

# UVA CS 6316/4501 – Fall 2016 Machine Learning

## Lecture 19: Unsupervised Clustering (I)

Dr. Yanjun Qi

University of Virginia

Department of  
Computer Science

# Where are we ? ➔

## major sections of this course

- ❑ Regression (supervised)
- ❑ Classification (supervised)
  - ❑ Feature selection
- ❑ Unsupervised models
  - ❑ Dimension Reduction (PCA)
  - ➔ ❑ Clustering (K-means, GMM/EM, Hierarchical )
- ❑ Learning theory
- ❑ ~~Graphical models~~

	$X_1$	$X_2$	$X_3$
$s_1$			
$s_2$			
$s_3$			
$s_4$			
$s_5$			
$s_6$			

# An unlabeled Dataset X

a data matrix of  $n$  observations on  $p$  variables  $x_1, x_2, \dots, x_p$

**Unsupervised learning** = learning from raw (unlabeled, unannotated, etc) data, as opposed to supervised data where a classification label of examples is given

- **Data/points/instances/examples/samples/records:** [ rows ]
- **Features/attributes/dimensions/independent variables/covariates/predictors/regressors:** [ columns ]

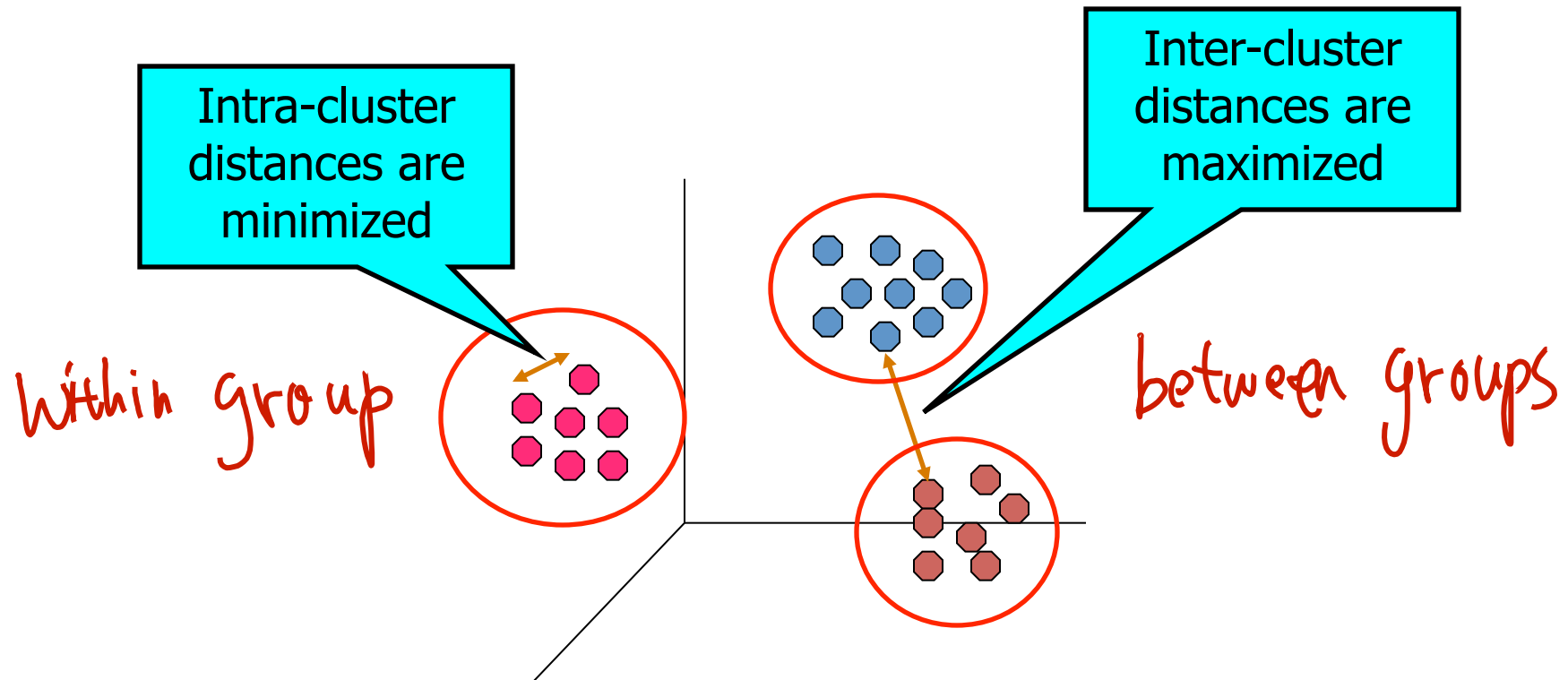
# Today: What is clustering?



- Are there any “groups”?
- What is each group ?
- How many ?
- How to identify them?

# What is clustering?

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

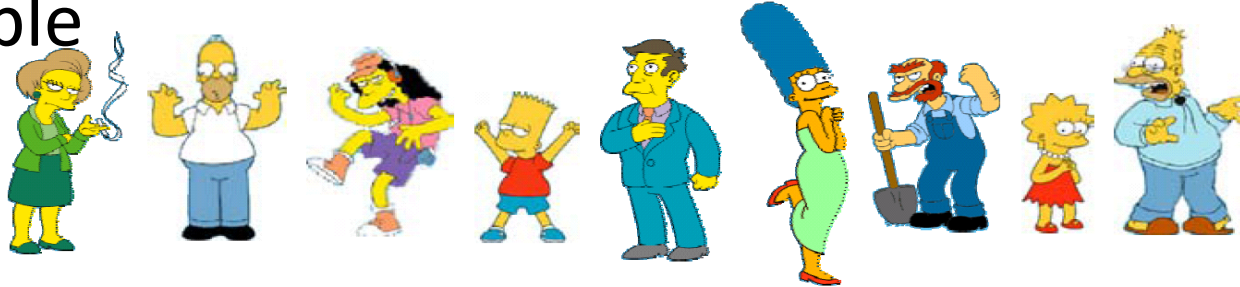


# What is clustering?

- Clustering: the process of grouping a set of objects into classes of similar objects
  - high intra-class similarity
  - low inter-class similarity
  - It is the commonest form of **unsupervised learning**
- A common and important task that finds many applications in Science, Engineering, information Science, and other places, e.g.
  - Group genes that perform the same function
  - Group individuals that has similar political view
  - Categorize documents of similar topics
  - Ideality similar objects from pictures

# Toy Examples

- People



- Images



- Language

*Piotr* *Pyotr* *Petros* *Pietro* *Pedro* *Pierre* *Piero* *Peter* *Peder* *Peka* *Peadar*

- species

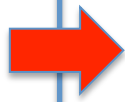


# Issues for clustering

- What is a natural grouping among these objects?
  - Definition of "groupness"
- What makes objects “related”?
  - Definition of "similarity/distance"
- Representation for objects
  - Vector space? Normalization?
- How many clusters?
  - Fixed a priori?
  - Completely data driven?
    - Avoid “trivial” clusters - too large or small
- Clustering Algorithms
  - Partitional algorithms
  - Hierarchical algorithms
- Formal foundation and convergence



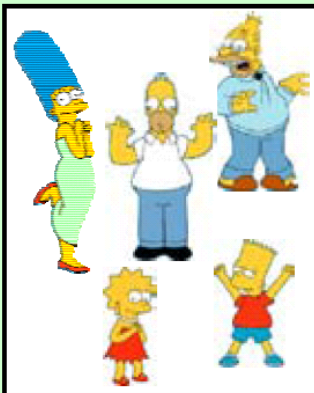
# Today Roadmap: clustering

- 
- Definition of "groupness"
  - Definition of "similarity/distance"
  - Representation for objects
  - How many clusters?
  - Clustering Algorithms
    - Partitional algorithms
    - Hierarchical algorithms
  - Formal foundation and convergence

