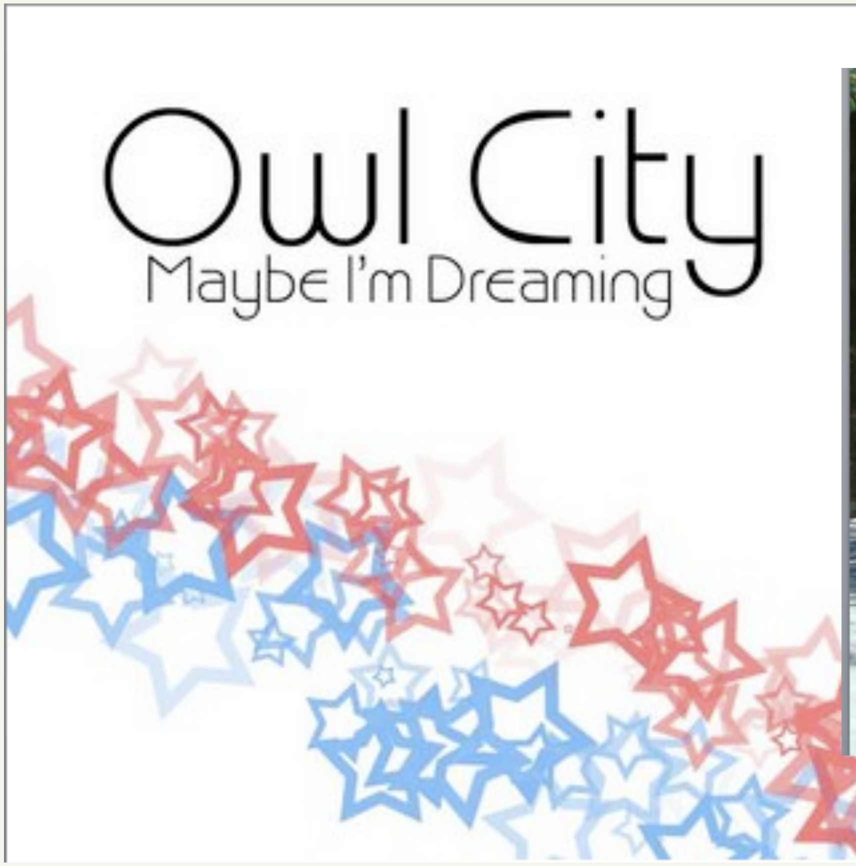


http://ps3.kombo.com/images/content/news/blurb_valve_store_camiseta_portal_2008-730.jpges

Fireflies

http://www.youtube.com/watch?v=Y6ljFaKRTrI

Hierarchical Clustering

David Kauchak
cs160
Fall 2009

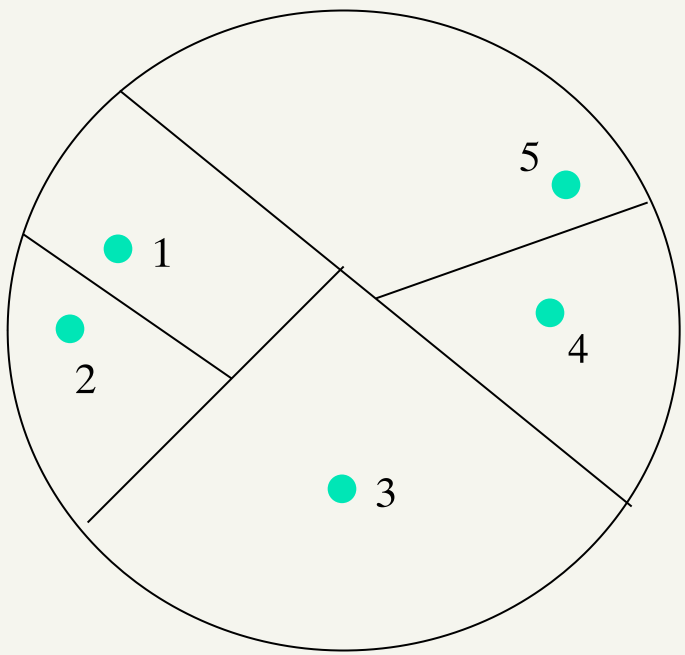
some slides adapted from:
http://www.stanford.edu/class/cs276/handouts/lecture17-clustering.ppt

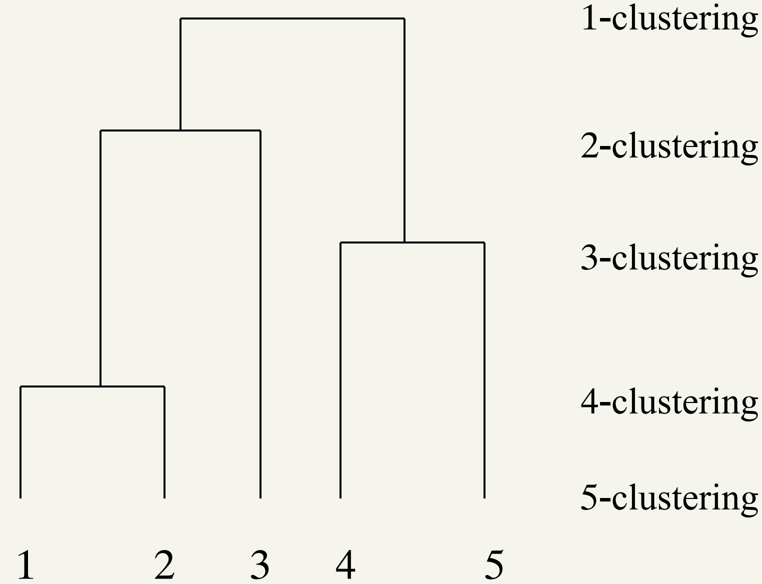
Administrative

- Project schedule
- Ethics in IR lecture
  - http://www.cs.pomona.edu/classes/cs160/ethics.html

Hierarchical Clustering

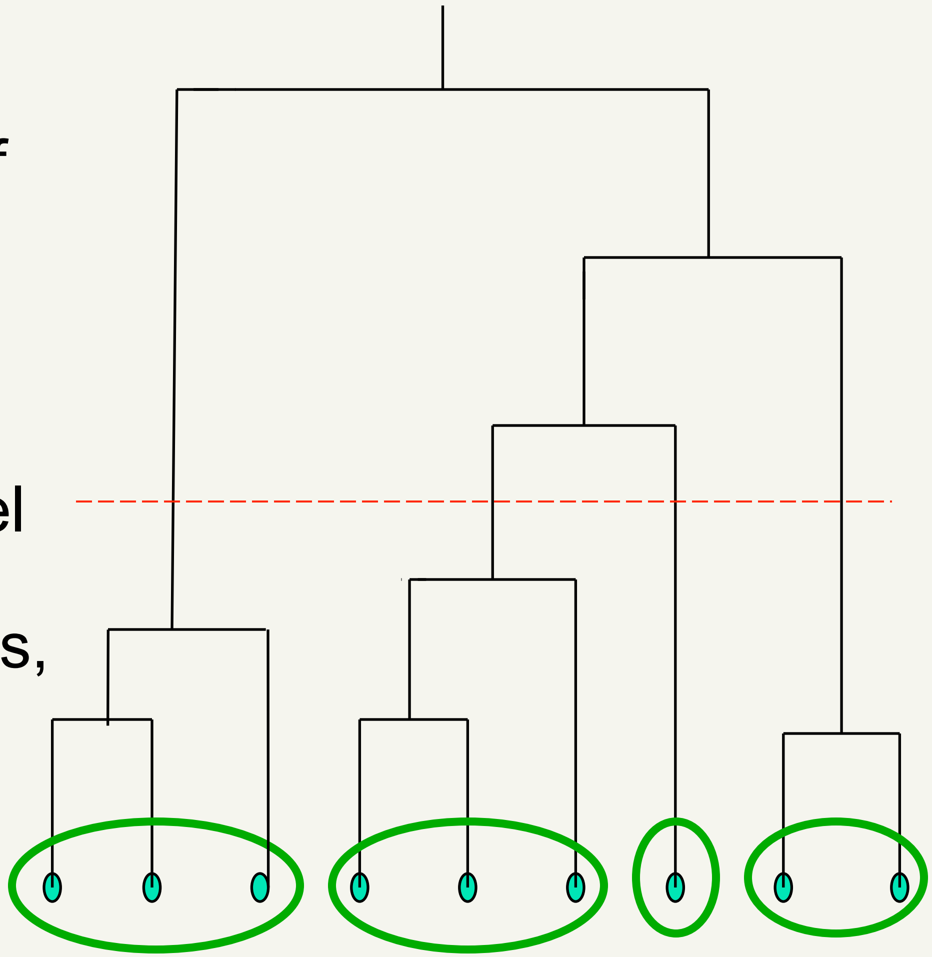
Recursive partitioning/merging of a data set





Dendogram

- Represents all partitionings of the data
- We can get a K clustering by looking at the **connected** components at any given level
- Frequently binary dendograms, but n-ary dendograms are generally easy to obtain with minor changes to the algorithms



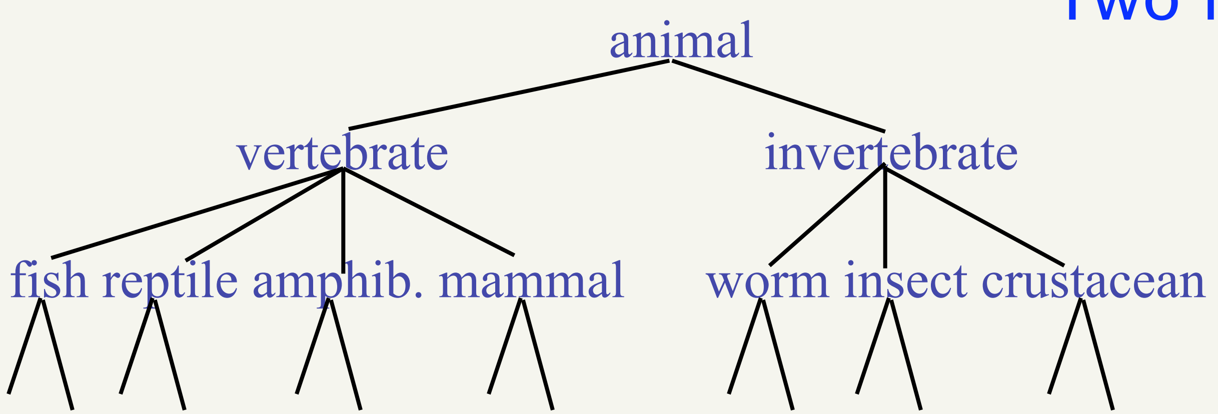
Advantages of hierarchical clustering

- Don’t need to specify the number of clusters
- Good for data visualization
  - See how the data points interact at many levels
  - Can view the data at multiple levels of granularity
  - Understand how all points interact
- Specifies all of the K clusterings/partitions

Hierarchical Clustering

- Common in many domains
  - Biologists and social scientists
  - Gene expression data
  - Document/web page organization
    - DMOZ
    - Yahoo directories

