EE 604
Digital Image Processing

IIT KANPUR
Indian Institute of Technology, Kanpur

Tanaya Guha

Aug - Nov 2017

Thresholding

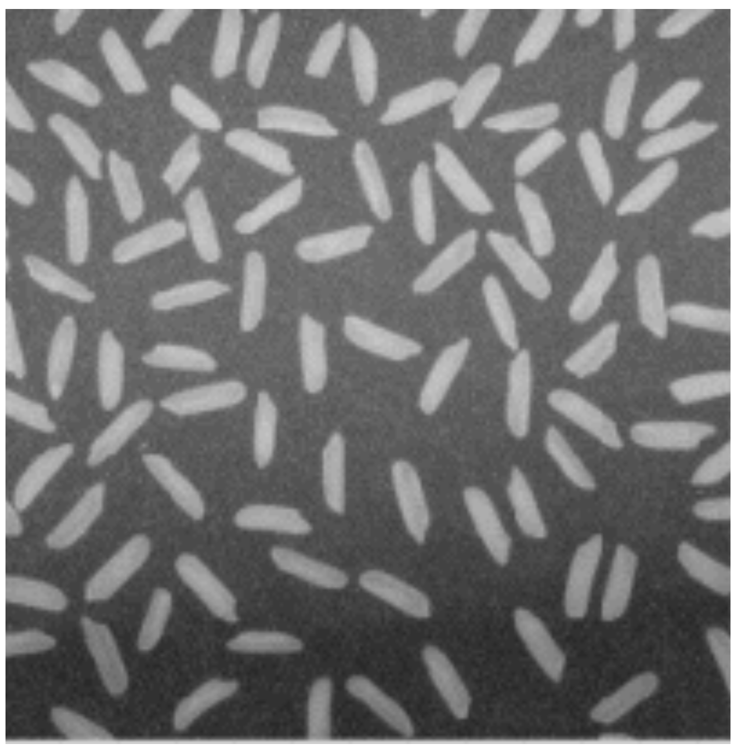
Thresholding

- Simplest form of image segmentation
- Usually, partitions an image f into a binary image $\boldsymbol{s}$ with 2 levels (foreground and background): above and below a chosen threshold $\boldsymbol{T}$
- If, $f(x,y) > T$, $s(x,y)=1$, else, $s(x,y) = 0$

- **Assumptions:**
    - Intensity values are different in different regions
    - Within a region (representing an object, say) intensity values are similar.

- Works well when the image histogram is bimodal

Thresholding



global

local

Global: single threshold T for the entire image

Local: divide image into sub-images, use a different T for each region

Adaptive: T changes based on image statistics

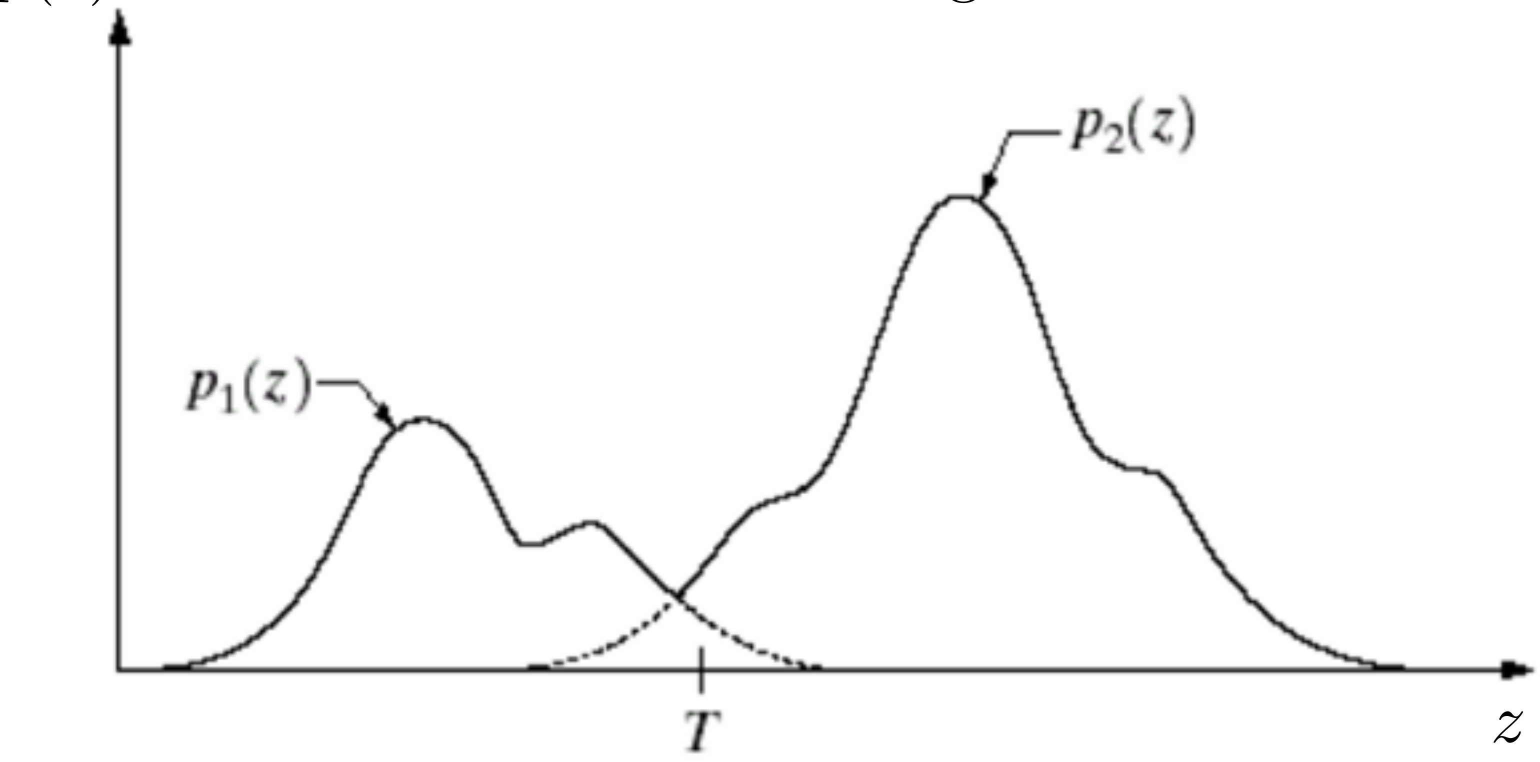
Basic global thresholding

- Initialize threshold ***T***
- Loop until converged
  - Partition image using ***T***
  - Compute background mean μ_b as the average intensity of all pixels below ***T***
  - Compute foreground mean μ_f as the average intensity of all pixels above ***T***
  - Update ***T***

