Clustering

Instructor: Jessica Wu -- Harvey Mudd College

The instructor gratefully acknowledges Andrew Ng (Stanford), Eric Eaton (UPenn), David Kauchak (Pomona), David Sontag (NYU), Piyush Rai (Uath), and the many others who made their course materials freely available online.

Clustering

Learning Goals

- Describe goal of clustering
- Describe common applications of clustering

Clustering

Unsupervised learning technique that **detects patterns**

- Informally: find natural groups among objects
- Formally: organize data into **clusters** such that there is
  - High intra-cluster similarity
  - Low inter-cluster similarity

Most frequently, when people think of unsupervised learning, they think of clustering

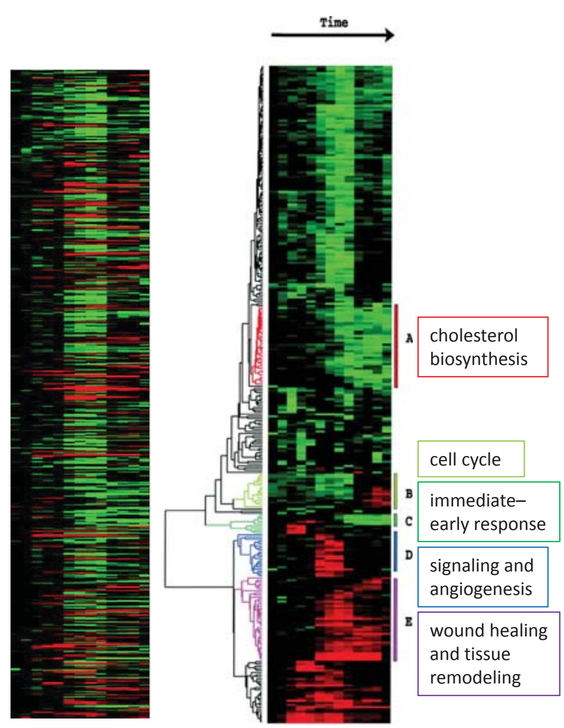
- Useful if you do not know what you are looking for
- But can produce gibberish

Based on slides by Eric Eaton and David Sontag

Gene Expression Data

Goal: Identify groups of genes with similar expression profiles

Cluster analysis and display of genome-wide expression patterns
MB Eisen, PT Spellman, PO Brown... - Proceedings of the ..., 1998 - National Acad Sciences
Abstract A system of cluster analysis for genome-wide expression data from DNA microarray hybridization is described that uses standard statistical algorithms to arrange genes according to similarity in pattern of gene expression. The output is displayed graphically, conveying the clustering and the underlying expression data simultaneously in a form intuitive for biologists. We have found in the budding yeast Saccharomyces cerevisiae that ...
Cited by 16411 Related articles All 128 versions Web of Science: 10393 Cite Save