# What is a natural grouping among these objects?



Clustering is subjective



Simpson's Family



School Employees

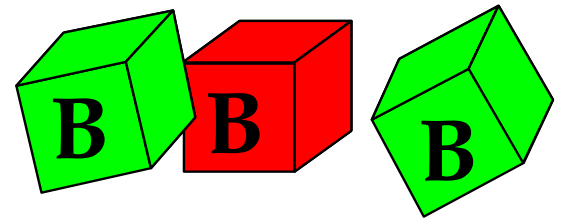
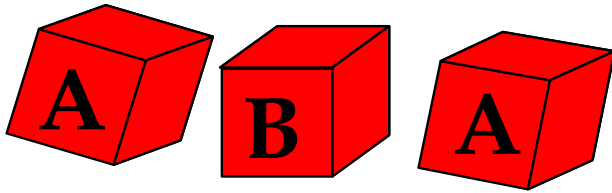


Females

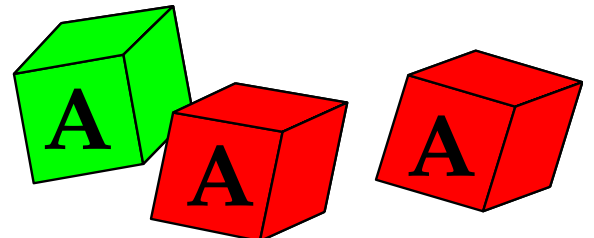
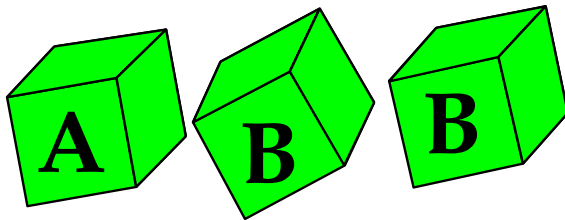


Males

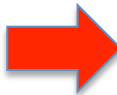
# Another example: clustering is subjective



Two possible Solutions...



# Today Roadmap: clustering

- 
- Definition of "groupness"
  - Definition of "similarity/distance"
  - Representation for objects
  - How many clusters?
  - Clustering Algorithms
    - Partitional algorithms
    - Hierarchical algorithms
  - Formal foundation and convergence

# What is Similarity?



Hard to define!  
But we know it  
when we see it

- The real meaning of similarity is a philosophical question. We will take a more pragmatic approach
- Depends on representation and algorithm. For many rep./alg., easier to think in terms of a distance (rather than similarity) between vectors.

# What properties should a distance measure have?

- $D(A,B) = D(B,A)$  *Symmetry*
- $D(A,A) = 0$  *Constancy of Self-Similarity*
- $D(A,B) = 0$  iff  $A = B$  *Positivity Separation*
- $D(A,B) \leq D(A,C) + D(B,C)$  *Triangular Inequality*

# Intuitions behind desirable properties of distance measure

- $D(A,B) = D(B,A)$  *Symmetry*
  - Otherwise you could claim "Alex looks like Bob, but Bob looks nothing like Alex"
- $D(A,A) = 0$  *Constancy of Self-Similarity*
  - Otherwise you could claim "Alex looks more like Bob, than Bob does"
- $D(A,B) = 0 \text{ iff } A = B$  *Positivity Separation*
  - Otherwise there are objects in your world that are different, but you cannot tell apart.
- $D(A,B) \leq D(A,C) + D(B,C)$  *Triangular Inequality*
  - Otherwise you could claim "Alex is very like Bob, and Alex is very like Carl, but Bob is very unlike Carl"

# Distance Measures: Minkowski Metric

- Suppose two object  $x$  and  $y$  both have  $p$  features

$$x = (x_1, x_2, \dots, x_p)$$

$$y = (y_1, y_2, \dots, y_p)$$

- The Minkowski metric is defined by

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

- Most Common Minkowski Metrics

1,  $r = 2$  (Euclidean distance )

$$d(x, y) = \sqrt{\sum_{i=1}^p |x_i - y_i|^2}$$

2,  $r = 1$  (Manhattan distance)

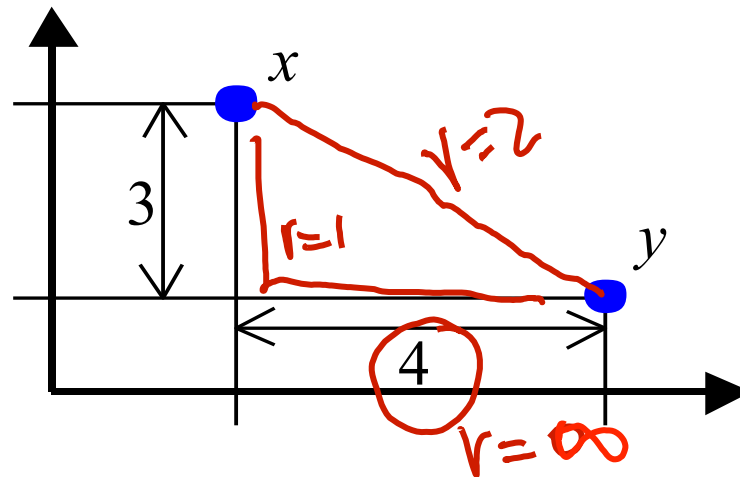
$$d(x, y) = \sum_{i=1}^p |x_i - y_i|$$

3,  $r = +\infty$  ("sup" distance )

$$d(x, y) = \max_{1 \leq i \leq p} |x_i - y_i|$$



# An Example



- 1: Euclidean distance:  $\sqrt{4^2 + 3^2} = 5.$
- 2: Manhattan distance:  $4 + 3 = 7.$
- 3: "sup" distance:  $\max\{4, 3\} = 4.$

# Hamming distance: binary features

- Manhattan distance is called *Hamming distance* when all features are binary.

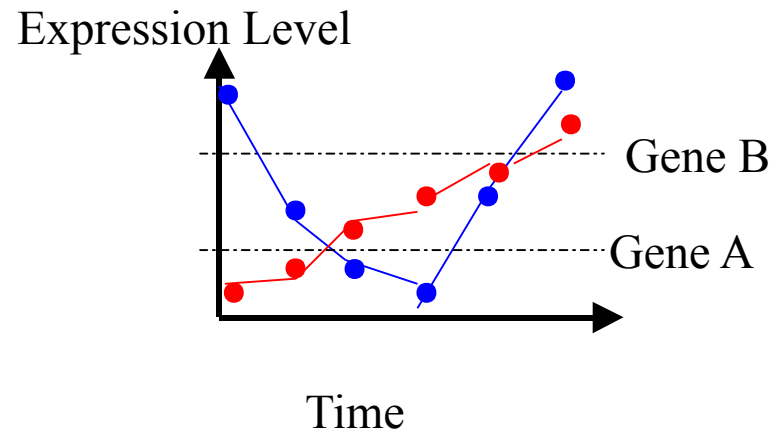
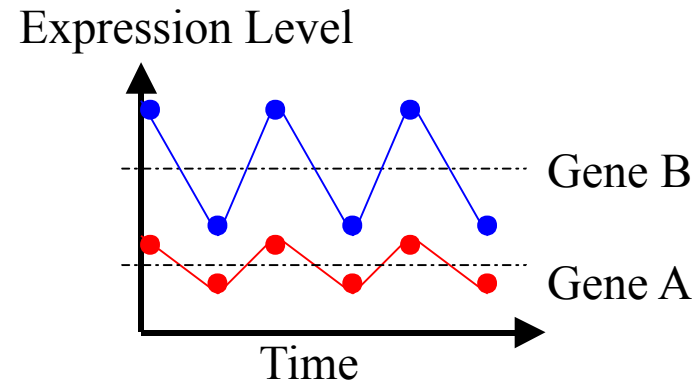
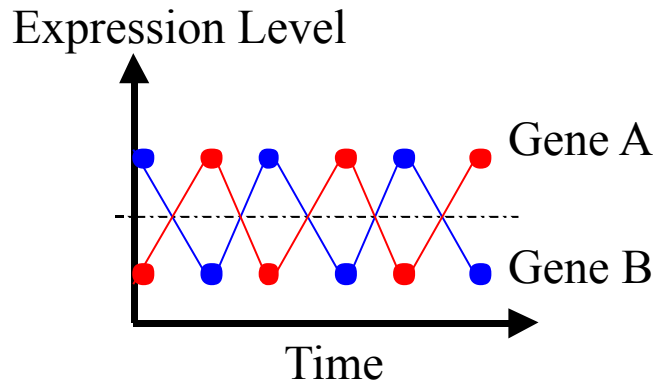
$$d(x, y) = \sum_{i=1}^p |x_i - y_i|$$

- 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
<i>GeneA</i>	0	1	1	0	0	1	0	0	1	0	0	1	1	1	0	0	1
<i>GeneB</i>	0	1	1	1	0	0	0	0	1	1	1	1	1	1	0	1	1

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

# Similarity Measures: Correlation Coefficient



Correlation is unit independent;

If you scale one of the objects ten times, you will get different euclidean distances and same correlation distances.