$$T = \frac{1}{2}(\mu_f + \mu_b)$$

Optimal global thresholding

$p(z)$ = PDF i.e. normalized histogram



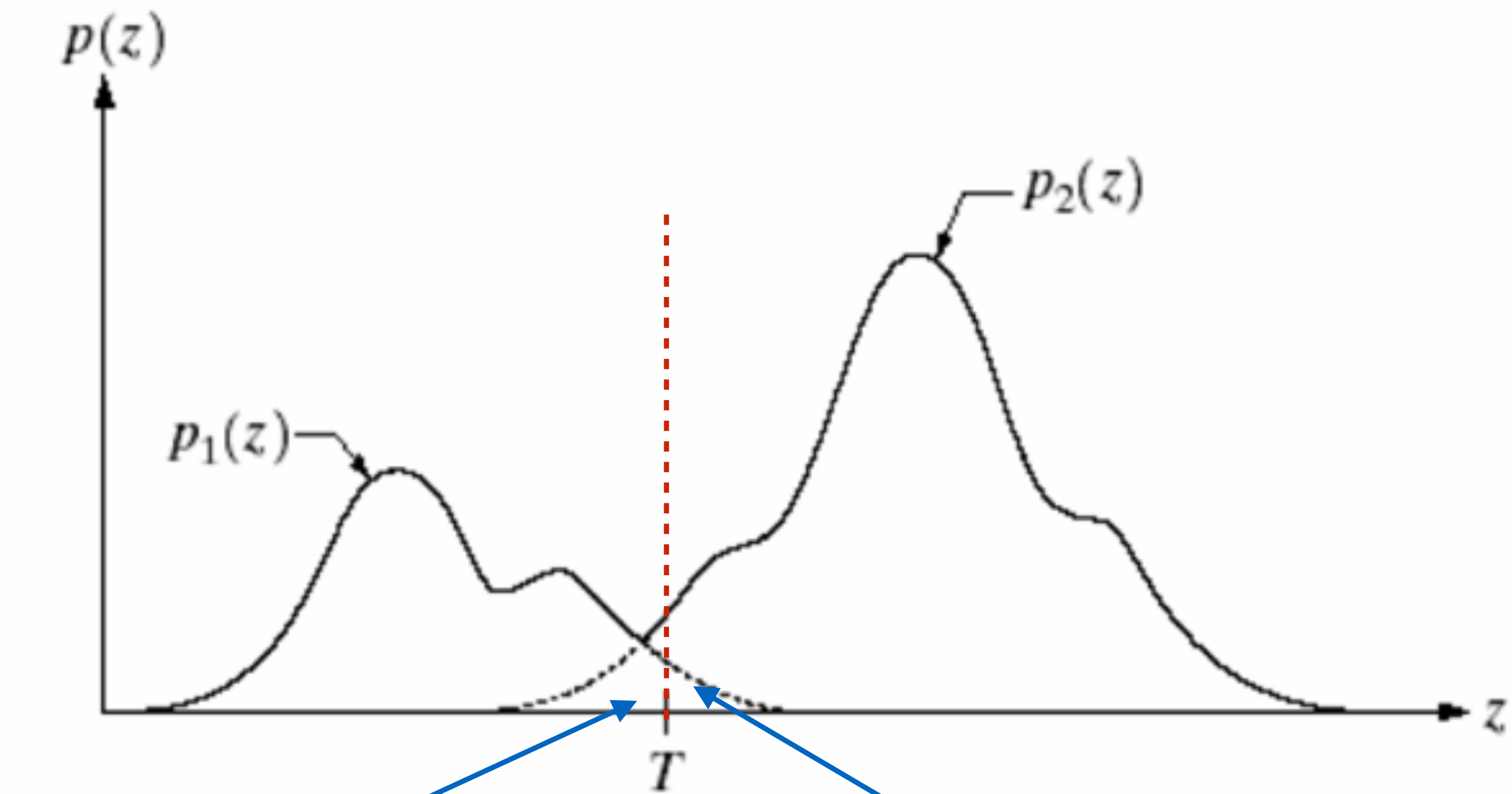
z = gray level

$$p(z) = P_1 p_1(z) + P_2 p_2(z)$$

P_1, P_2 : prob of occurrence of two classes of pixels

$$P_1 + P_2 = 1$$

Optimal global thresholding



$E_1(T)$ = prob of misclassifying class 2 as class 1

$E_2(T)$ =prob of misclassifying class 1 as class 2

Optimal global thresholding

$$E_1(T) = \int_{-\infty}^{T} p_2(z)dz \qquad E_2(T) = \int_{T}^{\infty} p_1(z)dz$$

Total error :

$$E(T) = P_2 E_1(T) + P_1 E_2(T)$$

$$T^* = \underset{T}{\text{argmin}} \; E(T)$$

Differentiating ***E(T)*** w.r.t ***T*** and setting to 0 results in

$$P_1 p_1(T^*) = P_2 p_2(T^*)$$

Optimal global thresholding

- Not easy to always find an analytical expression
- Requires the knowledge of the PDFs

- Expression when modeling using Gaussian (Homework: derive the expression)

$$T^* = \frac{\mu_1 + \mu_2}{2} + \frac{\sigma^2}{\mu_1 - \mu_2} \ln(\frac{P_2}{P_1})$$

μ_1, μ_2 : class mean

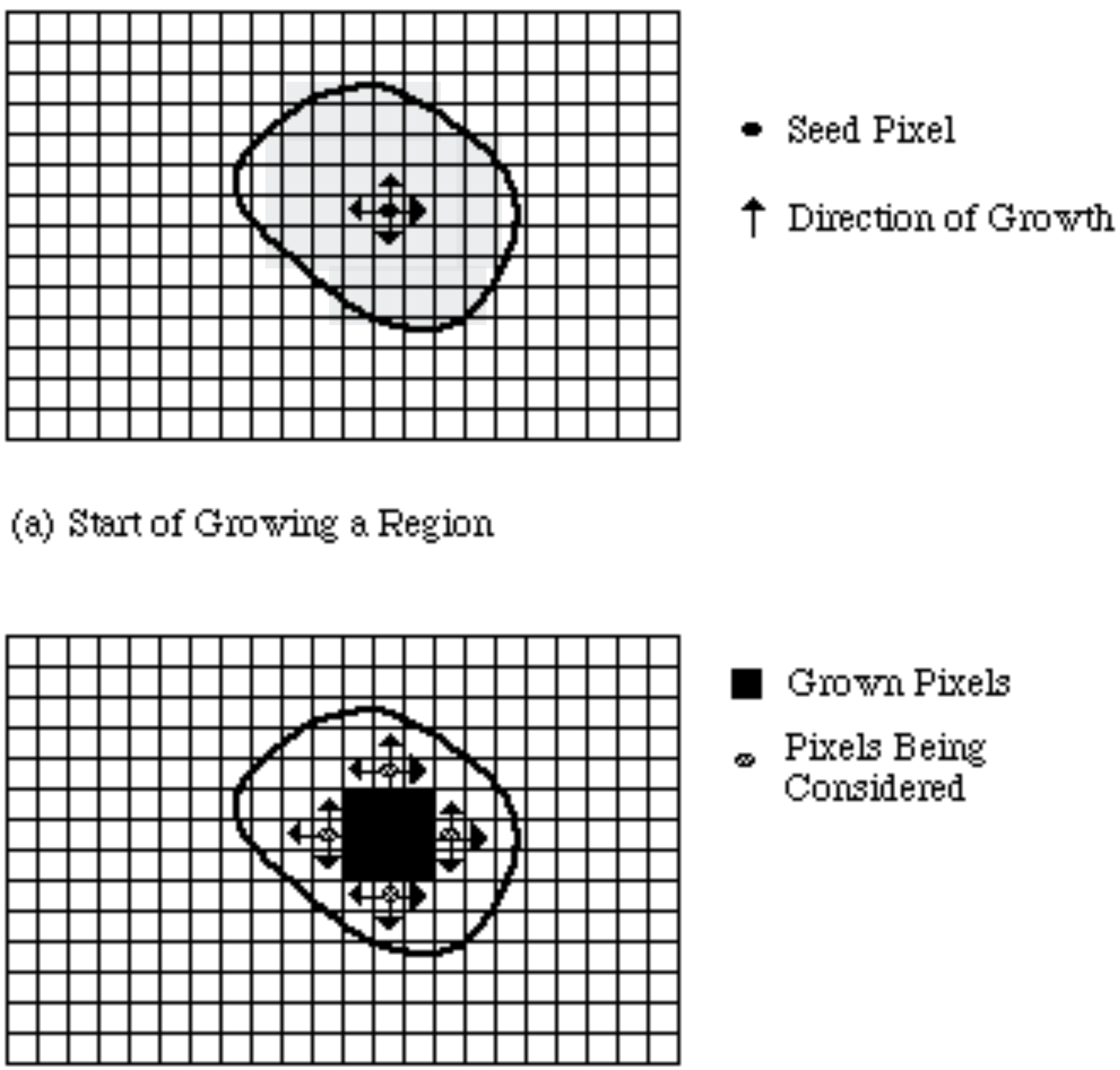
$\sigma = \sigma_1 = \sigma_2$: class variance

Region growing

Assumptions

- We wish to partition image $\mathcal{I}$ into $\boldsymbol{n}$ regions such that
- $\mathcal{I} = \bigcup_{i=1}^{n} R_i$ i.e. every pixel belongs to some region
- $R_i \bigcap R_j = \emptyset$ i.e. each pixel is assigned to only one region
- All pixels in a region share similar (predefined) property
- Pixels in different regions have different properties.

Region growing



(a) Start of Growing a Region

(b) Growing Process After a Few Iterations

What do we need?

- Seed points
- A measure of similarity
- A stopping criterion

Image from: cs.cf.ac.uk

A basic algorithm

- Start with a seed point s_j for region R_j
- For every s_j
  - Initialize mean intensity of each region: $\mu_j = s_j$
  - Initialize region: $R_j = \{s_j\}$
- For each point p in R_j
  - Get its 4-connect neighborhood: $\mathcal{N}_i(p), \; i = 1, 2, 3, 4$
  - If $|\mathcal{N}_i(p) - \mu_j| < \tau$, $\mathcal{N}_i(p) \notin \mathcal{R}_k \quad j \neq k$
    - $\mathcal{R}_j \leftarrow \mathcal{R}_j \bigcup \mathcal{N}_i(p)$
    - update μ_j
  - Stop growing when no neighborhood pixel matches
- Move to the next seed point, until the whole image is partitioned.