Face Clustering

Goal: Identify similar faces

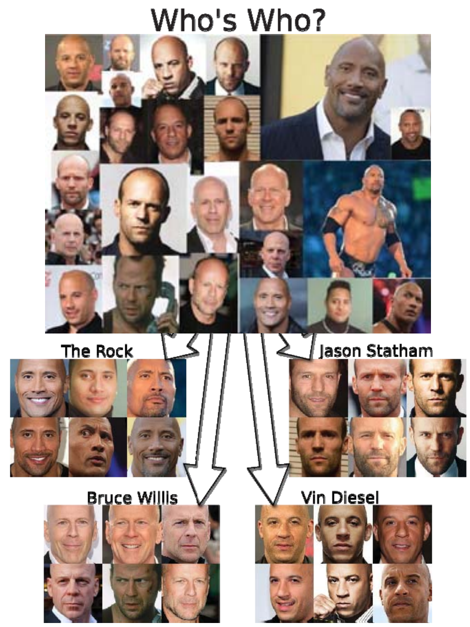
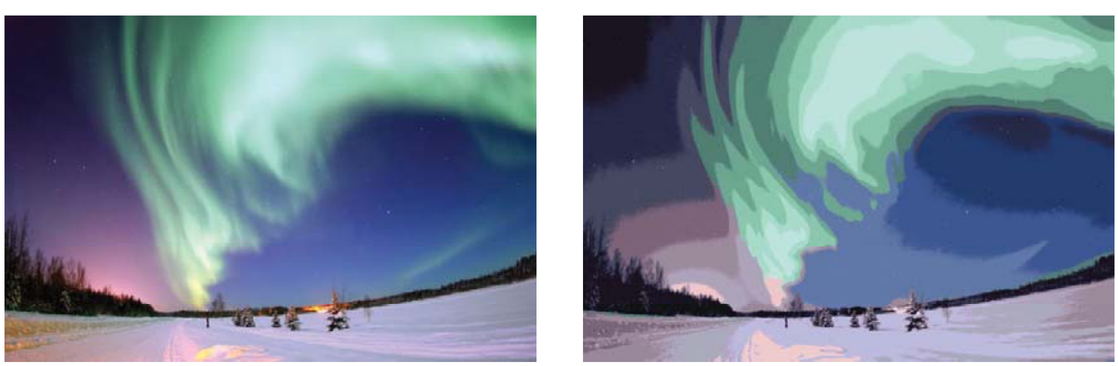


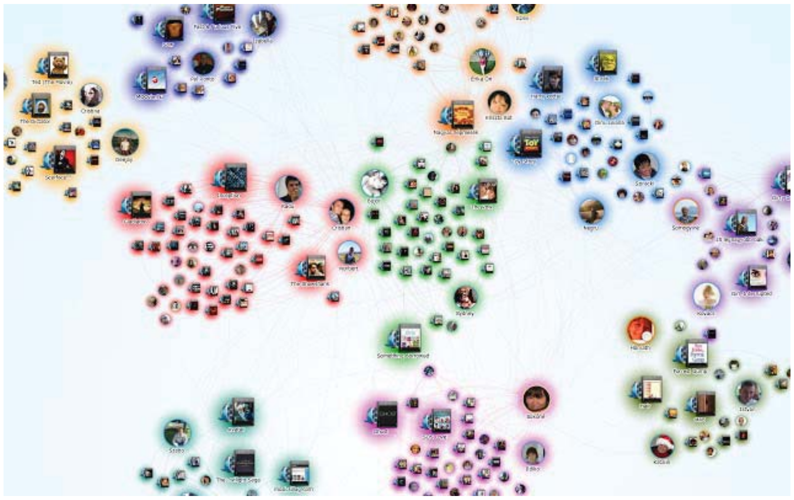
Image Segmentation

Goal: Break up image into meaningful or perceptually similar regions



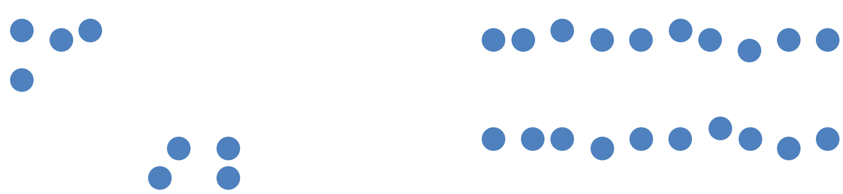
Social Graphs

Goal: Identify groups within Facebook friends.



Clustering

Group together similar points



Issues

- How do we represent an example?
- How do we compute similarity between two examples?
- How to cluster?
- How many clusters?

