to their color and texture
- ...

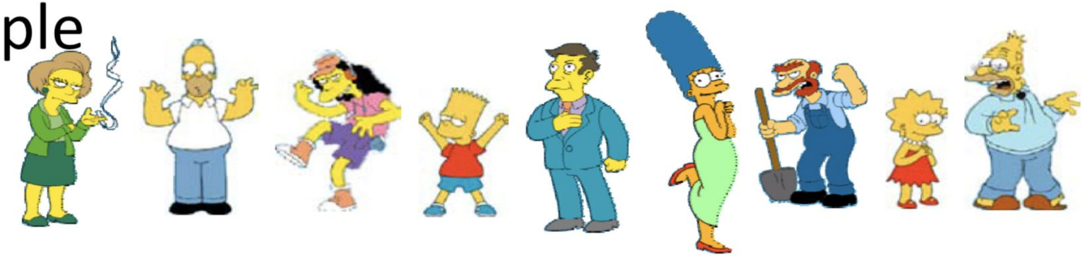
How to find good Clustering?

Find groups (clusters) of data points such that data points in a group will be similar (or related) to one another while different from (or unrelated to) the data points in other groups



Toy Examples

- People



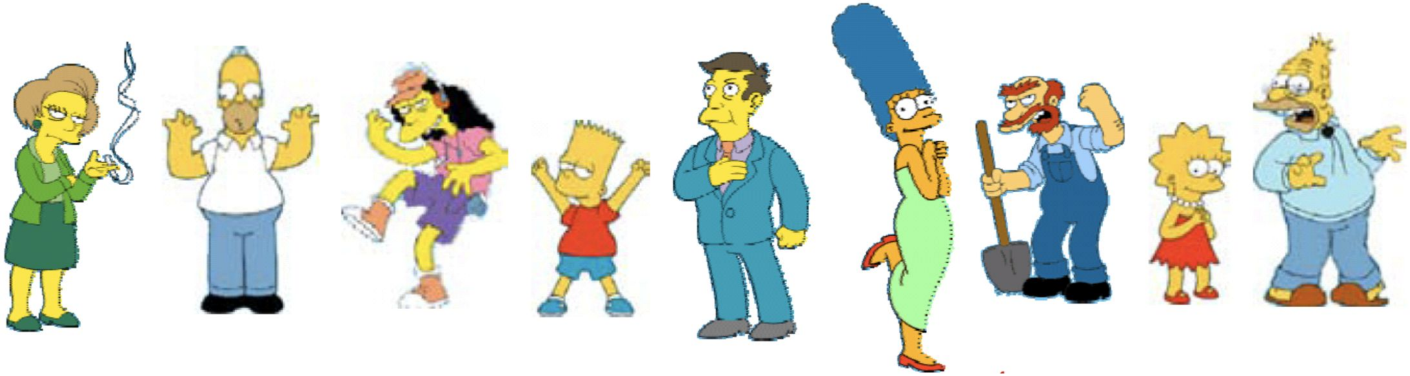
- Images

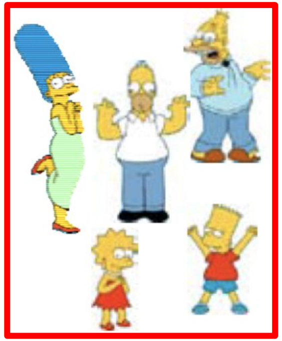
- Language

Piotr Pyotr Petros Pietro Pedro Pierre Piero Peter Peder Peka Peadar

- species

Natural Grouping ...is subjective





Family

Employees

Females

Males

Challenges of Clustering

- What is the natural grouping among these objects? (**groupness**)
- What makes objects “related”? (**distance of similarity**)
- Representation for objects? (**vector space/ normalization**)
- How many clusters? (**fixed number or data driven**)
- Clustering Algorithms? (**partitional or hierarchical**)
- Convergence? (**formal theory**)

2. Similarity Measures

Similarity

- Hard to define, but we know it when we see it!
- Depends on representation and algorithm. Easier to think in terms of a distance between two vectors.

Distance Measures

- **Symmetry**: `D(A,B) = D(B,A)`
  - Alex looks like Bob, then Bob looks like Alex
- **Self similarity**: `D(A,A) = 0`
  - Alex looks like Alex more than anyone else
- **Positivity Separation:** `D(A,B) = 0 iff A = B`
  - Otherwise you cannot tell anything apart
- **Triangular Inequality**: `D(A,B) ≤ D(A,C) + D(B,C)`
  - Loosely: Alex looks like Bob since Alex looks like Carl and Bob also looks like Carl

Minkowski (1864-1909) Metric

Suppose 2 objects x and y both have n features:

$$\mathbf{x} = (x_1, x_2, \ldots, x_n)$$
$$\mathbf{y} = (y_1, y_2, \ldots, y_n)$$

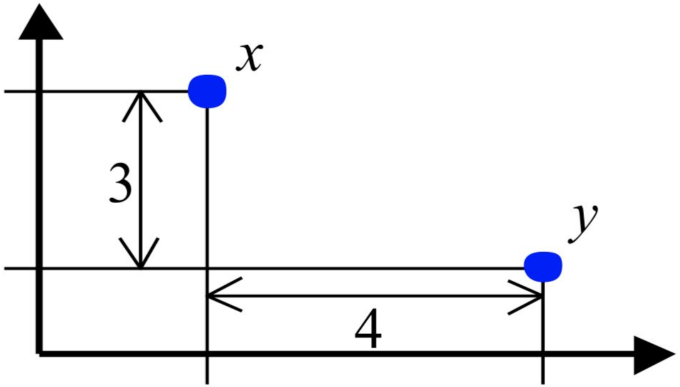
The Minkowski metric is defined by:

$$d(\mathbf{x}, \mathbf{y}) = \sqrt[p]{\sum_{i=1}^{n} |x_i - y_i|^p}$$

Most commonly used Minkowski Metrics:

1. Manhattan distance (p=1)
2. Euclidean distance (p =2)
3. “Sup” distance (p = ∞) $d(\mathbf{x}, \mathbf{y}) = \max_{1 \le i \le p} |x_i - y_i|$