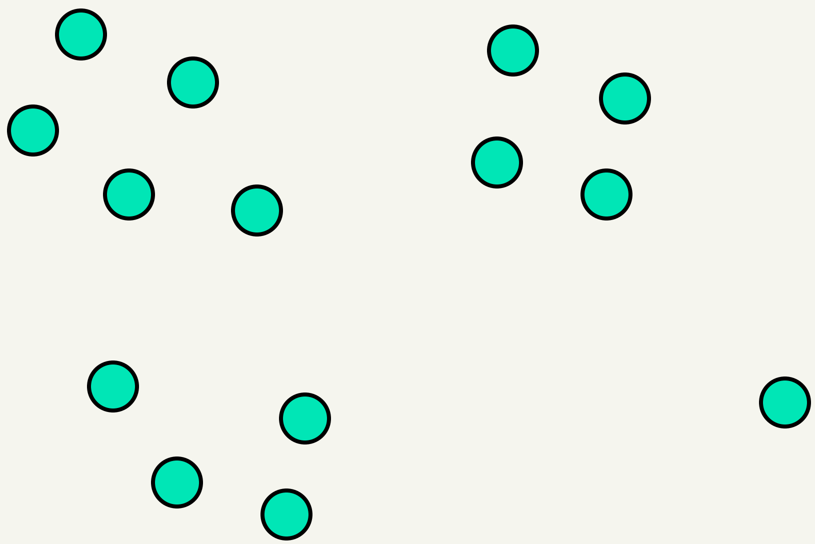
Two main approaches…



Divisive hierarchical clustering

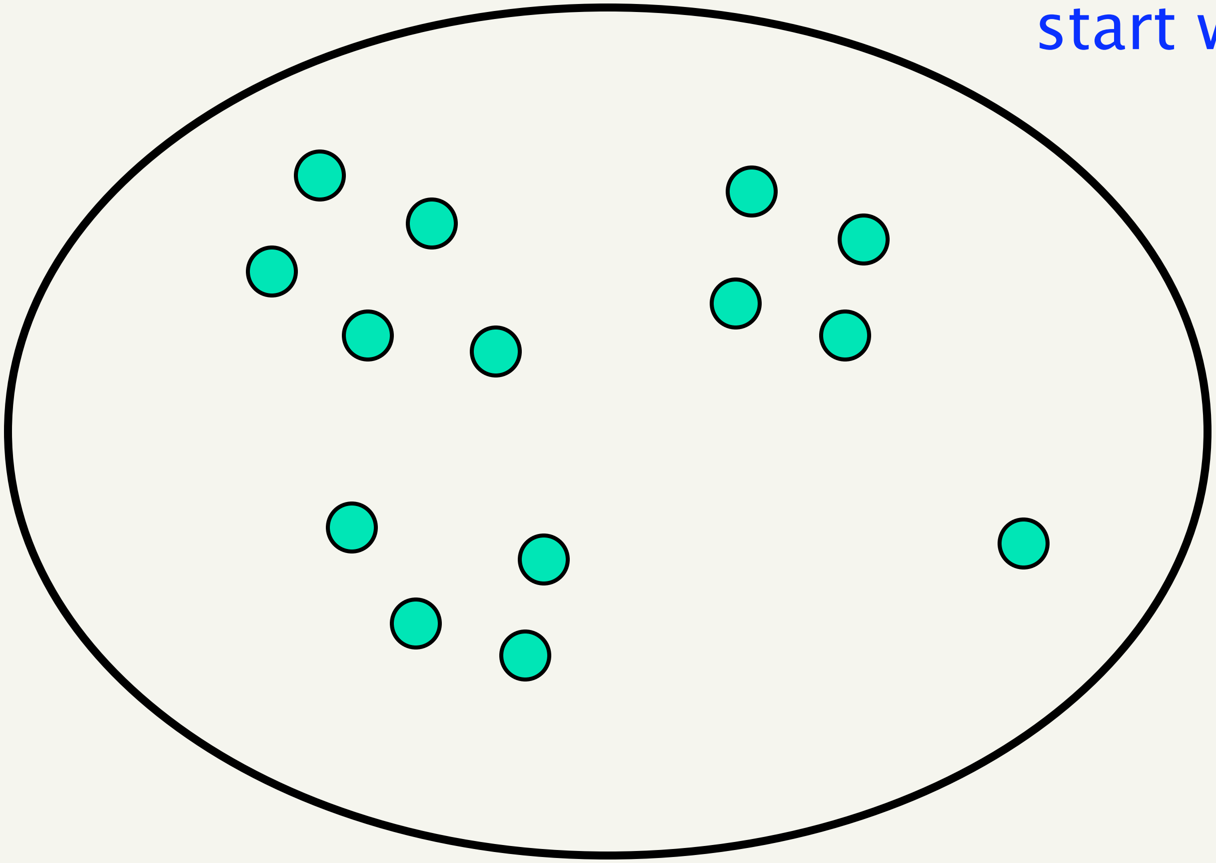
- Finding the best partitioning of the data is generally exponential in time
- Common approach:
  - Let **C** be a set of clusters
  - Initialize **C** to be the one-clustering of the data
  - While there exists a cluster *c* in **C**
    - remove *c* from **C**
    - partition *c* into 2 clusters using a flat clustering algorithm, c_1 and c_2
    - Add to c_1 and c_2 **C**
- Bisecting k-means