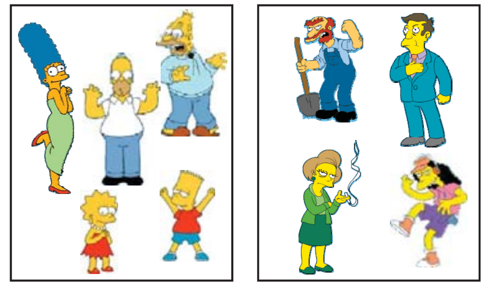
Domain knowledge – we will assume these are given / known

Based on slides by David Kauchak and David Sontag

Clustering Algorithms

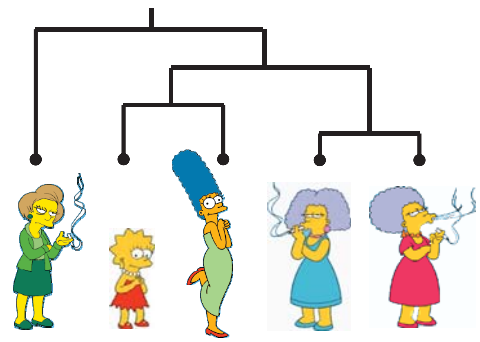
Flat / Partitional

- Construct various partitions then evaluate by some criterion
- Examples
  - k-means
  - mixture of Gaussians
  - spectral



Hierarchical

- Create hierarchical decomposition
- Examples
  - agglomerative (bottom-up)
  - divisive (top-down)



Based on slides by David Kauchak and David Sontag

(This slide intentionally left blank.)

K-Means Clustering

Learning Goals

- Describe *k*-means algorithm
- Describe *k*-means objective
- Describe *k*-means limitations and extensions

k-Means Algorithm

Given training set $\left\{\boldsymbol{x}^{(i)}\right\}_{i=1}^{n}$, $\boldsymbol{x}^{(i)} \in \mathbb{R}^d$, and number of clusters k

Goal: Group data into cohesive "clusters"

(1) Initialize cluster centers

(2) Repeat until convergence {

Assign each example to closest center

Update cluster centers

}

What details do we have to specify?

k-Means Algorithm

(1) Initialize k cluster centroids randomly

$$\boldsymbol{\mu}_1, \ldots, \boldsymbol{\mu}_k \in \mathbb{R}^d$$

(2) Repeat until convergence {

for $i = 1, \ldots, n$

set $c^{(i)} = \arg\min_{j=1,\ldots,k} \|\boldsymbol{x}^{(i)} - \boldsymbol{\mu}_j\|^2$

assign each training example $\boldsymbol{x}^{(i)}$ to closest cluster centroid $\boldsymbol{\mu}_j$

for $j = 1, \ldots, k$

set $\boldsymbol{\mu}_j = \dfrac{\sum_{i=1}^n \mathbb{I}\left[\left[c^{(i)} = j\right]\right] \boldsymbol{x}^{(i)}}{\sum_{i=1}^n \mathbb{I}\left[\left[c^{(i)} = j\right]\right]}$

move each cluster centroid $\boldsymbol{\mu}_j$ to mean of points assigned to it

}

Interactive Demo

https://www.naftaliharris.com/blog/visualizing-k-means-clustering/

Based on notes by Andrew Ng

(This slide intentionally left blank.)

Exercise

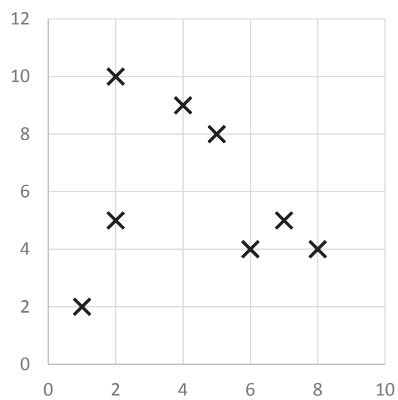
You are to cluster eight points:

$x^{(1)} = [2,10]^T \quad x^{(2)} = [2,5]^T \quad x^{(3)} = [8,4]^T \quad x^{(4)} = [5,8]^T$

$x^{(5)} = [7,\ 5]^T \quad x^{(6)} = [6,4]^T \quad x^{(7)} = [1,2]^T \quad x^{(8)} = [4,9]^T$

You assign $x^{(1)}$, $x^{(3)}$, and $x^{(7)}$ as initial centers (k = 3). Run *k*-means using Manhattan distance (ℓ_1-norm). Compute cluster centers and assignments for each round until convergence.