An Example



$1:$ Euclidean distance: $\sqrt[2]{4^2+3^2}=5.$

$2:$ Manhattan distance: $4+3=7.$

$3:$ "sup" distance: $\max\{4,3\}=4.$

Hamming Distance

Manhattan distance is called the Hamming distance when all features are binary or discrete:

$$d_{Hamming}(\mathbf{x}, \mathbf{y}) = \sum_{i=1}^{n} |x_i - y_i|$$

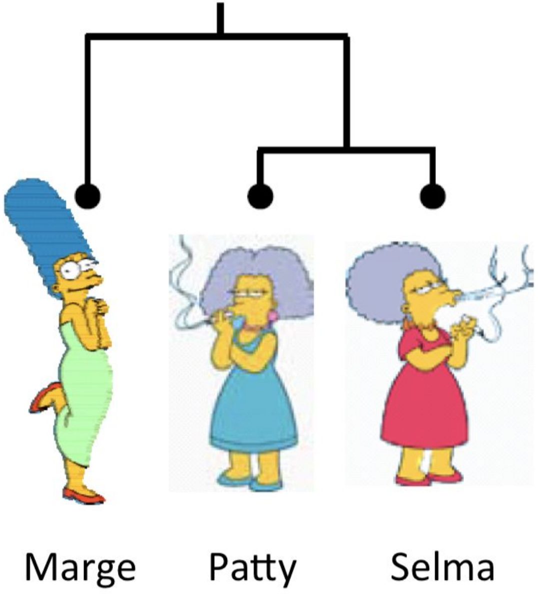
E.g., Gene Expression Levels Under 17 Conditions (1-High,0-Low)

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17
GeneA	0	1	1	0	0	1	0	0	1	0	0	1	1	1	0	0	1
GeneB	0	1	1	1	0	0	0	0	1	1	1	1	1	1	0	1	1

Hamming Distance: $\#(01) + \#(10) = 4 + 1 = 5.$

Edit (Transform) Distance

To measure the similarity between two objects, **transform** one object into the other, and measure how much effort it takes.



The distance between Patty and Selma.

Change dress color, 1 point
Change earring shape, 1 point
Change hair part, 1 point

D(Patty,Selma) = **3**

The distance between Marge and Selma.

Change dress color, 1 point
Add earrings, 1 point
Decrease height, 1 point
Take up smoking, 1 point
Lose weight, 1 point

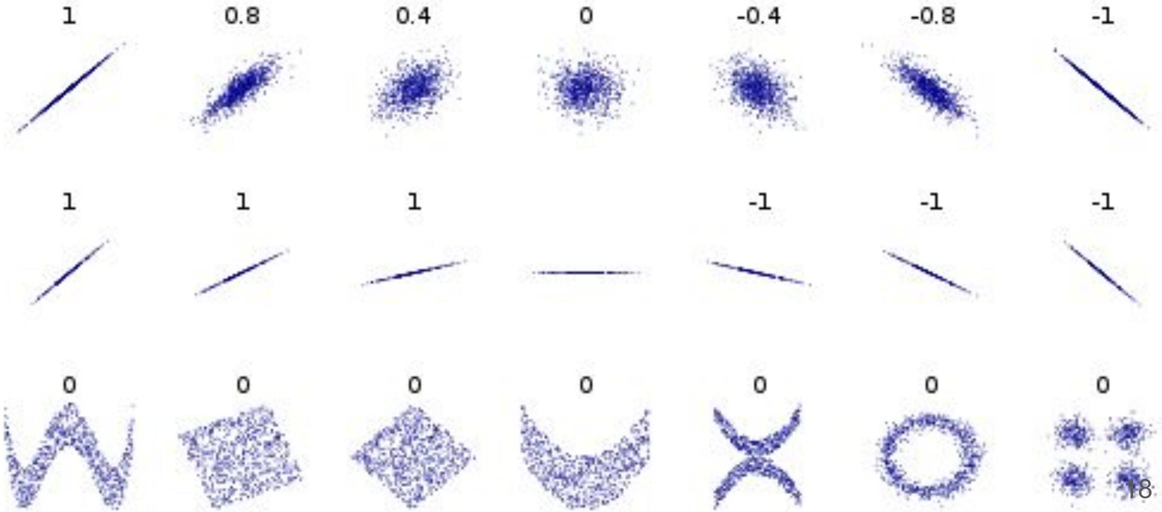
D(Marge,Selma) = **5**

Correlation to measure similarity

Pearson Correlation Coefficient to measure the linear correlation between x and y

Giving a value between -1 and 1 where 1 is total positive correlation, 0 is no correlation, and -1 is total negative correlation (but you all know this already)

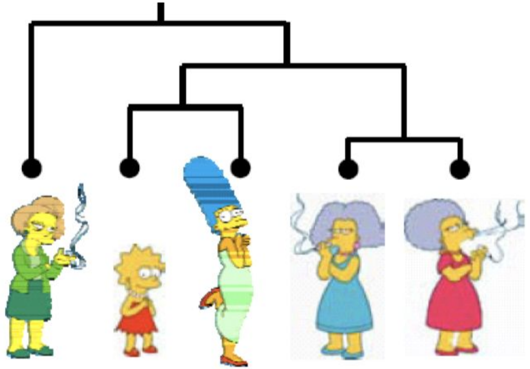
$$d(\mathbf{x},\mathbf{y}) = \frac{\sum_i (x^{(i)}-\bar{x})(y^{(i)}-\bar{y})}{\sqrt{\sum_i (x^{(i)}-\bar{x})^2 \sum_i (y^{(i)}-\bar{y})^2}}$$