Remarks

- Seed point selection is important.
    - Different choice of seeds will lead to different segmentation
    - Can be difficult to decide
    - A region may not grow at all (if the seed point is bad)

- Similarity criteria: intensity, color, texture, shape, motion, size

- What happens if you let one region grow completely before others?
    - Chosen region may dominate

- **Simultaneous region growing:** Let adjacent regions grow simultaneously
    - May result into over segmentation

A better algorithm

Key Idea:

- Fit an appropriate low-order surface through the pixels in a region

$$f(x, y, a, m) = \sum_{(i+j) \leq m} a_{ij} x^i y^j$$

- Check approximation error:

$$E(R, a, m) = \sum_{(x,y) \in R} \{g(x, y) - f(x, y, a, m)\}^2$$

- For pixel values belonging to the same region, E should be small.

Besl and Jain, Segmentation Through Variable-Order Surface Fitting, IEEE Transactions on Pattern Analysis and Machine Intelligence, 1988.

A better algorithm

- Partition the image into initial seed **regions** $R_i^{(0)}$
  - E.g. split the image into ***11x11*** regions

- Fit a surface to each seed region.
  - If $E(R_i^{(0)}, a, m)$ is smaller than some predefined value
    - accept $R_i^{(0)}$ as a seed region
    - accept its model
  - else, reject $R_i^{(0)}$

- Let's say we selected ***n*** seed regions

Besl and Jain, Segmentation Through Variable-Order Surface Fitting, IEEE Transactions on Pattern Analysis and Machine Intelligence, 1988.

A better algorithm

- For a seed region:
  - Find all *compatible* points $C_i^{(k)}$ in its neighborhood:

$$C_i^{(k)} = \{(x,y)|(g(x,y) - f(x,y,a,m))^2 \leq \epsilon\}$$

$$(x,y) \text{ is a neighbor of } R_i^{(k)}$$

  - If $C_i^{(k)}$ is empty, stop growing $R_i^{(k)}$
  - Else $R_i^{(k+1)} = R_i^{(k)} \bigcup C_i^{(k)}$
  - Refit the model to $R_i^{(k+1)}$, and compute $E(R_i^{(k+1)}, a, m)$
  - If $(E(R_i^{(k+1)}, a, m) - E(R_i^{(k)}, a, m)) \leq \tau$, compute $C_i^{(k+1)}$

Besl and Jain, Segmentation Through Variable-Order Surface Fitting, IEEE Transactions on Pattern Analysis and Machine Intelligence, 1988.

Region growing

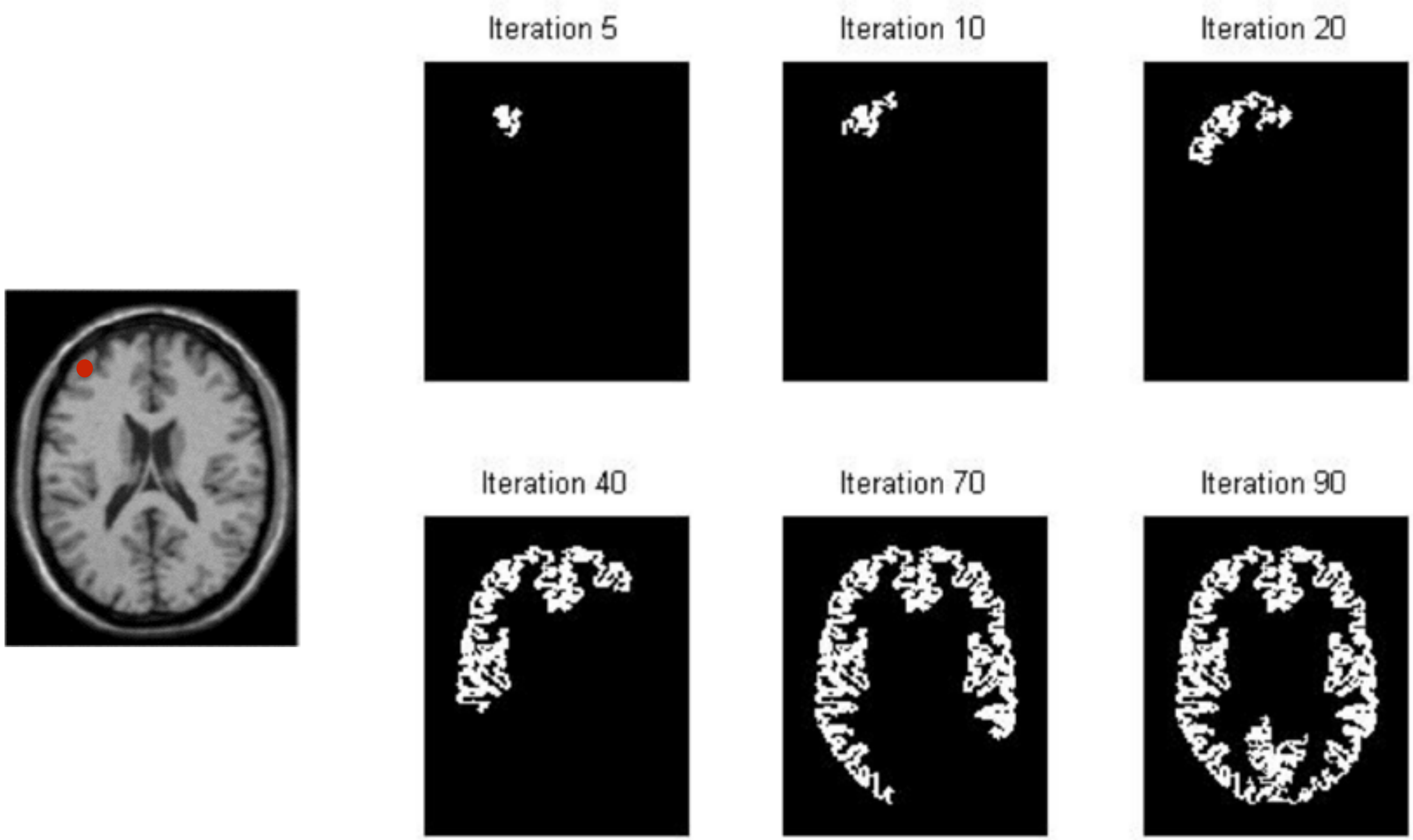


Image from: mathworks.com

Clustering

Segmentation as clustering

- Segmentation can be seen as a clustering problem (unsupervised learning).
    - Group pixels into a number of clusters based on predefined criteria

- Given a pixel ***(x,y)*** in an image ***I***, we can create a vector

$$\mathbf{F} = \begin{bmatrix} (x,y) \\ I(x,y) \\ L(x,y) \end{bmatrix}$$

location

intensity

any other information

- For **N** pixels in an image, we get $\mathbf{F}_1, \mathbf{F}_2, \ldots, \mathbf{F}_\mathrm{N}$

- <u>**Task:**</u> cluster **N** vectors into ***k*** clusters

Segmentation as clustering

- **<u>Task:</u>** cluster **N** vectors into ***C*** clusters

- A large number of algorithms exist for this task
    - k-means clustering
    - Gaussian mixture model
    - Mean shift clustering
    - Spectral clustering
    - Hierarchical clustering
    - Many more …

Acknowledgement (next 15 slides adapted from…)