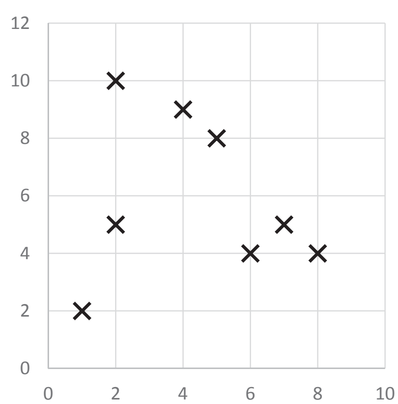
(hint: you may not need all the plots)



cluster 1 $\Rightarrow$

cluster 2 $\Rightarrow$

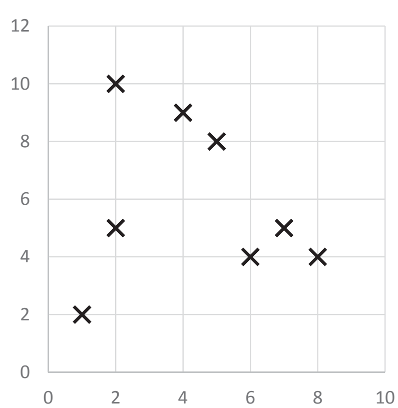
cluster 3 $\Rightarrow$



cluster 1 $\Rightarrow$

cluster 2 $\Rightarrow$

cluster 3 $\Rightarrow$



cluster 1 $\Rightarrow$

cluster 2 $\Rightarrow$

cluster 3 $\Rightarrow$

Optimization Objective

Is k-means guaranteed to converge? yes

Define **distortion function**

$$J(\boldsymbol{c}, \boldsymbol{\mu}) = \sum_{i=1}^{n} \|\boldsymbol{x}^{(i)} - \boldsymbol{\mu}_{c^{(i)}}\|^2$$

where $\boldsymbol{c} = [c^{(1)}, \ldots, c^{(n)}]^T$ and $\boldsymbol{\mu} = [\boldsymbol{\mu}_1, \ldots, \boldsymbol{\mu}_k]^T$.

J measures sum of squared distances between each training example $\boldsymbol{x}^{(i)}$ and the cluster centroid to which it has been assigned. In other words, it measures intra-class variance.

Optimization Objective

Claim: k-means is **coordinate descent** on J.

Inner loop of k-means algorithm repeatedly…

- holds $\boldsymbol{\mu}$ fixed and minimizes $J(\boldsymbol{c},\boldsymbol{\mu})$ w.r.t. $\boldsymbol{c}$, and then
- holds $\boldsymbol{c}$ fixed and minimizes $J(\boldsymbol{c},\boldsymbol{\mu})$ w.r.t. $\boldsymbol{\mu}$.

Thus

- J must monotonically decrease, and
- the value of J must converge.

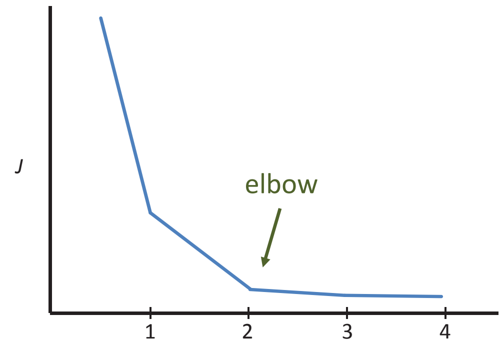
Usually, this implies that c and μ will converge, too. In theory, it is possible for k-means to oscillate between a few different clusterings (a few different values for $\boldsymbol{c}$ and/or μ that have exactly the same value of J), but this almost never happens in practice.