2 types of Clustering Algorithms

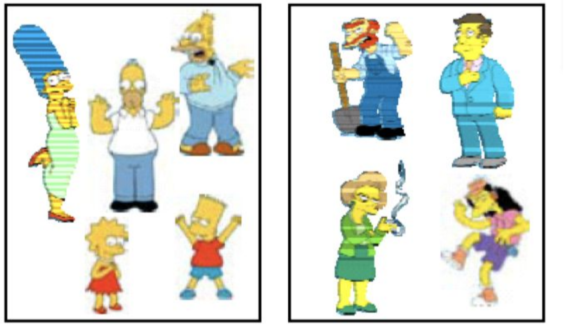
Hierarchical algorithms

- Bottom-up: agglomerative
- Top-down: divisive



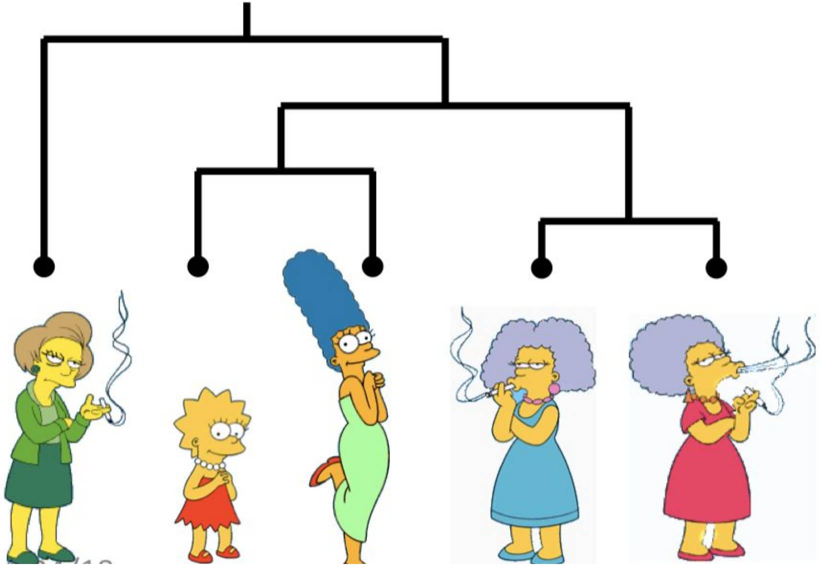
Partitional algorithms

- Usually start with a random (partial) partitioning
- Refine it iteratively (K-means)



3. Hierarchical Clustering (HAC)

Hierarchical Clustering



The number of dendrograms with n leafs $= (2n-3)!/[(2^{(n-2)})\,(n-2)!]$

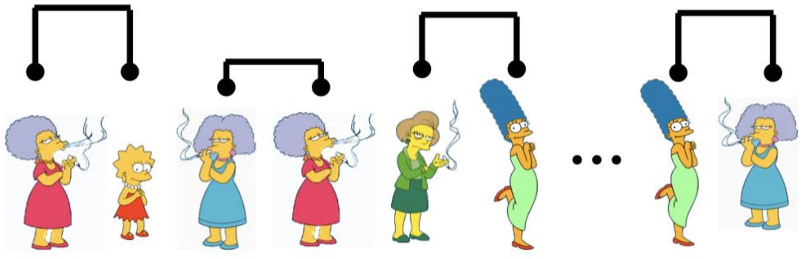
Number of Leafs	Number of Possible Dendrograms
2	1
3	3
4	15
5	105
...	…
10	34,459,425

Bottom-up: Starting with each item in its own cluster, find the best pair to merge into a new cluster. Repeat until all clusters are fused together.

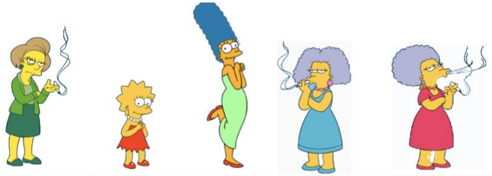
→ **A greedy local optimal solution!**

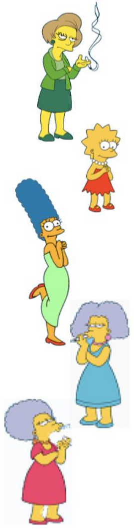
Bottom up Approach

- Begin with a distance matrix which contains the distances between every pair of objects
- Consider all possible merges



- Choose the best:



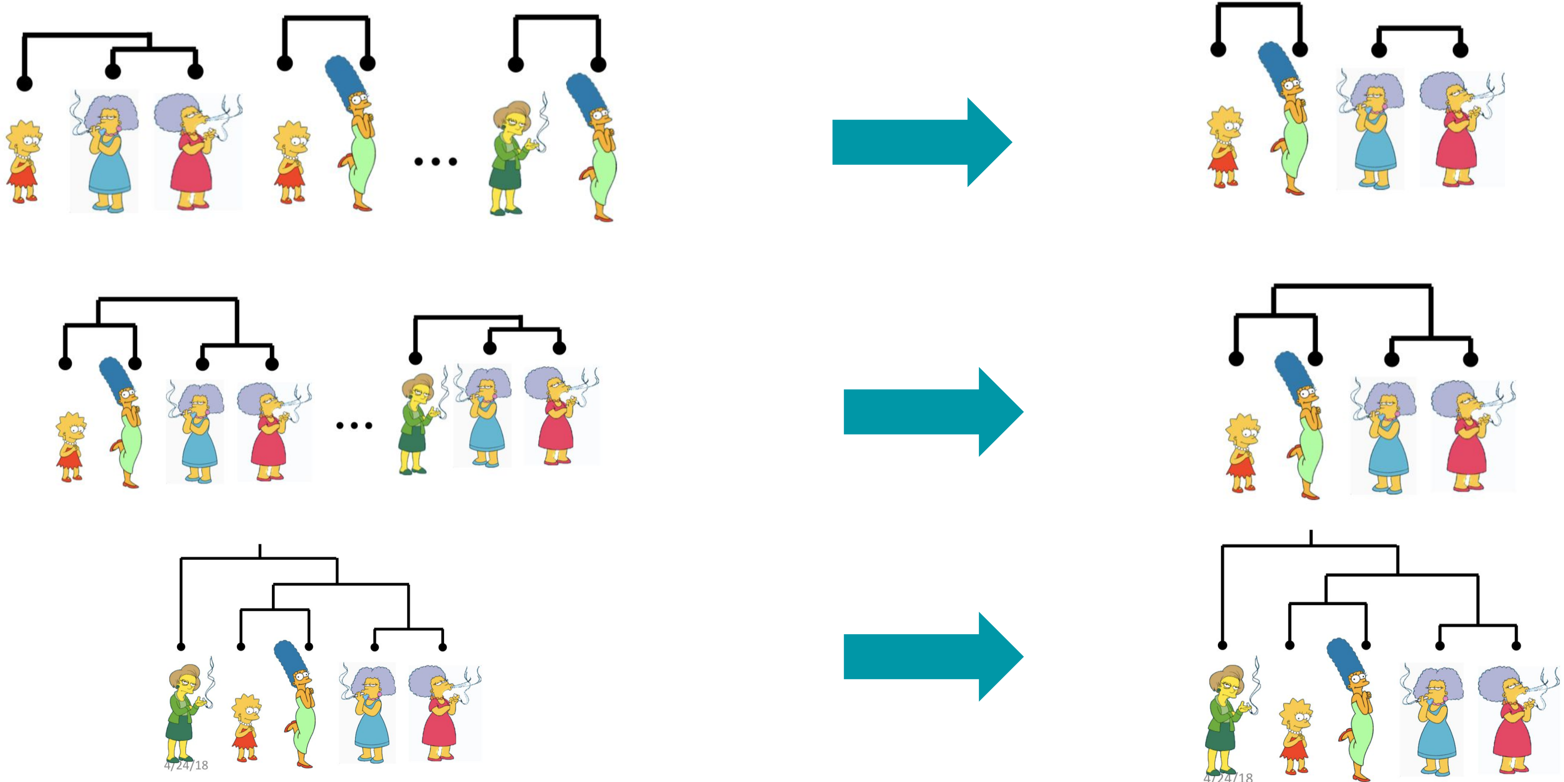


0	8	8	7	7
	0	2	4	4
		0	3	3
			0	1
				0

34

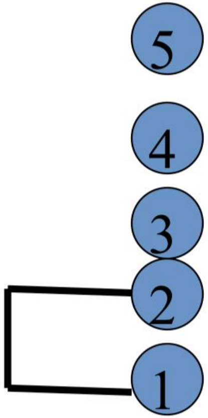
Bottom up Approach

- Continue to consider all possible mergers and choose the best:



Numerical Example

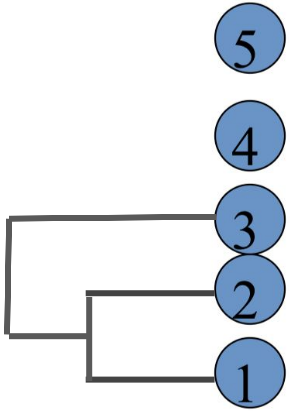
$$
\begin{array}{c|ccccc}
 & 1 & 2 & 3 & 4 & 5 \\
1 & 0 & & & & \\
2 & 2 & 0 & & & \\
3 & 6 & 3 & 0 & & \\
4 & 10 & 9 & 7 & 0 & \\
5 & 9 & 8 & 5 & 4 & 0
\end{array}
$$



Numerical Example

$$\begin{array}{c|c} & \begin{array}{ccccc} 1 & 2 & 3 & 4 & 5 \end{array} \\ \begin{array}{c} 1 \\ 2 \\ 3 \\ 4 \\ 5 \end{array} & \begin{bmatrix} 0 & & & & \\ 2 & 0 & & & \\ 6 & 3 & 0 & & \\ 10 & 9 & 7 & 0 & \\ 9 & 8 & 5 & 4 & 0 \end{bmatrix} \end{array} \Rightarrow \begin{array}{c|c} & \begin{array}{cccc} (1,2) & 3 & 4 & 5 \end{array} \\ \begin{array}{c} (1,2) \\ 3 \\ 4 \\ 5 \end{array} & \begin{bmatrix} 0 & & & \\ 3 & 0 & & \\ 9 & 7 & 0 & \\ 8 & 5 & 4 & 0 \end{bmatrix} \end{array}$$

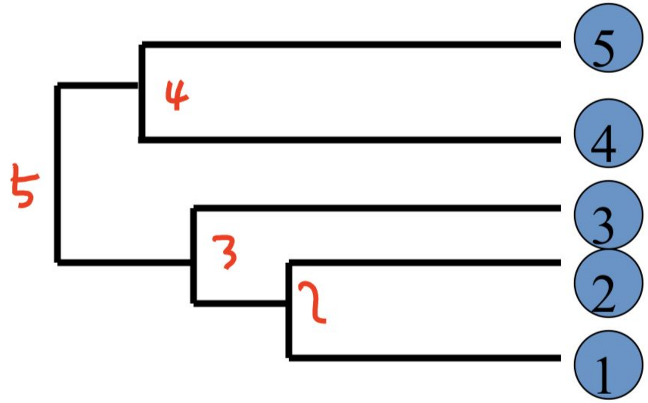
$$d_{(1,2),3} = \min\{ d_{1,3}, d_{2,3} \} = \min\{ 6,3 \} = 3$$
$$d_{(1,2),4} = \min\{ d_{1,4}, d_{2,4} \} = \min\{ 10,9 \} = 9$$
$$d_{(1,2),5} = \min\{ d_{1,5}, d_{2,5} \} = \min\{ 9,8 \} = 8$$



Numerical Example

$$\begin{array}{c|c} & \begin{array}{ccccc} 1 & 2 & 3 & 4 & 5 \end{array} \\ \begin{array}{c} 1 \\ 2 \\ 3 \\ 4 \\ 5 \end{array} & \begin{bmatrix} 0 & & & & \\ 2 & 0 & & & \\ 6 & 3 & 0 & & \\ 10 & 9 & 7 & 0 & \\ 9 & 8 & 5 & 4 & 0 \end{bmatrix} \end{array} \Rightarrow \begin{array}{c|c} & \begin{array}{cccc} (1,2) & 3 & 4 & 5 \end{array} \\ \begin{array}{c} (1,2) \\ 3 \\ 4 \\ 5 \end{array} & \begin{bmatrix} 0 & & & \\ 3 & 0 & & \\ 9 & 7 & 0 & \\ 8 & 5 & 4 & 0 \end{bmatrix} \end{array} \Rightarrow \begin{array}{c|c} & \begin{array}{ccc} (1,2,3) & 4 & 5 \end{array} \\ \begin{array}{c} (1,2,3) \\ 4 \\ 5 \end{array} & \begin{bmatrix} 0 & & \\ 7 & 0 & \\ 5 & 4 & 0 \end{bmatrix} \end{array}$$

$$d_{(1,2,3),(4,5)} = \min\{ d_{(1,2,3),4}, d_{(1,2,3),5} \} = 5$$



HAC Computational Complexity

In the first iteration, all hierarchical methods need to compute the similarity of all pairs of m individual instances which is $O(m^2n)$