Divisive clustering



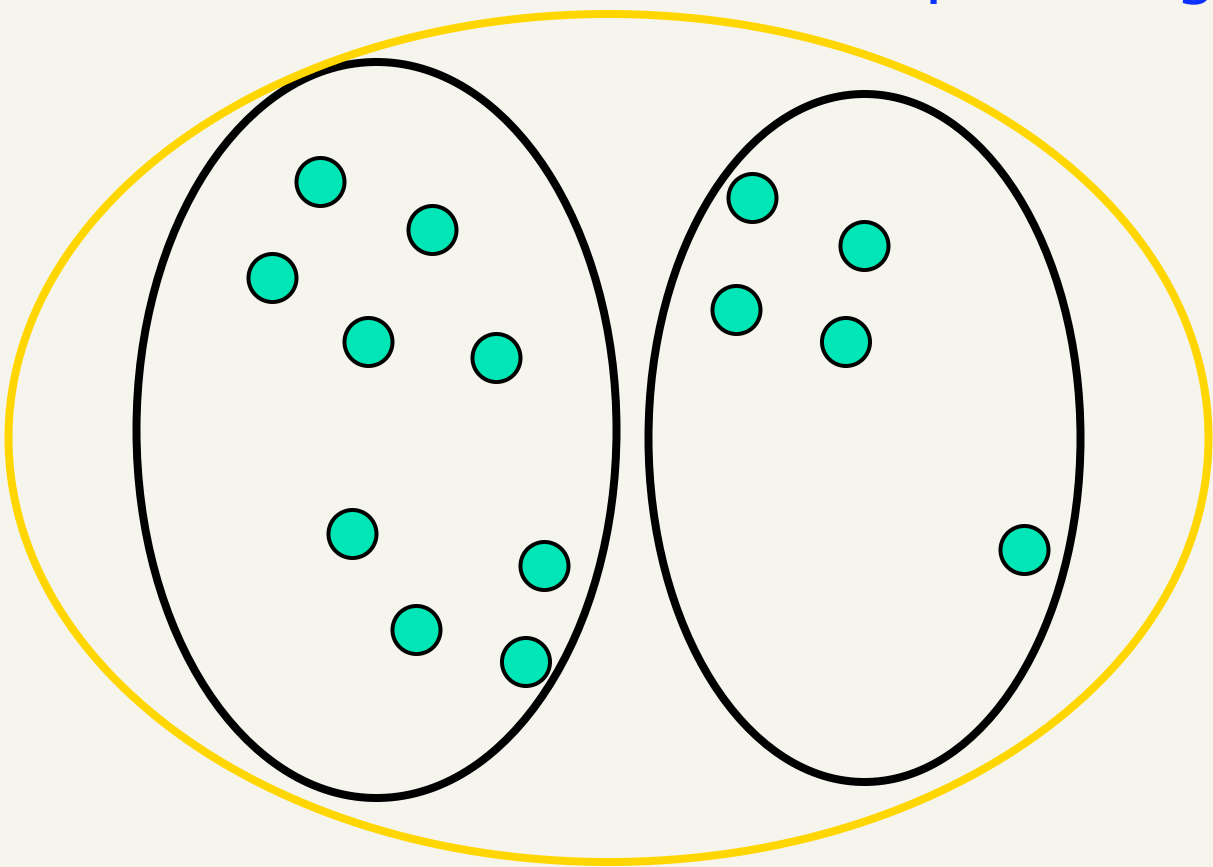
Divisive clustering

start with one cluster

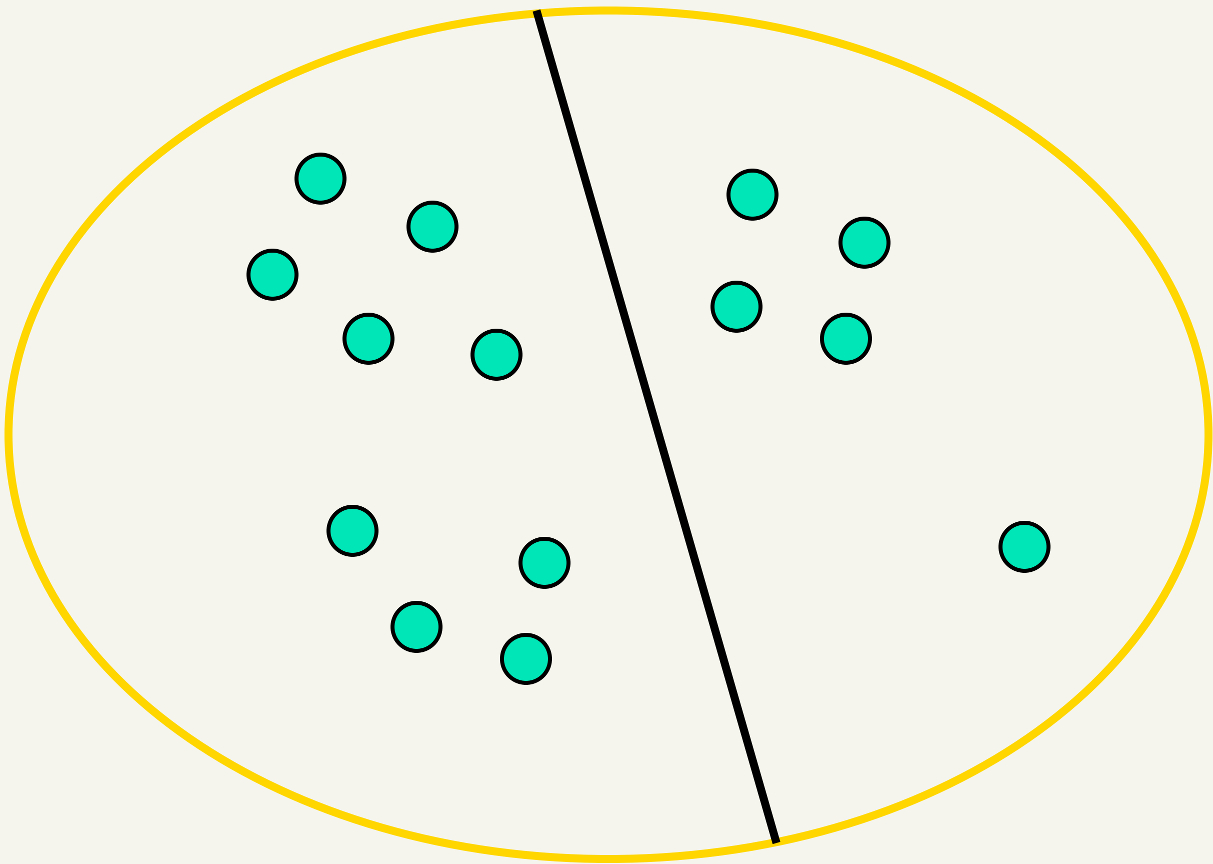


Divisive clustering

split using flat clustering

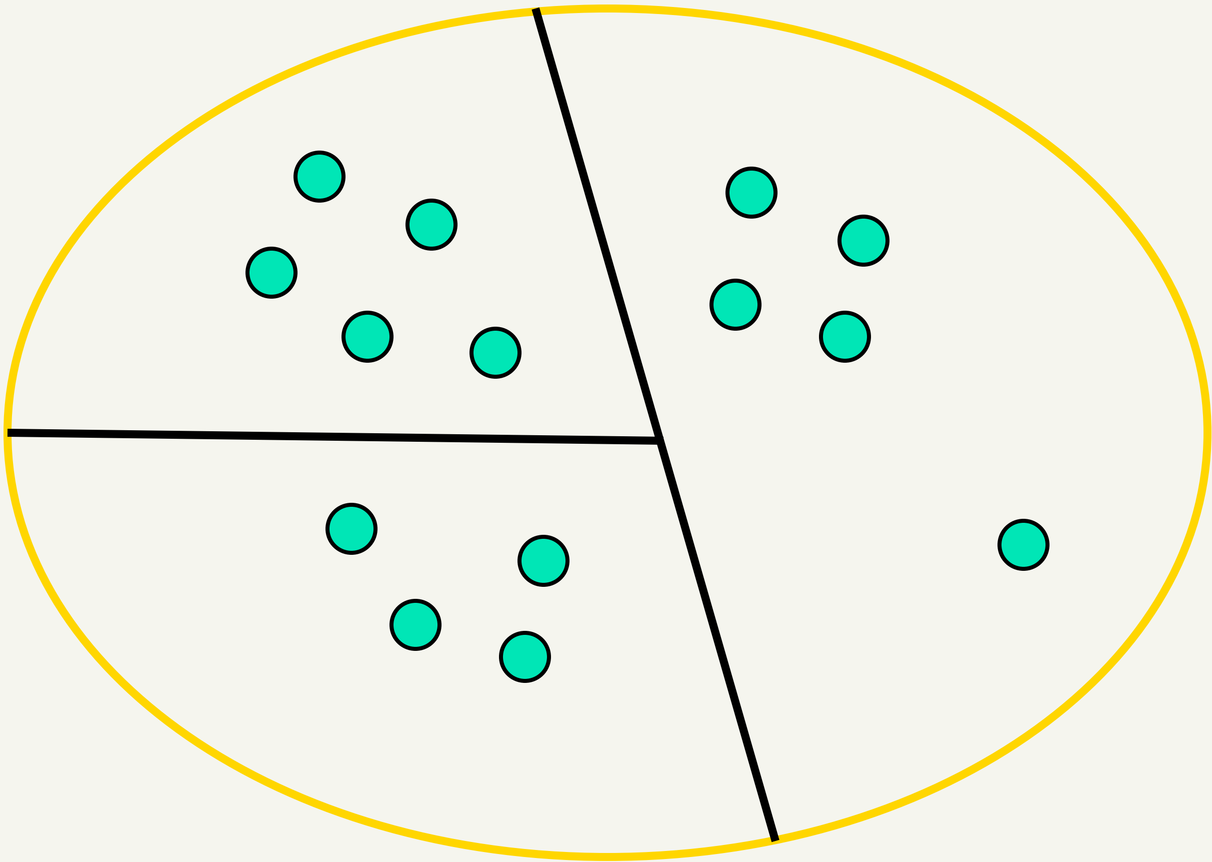


Divisive clustering



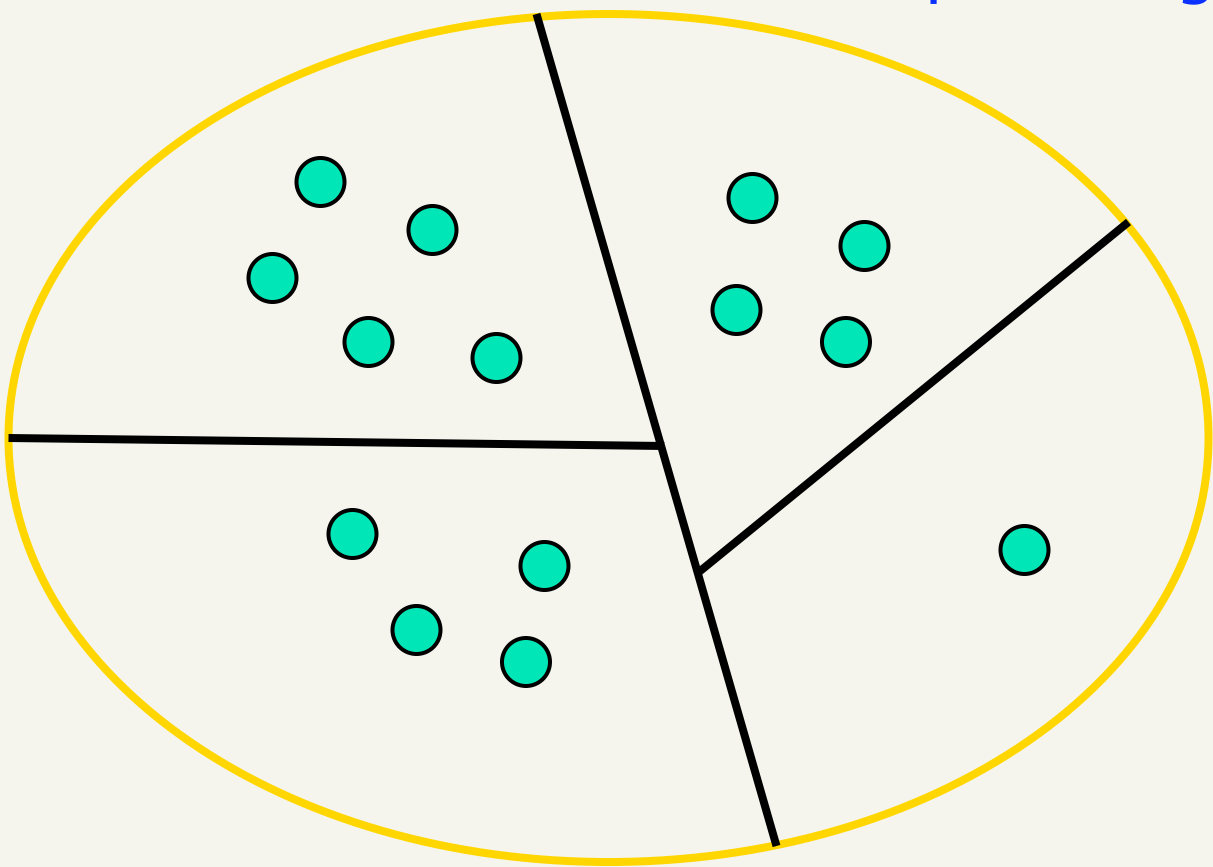
Divisive clustering

split using flat clustering

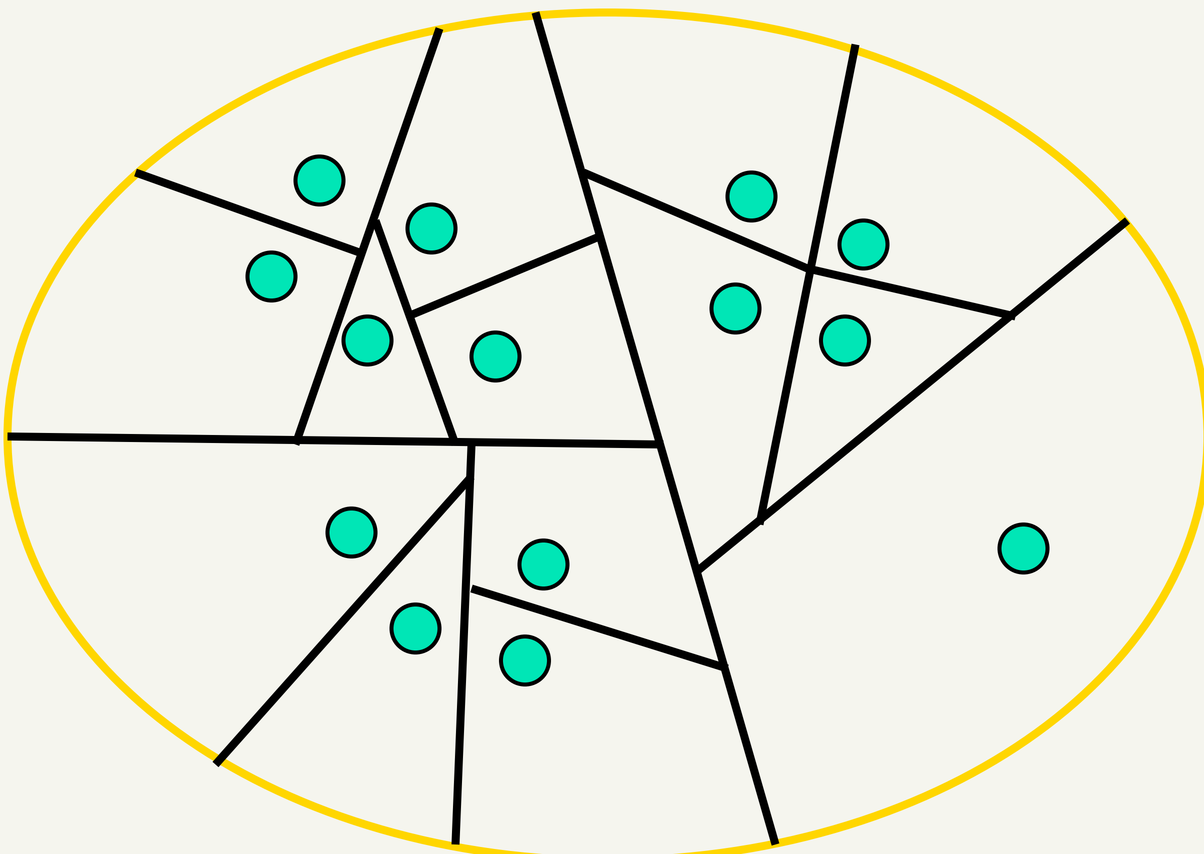


Divisive clustering

split using flat clustering



Divisive clustering



Note, there is a “temporal” component not seen here

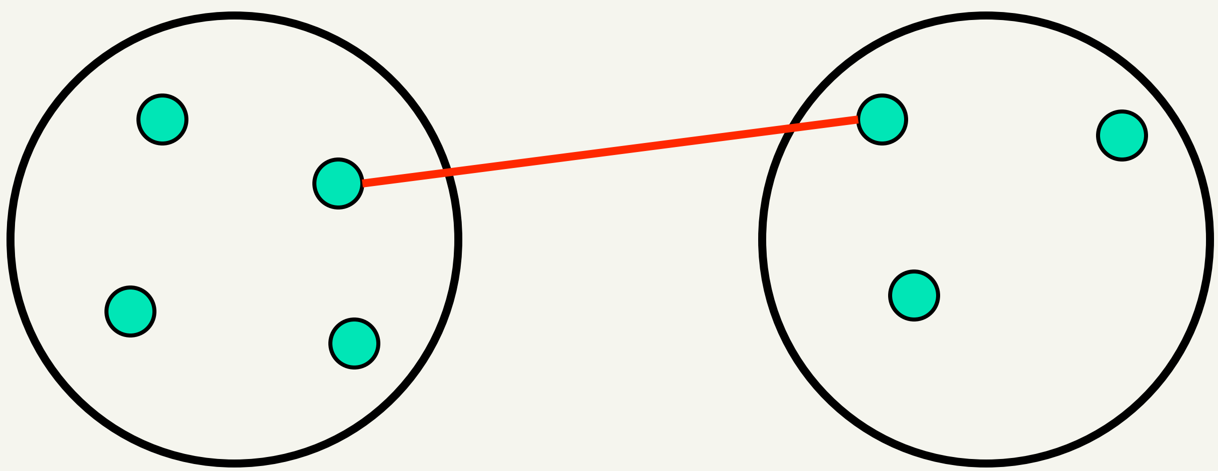
Hierarchical Agglomerative Clustering (HAC)

- Let **C** be a set of clusters
- Initialize **C** to be all points/docs as separate clusters
- While **C** contains more than one cluster
  - find c_1 and c_2 in **C** that are closest together
  - remove c_1 and c_2 from **C**
  - merge c_1 and c_2 and add resulting cluster to **C**
- The history of merging forms a binary tree or hierarchy

- How do we measure the distance between clusters?