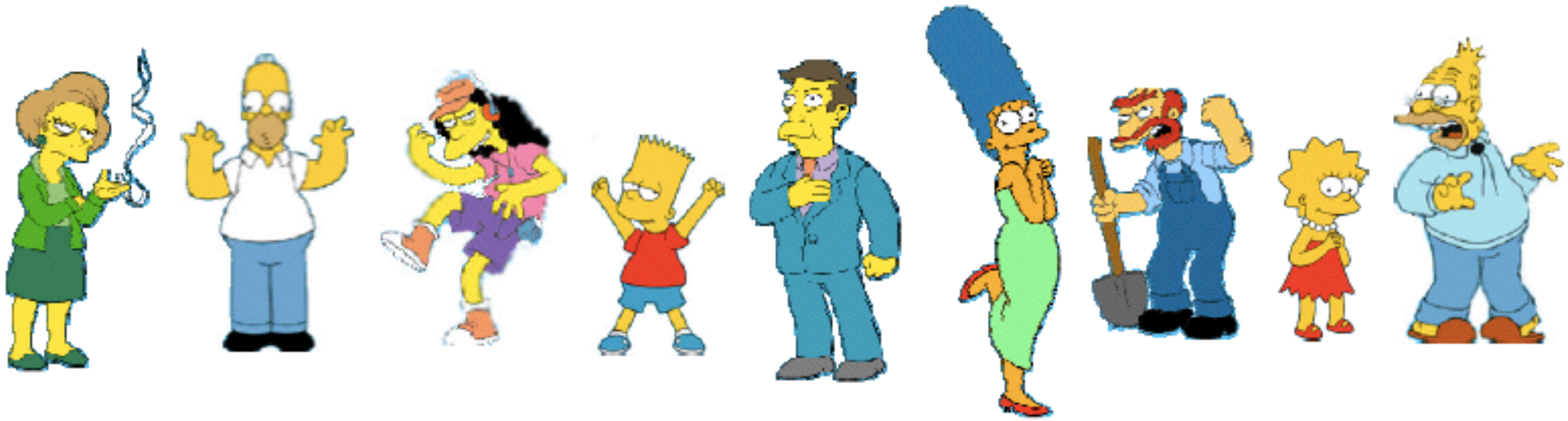
Clustering

15-381 Artificial Intelligence

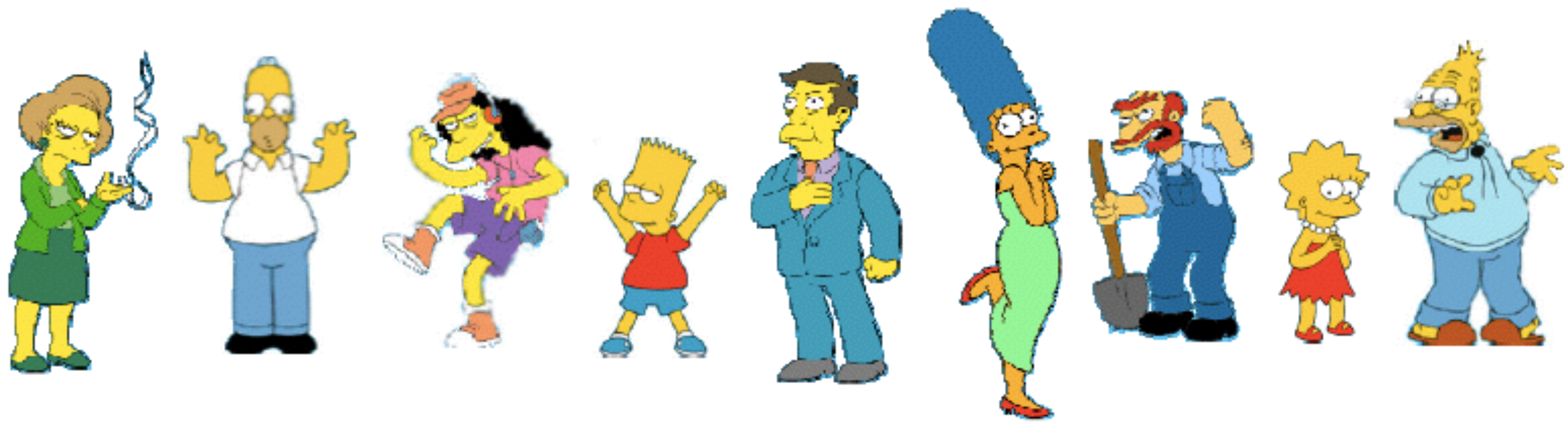
Henry Lin

Modified from excellent slides of Eamonn Keogh, Ziv Bar-Joseph, and Andrew Moore

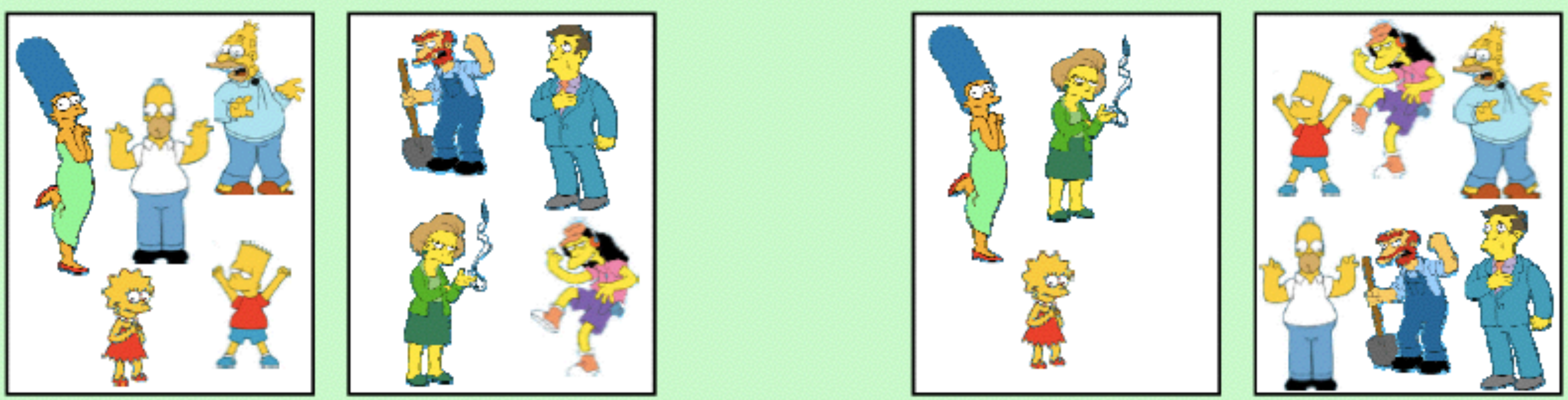
What is a natural grouping among these objects?



What is a natural grouping among these objects?



Clustering is subjective



Simpson's Family | School Employees | Females | Males

What is similarity?

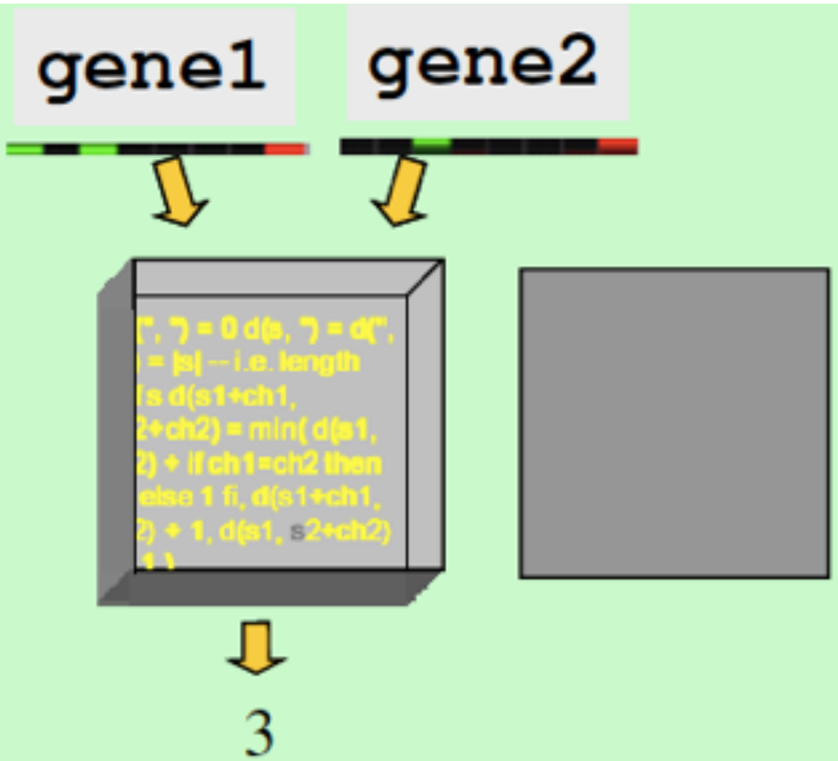
The quality or state of being similar; likeness; resemblance; as, a similarity of features.

Webster's Dictionary

Similarity is 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.

What is similarity?