# Similarity Measures: Correlation Coefficient

- Pearson correlation coefficient

$$s(x, y) = \frac{\sum_{i=1}^p (x_i - \bar{x})(y_i - \bar{y})}{\sqrt{\sum_{i=1}^p (x_i - \bar{x})^2 \times \sum_{i=1}^p (y_i - \bar{y})^2}}$$

$$\text{where } \bar{x} = \frac{1}{p} \sum_{i=1}^p x_i \text{ and } \bar{y} = \frac{1}{p} \sum_{i=1}^p y_i.$$

$$|s(x, y)| \leq 1$$

Correlation is unit independent

- Measuring the **linear correlation** between two sequences,  $x$  and  $y$ ,
- giving a value between  $+1$  and  $-1$  inclusive, where  $1$  is total positive **correlation**,  $0$  is no **correlation**, and  $-1$  is total negative **correlation**.

- Special case: cosine distance

$$s(x, y) = \frac{\vec{x} \cdot \vec{y}}{|\vec{x}| \cdot |\vec{y}|}$$

# Edit Distance:

A generic technique for measuring similarity

- To measure the similarity between two objects, transform one of the objects into the other, and **measure how much effort it took**. The measure of effort becomes the distance measure.

The distance between Patty and Selma.

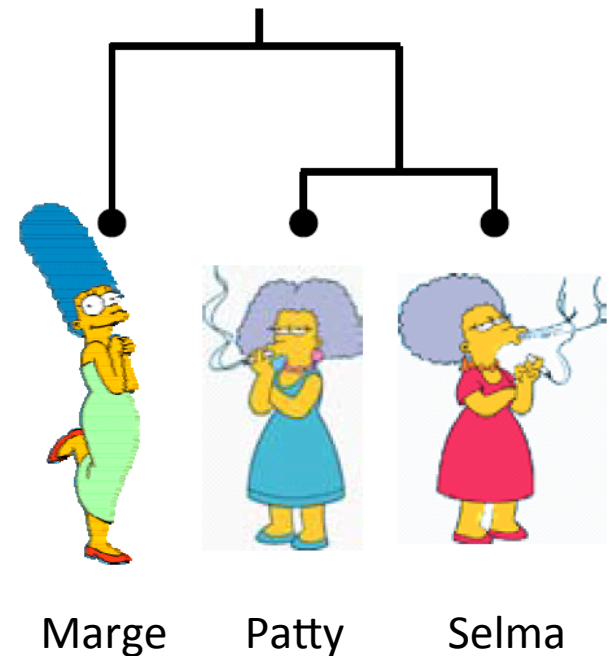
Change dress color, 1 point  
Change earring shape, 1 point  
Change hair part, 1 point

$$D(\text{Patty}, \text{Selma}) = 3$$

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

$$D(\text{Marge}, \text{Selma}) = 5$$



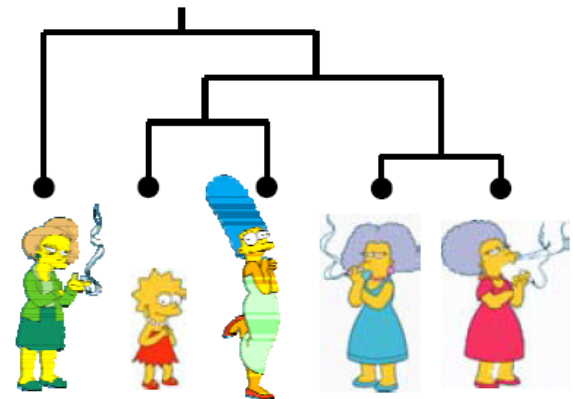
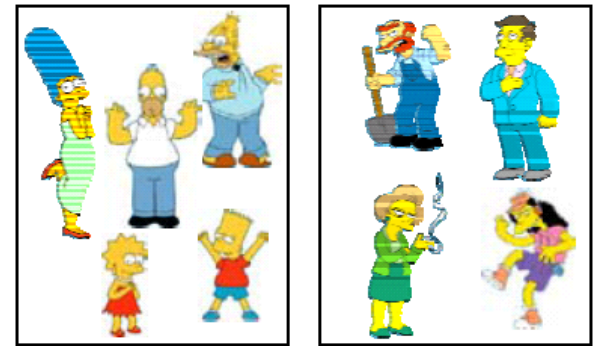
This is called the Edit distance  
or the Transformation distance

# Today Roadmap: clustering

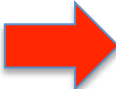
- Definition of "groupness"
- Definition of "similarity/distance"
- Representation for objects
- How many clusters?
- Clustering Algorithms
  - Partitional algorithms
  - Hierarchical algorithms
- Formal foundation and convergence

# Clustering Algorithms

- Partitional algorithms
  - Usually start with a random (partial) partitioning
  - Refine it iteratively
    - K means clustering
    - Mixture-Model based clustering
- Hierarchical algorithms
  - Bottom-up, agglomerative
  - Top-down, divisive



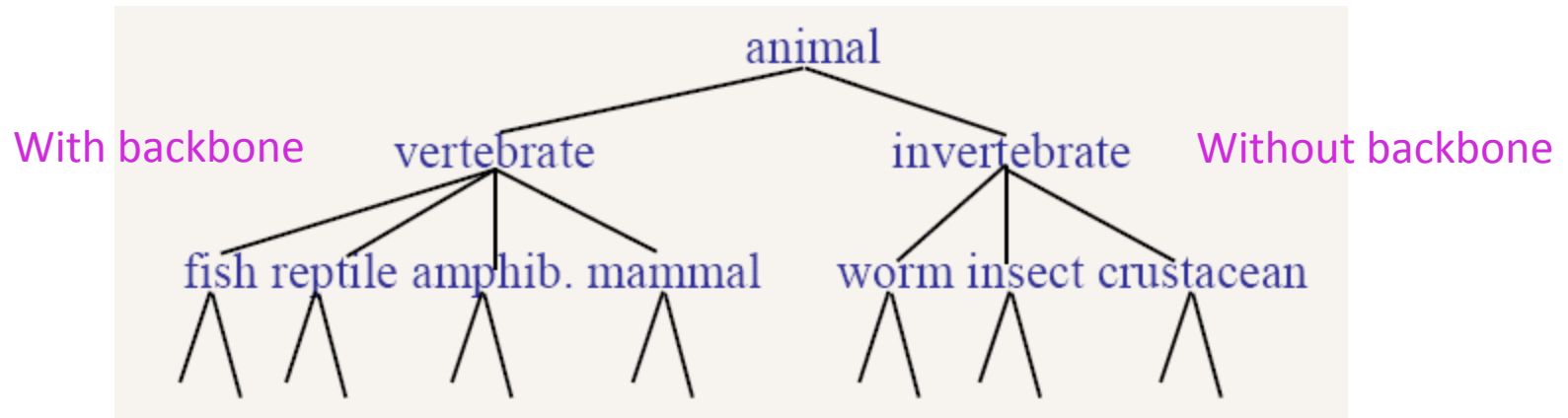
# Today Roadmap: clustering

- Definition of "groupness"
- Definition of "similarity/distance"
- Representation for objects
- How many clusters?
- Clustering Algorithms
  - Partitional algorithms
  -  ■ Hierarchical algorithms
  - Formal foundation and convergence



# Hierarchical Clustering

- Build a tree-based hierarchical taxonomy (**dendrogram**) from a set of objects, e.g. organisms, documents.