Distance between clusters

- **Single-link**
  - Similarity of the *most* similar (single-link)

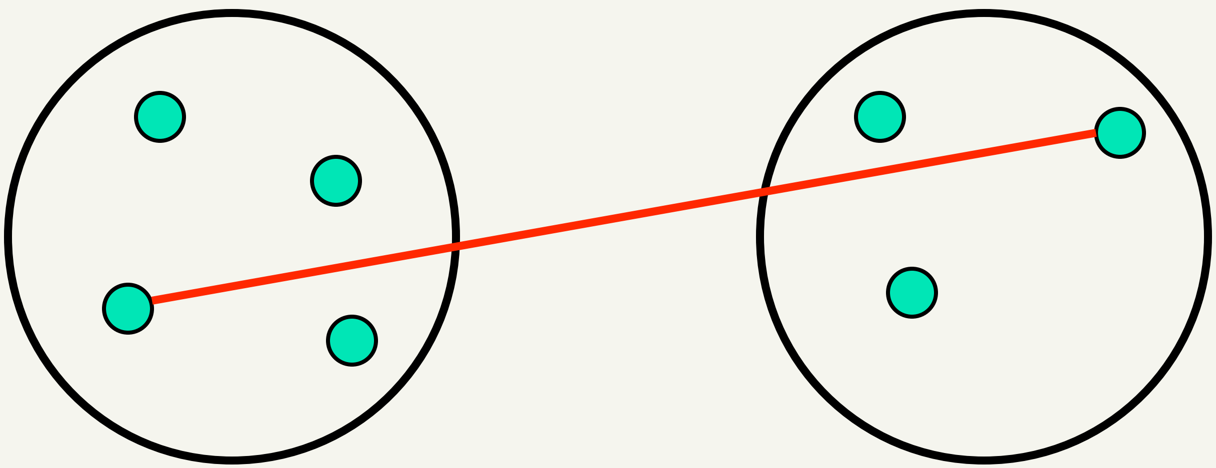


$$\max_{l \in L, r \in R} sim(l,r)$$

Distance between clusters

- **Complete-link**
  - Similarity of the "furthest" points, the *least* similar

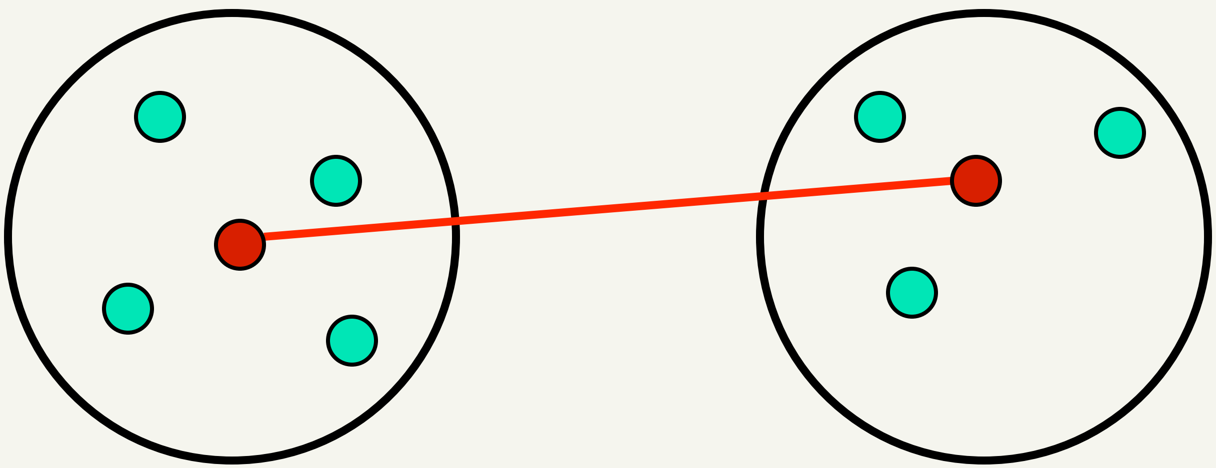
$$\min_{l \in L, r \in R} sim(l,r)$$



Why are these "local" methods used? efficiency

Distance between clusters

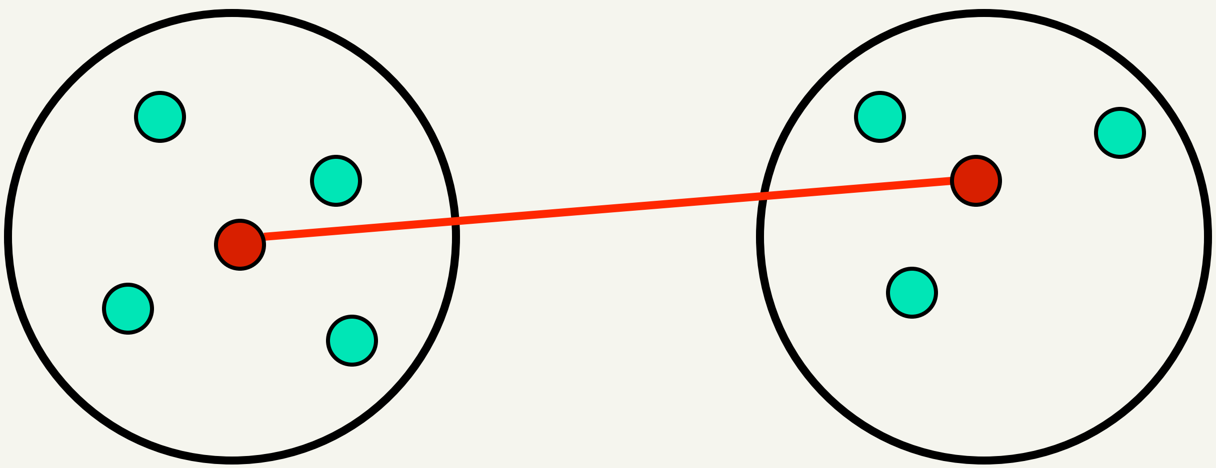
- **Centroid**
  - Clusters whose centroids (centers of gravity) are the most similar



$$\left\| \mu(L) - \mu(R) \right\|^2$$

Distance between clusters

- **Centroid**
  - Clusters whose centroids (centers of gravity) are the most similar



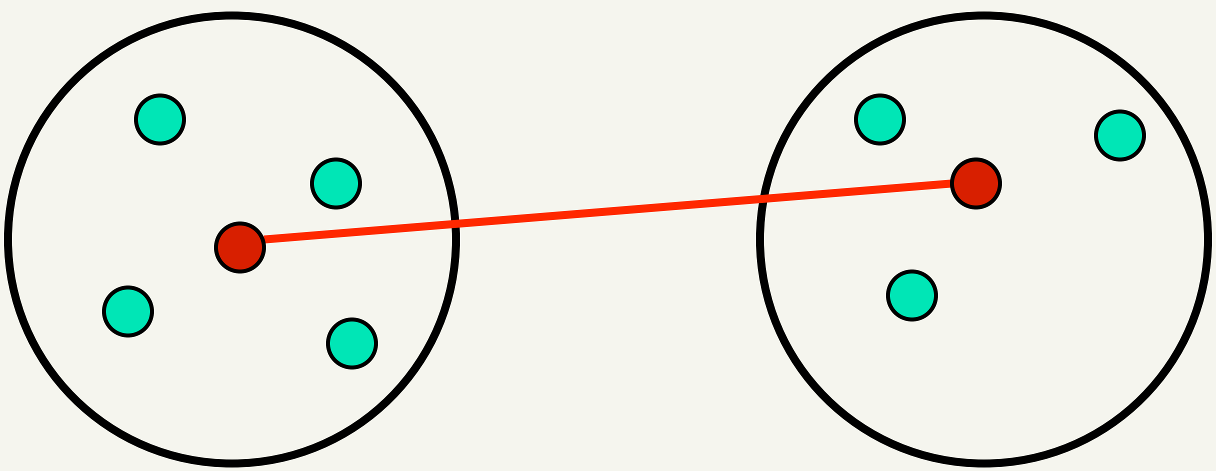
$$\frac{|L| \cdot |R|}{|L| + |R|} \|\mu(L) - \mu(R)\|^2$$

Ward's method

What does this do?

Distance between clusters

- **Centroid**
  - Clusters whose centroids (centers of gravity) are the most similar



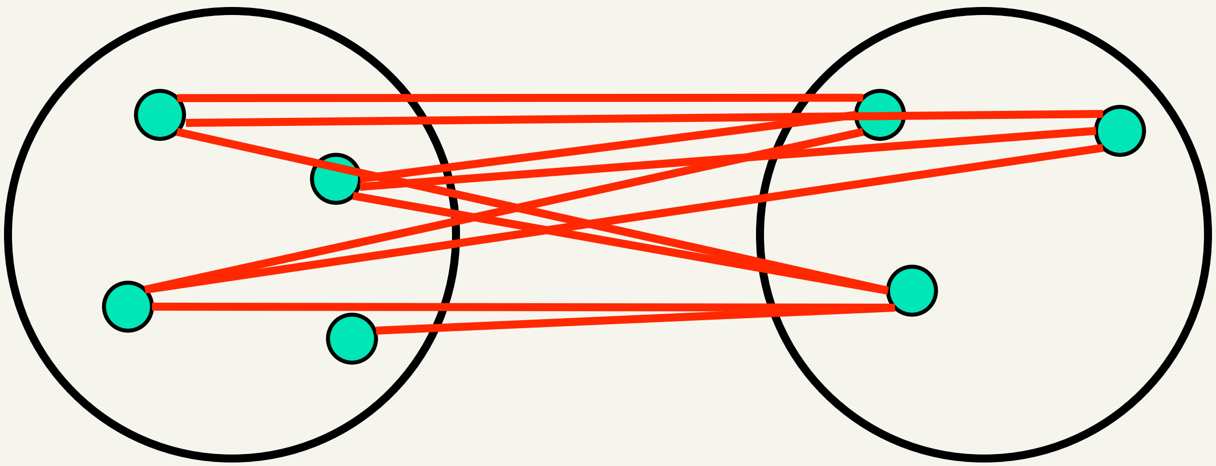
$$\frac{|L| \cdot |R|}{|L| + |R|} \| \mu(L) - \mu(R) \|^2$$