Inside these black boxes:
some function on two variables
(might be simple or very
complex)

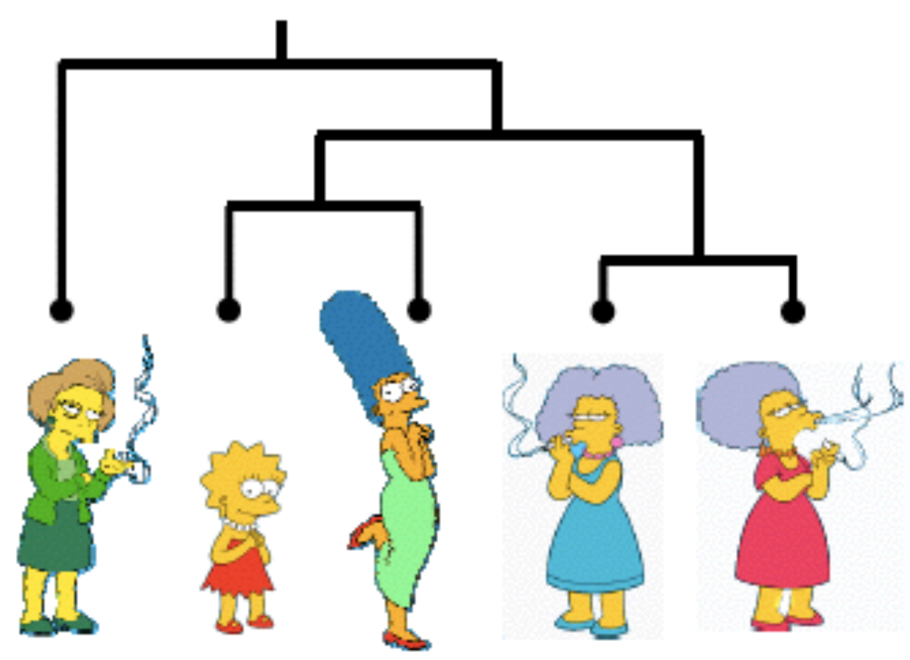
What properties should a distance measure have?

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

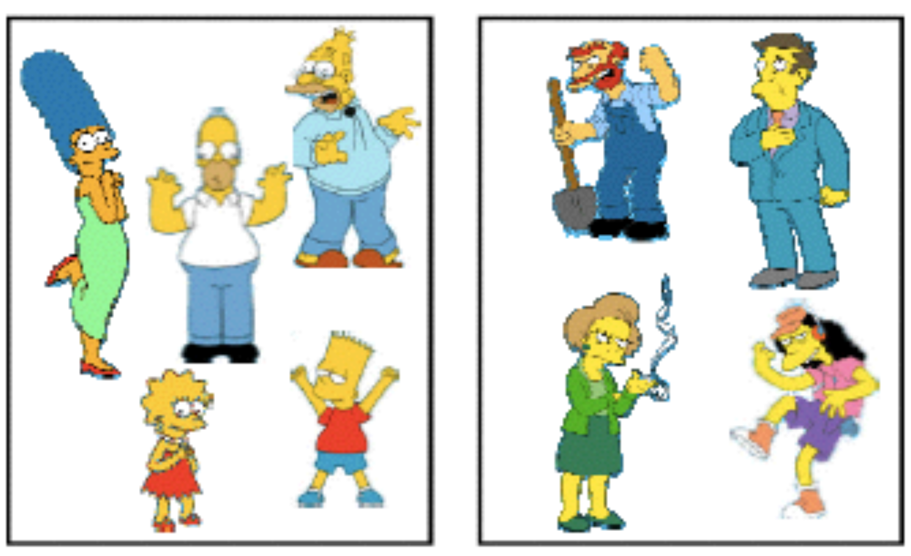
Types of clustering

- **Partitional algorithms:** Construct various partitions and then evaluate them by some criterion (we will see an example called BIRCH)
- **Hierarchical algorithms:** Create a hierarchical decomposition of the set of objects using some criterion

Hierarchical



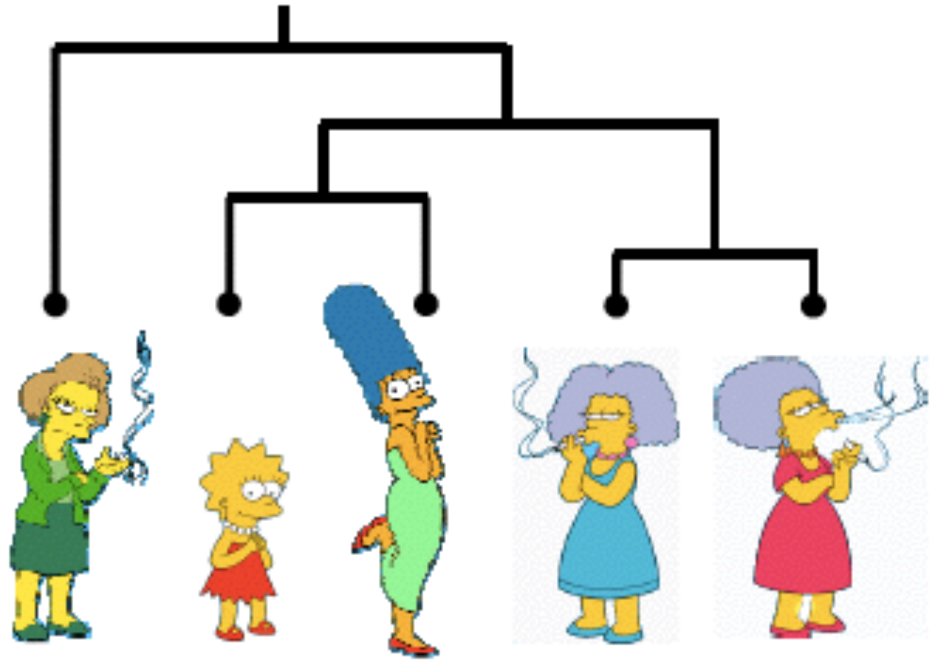
Partitional



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



Since we cannot test all possible trees we will have to heuristic search of all possible trees. We could do this..

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.

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.