In each subsequence n-2 merging, compute the distance between the most recently created cluster and all existing clusters $O(mn)$

It does **not** scale well: $O(m^2n)$ with a local optima

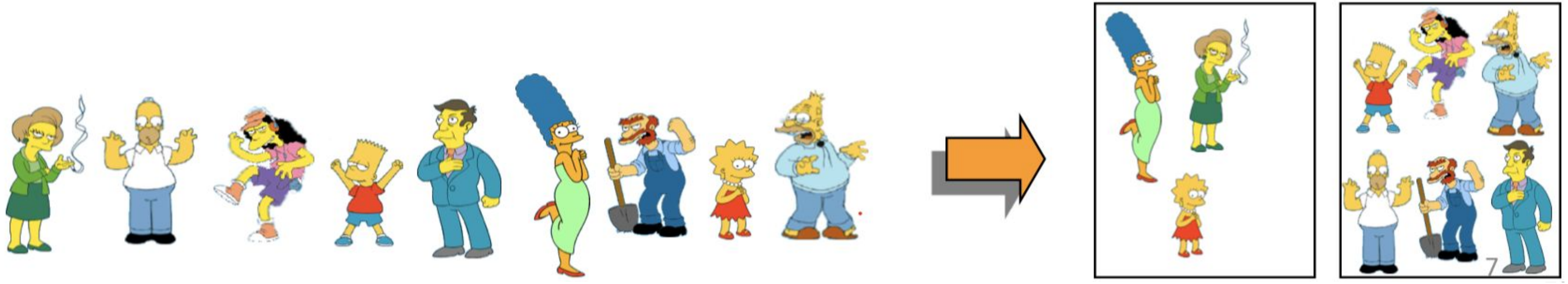
4. Partition Clustering (K-means)

Partitional Clustering

Non-hierarchical

Construct a partition of n objects into a set of k clusters

User has to specify the desired number of clusters k



Partitioning Algorithms

Given: a set of objects and the number k

Find: a partition of k clusters that optimizes a chosen partitioning criterion

- **Globally optimal:** exhaustively enumerate all partition (often too expensive)
- **Effective Heuristic methods:** k-means

K-means

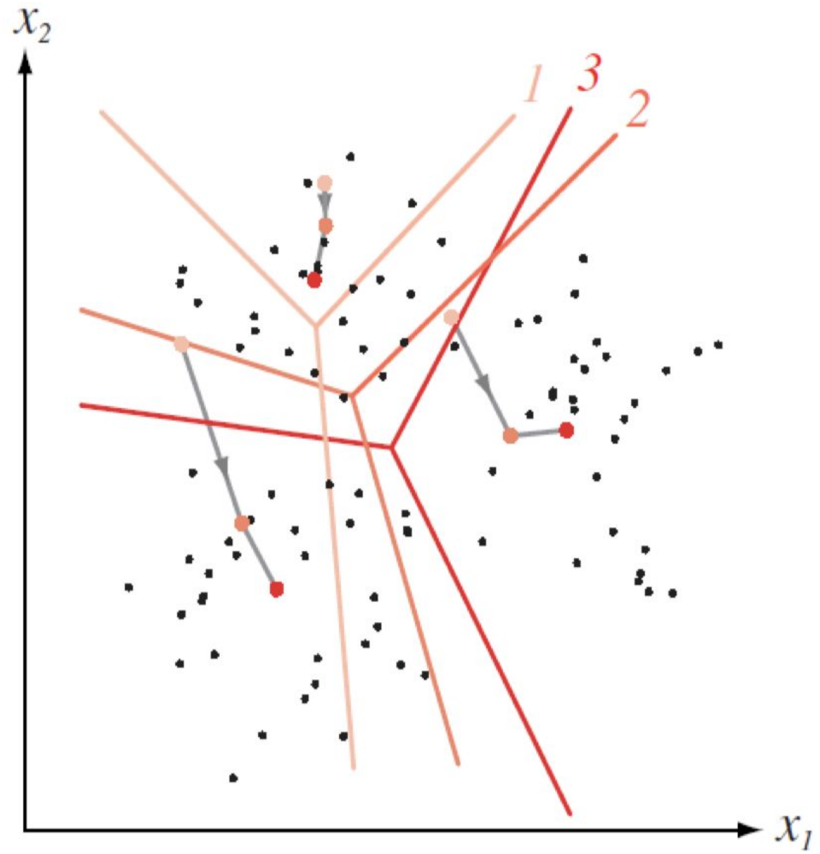
Proposed by Stuart Lloyd at Bell Labs (published in 1982)

1. Decide on a value for `k`
2. Initialize `k` cluster centers randomly
3. Decide the class memberships of the `m` objects by assigning them to the nearest cluster centroid (*aka* the mean)
4. Re-estimate the `k` cluster centroid, by assuming the membership found above are correct.
5. If none of the `m` objects change membership in the last iteration, exit. Otherwise, go to 3.

