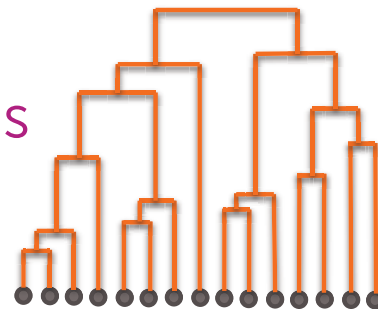




Bonus content: Hierarchical clustering

Why hierarchical clustering?

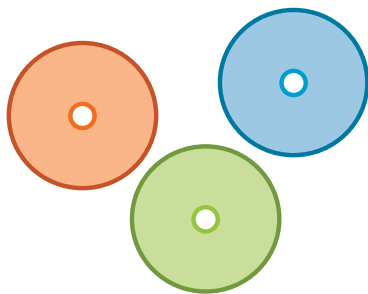
- Avoid choosing # clusters beforehand
- **Dendrograms** help visualize different clustering **granularities**
 - No need to rerun algorithm
- Most algorithms allow user to **choose any distance metric**
 - k-means restricted us to Euclidean distance



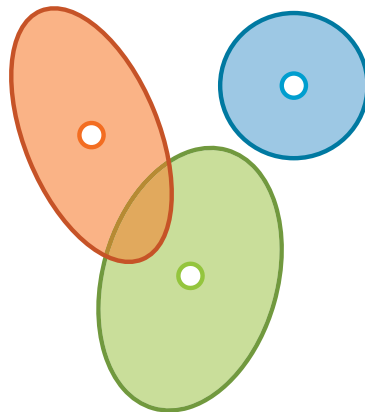
Why hierarchical clustering?

Can often find more **complex shapes** than k-means or Gaussian mixture models

k-means: spherical clusters



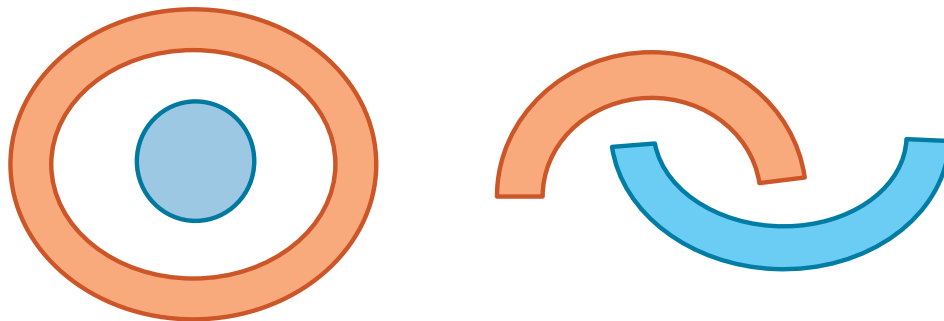
Gaussian mixtures: ellipsoids



Why hierarchical clustering?

Can often find more **complex shapes** than k-means or Gaussian mixture models

What about these?





Two main types of algorithms

Divisive, *a.k.a top-down*: Start with all data in one big cluster and recursively split.

- Example: **recursive k-means**

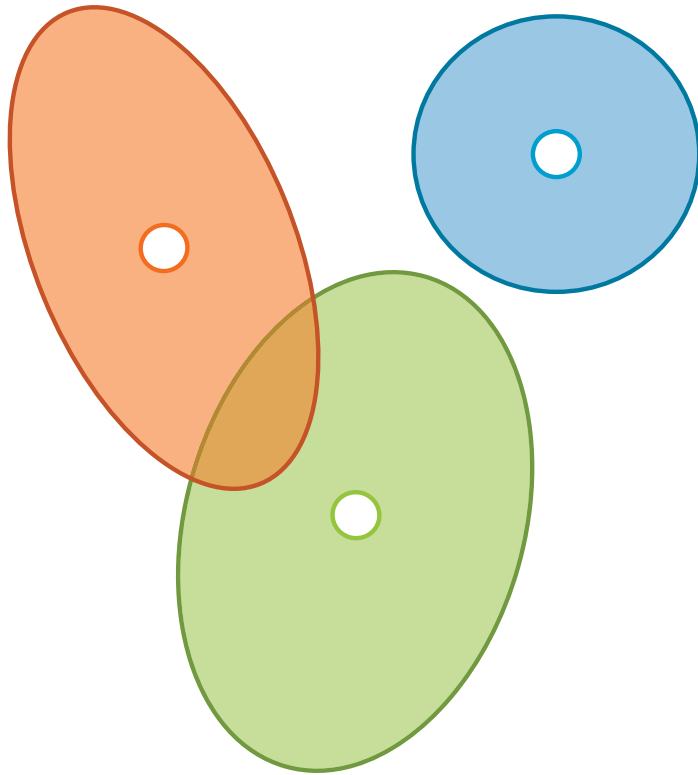
Agglomerative *a.k.a. bottom-up*: Start with each data point as its own cluster. Merge clusters until all points are in one big cluster.