- Note that hierarchies are commonly used to organize information, for example in a web portal.
  - Yahoo! hierarchy is manually created, we will **focus on automatic creation of hierarchies in data mining.**

# (How-to) Hierarchical Clustering

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

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

NP

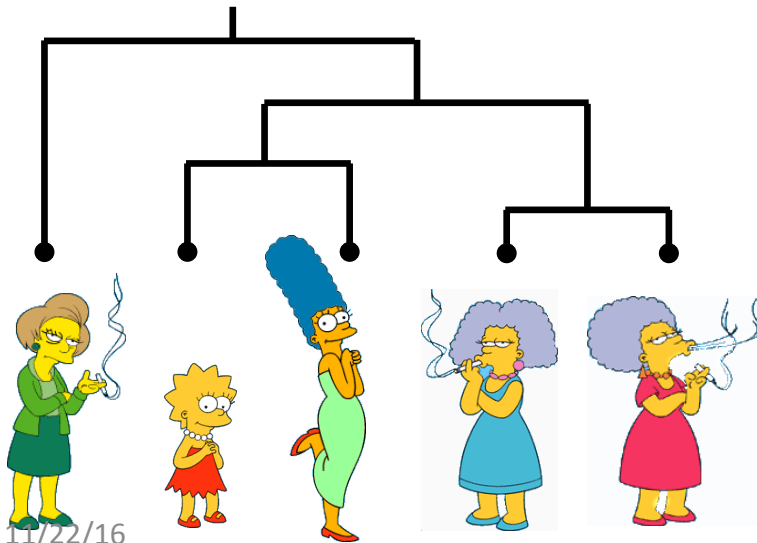
## Bottom-Up (agglomerative):

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

Clustering: the process of grouping a set of objects into classes of similar objects →

high intra-class similarity  
low inter-class similarity



We begin with a distance matrix which contains the distances between every pair of objects in our database.










$$\begin{cases} D(A, A) = 0 \\ D(A, B) = D(B, A) \end{cases}$$



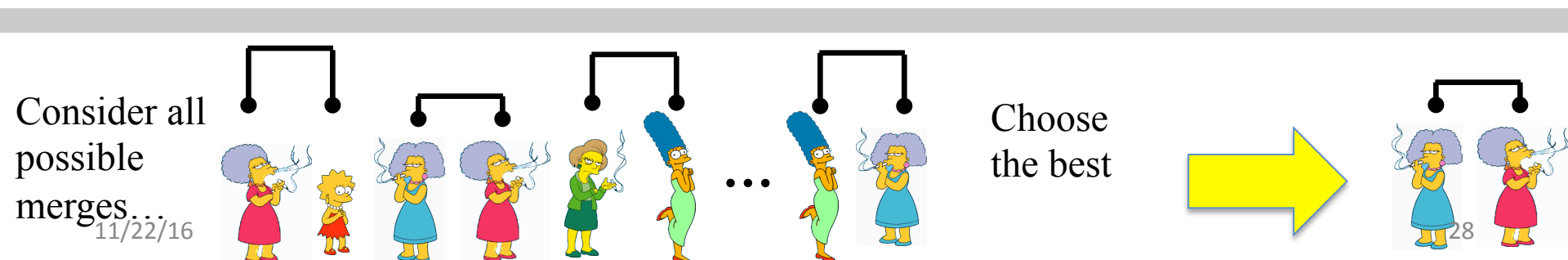
$$D(\text{Mrs. Simpson}, \text{Lisa Simpson}) = 8$$

$$D(\text{Marge Simpson}, \text{Bart Simpson}) = 1$$

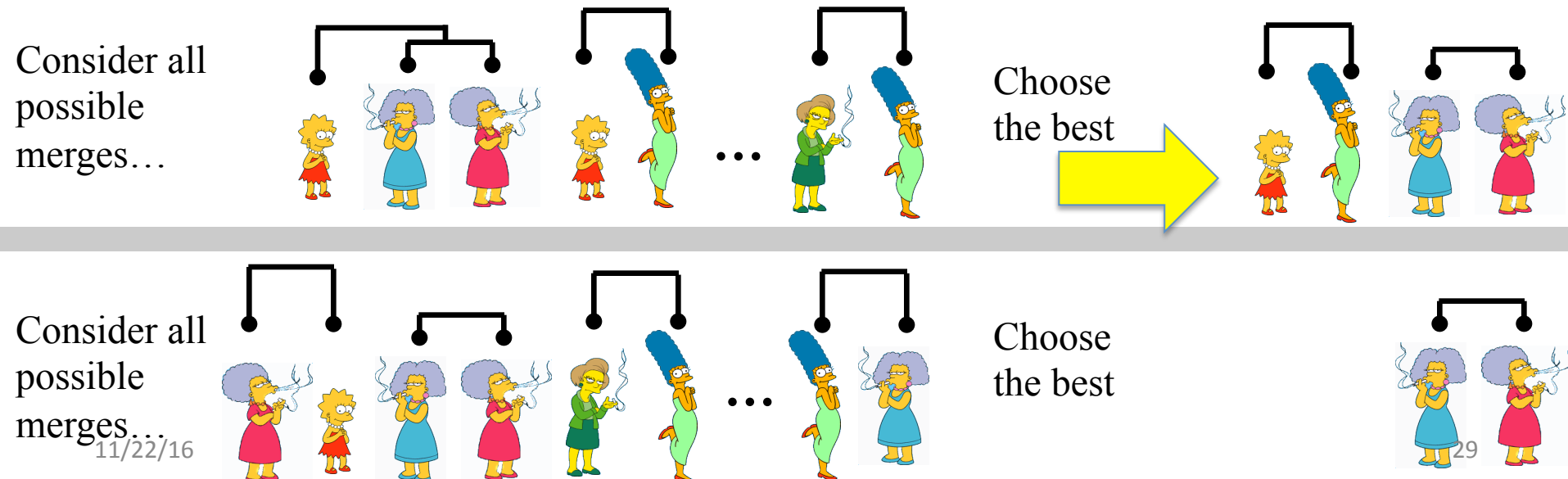
11/22/16

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

**Bottom-Up (agglomerative):** 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.

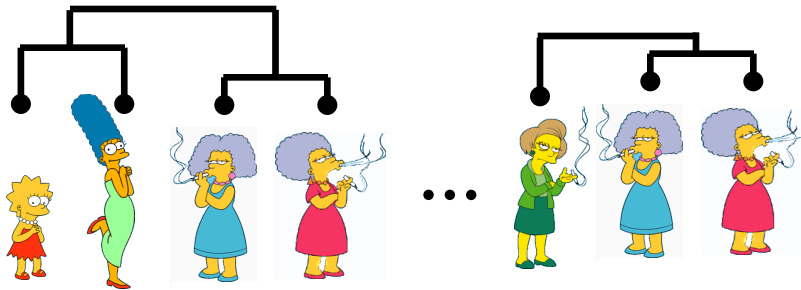


**Bottom-Up (agglomerative):** Starting with each item in its own cluster, find the best pair to merge into a new cluster. Repeat until all clusters are fused together.

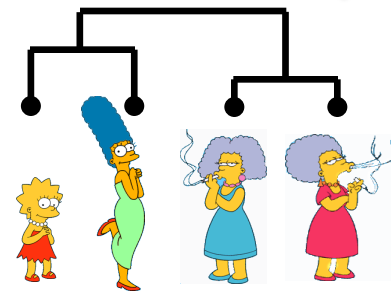


**Bottom-Up (agglomerative):** Starting with each item in its own cluster, find the best pair to merge into a new cluster. Repeat until all clusters are fused together.