K-means Demo

http://stanford.edu/class/ee103/visualizations/kmeans/kmeans.html

How K-means partition

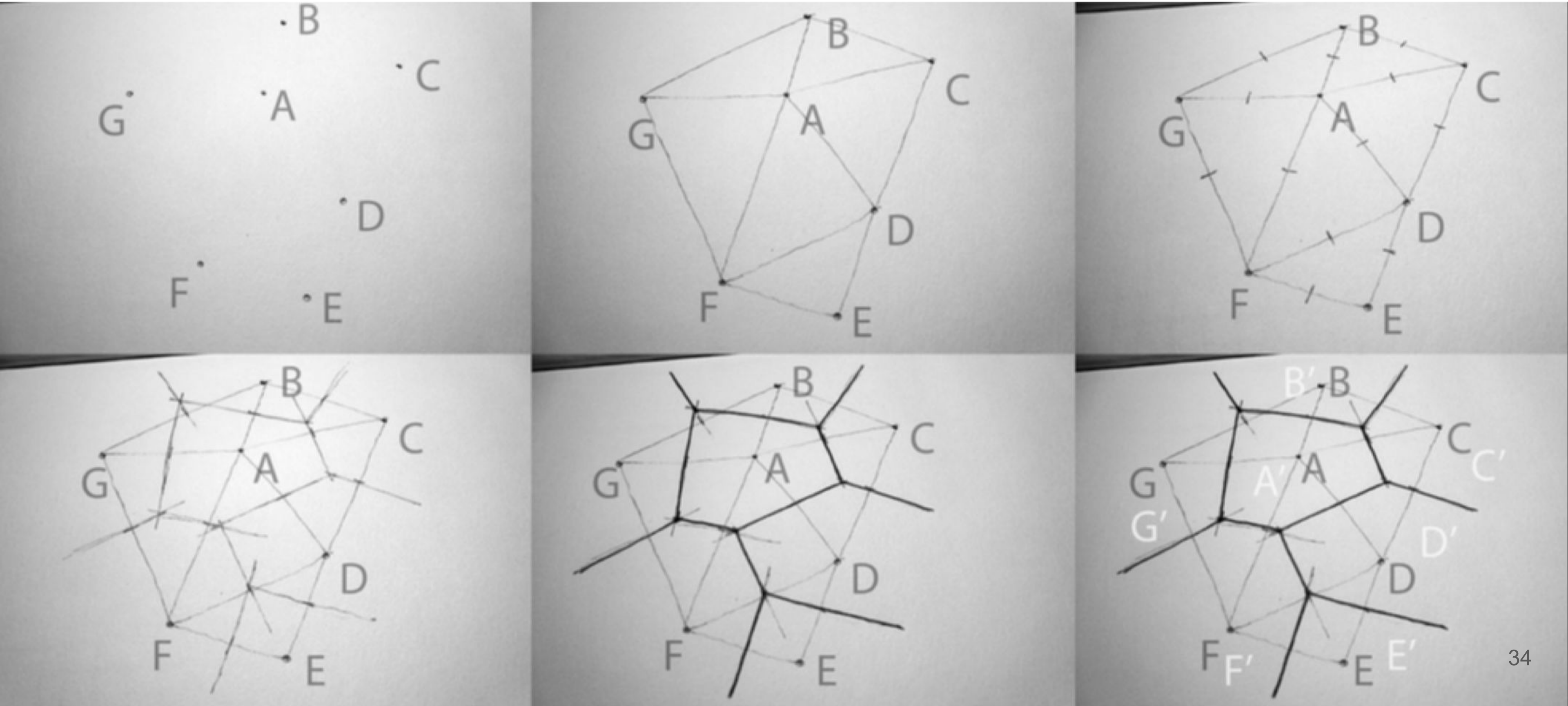


When K centroids are set, they partition the whole data space into K mutually exclusive subspace to form a partition.

A partition amounts to a Voronoi Diagram

Changing positions of centroids leads to a new partitioning.

How to draw Voronoi Diagram



Evaluation

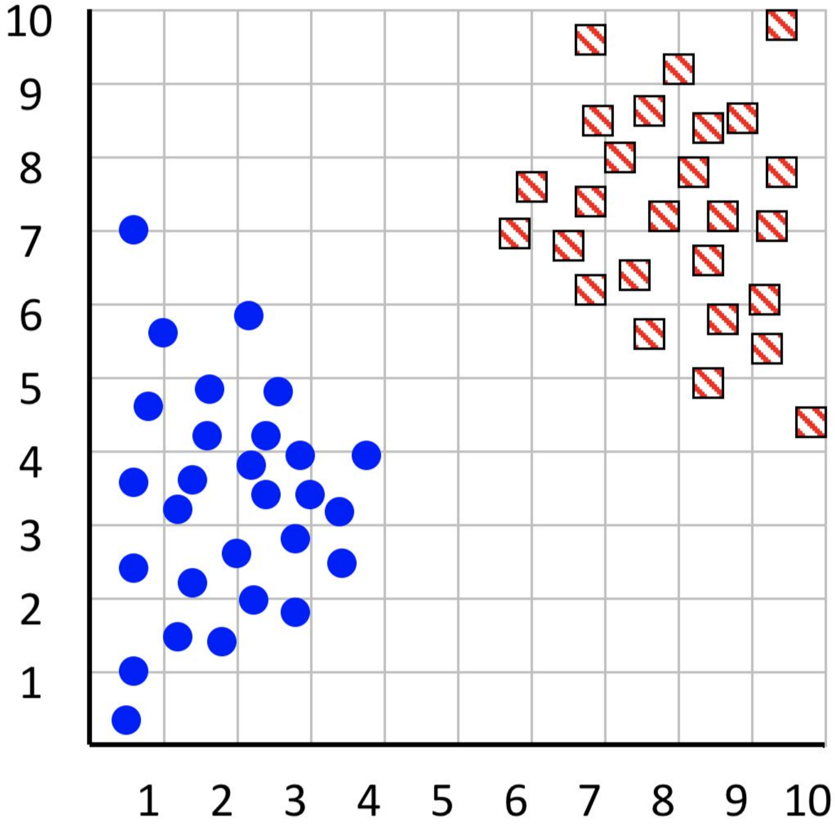
Internal criterion: A good clustering will produce high quality clusters in which:

- The intra cluster similarity is high
- The inter cluster similarity is low
- Depends on both the data representation and the similarity measure used

External Criteria for clustering quality

- Quality measured by its ability to discover some of the hidden patterns in data
- **If** given labels data, assess a clustering with respect to ground truth
    - **Purity**
    - **Entropy**