- Example: **single linkage**

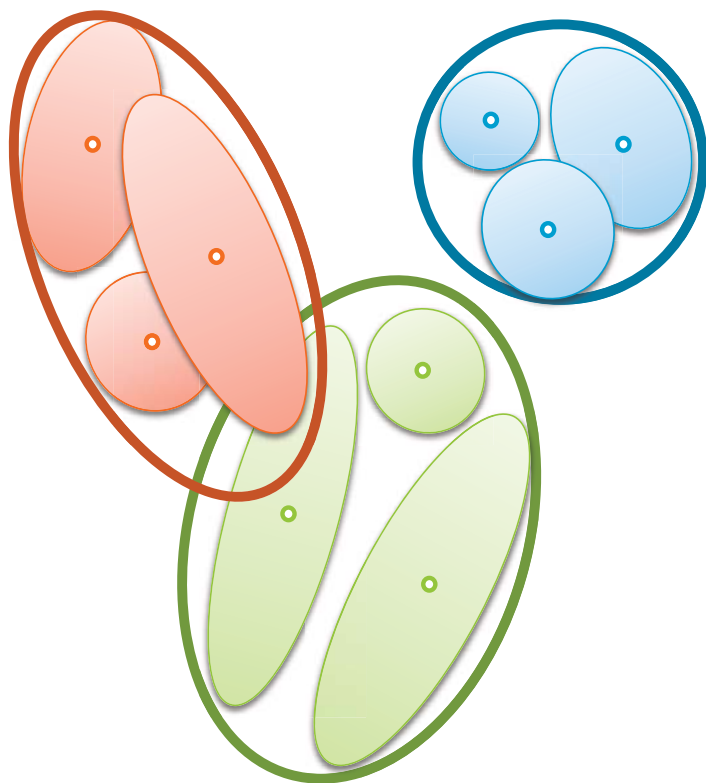


Divisive clustering

Divisive in pictures – level 1



Divisive in pictures – level 2

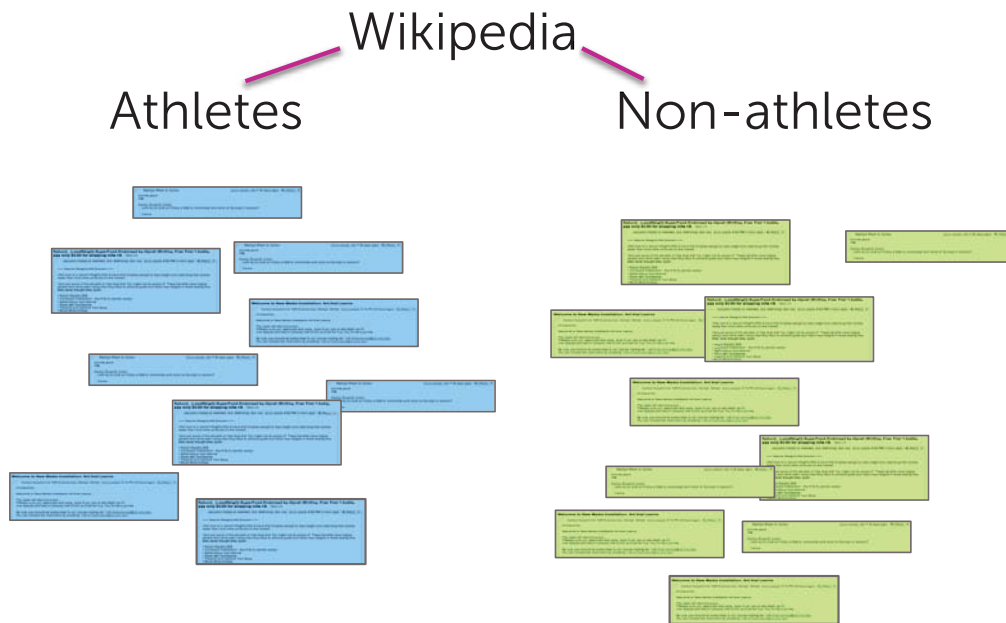


Divisive: Recursive k-means

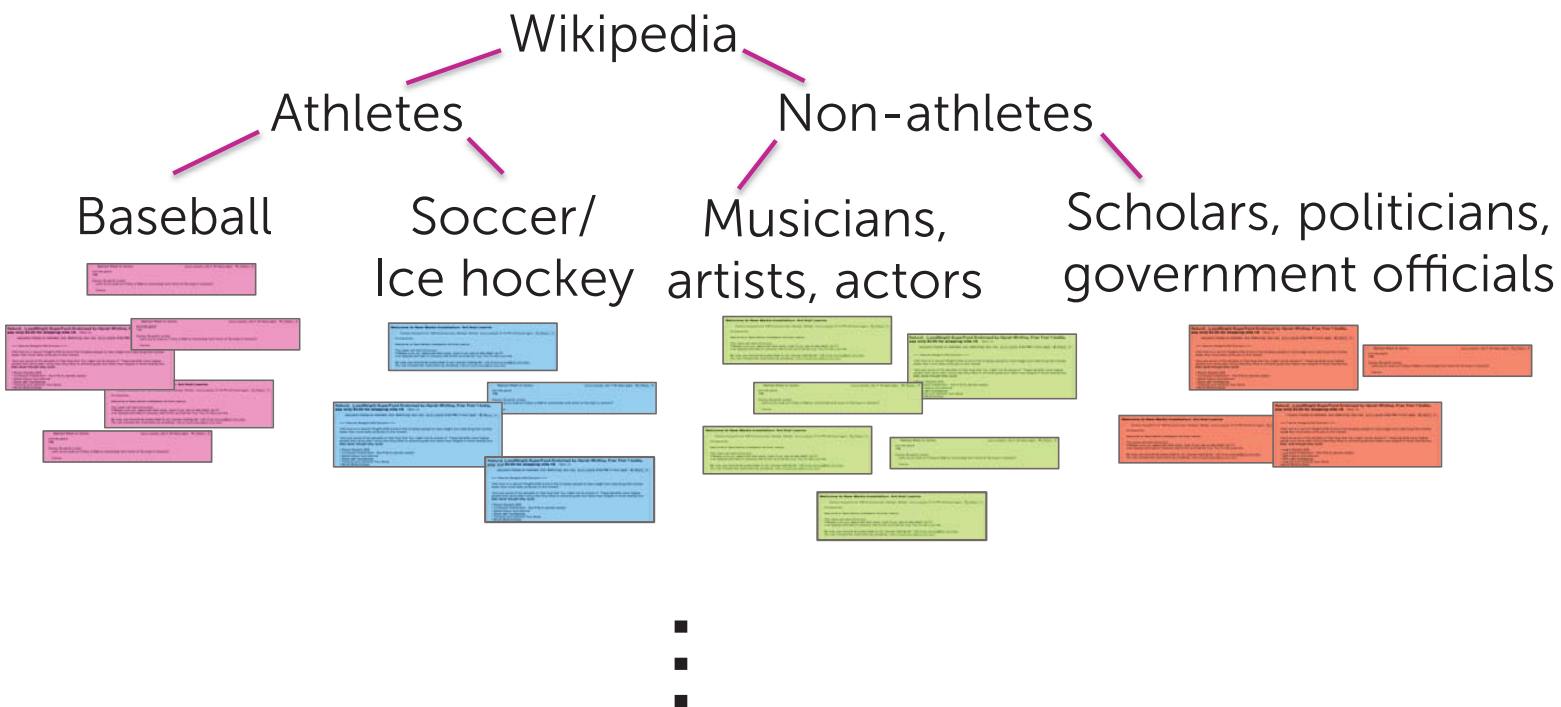
Wikipedia



Divisive: Recursive k-means



Divisive: Recursive k-means





Divisive choices to be made

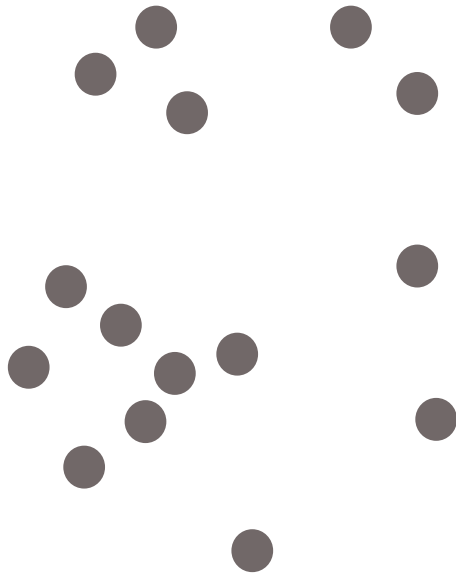
- Which algorithm to recurse
- How many clusters per split
- When to split vs. stop
 - Max cluster size:
number of points in cluster falls below threshold
 - Max cluster radius:
distance to furthest point falls below threshold
 - Specified # clusters:
split until pre-specified # clusters is reached