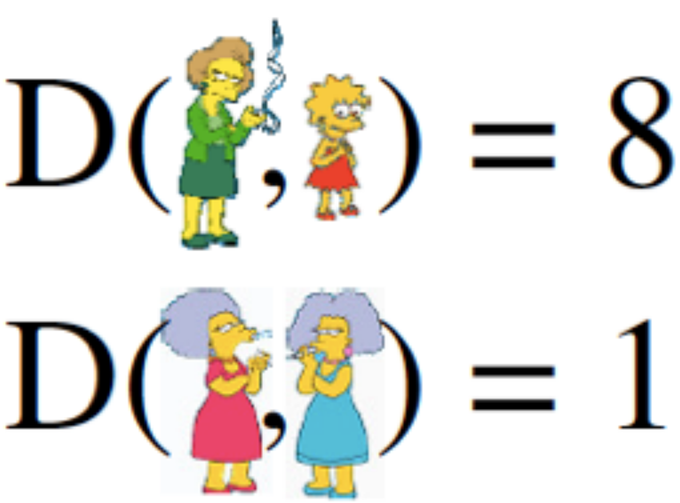
Hierarchical clustering

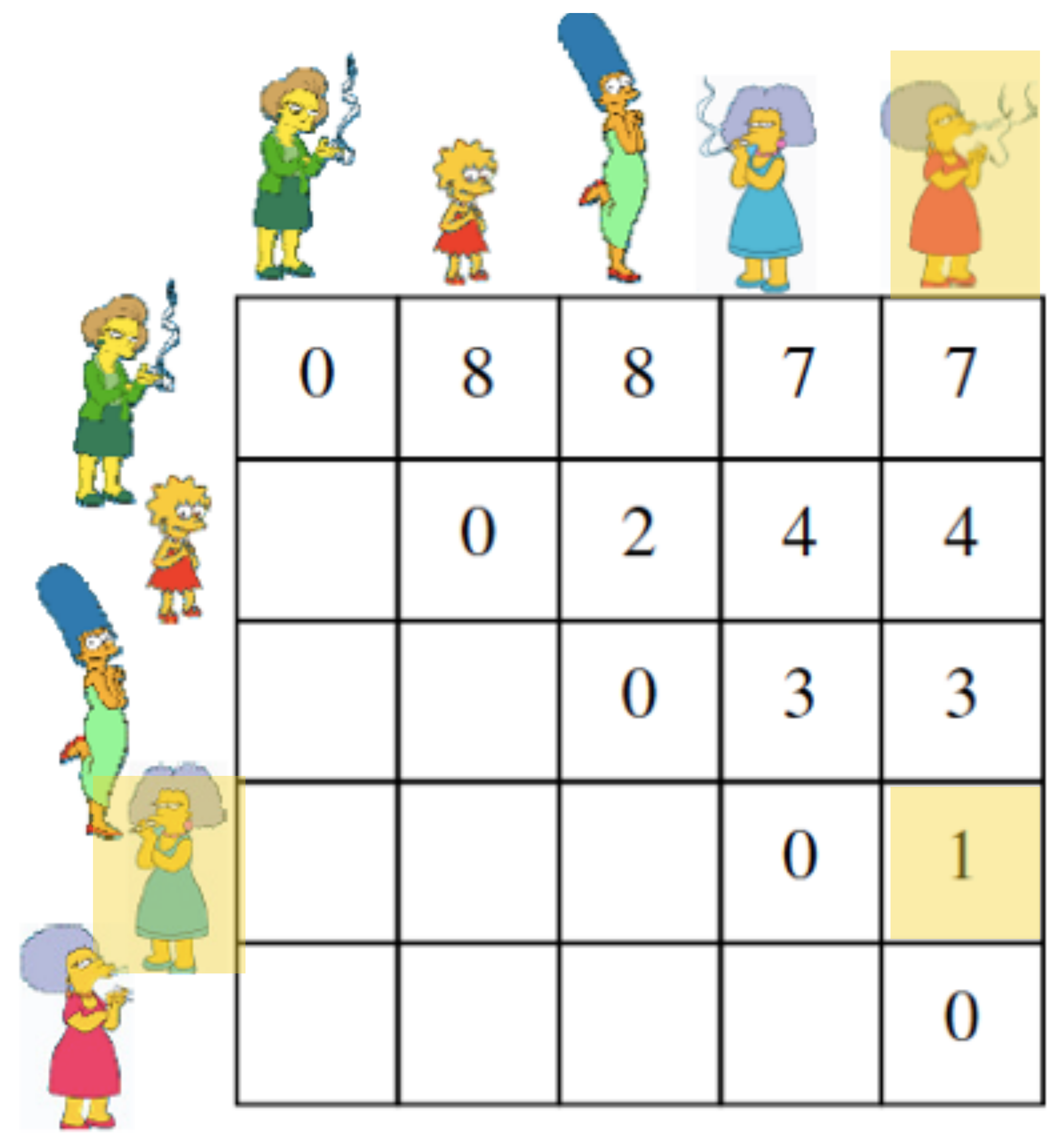
D() = 8

D() = 1

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

Hierarchical clustering



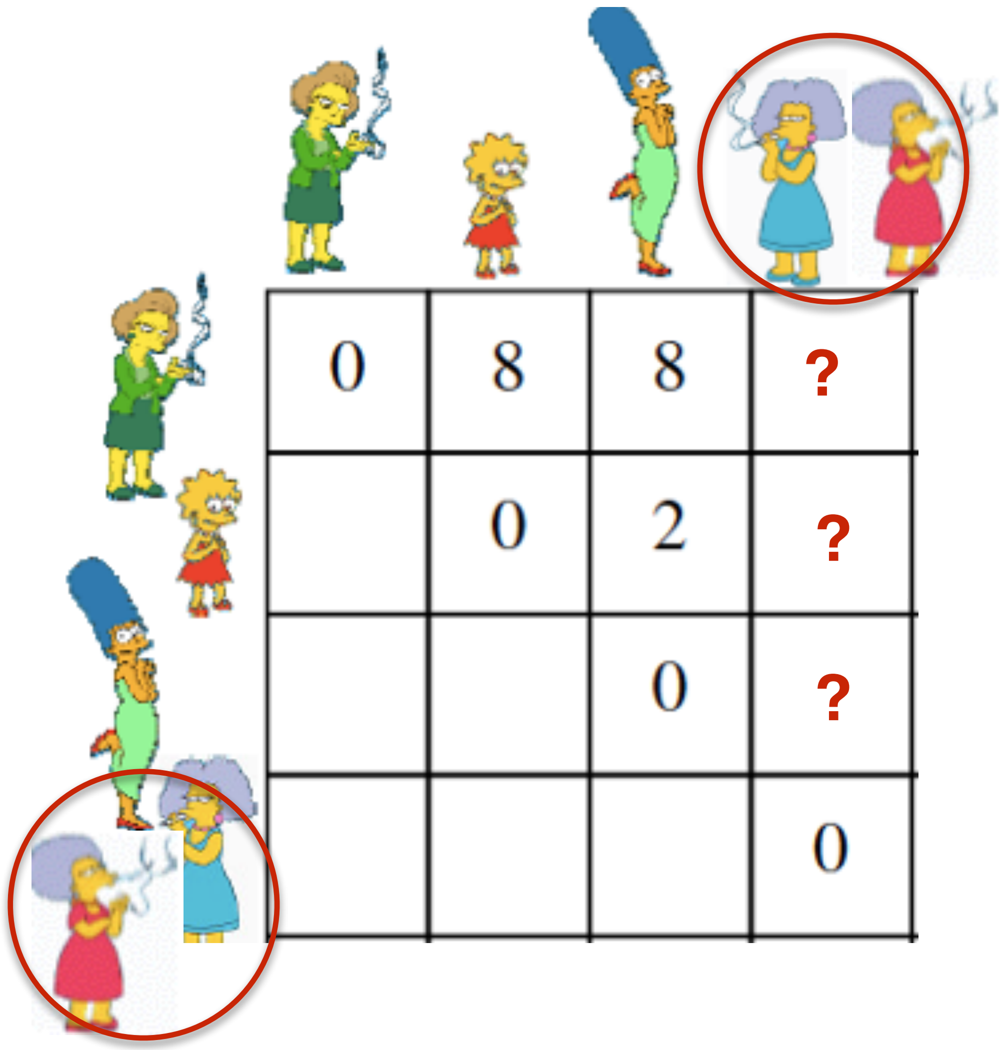


0	8	8	7	7
	0	2	4	4
		0	3	3
			0	1
				0

Consider all possible merges…

Choose the best

Hierarchical clustering

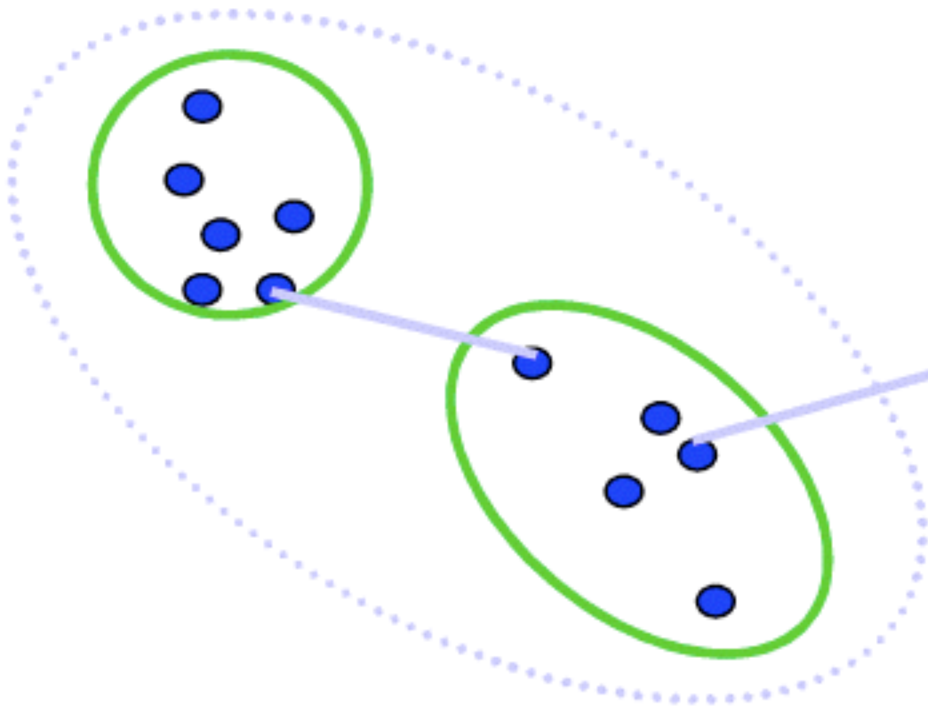


0	8	8	?
	0	2	?
		0	?
			0

Similarity between clusters

Single link

- cluster similarity = similarity of two most similar members

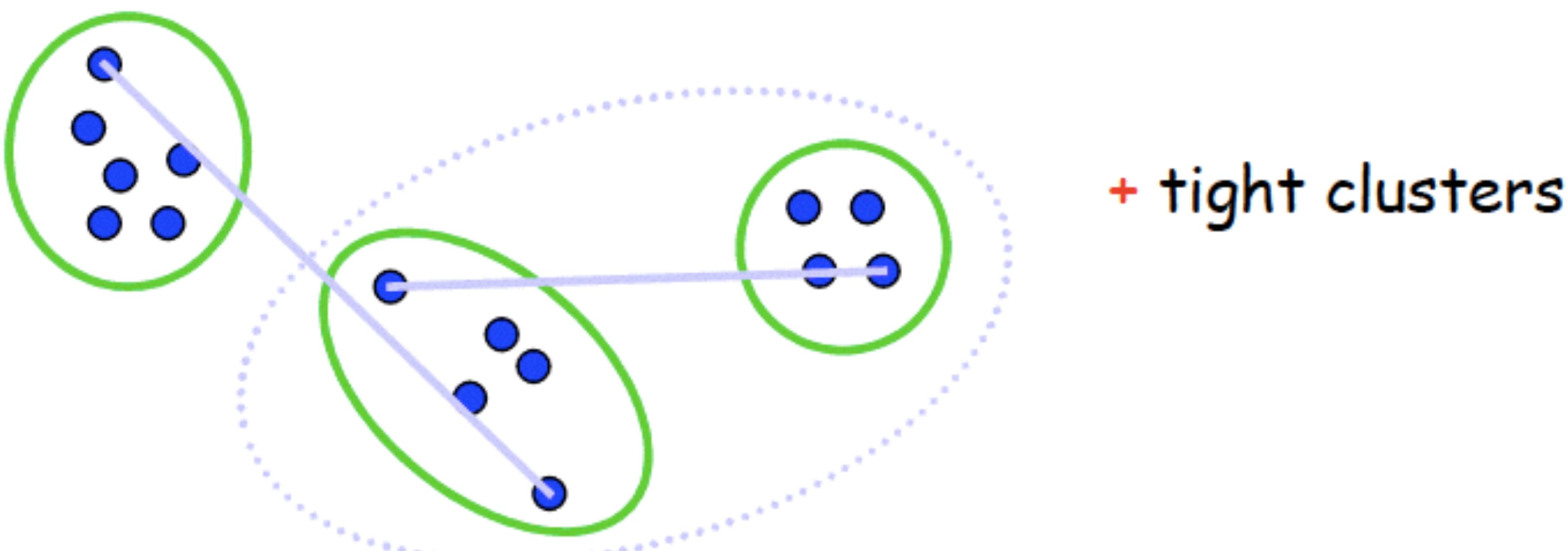


- Potentially long and skinny clusters

Similarity between clusters