Ward's method

Encourages similar sized clusters

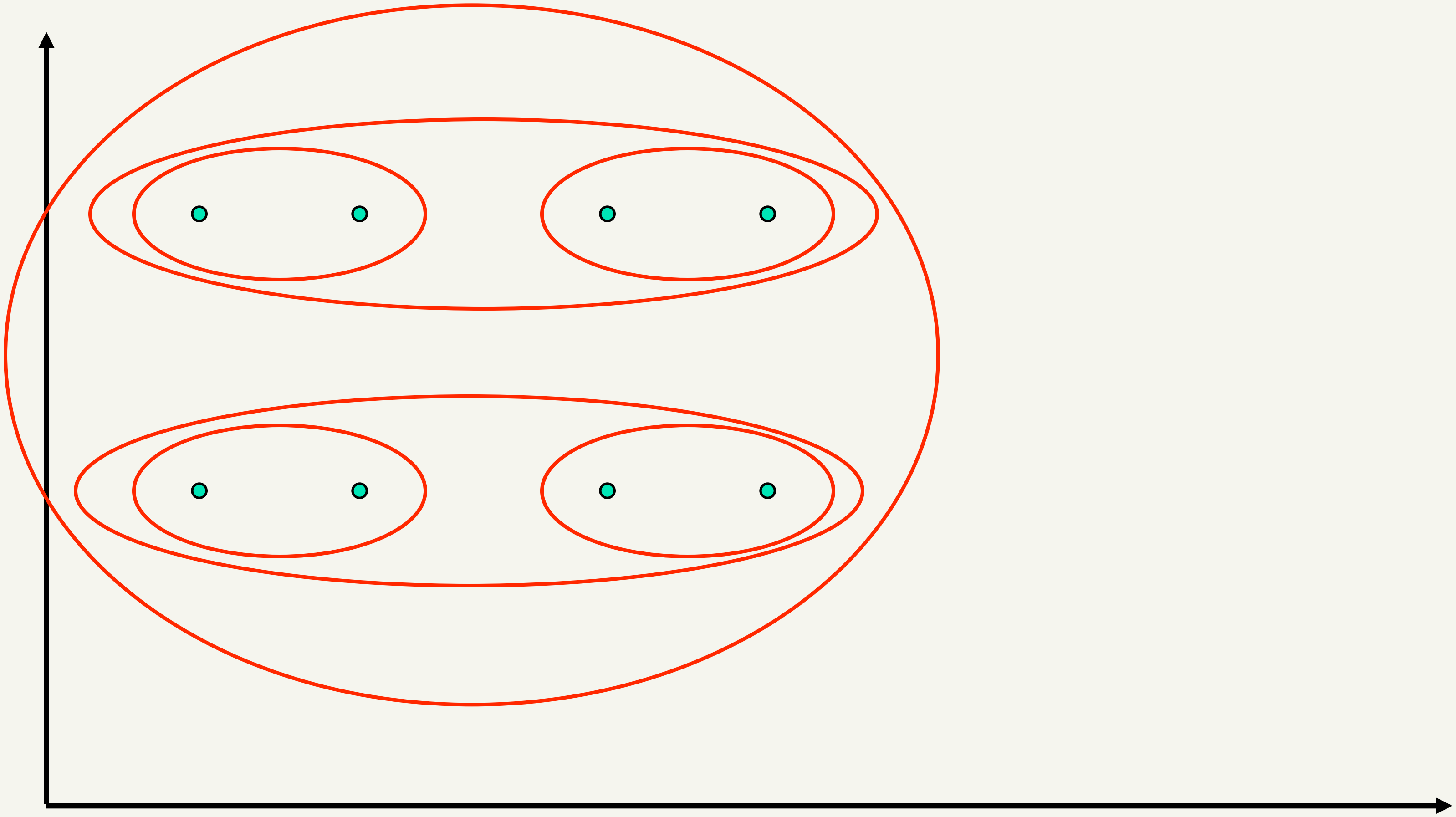
Distance between clusters

- **Average-link**
  - Average similarity between all pairs of elements

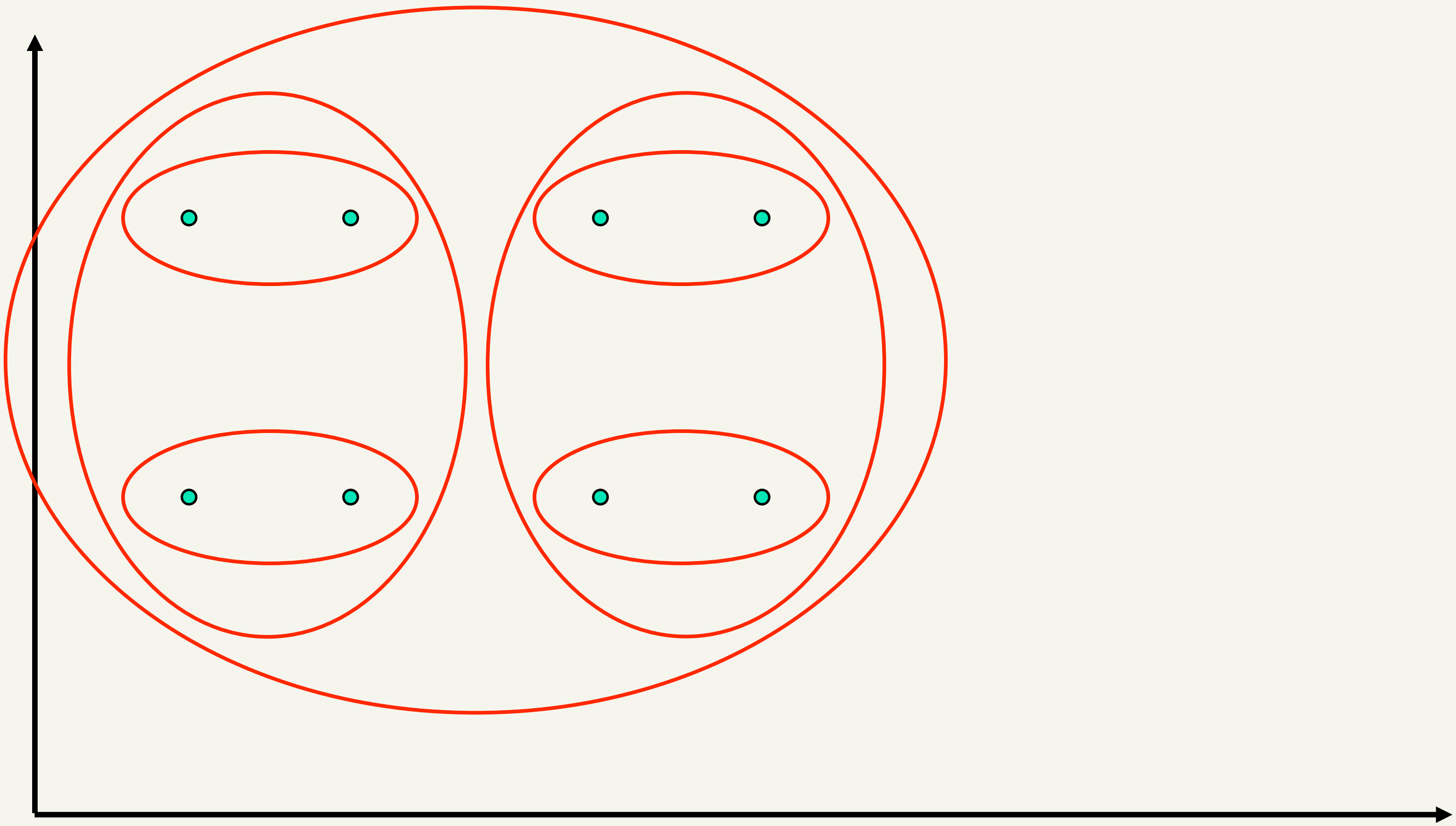


$$\frac{1}{|L| \cdot |R|} \sum_{x \in L,\, y \in R} \|x - y\|^2$$

Single Link Example

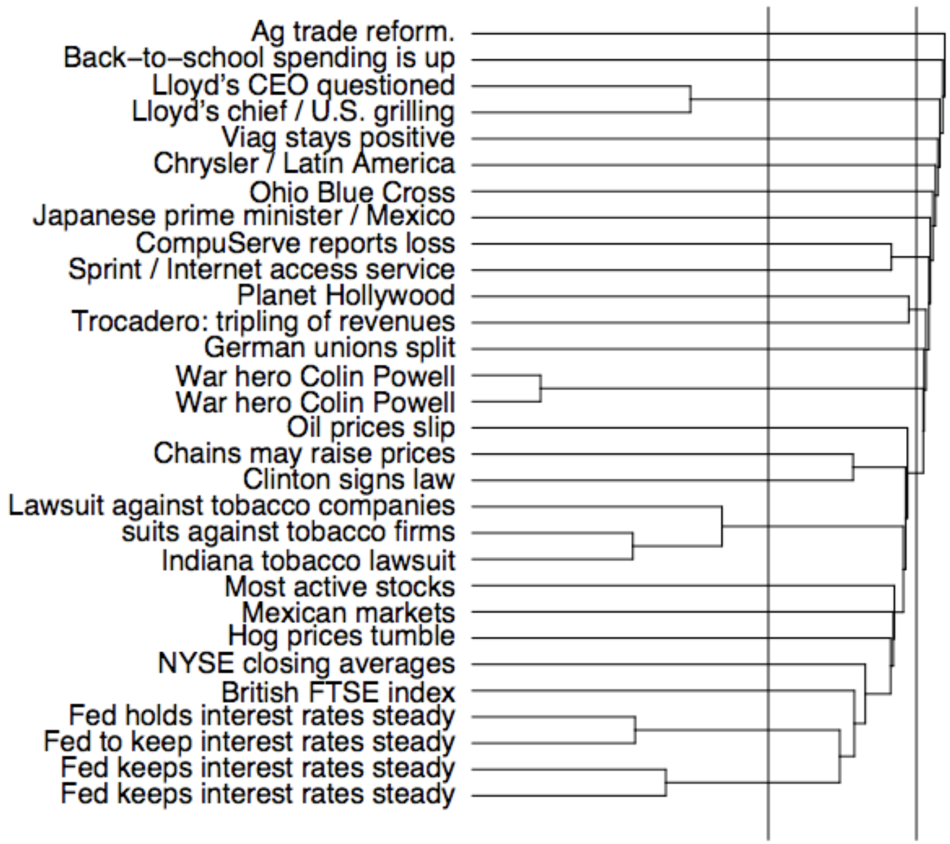


Complete Link Example



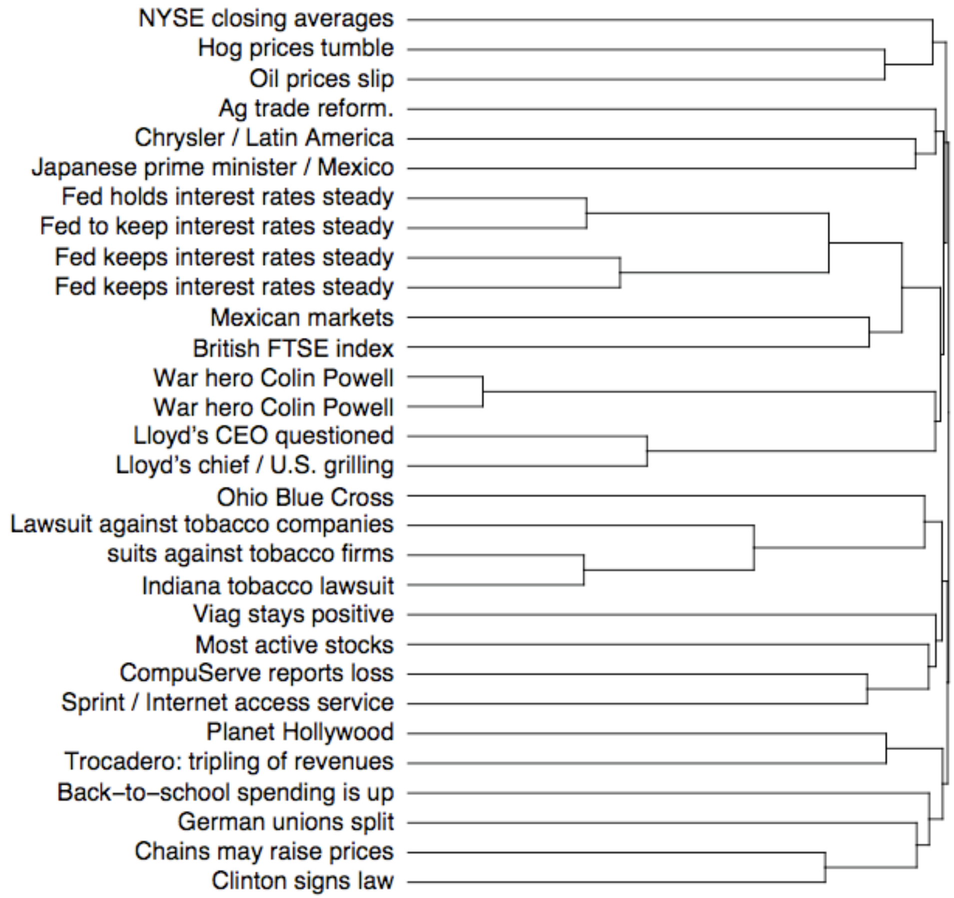
Computational Complexity

- For
  - m dimensions
  - n documents/points
- How many iterations?
  - $n-1$ iterations
- First iteration
  - Need to compute similarity of all pairs of n points/documents: $O(n^2m)$
- Remaining n-2 iterations
  - compute the distance between the most recently created cluster and all other existing clusters: $O(nm)$
  - Does depend on the cluster similarity approach
- Overall run-time: $O(n^2m)$ – generally slower than flat clustering!



single linkage

complete linkage

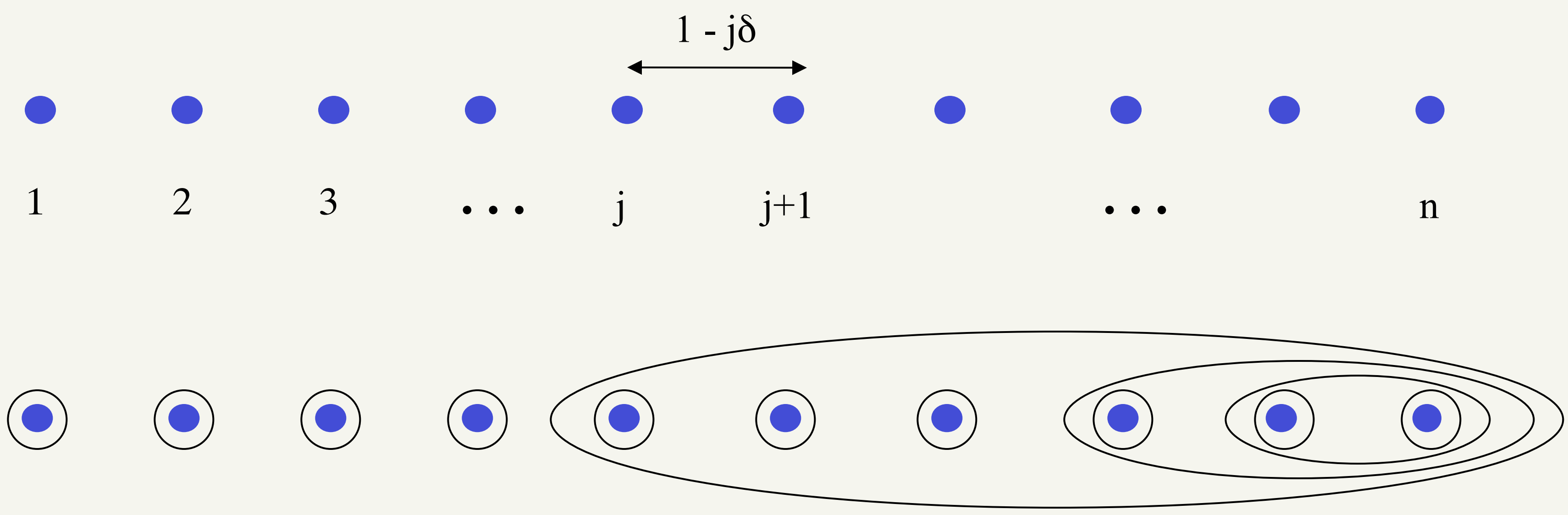


Problems with hierarchical clustering

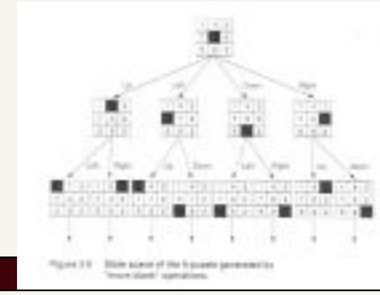
Problems with hierarchical clustering

- Locally greedy: once a merge decision has been made it cannot be changed

Single-linkage: chaining effect



State space search approach



- View hierarchical clustering problem as a state space search problem
- Each hierarchical clustering represents a state
- Goal is to find a state that minimizes some criterion function
- Avoids problem of traditional greedy methods