Exact optimization of J is NP-hard. K-means is heuristic that converges to local optimum.

Choosing the Number of Clusters

Elbow method

- Try different values of *k*
- Plot objective vs *k*
- Look for "elbow"



Application-Specific

- If running *k*-means to use clusters for some later purpose
- Evaluate *k*-means based on metric for how well it performs for that later purpose
- E.g. t-shirt size, clusters using height and weight as features
  - 3 clusters $\Rightarrow$ S, M, L $\Rightarrow$ fewer sizes, can make shirts more cheaply
  - 5 clusters $\Rightarrow$ XS, S, M, L, XL $\Rightarrow$ more sizes, can make better-fitting shirts

Based on slides by Andrew Ng

k-Means Time Complexity

(1) Initialize *k* cluster centroids randomly

$$\boldsymbol{\mu}_1, \ldots, \boldsymbol{\mu}_k \in \mathbb{R}^d$$

(2) Repeat until convergence {

for $i = 1, \ldots, n$

set $c^{(i)} = \underset{j=1,\ldots,k}{\arg\min} \|\boldsymbol{x}^{(i)} - \boldsymbol{\mu}_j\|^2$

for $j = 1, \ldots, k$

set $\boldsymbol{\mu}_j = \dfrac{\sum_{i=1}^n \mathbb{I}\left[\left[c^{(i)} = j\right]\right] \boldsymbol{x}^{(i)}}{\sum_{i=1}^n \mathbb{I}\left[\left[c^{(i)} = j\right]\right]}$

}

What is time complexity?
(*k* clusters, *n* training examples, *d* features, *i* iterations)

Initialization Issues

Problem

- Often randomly pick $k < n$ examples as starting centers
- But *k*-means is extremely sensitive to cluster center initialization

Reasoning

- Distortion function J is non-convex function
- *k*-means can be susceptible to local optima

Bad initialization can lead to

- Poor convergence speed
- Bad overall clusterings

Based on notes by Andrew Ng

Initialization Issues

In practice

- Very often *k*-means works fine and comes up with very good clusterings despite local minima

Safeguarding approaches

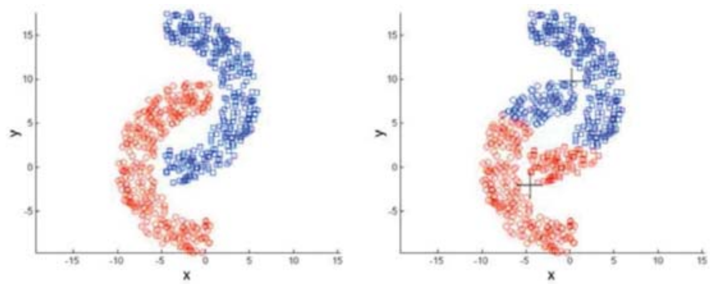
- Run *k*-means many times (using different random initializations), then choose clustering that gives lowest distortion
- *k*-means++: spread out cluster centers
  - choose first center uniformly at random from examples
  - choose remaining centers from remaining examples with probability proportional to squared distance from example's closest cluster center