Complete link

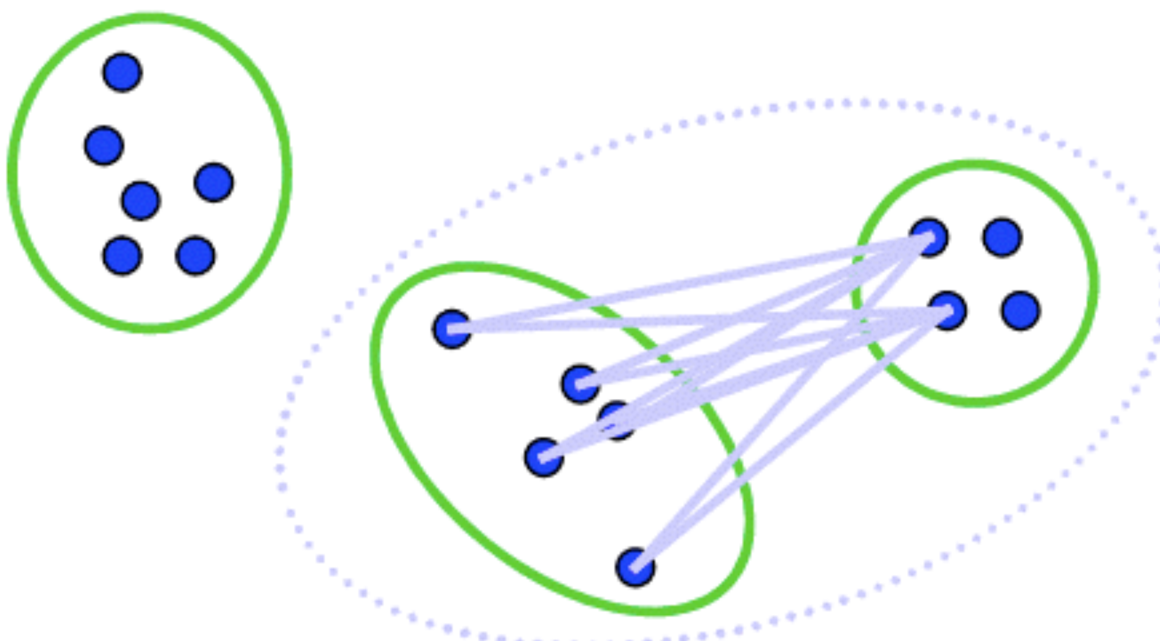
- cluster similarity = similarity of two least similar members



Similarity between clusters

Average link

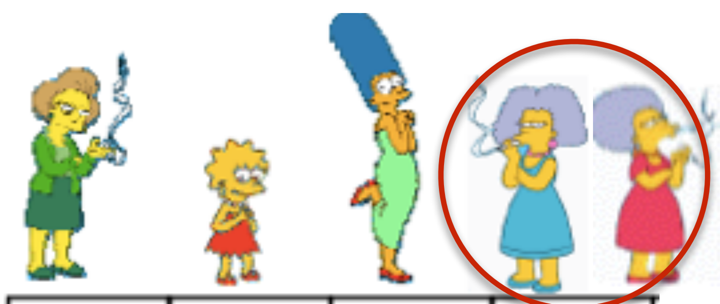
- cluster similarity = average similarity of all pairs



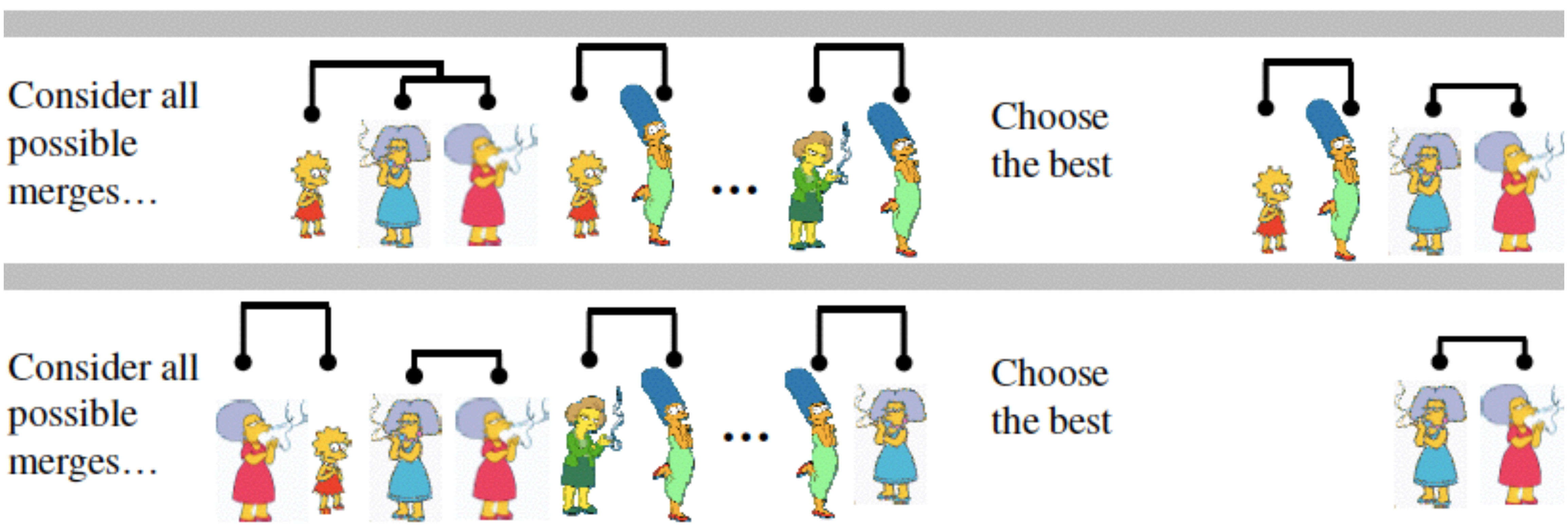
the most widely used similarity measure

Robust against noise

Hierarchical clustering

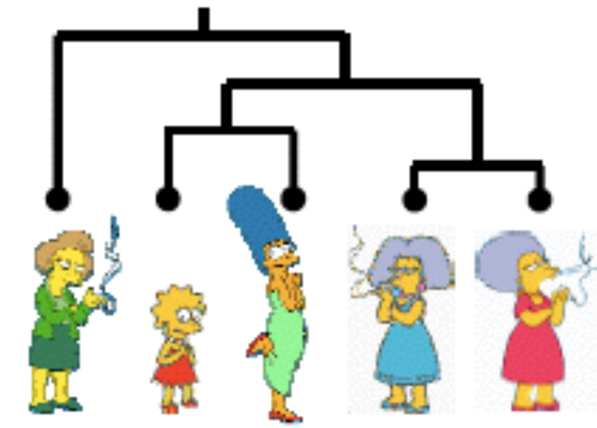


0	8	8	5
	0	2	3
		0	4
			0

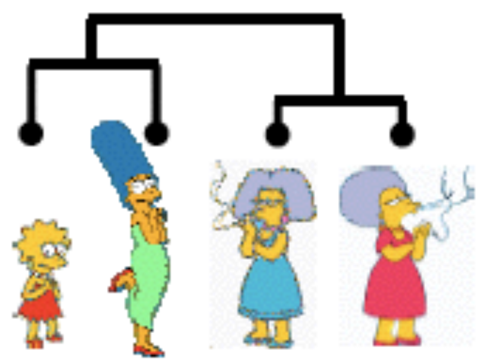

Consider all possible merges…
Choose the best
Consider all possible merges…
Choose the best

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.