Agglomerative clustering

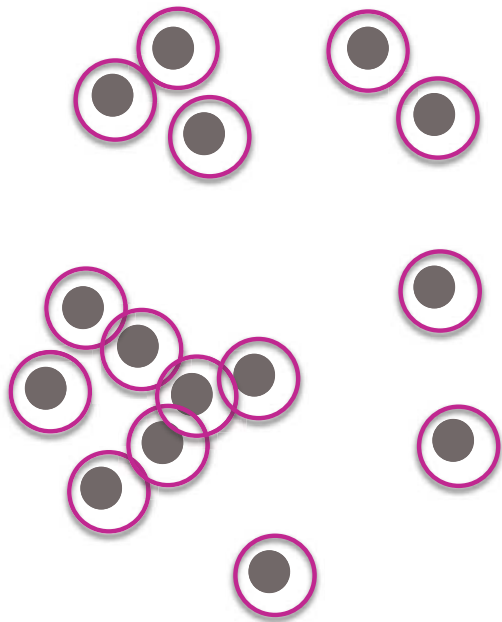
Agglomerative: Single linkage

1. Initialize each point to be its own cluster



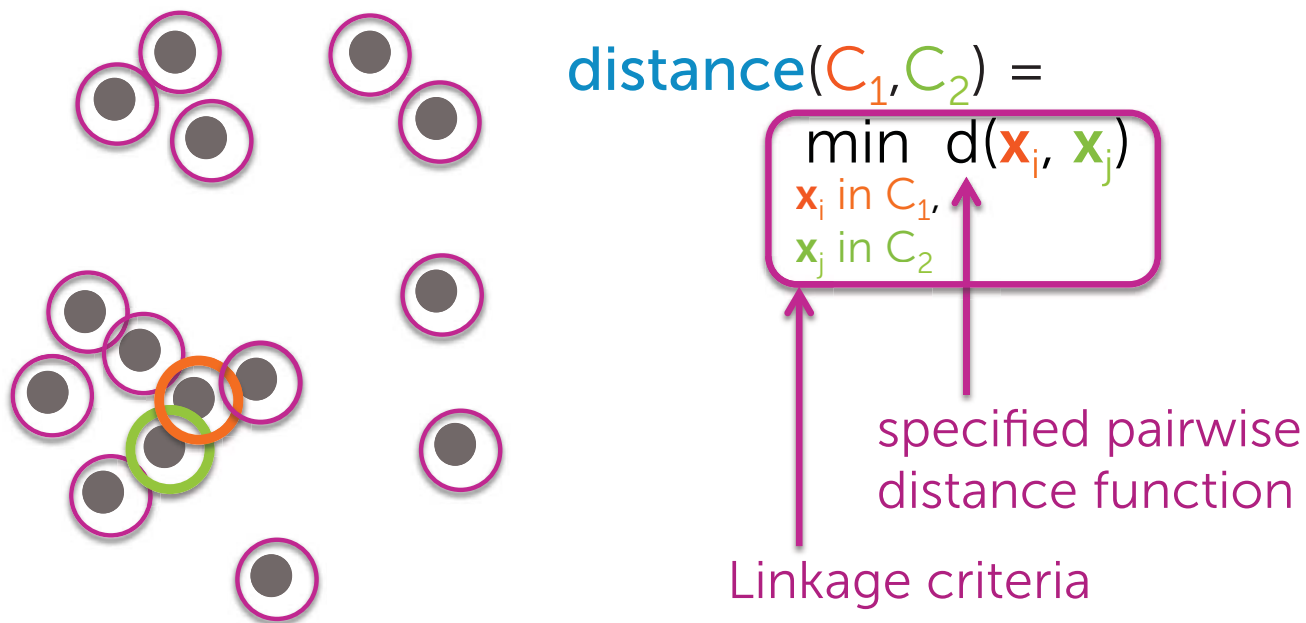
Agglomerative: Single linkage

1. Initialize each point to be its own cluster



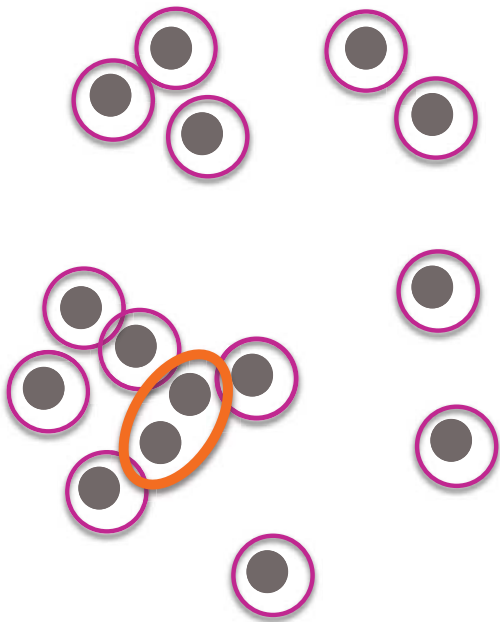
Agglomerative: Single linkage

2. Define distance between clusters to be:



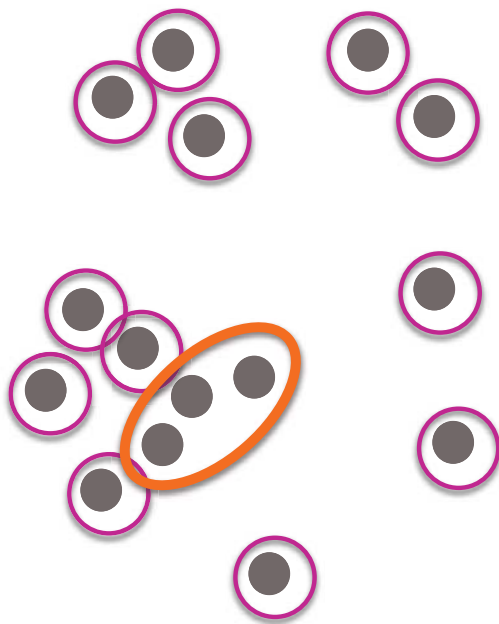
Agglomerative: Single linkage

3. Merge the two closest clusters



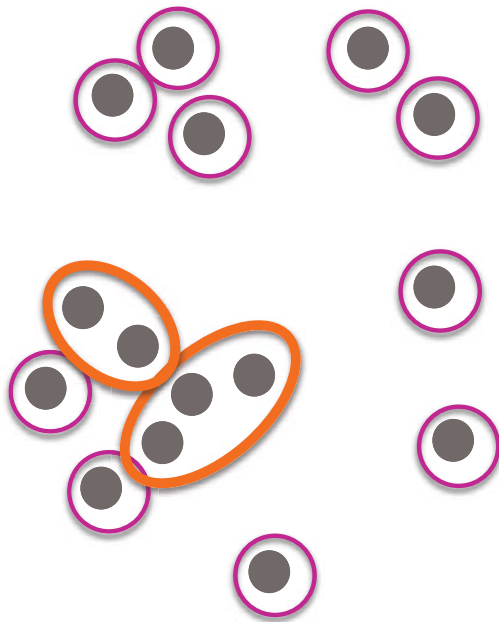
Agglomerative: Single linkage

4. Repeat step 3 until all points are in one cluster