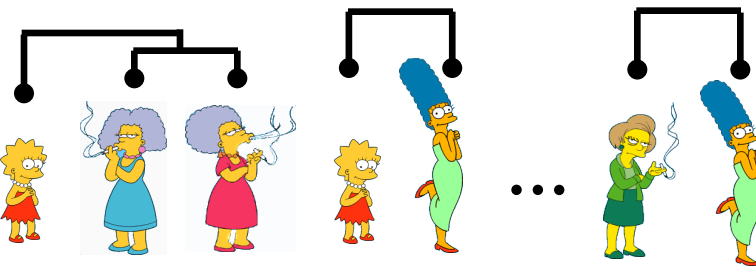
Consider all possible merges...



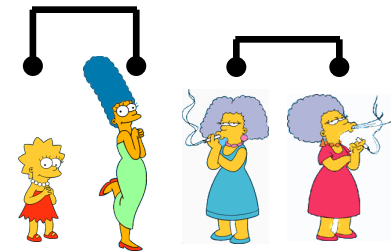
Choose the best



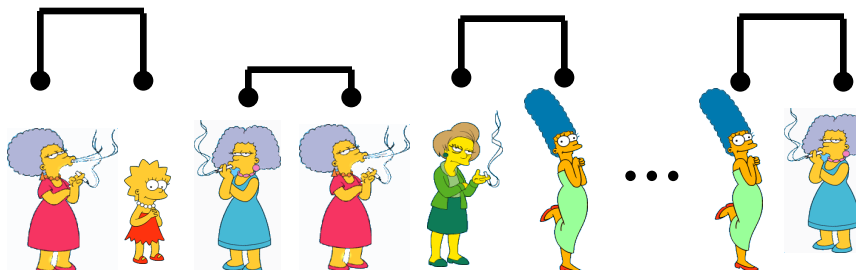
Consider all possible merges...



Choose the best



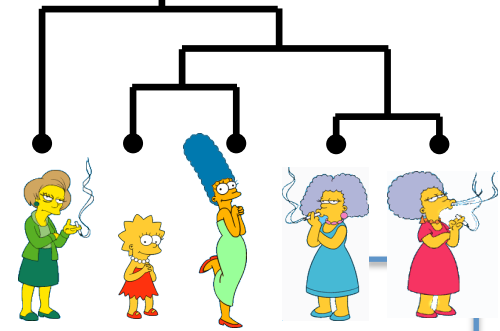
Consider all possible merges...



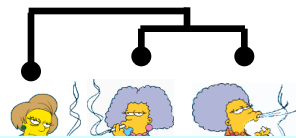
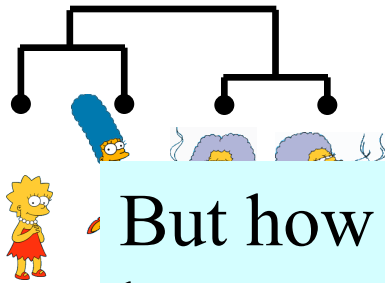
Choose the best



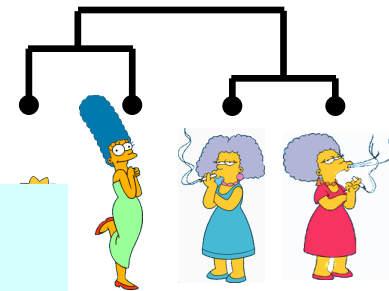
**Bottom-Up (agglomerative):** Starting with each item in its own cluster, find the best pair to merge into a new cluster. Repeat until all clusters are fused together.



Consider all possible merges...

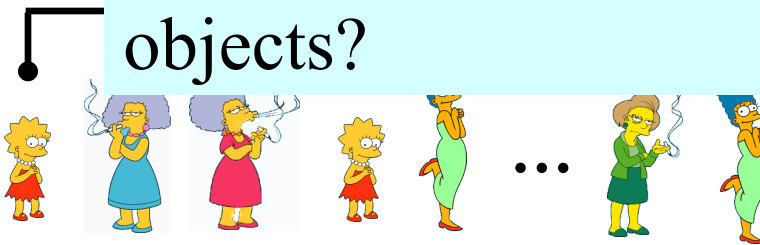


Choose

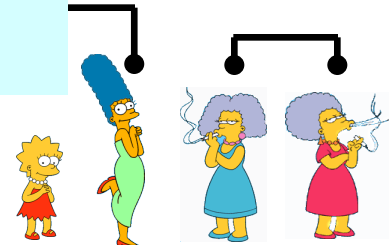


But how do we compute distances between clusters rather than objects?

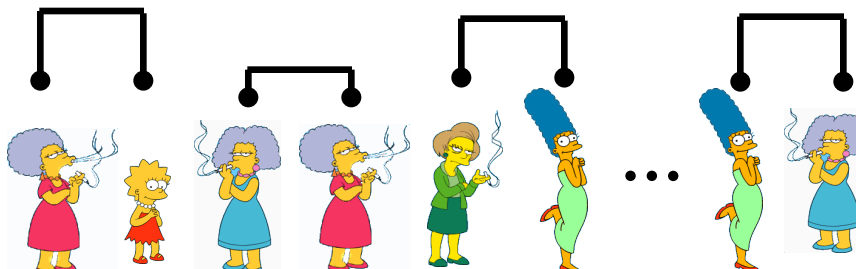
Consider all possible merges...



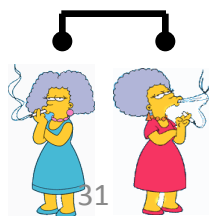
the best



Consider all possible merges...



Choose the best



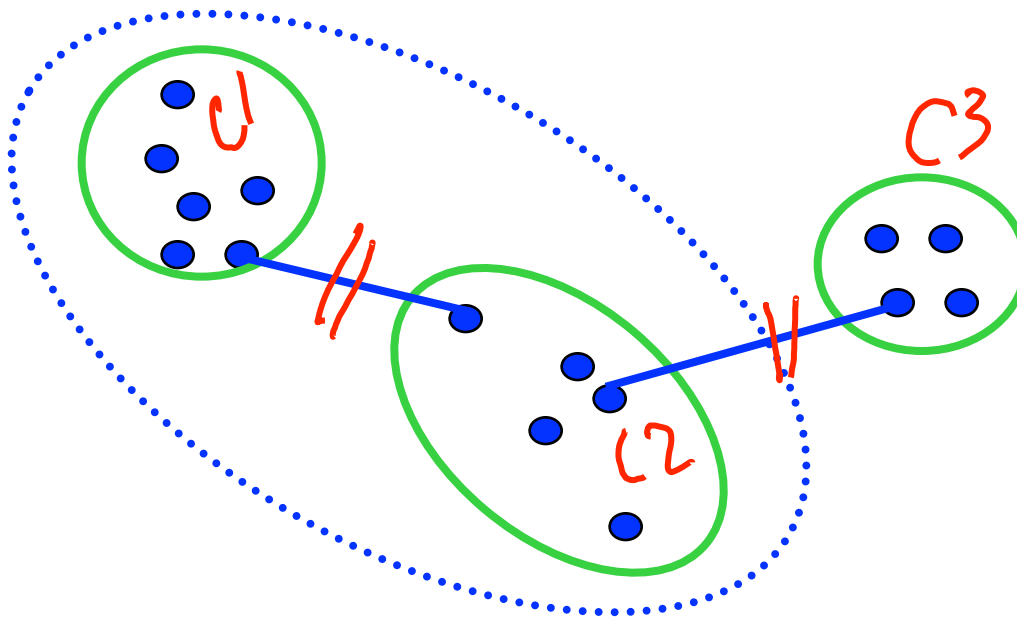
# How to decide the distances between clusters ?