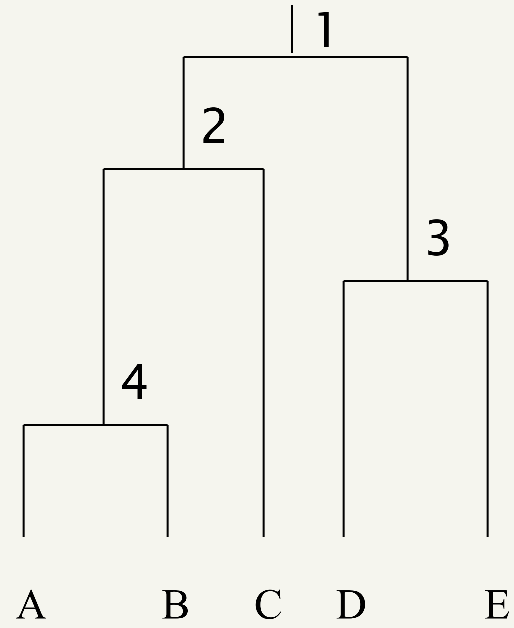
Basic state space search algorithm

- Start at some initial state
- Repeat
  - List all next states
  - Evaluate all next states using some criterion function
  - Pick choice based on some search method/criterion

State space search components

- State space
  - What is a state?
  - How to move between states?
- Search
  - State criterion function
  - Method to choose next state

State space

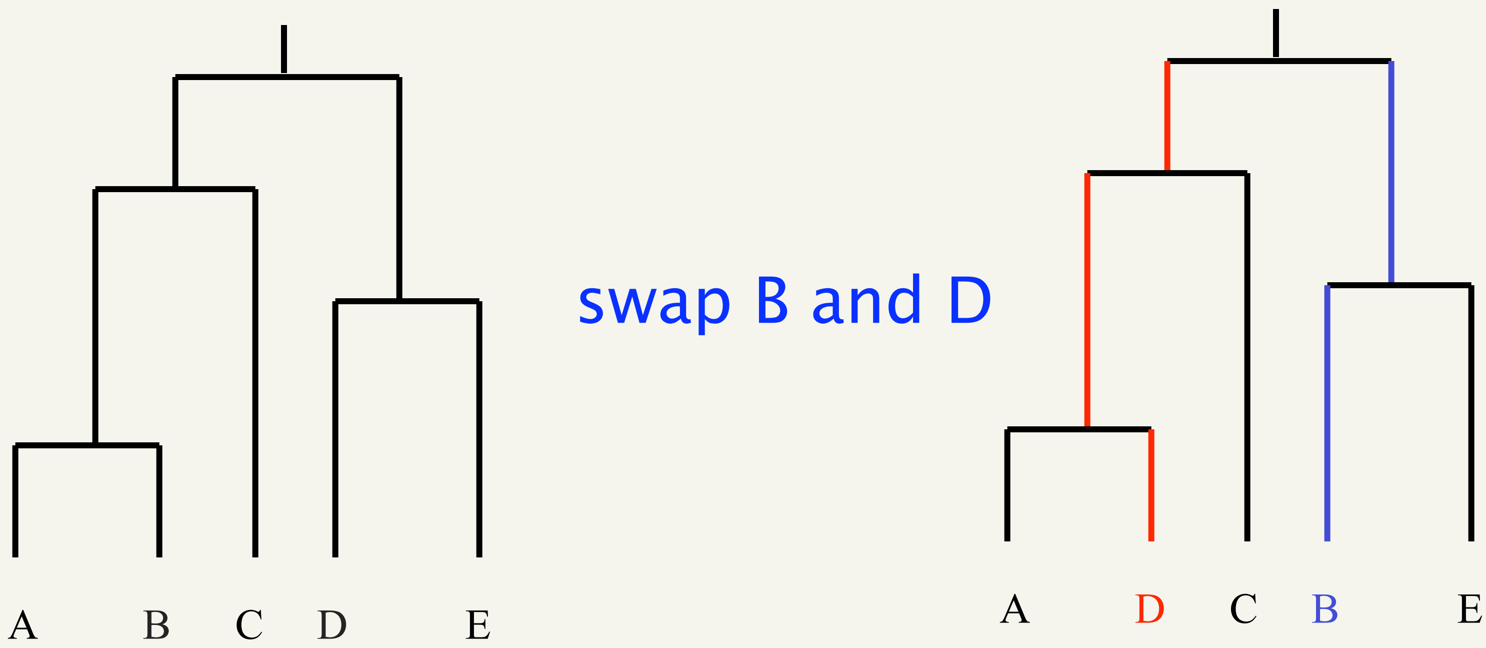


- Each state is a hierarchical clustering
- *n* points
- *n* –1 sub-clusters labeled with temporal component (i.e. split order or inverse merge order)
- Huge state space!

Moving between states

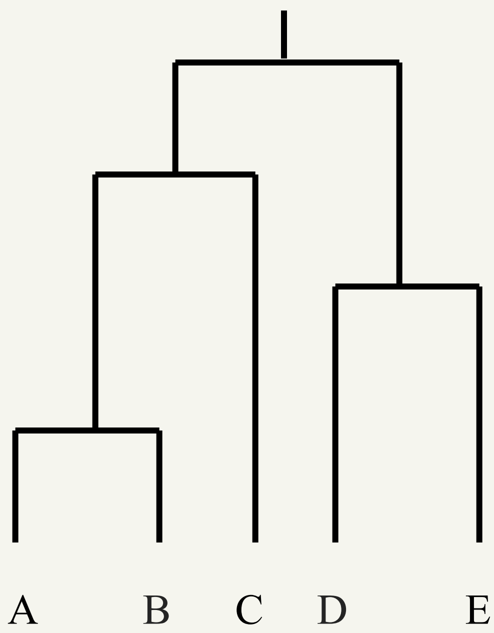
- Move should be:
  - Simple/Straightforward
  - Well motivated
  - Traverse entire state space (state space complete)
- Ideas?
- 2 options
  - node swap
  - node graft
- Also include a temporal swap

Swap without temporal constraints, example 1

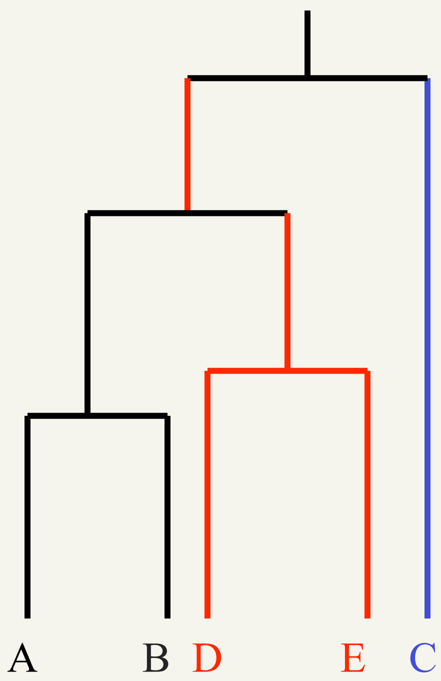


no change to the structure

Swap without temporal constraints, example 2



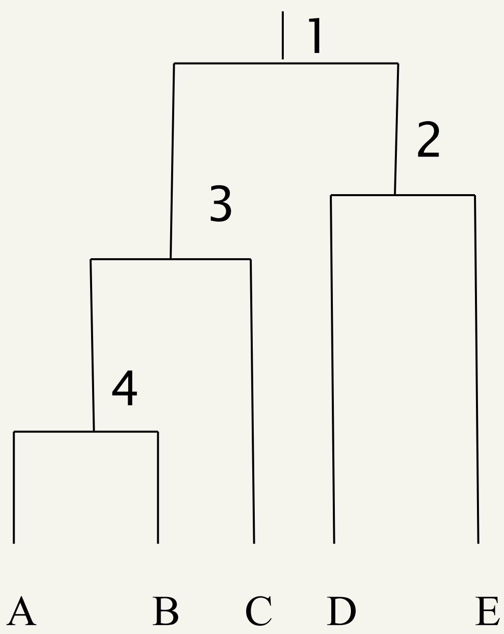
swap (D,E) and C



structure changed!

Swap with temporal constraints

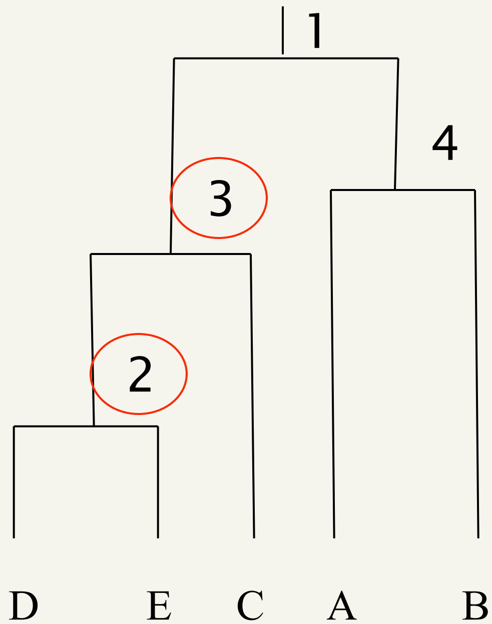
- Move split numbers with sub-clusters (nodes)
- Some swap moves don't result in legal hierarchies



What would be an illegal swap?