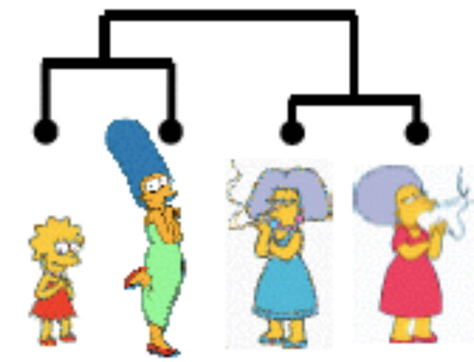


Consider all possible merges…

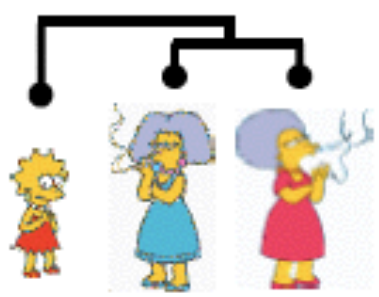


…

Choose the best

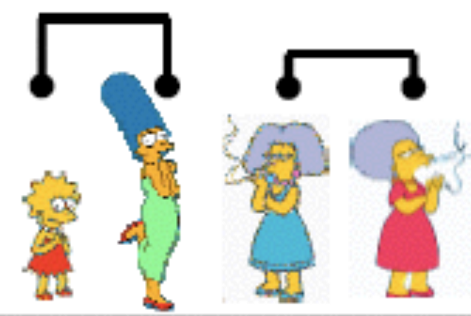


Consider all possible merges…

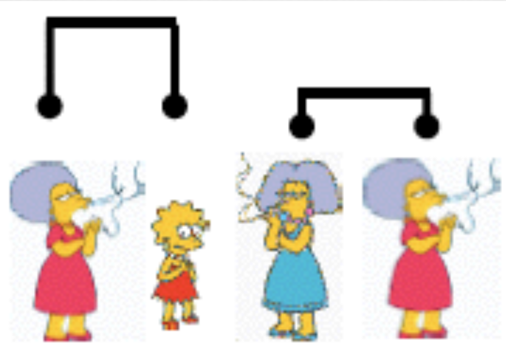


…

Choose the best



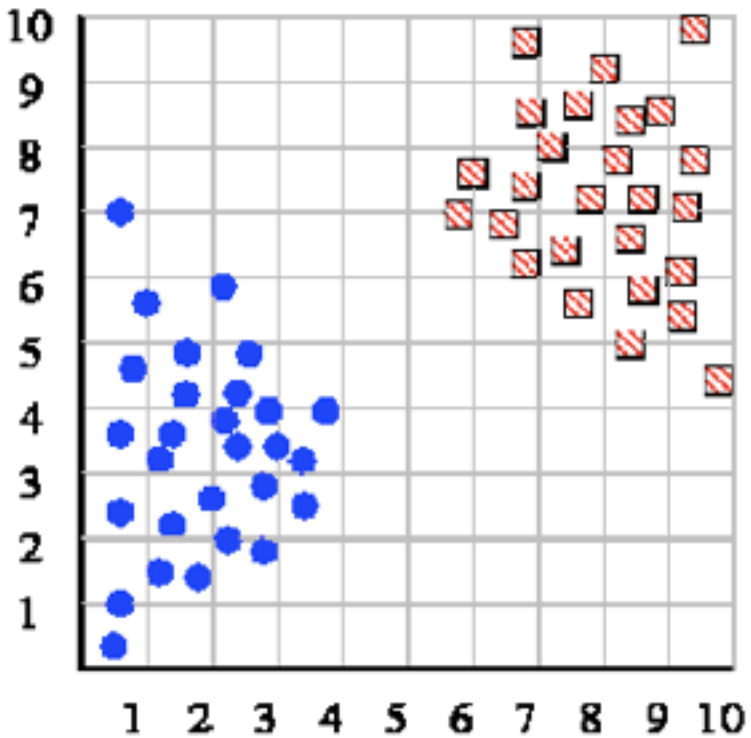
Consider all possible merges…

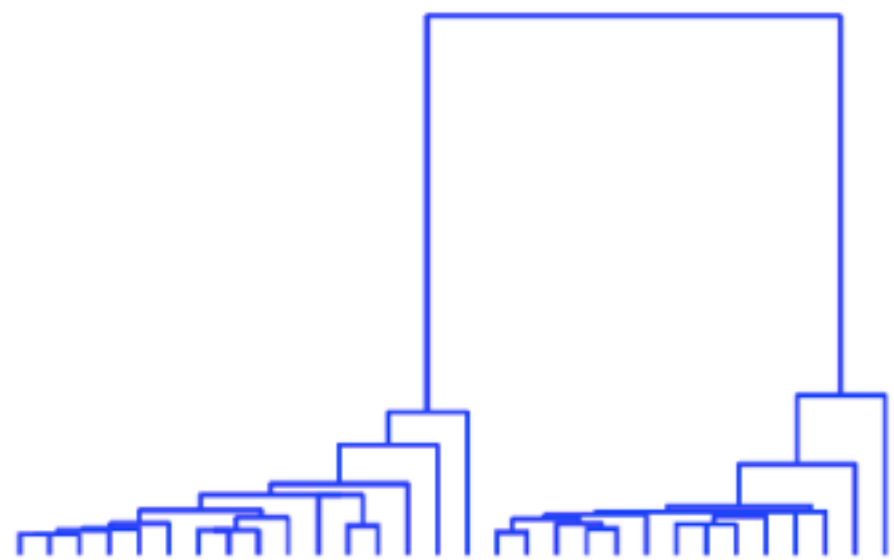


…

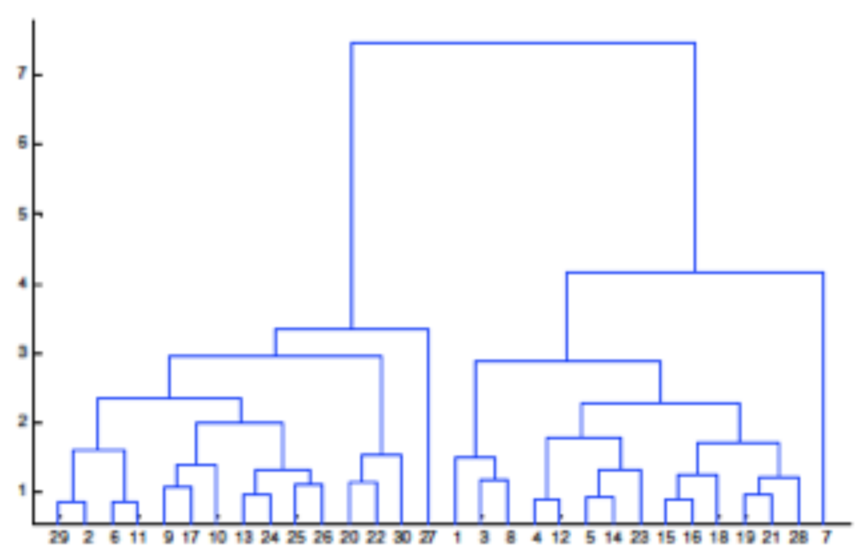
Choose the best

Agglomerative clustering





Single linkage



Average linkage

Agglomerative clustering

- No need to specify the numb rod clusters in advance
- Hierarchical structure is often more intuitive
- Do not scale well : Complexity is $\boldsymbol{O(n\,\hat{}\,2)}$ for n data points
- Interpretation is very subjective