Based on notes by Andrew Ng

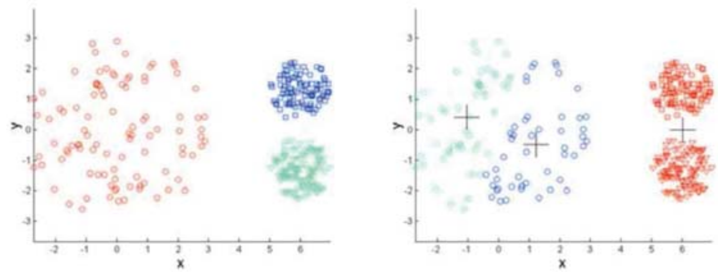
Limitations

k-means works well only for round-shaped, roughly equal size / density clusters

non-convex / non-round shaped clusters

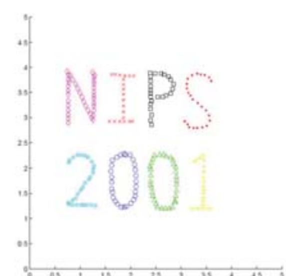


different density clusters



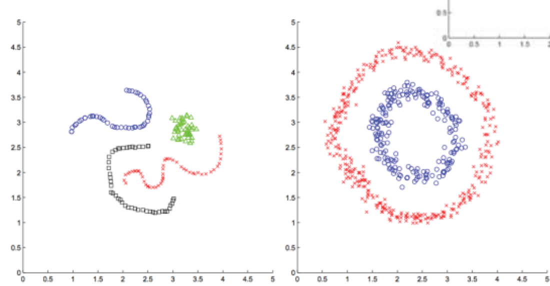
Based on slides by Piyush Rai [Images by Christof Monz, Queen Mary, Univ of London]

Extensions



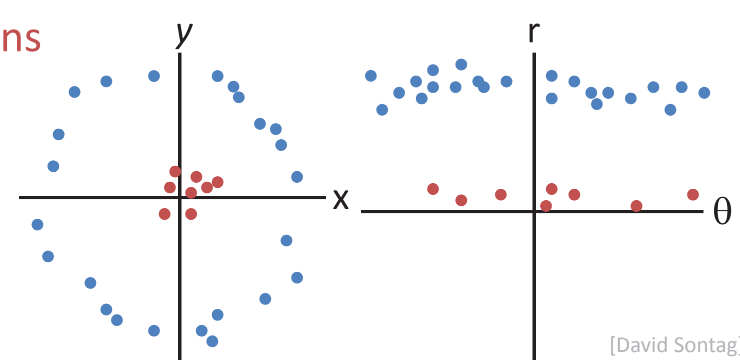
Problem: clusters have non-convex shapes

- spectral clustering (value connectivity over compactness)



[Andrew Ng]

- kernelized *k*-means



[David Sontag]

Extensions

Problem: sensitive to outlier examples

- *k*-medians
  - median more robust than mean in presence of outliers

Problem: cluster centers not an actual example

- *k*-medoids
  - medoid = element of cluster whose average dissimilarity to all elements in cluster is minimal ("most centrally located point in cluster")

Based on slides by Piyush Rai

Extensions

Problem: makes hard assignments of points to clusters

- a point completely belongs to cluster or not at all
- no notion of soft assignment

- fuzzy *k*-means
  - let weight $w_{ij} \in [0,1]$ measure "degree of belonging" (degree to which element $\boldsymbol{x}^{(i)}$ belongs to cluster c_j)

$$\boldsymbol{\mu}_j = \frac{\sum_{i=1}^{n} w_{ij} \boldsymbol{x}^{(i)}}{\sum_{i=1}^{n} w_{ij}} \qquad J(\boldsymbol{c}, \boldsymbol{\mu}) = \sum_{i=1}^{n} \sum_{j=1}^{k} w_{ij} \|\boldsymbol{x}^{(i)} - \boldsymbol{\mu}_{c^{(i)}}\|^2$$

w_{ij} typically inversely related to distance from cluster center
compare standard (hard assignment) *k*-means: $w_{ij} = \mathbb{I}\left[\left[c^{(i)} = j\right]\right]$

- Gaussian Mixture Models (GMMs)
  - more statistically formalized method
  - probabilistic assignments to clusters and multivariate Gaussian distributions
  - next time…

Hierarchical Clustering

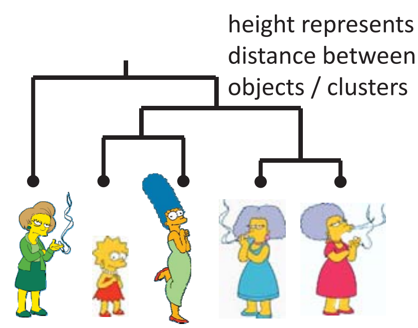
Learning Goals

- Describe agglomerative clustering algorithm
- Define single link, complete link, and average link and describe how they affect clustering