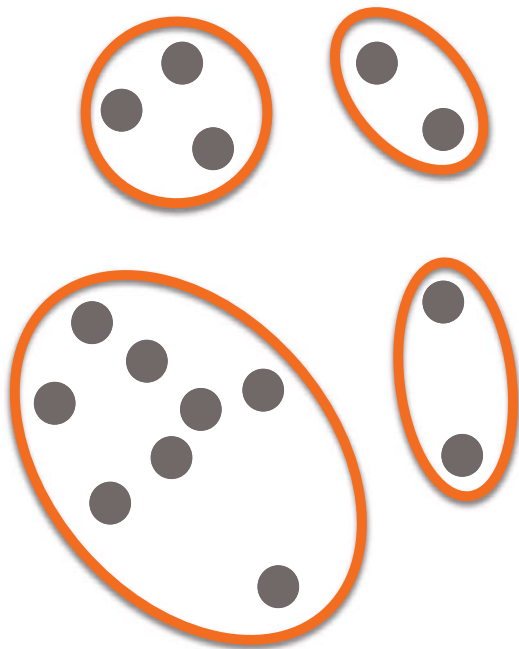
Agglomerative: Single linkage

4. Repeat step 3 until all points are in one cluster



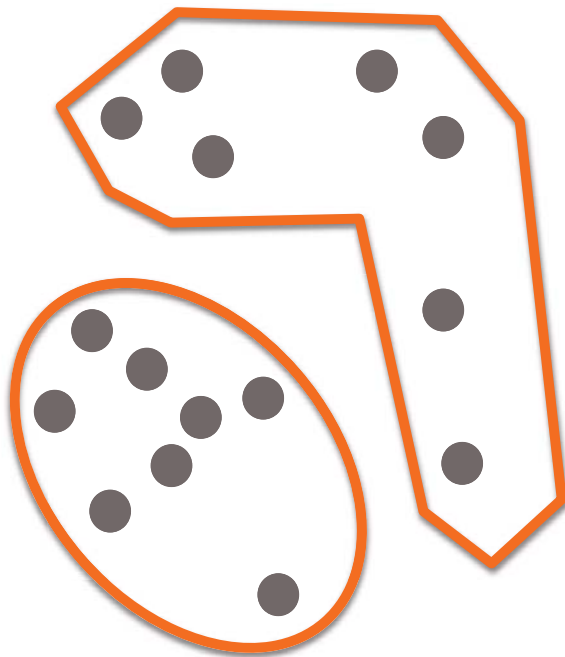
Agglomerative: Single linkage

4. Repeat step 3 until all points are in one cluster



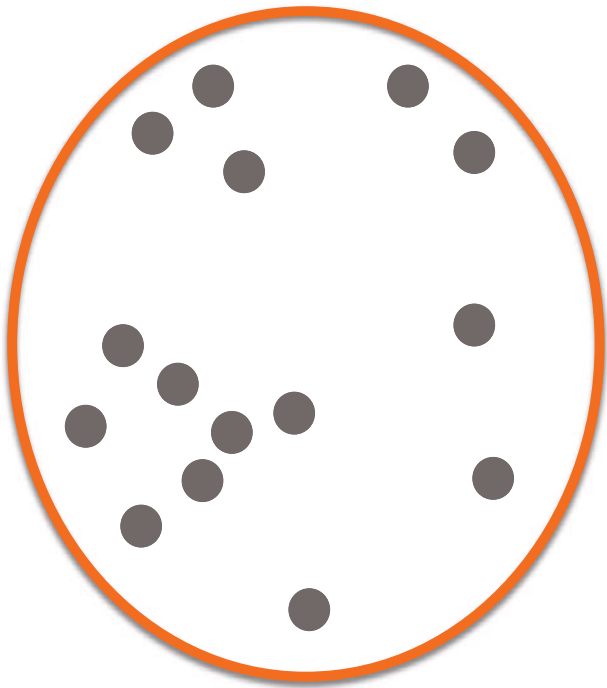
Agglomerative: Single linkage

4. Repeat step 3 until all points are in one cluster



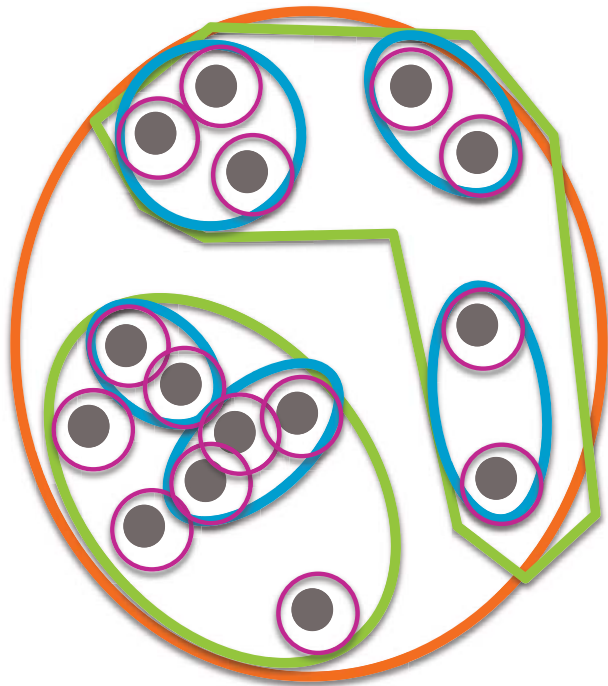
Agglomerative: Single linkage


4. Repeat step 3 until all points are in one cluster



Clusters of clusters

Just like our picture for divisive clustering...