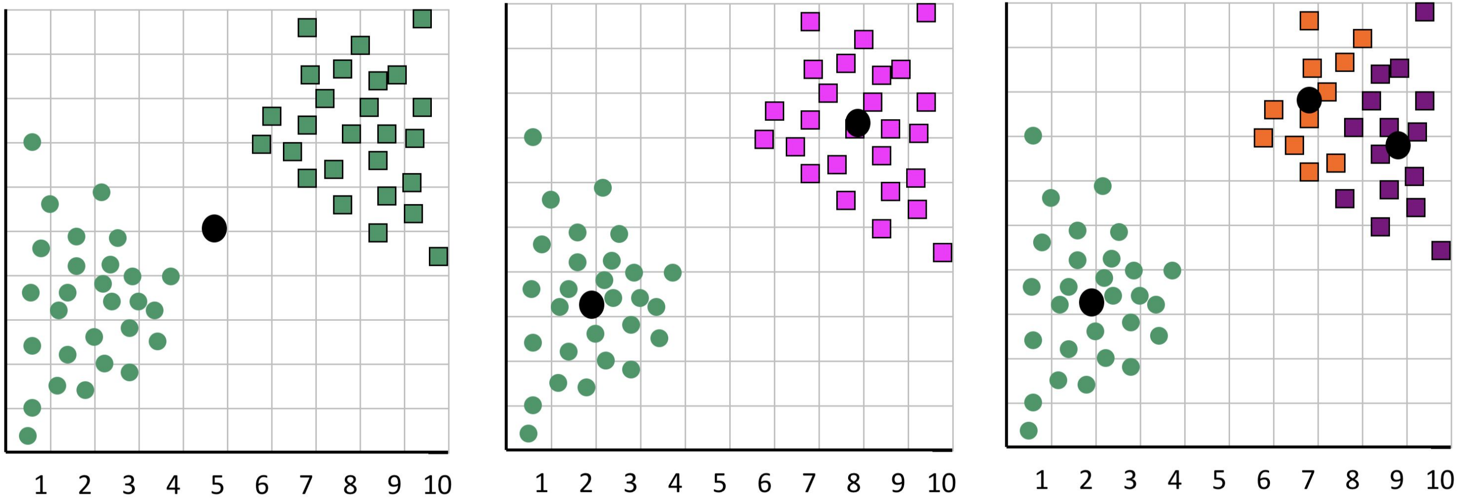
How to determine the right number of clusters



This is an **unsolved** problem, but we can still approximate by trying different k and measure the SSE (**inertia**) following objective function:

$$J = \arg\min_{C_j} \sum_{j=1}^{k} \sum_{i=1}^{m} ||x^{(i)} - C_j||^2$$

Different values of K



$k = 1$
$\Rightarrow J = 873.0$

$k = 2$
$\Rightarrow J = 173.1$

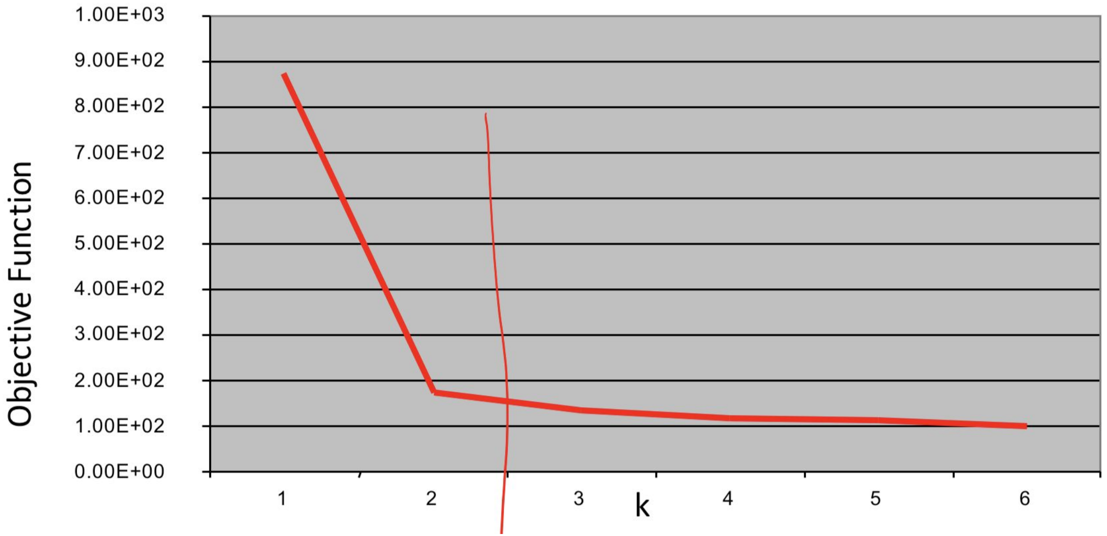
$k = 3$
$\Rightarrow J = 133.6$

$k = m$
$\Rightarrow J = 0$

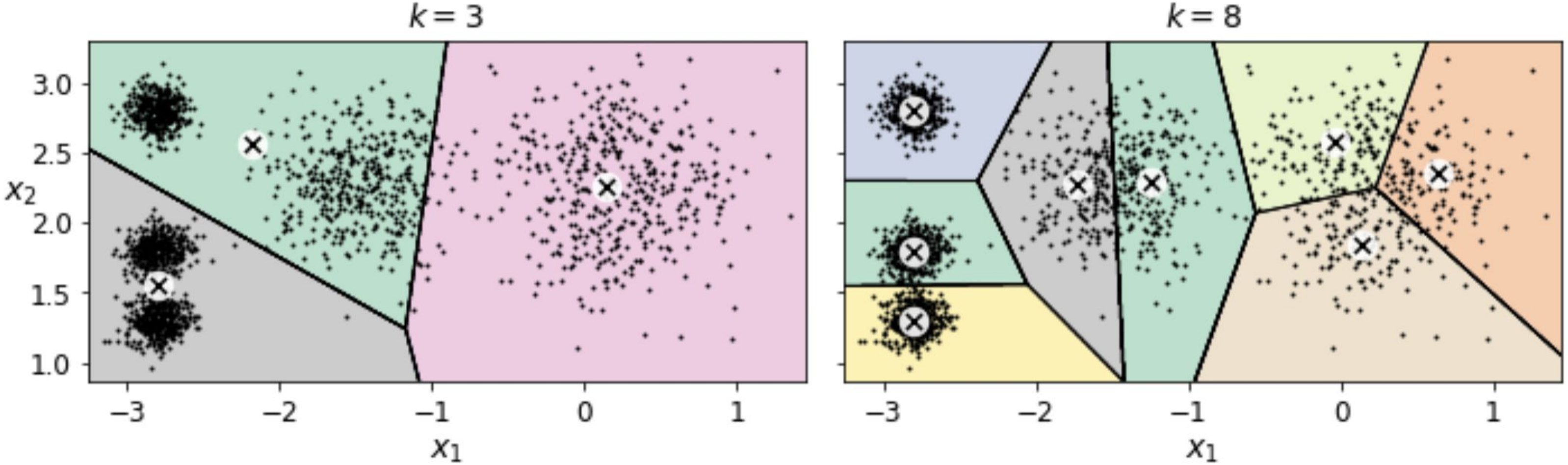
Finding the “Elbow”