Agglomerative (aka bottom-up hierarchical) Clustering

Idea

- Start with each item in its own cluster
- Find best pair to merge into new cluster
- Repeat until all clusters are merged



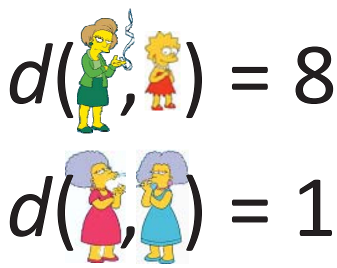
Properties

- Produces not one clustering but a family of clusterings represented by a dendrogram
- # of dendograms with n leaves = $\frac{(2n-3)!}{(2^{n-2})(n-2)!}$
- Compare to agglomerative algorithm
  n loops, on each loop compute n^2 distances
  time complexity: $O(n^3)$

n	# dendrograms
2	1
3	3
4	15
5	105
…	
10	34,459,425

Distance Matrix

Contains distances between every pair of objects in training set

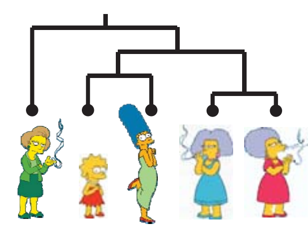


0	8	8	7	7
	0	2	4	4
		0	3	3
			0	1
				0

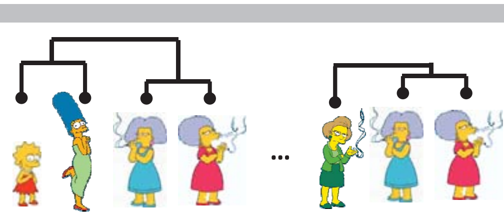
Based on slides by Ziv Bar-Joseph

Bottom-up (agglomerative)

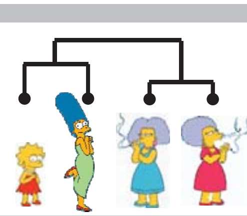
- Starting with each item in its own cluster
- Find best pair to merge into new cluster
- Repeat until all clusters are merged



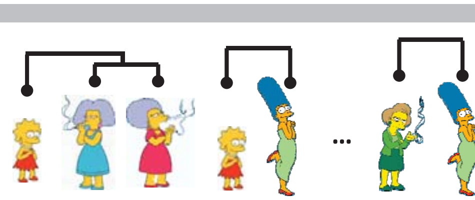
Consider all possible merges…



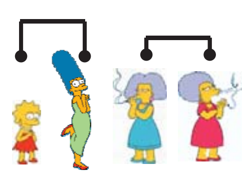
Choose the best



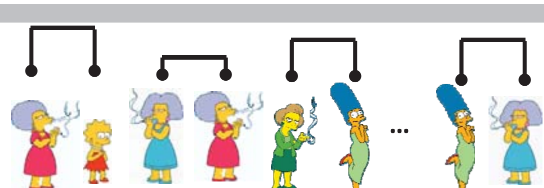
Consider all possible merges…



Choose the best



Consider all possible merges…



Choose the best

Based on slides by Ziv Bar-Joseph

Distance between Clusters

Match the linkage type to the behavior and picture (separately)

single linkage
  use closest pair

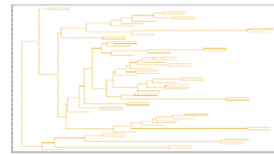
complete linkage
  use farthest pair

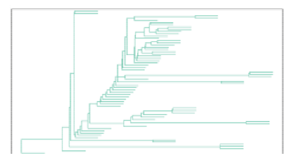
average linkage
  use average of
    all pairs

tight (small, round)
  clusters

robust against noise

potentially long and
  skinny clusters





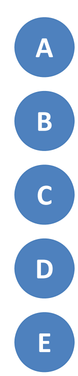
Based on slides by Ziv Bar-Joseph and Piyush Rai

[Mouse tumor data, Hastie et al.]