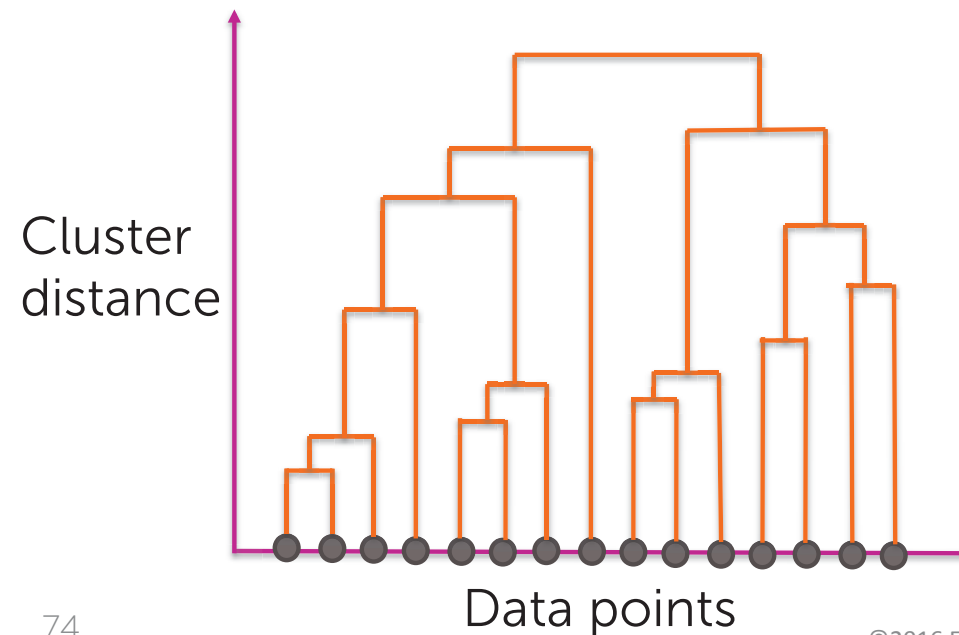




The dendrogram for agglomerative clustering

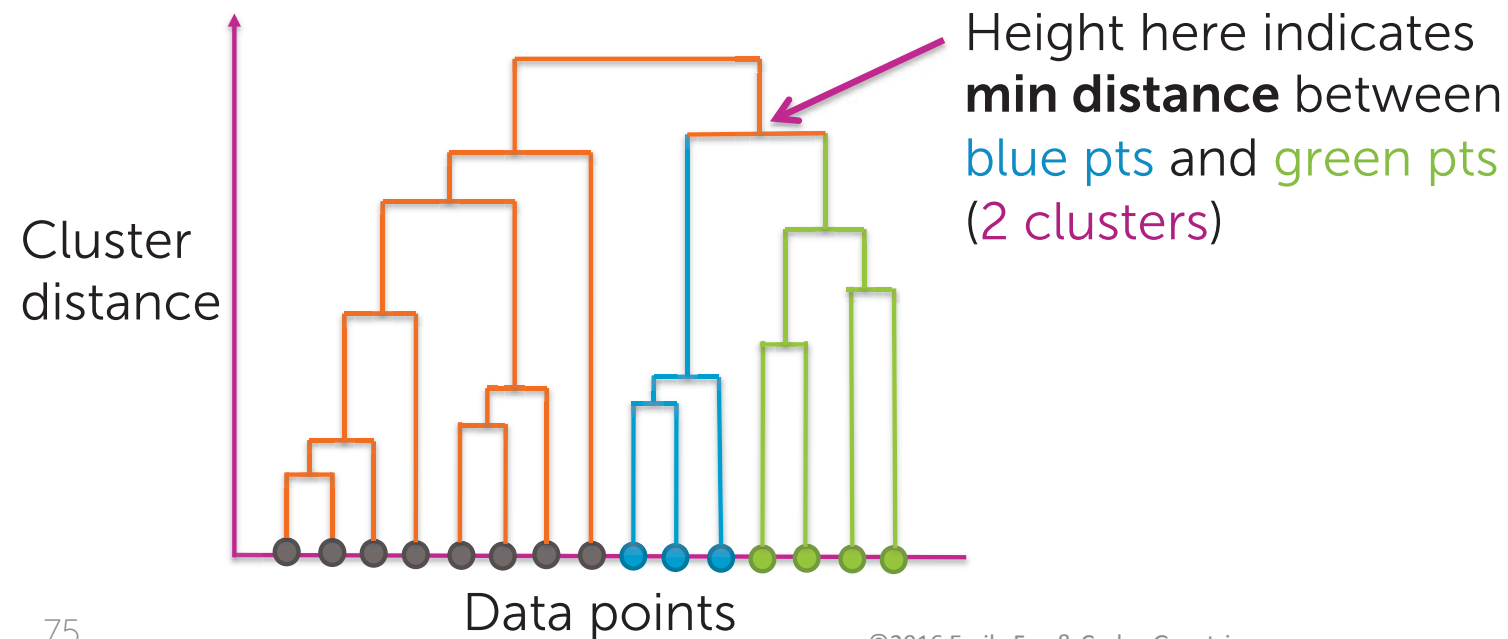
The dendrogram

- x axis shows data points (carefully ordered)
- y-axis shows distance between pair of clusters