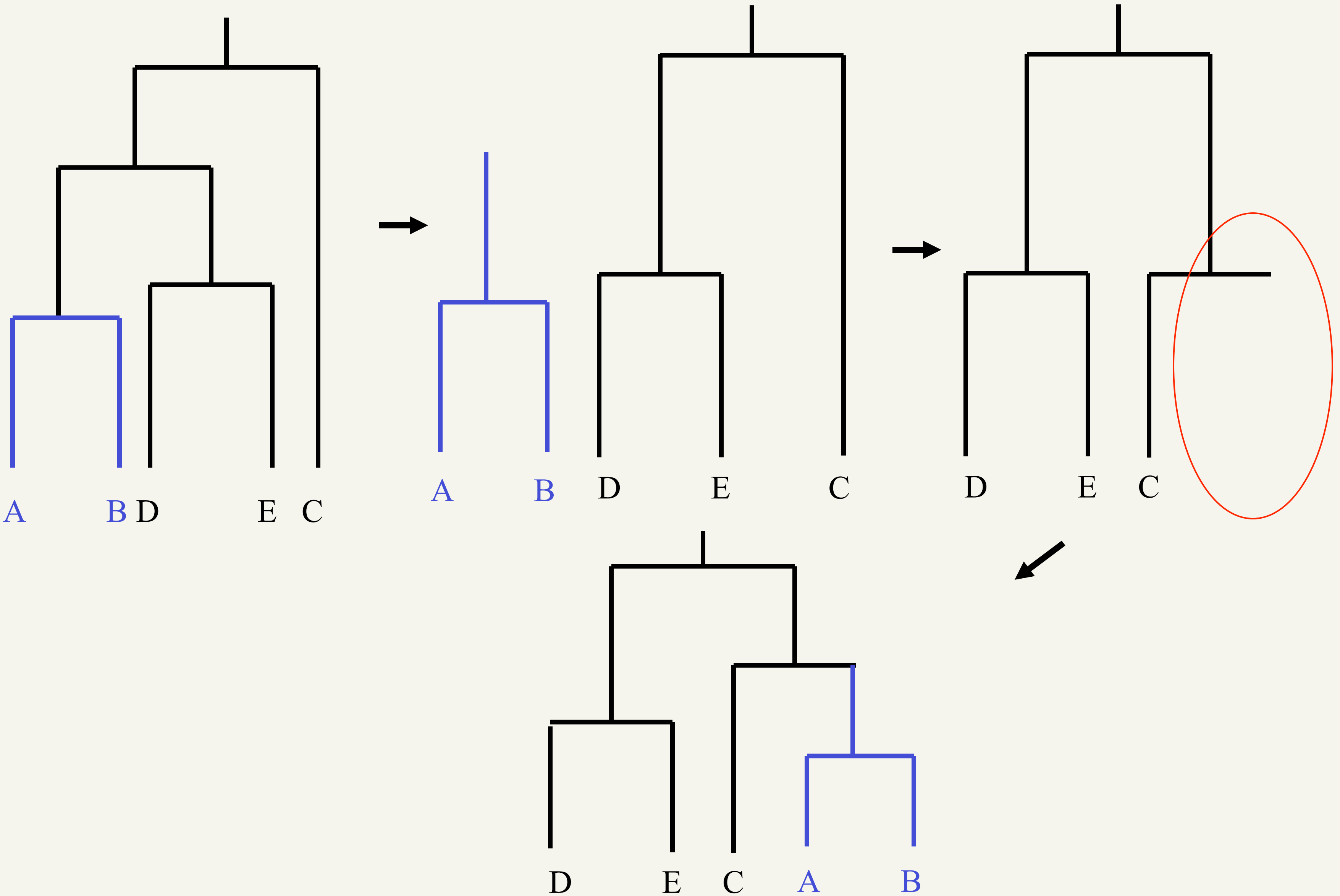
Swap with temporal constraints

- Move split numbers with sub-clusters (nodes)
- Some swap moves don't result in legal hierarchies
- The split number of the parent must be less than the split number of the child



cannot swap 2 and 4

Graft without temporal constraints

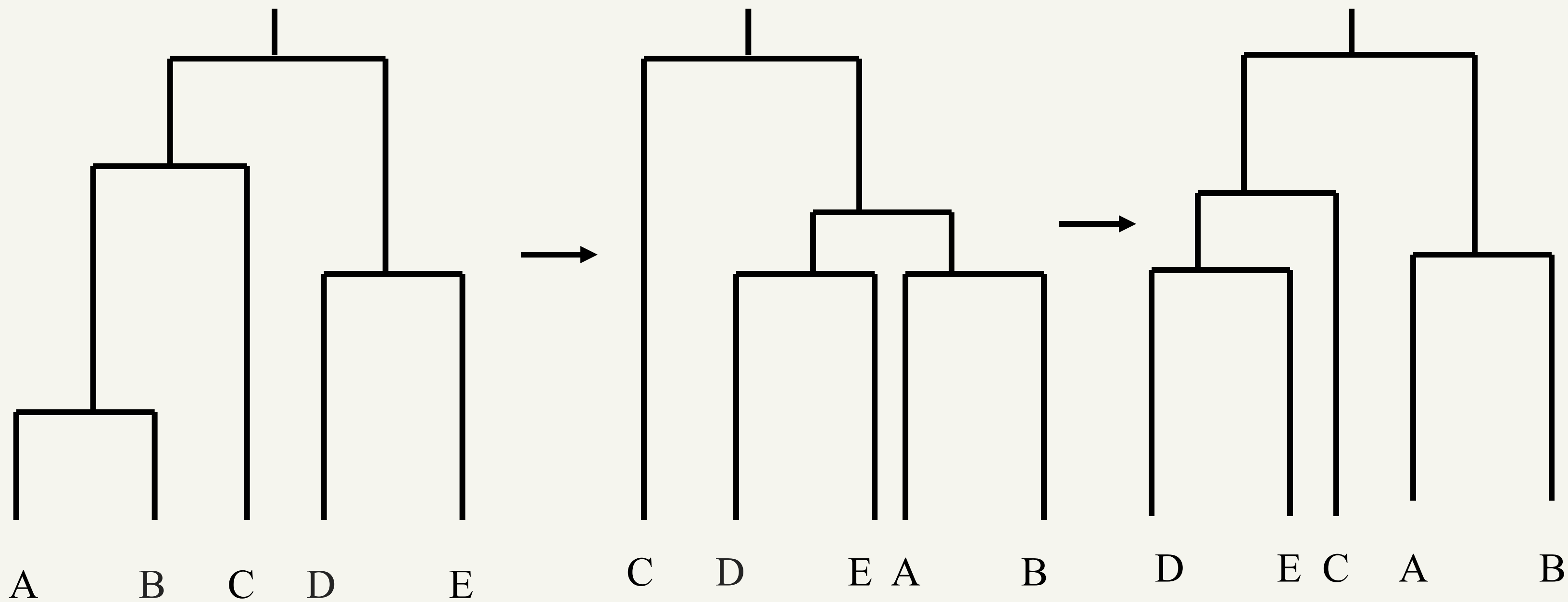


Graft with temporal constraints

- Move split number with sub-cluster
- Same as swap, only allow swaps that satisfy parent < child

Swap using grafts

Emulate:
swap (A,B) and (D,E)

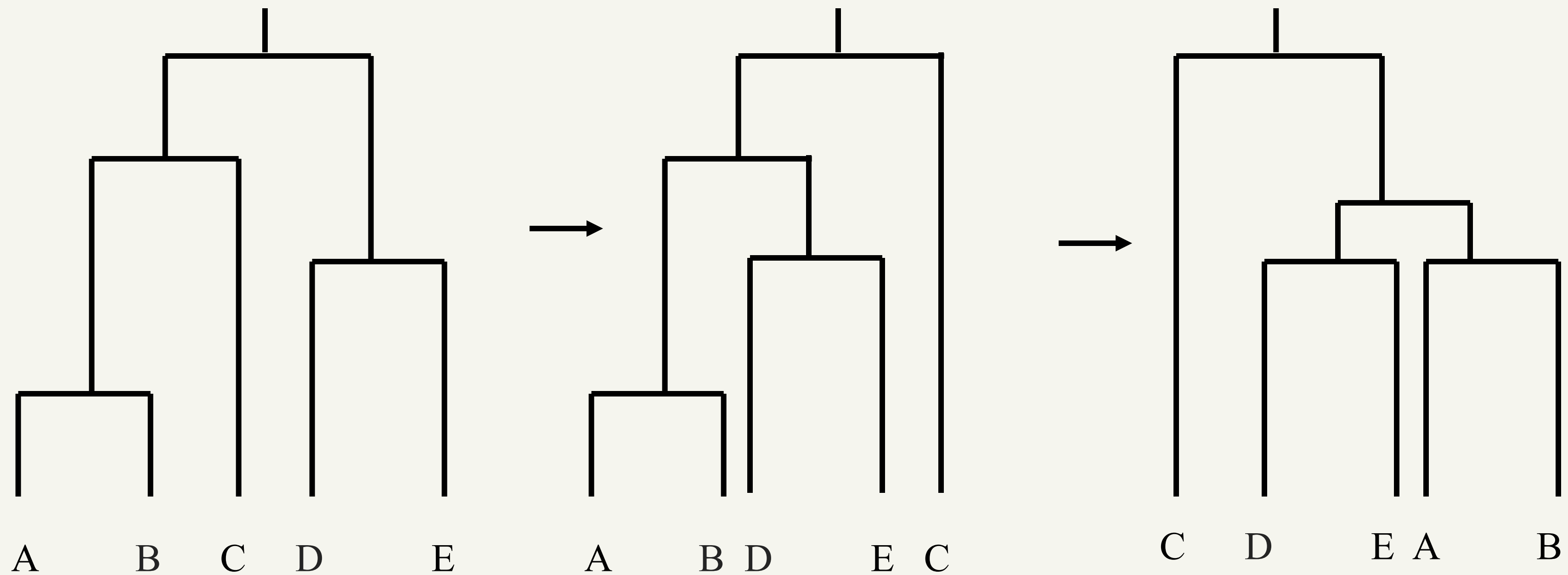


graft (A,B)

graft (D,E)

Graft using swaps

Emulate: graft (A,B) to above (D,E)

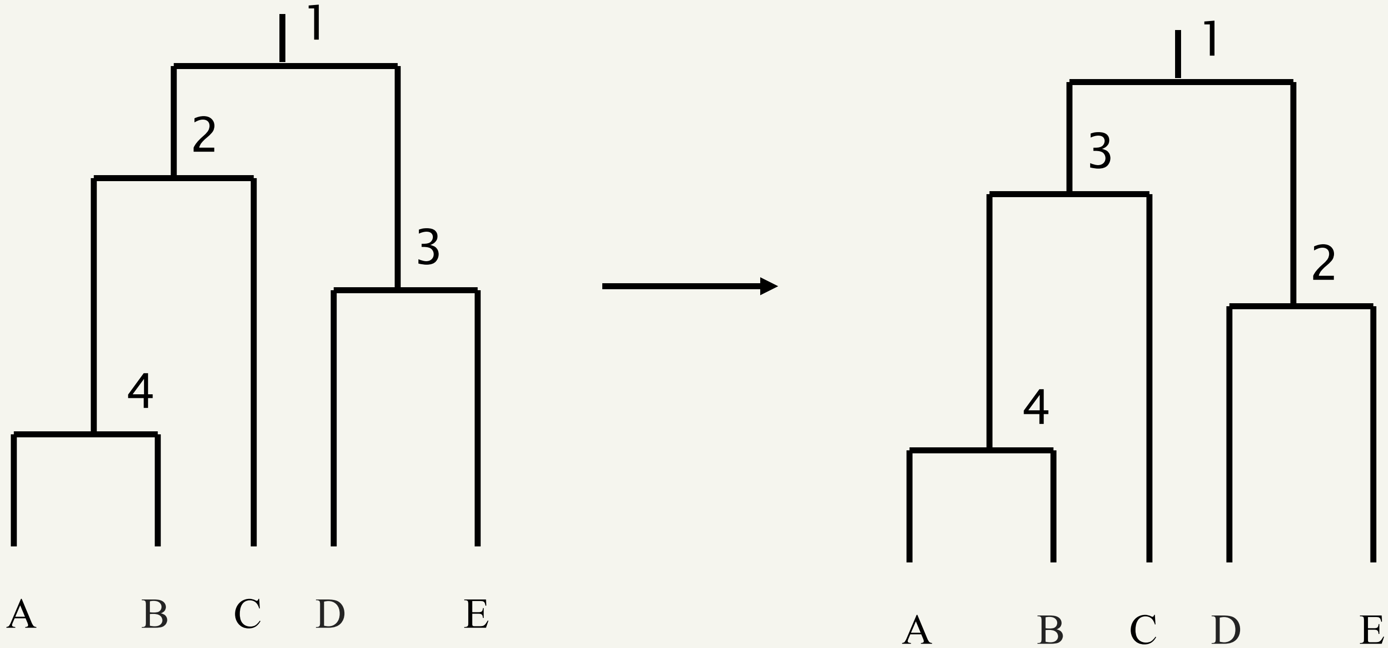


swap sibling of one with other: swap C with (D,E)

swap C and (A,B,D,E)

Temporal swap

- Must obey parent/child constraints
- In general, could swap any two that satisfy constraints
- Only consider adjacent numbers (i.e. 2, 3 or 3, 4)



Evaluating states

- For a given *k*-clustering, the k-means criterion function is the squared difference between a point and it's assigned center for all points and all centers

$$\text{cost}(C_k) = \sum_{j=1}^{k} \sum_{x \in S_j} \left\| x - \mu(S_j) \right\|^2$$