- Single-Link
  - Nearest Neighbor: their closest members.
- Complete-Link
  - Furthest Neighbor: their furthest members.
- Average:
  - average of all cross-cluster pairs.



# Computing distance between clusters: Single Link

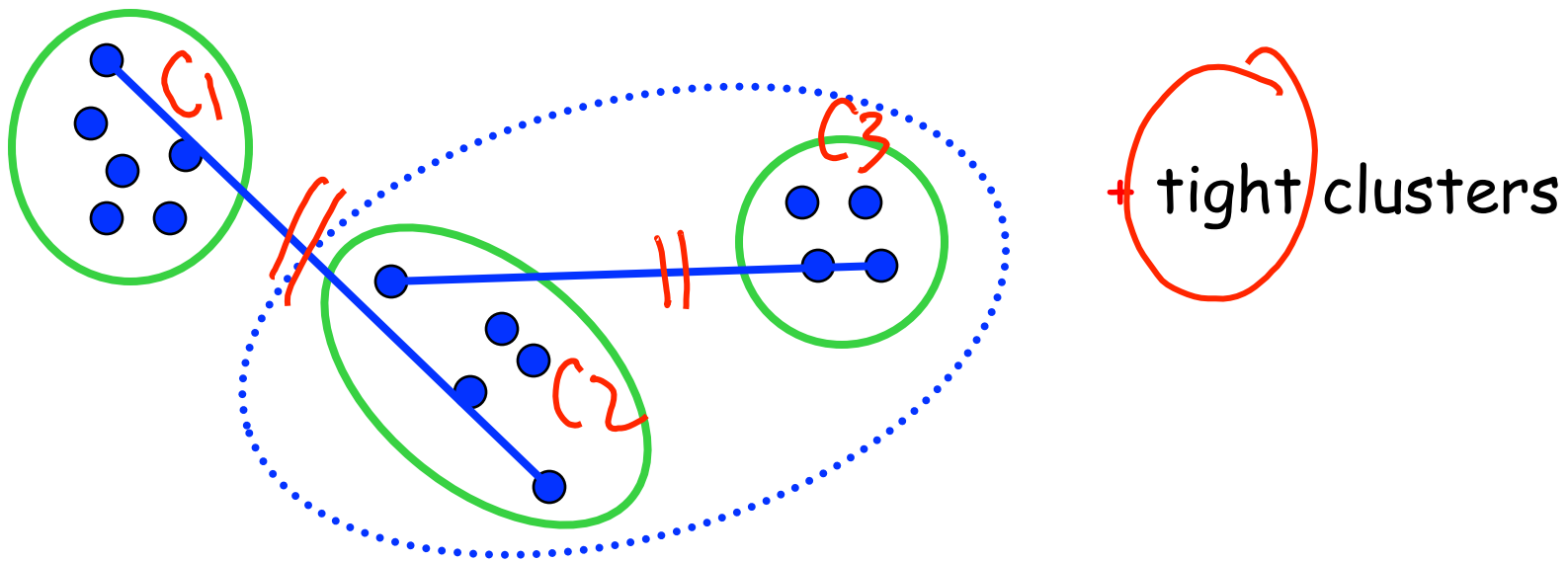
- cluster distance = distance of two **closest** members in each class



- Potentially long and skinny clusters

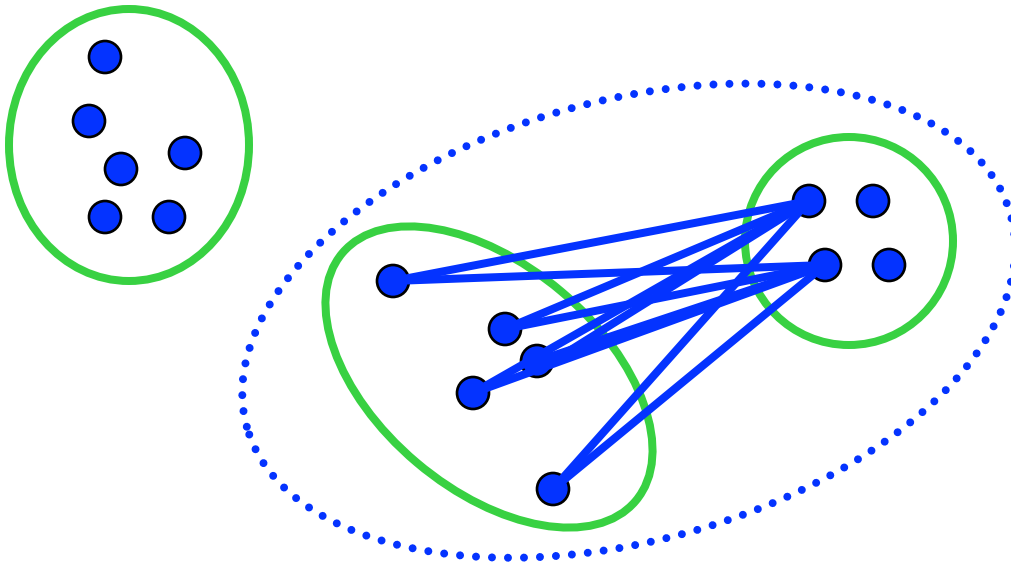
# Computing distance between clusters: : Complete Link

- cluster distance = distance of two farthest members



# Computing distance between clusters: Average Link

- cluster distance = **average distance** of all pairs



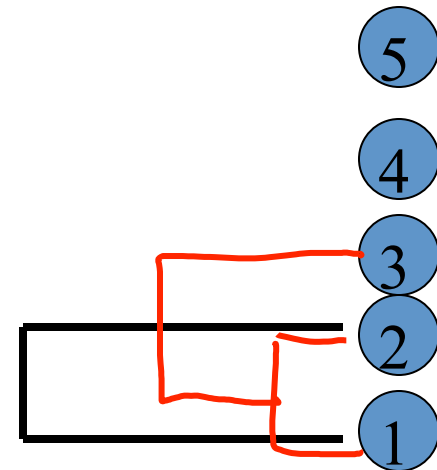
**the most widely  
used measure**

**Robust against  
noise**

# Example: single link

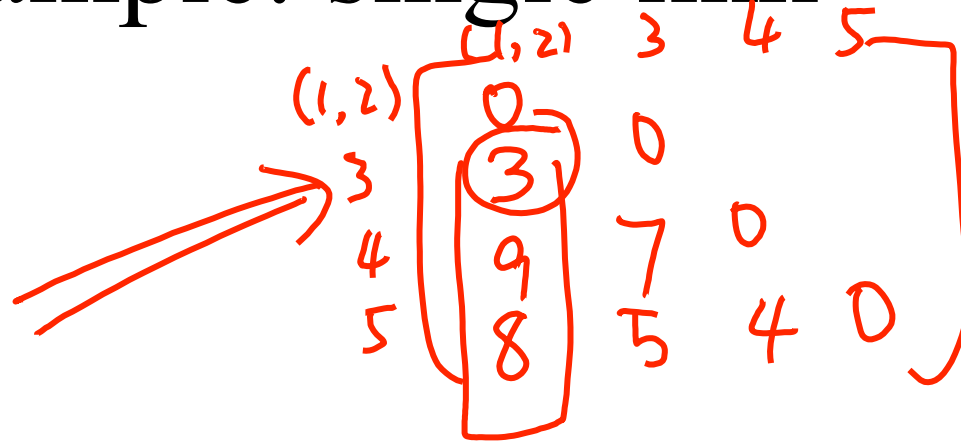
	1	2	3	4	5
1	0				
2	2	0			
3	6	3	0		
4	10	9	7	0	
5	9	8	5	4	0