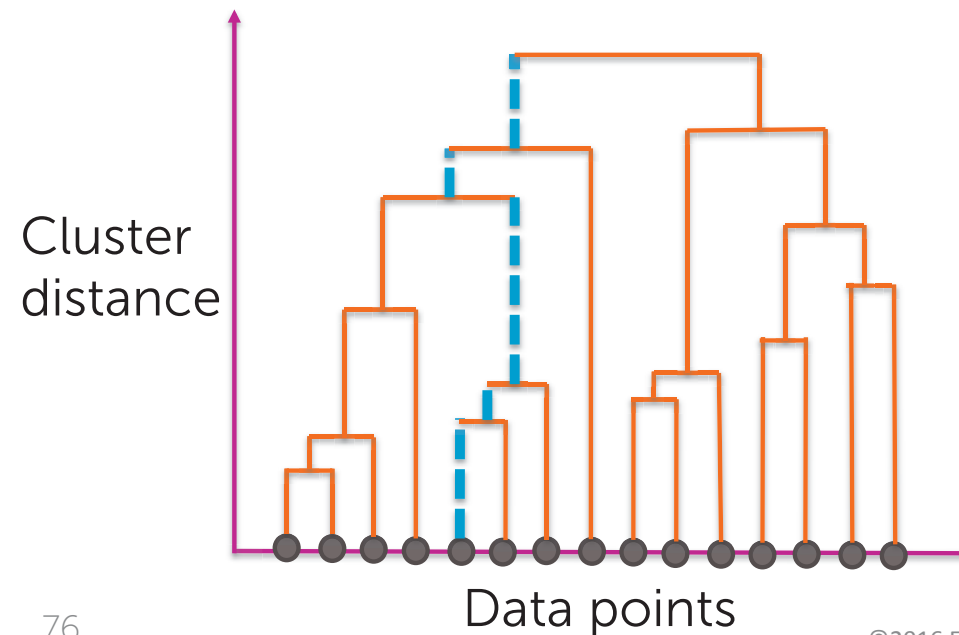
The dendrogram

- x axis shows data points (carefully ordered)
- y-axis shows distance between pair of clusters



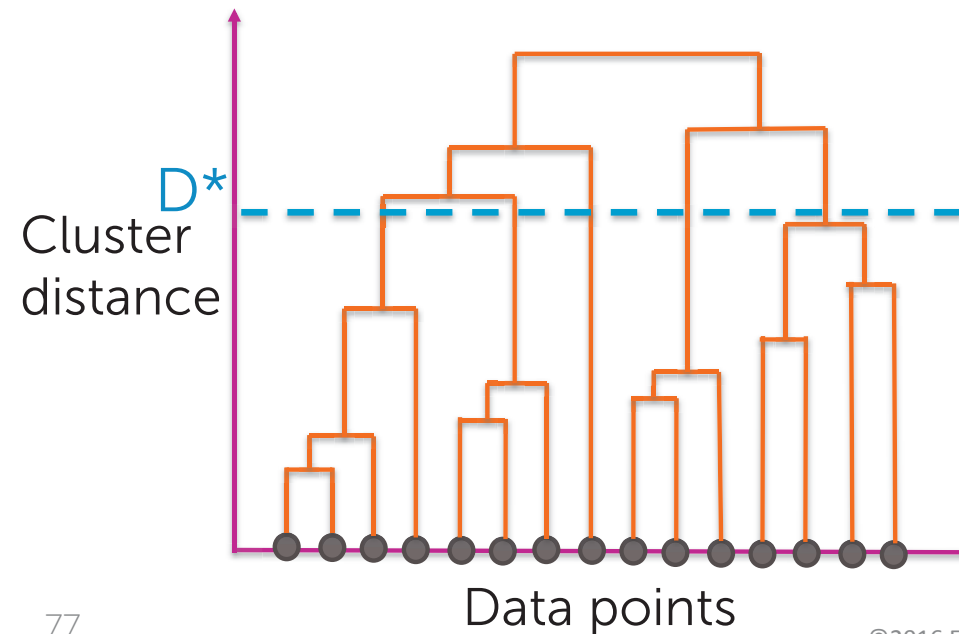
The dendrogram

Path shows all clusters to which a point belongs and the order in which clusters merge