Leveraging k-means criterion

- For a hierarchical clustering, calculate a weighted sum of the k-means criterion function for all *n* -1 clusterings represented by the hierarchy

$$\text{hcost} = \sum_{i=1}^{n} w_k \, \text{cost}(C_k)$$

Calculating criterion function

- How long does it take to calculate k-means cost?
  - O(*nm*)

$$\text{cost}(C_k) = \sum_{j=1}^{k} \sum_{x \in S_j} \left\| x - \mu(S_j) \right\|^2$$

- How long then for the overall cost?
  - *n* –1 clusterings: O(n^2m)

$$\text{hcost} = \sum_{i=1}^{n} w_k \, \text{cost}(C_k)$$

- We can do better!
  - Using a few tricks… O(nm) time

How to pick the next state

- Greedy: Pick best choice
- ε-greedy: Pick best choice with probability ε, otherwise choose randomly
- Soft-max: Choose *option* with probability

$$p(option) = \frac{e^{\text{hcost}(option)/\tau}}{\sum_{\text{all options}} e^{\text{hcost}(option_i)/\tau}}$$

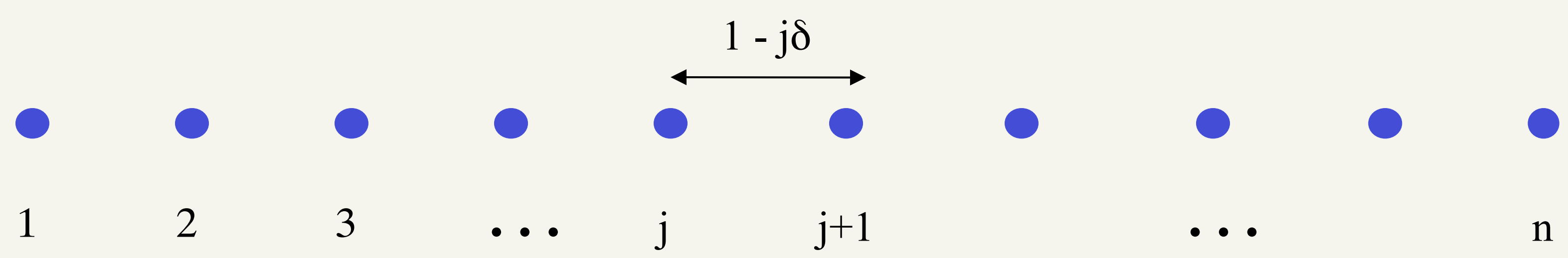
- Simulated annealing: Vary parameters for above algorithms over time

Overall run-time ☹

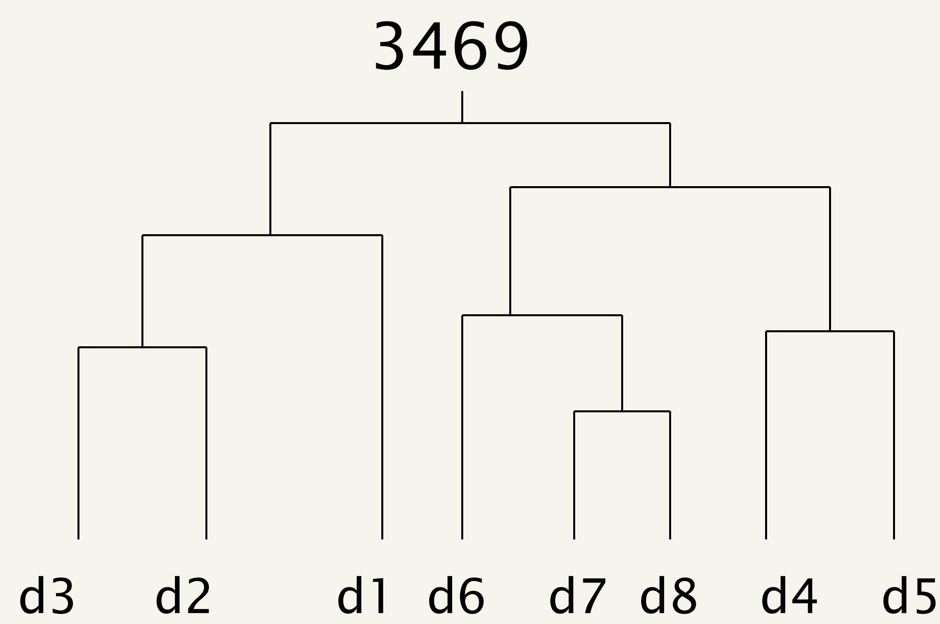
- List all next states
  - How many next states are there?
    - All combinations of n data points and $n - 1$ sub-clusters
    - $O(n^2)$
- Evaluate all next states using criterion function
  - $O(nm)$
- Pick choice based on some search method/criterion

$O(n^3)$ per iteration

Bad case for single linkage



- Examined n = 8
- Greedy method
- Using simulated annealing "best" was found 3 out of 10
- Lowest criterion value is "best" clustering (3304)

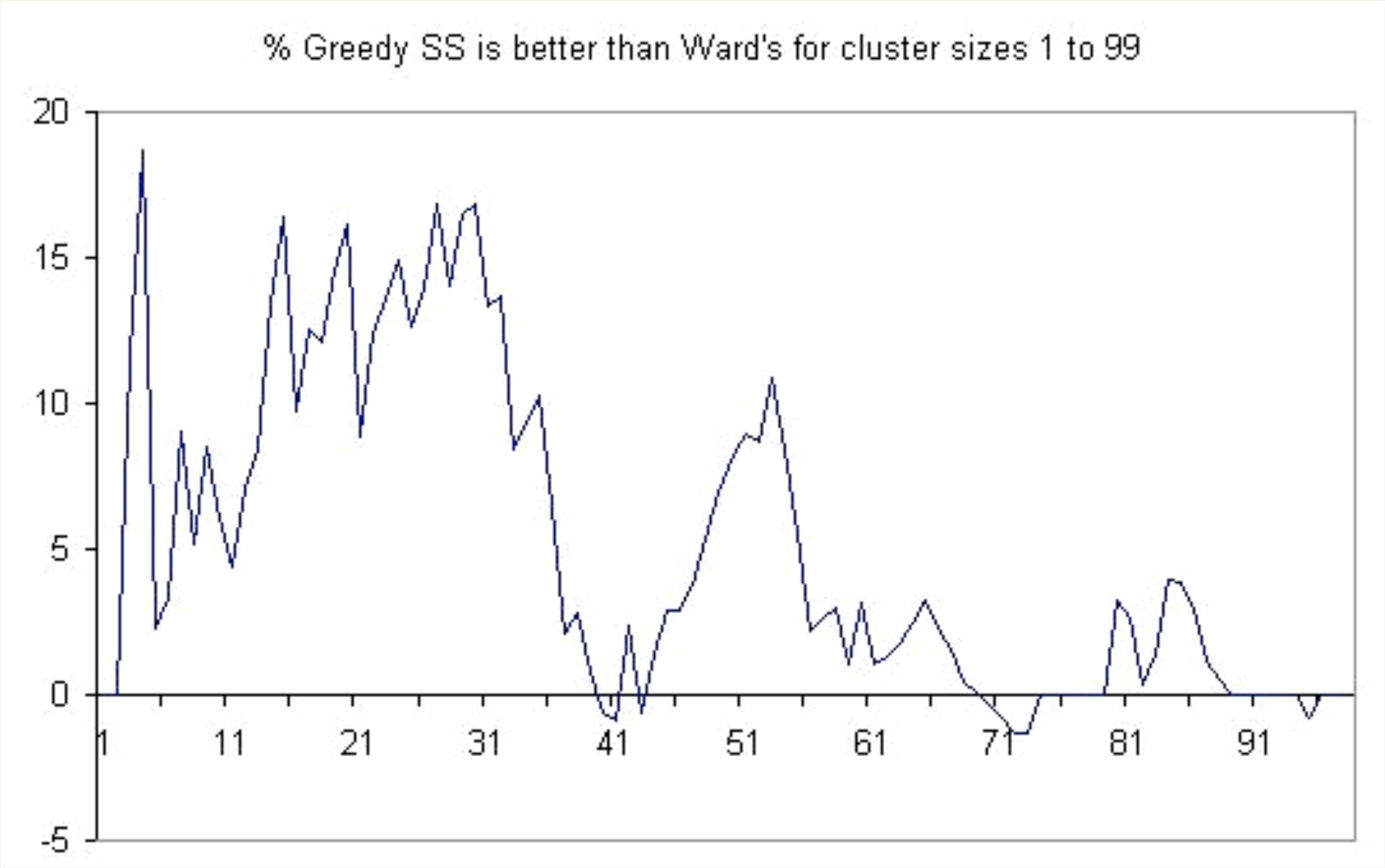


SS-Hierarchical vs. Ward's

Yeast gene expression data set

	SS-Hierarchical Greedy, Ward's initialize	Ward's
20 points	21.59 8 iterations	21.99
100 points	411.83 233 iterations	444.15
500 points	5276.30 ? iterations	5570.95

SS-Hierarchical vs. Ward's: Individual clusters



What Is A Good Clustering?

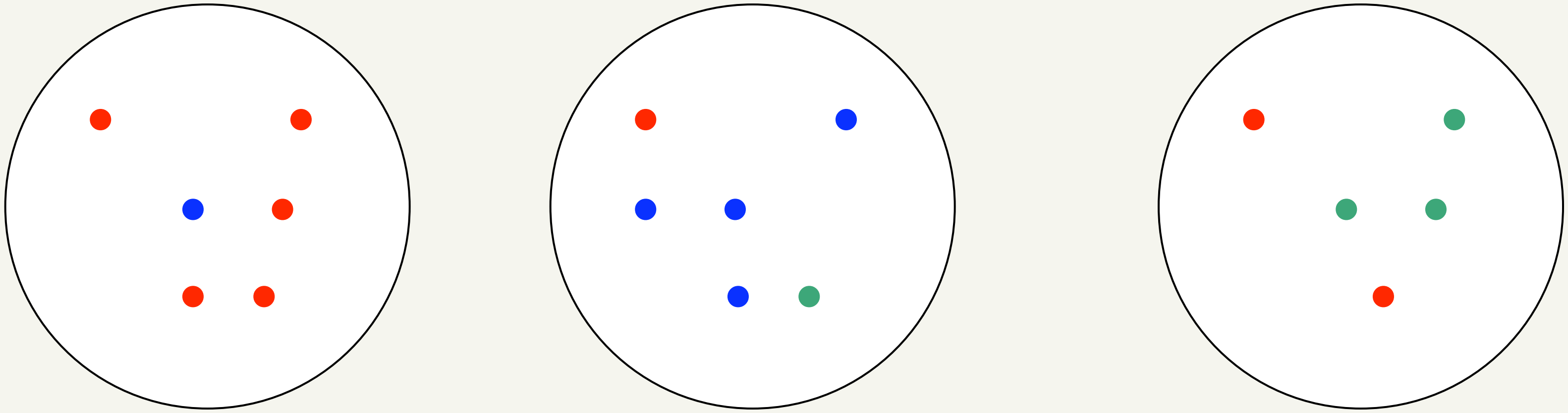
- Internal criterion: A good clustering will produce high quality clusters in which:
  - the <u>intra-class</u> (that is, intra-cluster) similarity is high
  - the <u>inter-class</u> similarity is low

How would you evaluate clustering?

Common approach: use labeled data

- Use data with known classes
  - For example, document classification data
- Measure how well the clustering algorithm reproduces class partitions
- **Purity**, the proportion of the dominant class in the cluster
  - Good for comparing two algorithms, but not understanding how well a single algorithm is doing, why?
    - Increasing the number of clusters increases purity
- **Average entropy** of classes in clusters
  - for example, prefers 50/50 vs. 50/25/25

Purity example



Cluster I Cluster II Cluster III

Cluster I: Purity = 1/6 (max(5, 1, 0)) = 5/6

Cluster II: Purity = 1/6 (max(1, 4, 1)) = 4/6

Cluster III: Purity = 1/5 (max(2, 0, 3)) = 3/5

Googlenomics

http://www.wired.com/culture/culturereviews/magazine/17-06/nep_googlenomics

- The article mentions the “quality score” as an important ingredient to the search. How is it important/useful?
- What are the drawbacks to this algorithm?