Exercise

Given distance matrix, run single-link (closest pair) clustering.

	A	B	C	D	E
A	0				
B	2	0			
C	6	3	0		
D	10	9	7	0	
E	9	8	5	4	0



Summary Comparison

Flat	Hierarchical
• Partitions independent of one another	• Produces different partitionings depending on level of granularity (refine or coarsen clusters by picking different *k*)
• Produces single partitioning	• Partitions nested within one another
• Requires *k* as input	• Can pick number of clusters after clustering
• More efficient runtime-wise	• Can be slow (has to make several merge/split decisions)

No clear consensus on which produces better clustering

Evaluation

Learning Goals

- Describe how to validate clusters produced by algorithms

Validation

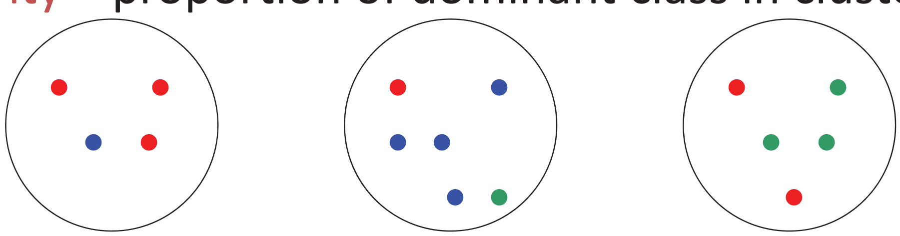
How “good” are our clusters?

- external validation
  - match to known categories (cluster data without labels, see how well we reproduce labels)
  - more common
- internal validation
  - no external labels

Based on slides by Ziv Bar-Joseph

External Validation

purity = proportion of dominant class in cluster



Cluster I   Cluster II   Cluster III

I: purity = (max(3, 1, 0)) / 4 = 3/4
II: purity = (max(1, 4, 1)) / 6 = 4/6
III: purity = (max(2, 0, 3)) / 5 = 3/5

overall purity

cluster average

$$\frac{\frac{3}{4}+\frac{4}{6}+\frac{3}{5}}{3} = 0.672$$

weighted average

$$\frac{4\left(\frac{3}{4}\right)+6\left(\frac{4}{6}\right)+5\left(\frac{3}{5}\right)}{15} = \frac{3+4+3}{15} = \frac{2}{3}$$

Based on slides by David Kauchak

Internal Validation

stability: if clusters capture real structure, they should be stable to minor perturbation (e.g. subsampling) of data

- Need measure of similarity between two *k*-clusterings
  - For any set of clusters C, define $L(C)$ as matrix of 0/1 labels
    - $L(C)_{ij} = 1$ if examples $\boldsymbol{x}^{(i)}$ and $\boldsymbol{x}^{(j)}$ belong to same cluster
    - $L(C)_{ij} = 0$ otherwise
  - Let $S = sim(L(C),\, L(C'))$ be similarity between two matrices
    - e.g. fraction of identical elements in matrices
- Let…
  - C denote clusters from all samples
  - $C_i{}'$ denote clusters from i^{th} randomly chosen subset of samples
  - S_n denote average of n scores between $L(C)$ and $L(C_i{}')$
- Have high confidence in C if $S_n \to 1$ with high probability
  (where comparison is done over samples common to both)

Based on slides by Ziv Bar-Joseph

Take-Aways

- Clustering basics
  - what it is
  - why it is useful
- Clustering algorithms
  - *k*-means
    - algorithm
    - objective (distortion function)
    - issues (initialization, choosing *k*, convex clusters)
    - extensions (*k*-medians, *k*-medoids)
  - agglomerative
    - algorithm
    - single-linkage, complete-linkage, average-linkage
- Clustering metrics
  - external
  - internal