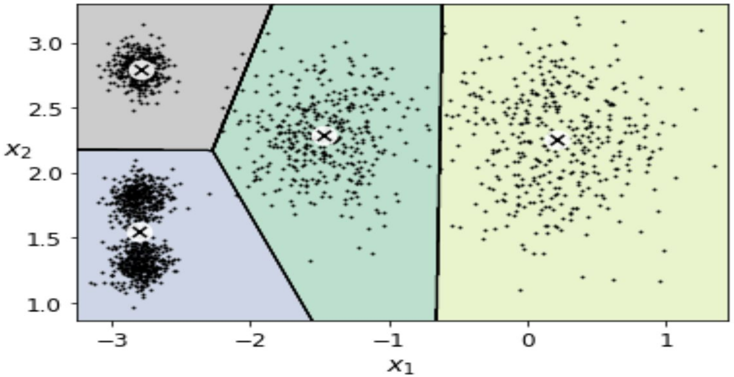
Plot the objective function for k from 1 to 6

The abrupt change at `k=2` highly suggestive of two clusters in the data, but the results are not always as clear cut in this toy example

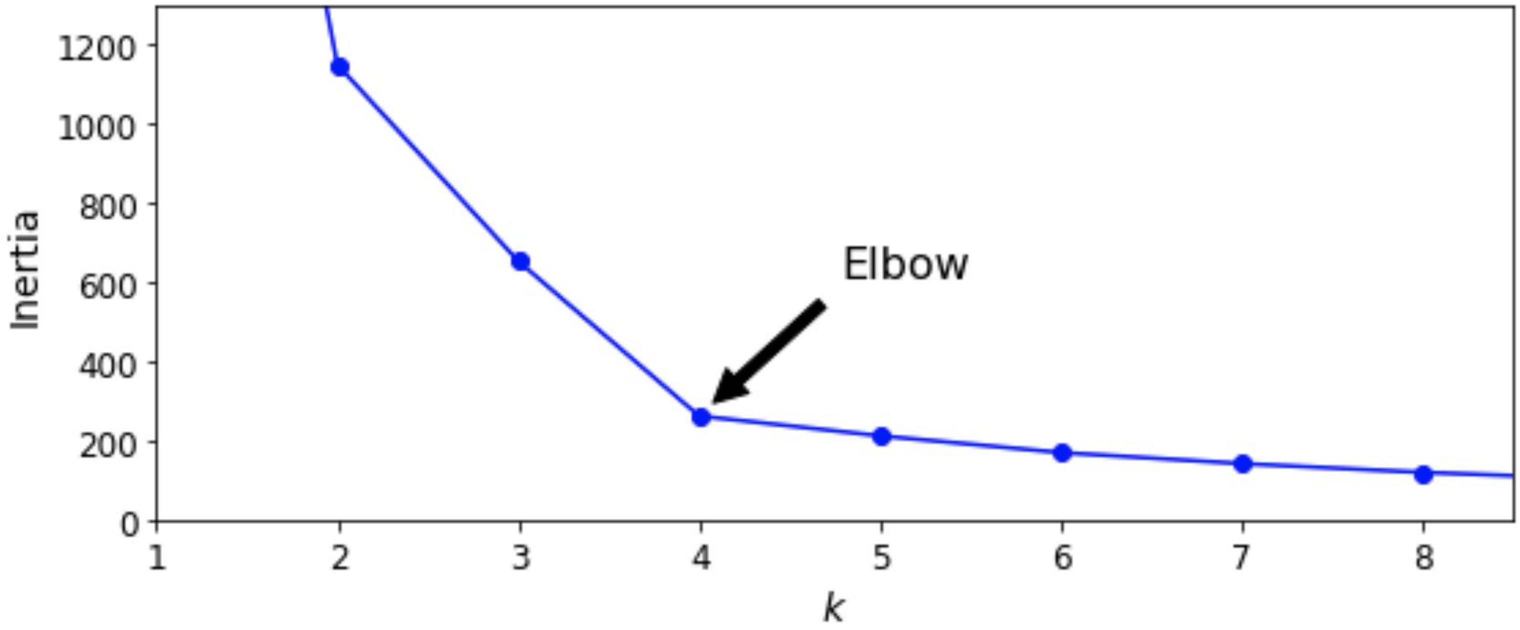


Another example





Another example



Time Complexity

Compute distance between two objects is `O(n)` where n is the dimensionality of vectors.

Reassigning k clusters is `O(kmn)` for distance computations

Recomputing centroids: Each object gets added once into a centroid `O(mn)`

Assuming reassigning and recomputing are each done for t iterations: `O(tkmn)`

Convergence

When K-means ever reach a fixed point?

- A state in which clusters **no longer change**

K-means is a special case of a general procedure known as **Expectation Maximization (EM)** algorithm.

- EM is known to converge
- Number of iterations could be large

Today: Learning Objectives

- ✓ Discuss Unsupervised Learning methods using the Simpsons!
- ✓ Represent “group” with distance of similarity
- ✓ Learning some clustering algorithms: partitional and hierarchical
- ✓ See how K-means algorithm works

Acknowledgements

Big thanks to these professors for providing part of the content for this lecture:

- Yanyun Qi, UVA
- Eric Xing, CMU
- Rong Jin, MSU