	(1,2)	3	4	5
(1,2)	0			
3	3	0		
4	9	7	0	
5	8	5	4	0

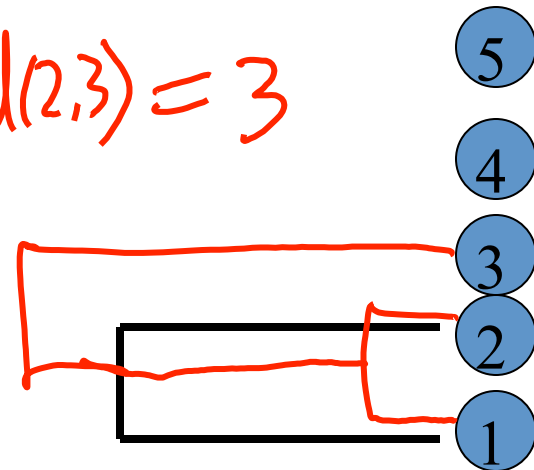


# Example: single link

	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

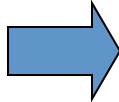


$$d((1,2), 3) = \min(d(1,3), d(2,3)) = 3$$



# Example: single link

	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

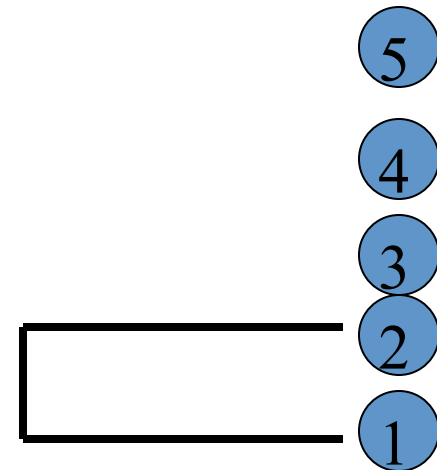


	(1,2)	3	4	5
(1,2)	0			
3	3	0		
4	9	7	0	
5	8	5	4	0

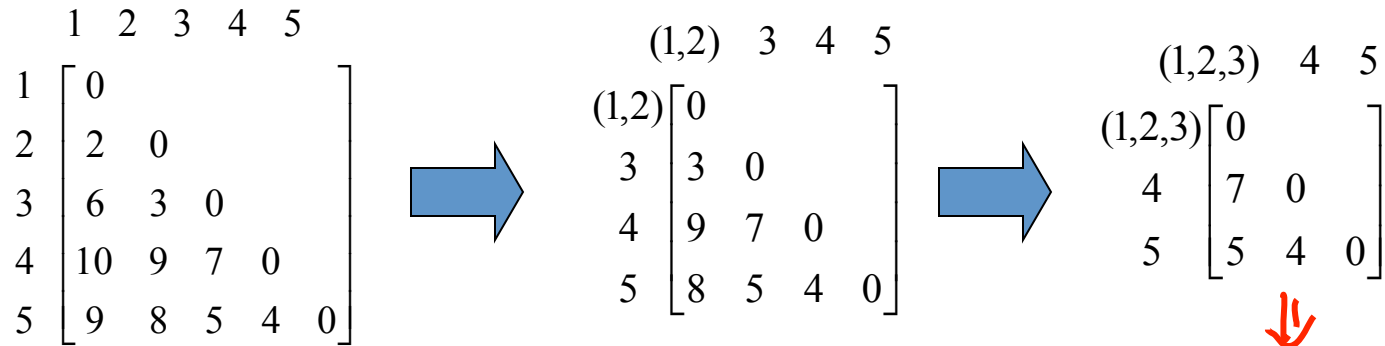
$$d_{(1,2),3} = \min\{d_{1,3}, d_{2,3}\} = \min\{6, 3\} = 3$$

$$d_{(1,2),4} = \min\{d_{1,4}, d_{2,4}\} = \min\{10, 9\} = 9$$

$$d_{(1,2),5} = \min\{d_{1,5}, d_{2,5}\} = \min\{9, 8\} = 8$$

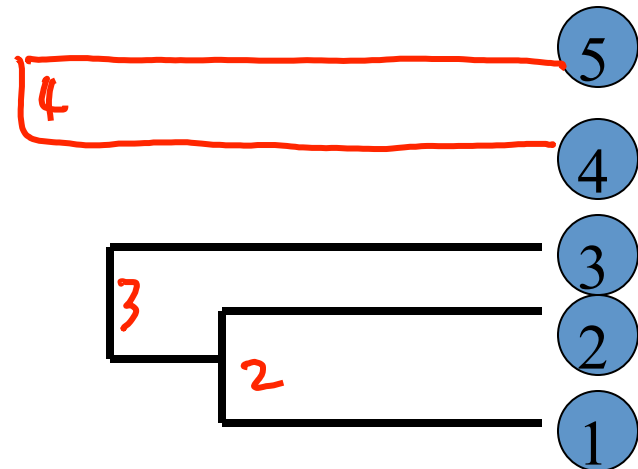


# Example: single link



$$d_{(1,2,3),4} = \min\{d_{(1,2),4}, d_{3,4}\} = \min\{9, 7\} = 7$$

$$d_{(1,2,3),5} = \min\{d_{(1,2),5}, d_{3,5}\} = \min\{8, 5\} = 5$$



# Example: single link

	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

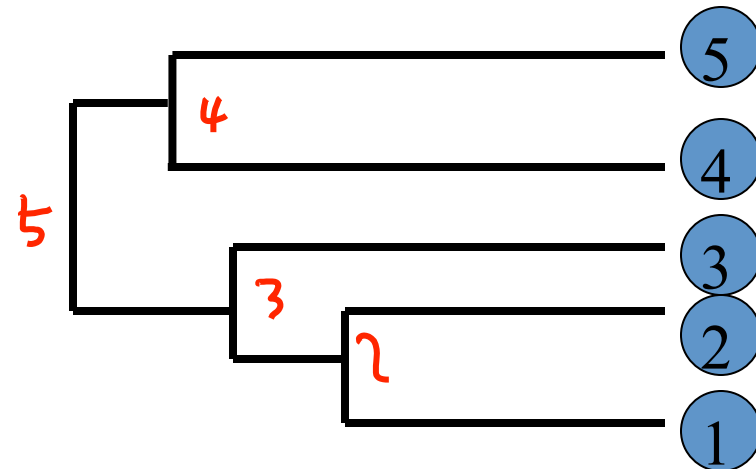
→

	(1,2)	3	4	5
(1,2)	0			
3	3	0		
4	9	7	0	
5	8	5	4	0

→

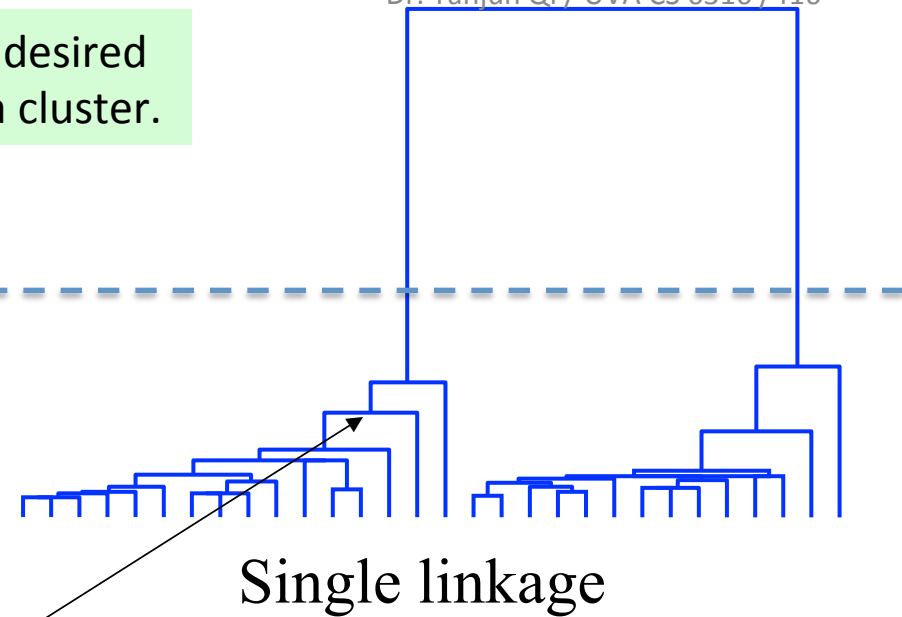
	(1,2,3)	4	5
(1,2,3)	0		
4	7	0	
5	5	4	0

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

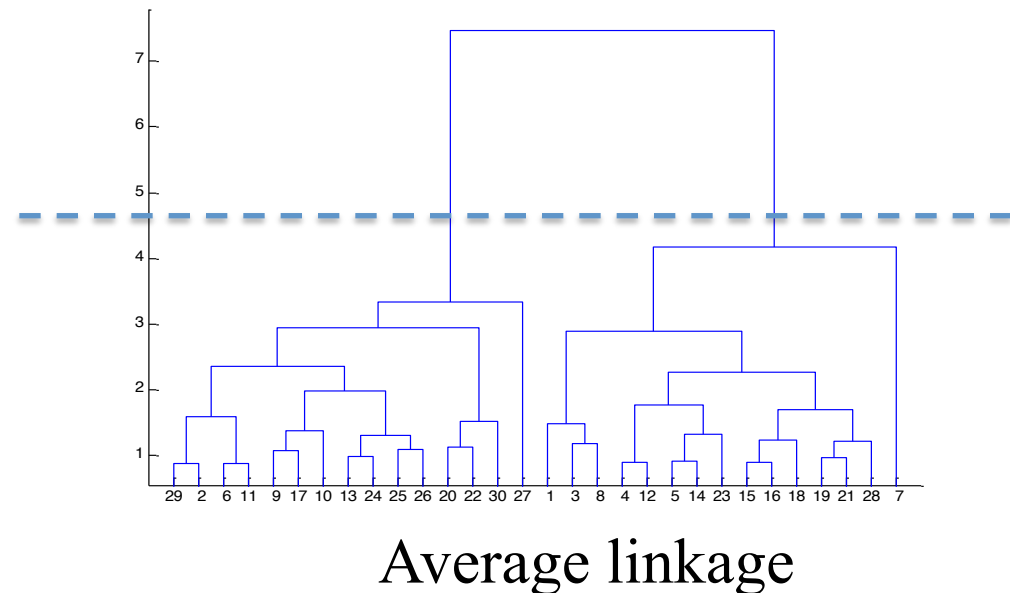




Partitions by cutting the dendrogram at a desired level: each connected component forms a cluster.



Height represents distance between objects / clusters



# Hierarchical Clustering

- **Bottom-Up** Agglomerative Clustering
  - Starts with **each** object in **a separate cluster**
  - then **repeatedly joins** the **closest** pair of clusters,
  - until there is only one cluster.

The history of merging forms a binary tree or hierarchy (dendrogram)

- **Top-Down divisive**
  - Starting with all the data in a single cluster,
  - Consider every possible way to divide the cluster into two. Choose the best division
  - And recursively operate on both sides.

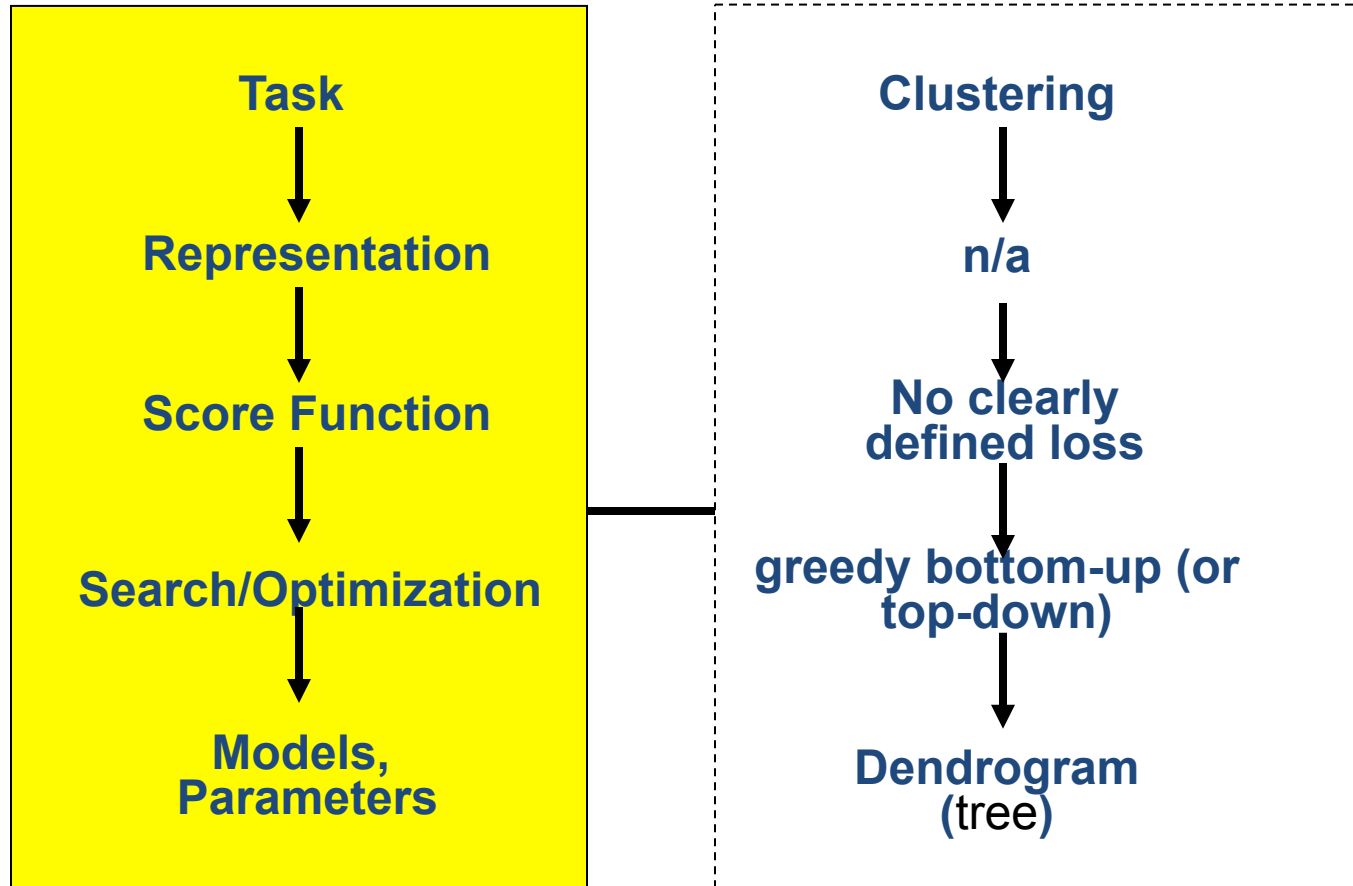
# Computational Complexity

- In the first iteration, all HAC methods need to compute similarity of all pairs of  $n$  individual instances which is  $O(n^2p)$ .  
*(Matrix)*
- In each of the subsequent  $n-2$  merging iterations, compute the distance between the most recently created cluster and all other existing clusters.
- For the subsequent steps, in order to maintain an overall  $O(n^2)$  performance, computing similarity to each other cluster must be done in constant time. Else  $O(n^2 \log n)$  or  $O(n^3)$  if done naively

# Summary of Hierarchical Clustering Methods

- No need to specify the number of clusters in advance.
- Hierarchical structure maps nicely onto human intuition for some domains
- They do not scale well: time complexity of at least  $O(n^2)$ , where  $n$  is the number of total objects.
- Like any heuristic search algorithms, local optima are a problem.
- Interpretation of results is (very) subjective.

# Hierarchical Clustering



# References

- ❑ Hastie, Trevor, et al. *The elements of statistical learning*. Vol. 2. No. 1. New York: Springer, 2009.
- ❑ Big thanks to Prof. Eric Xing @ CMU for allowing me to reuse some of his slides
- ❑ Big thanks to Prof. Ziv Bar-Joseph @ CMU for allowing me to reuse some of his slides