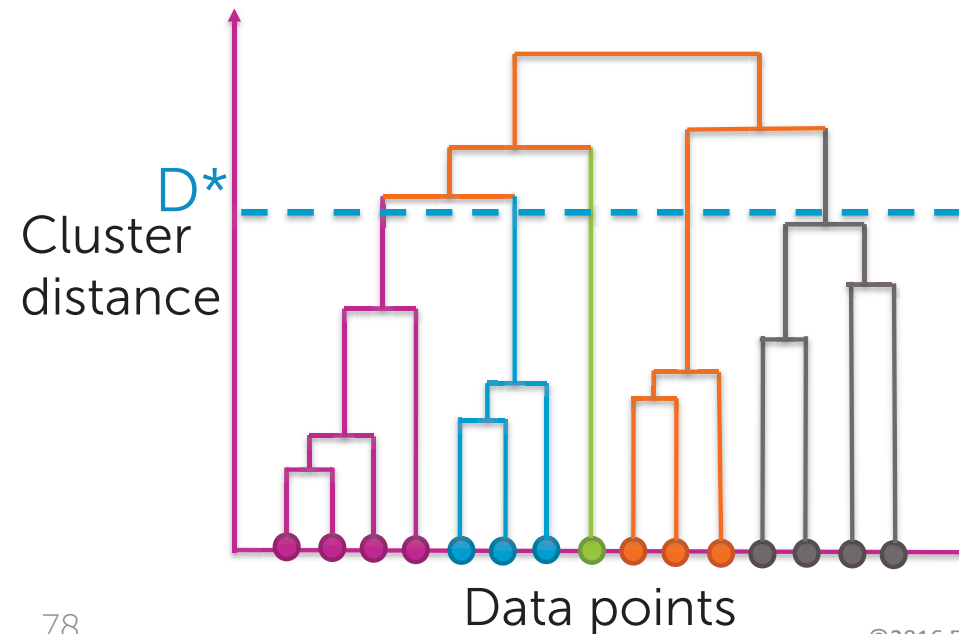
Extracting a partition

Choose a distance D^* at which to cut dendrogram



Extracting a partition

Every branch that crosses D^* becomes a separate cluster



Extracting a partition

Every branch that crosses D^*
becomes a separate cluster

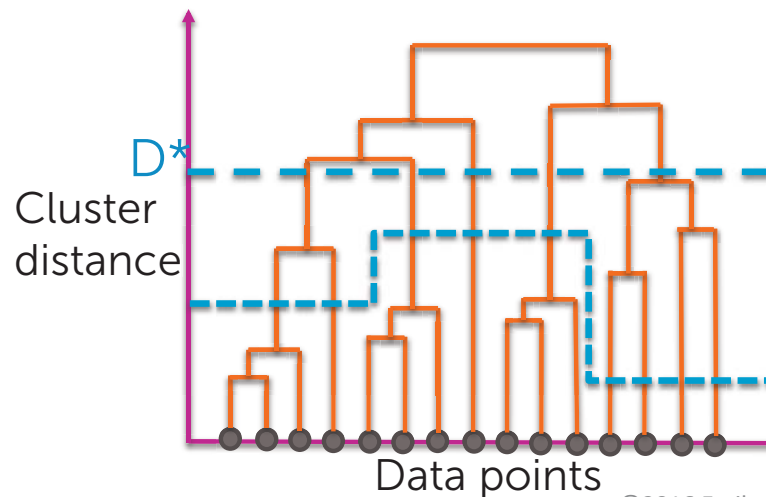




Agglomerative clustering details

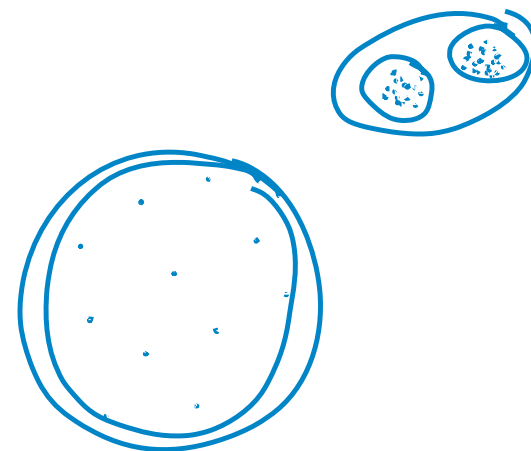
Agglomerative choices to be made

- Distance metric: $d(\mathbf{x}_i, \mathbf{x}_j)$
- Linkage function: e.g., $\min_{\substack{\mathbf{x}_i \in C_1, \\ \mathbf{x}_j \in C_2}} d(\mathbf{x}_i, \mathbf{x}_j)$
- Where and how to cut dendrogram




More on cutting dendrogram

- For visualization, smaller # clusters is preferable
- For tasks like outlier detection, cut based on:
 - Distance threshold
 - Inconsistency coefficient
 - Compare height of merge to average merge heights below
 - If top merge is substantially higher, then it is joining two subsets that are relatively far apart compared to the members of each subset internally
 - Still have to **choose a threshold** to cut at, but now in terms of "inconsistency" rather than distance
- No cutting method is "incorrect", some are just more useful than others

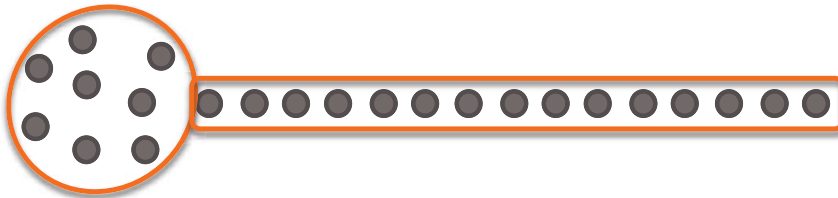


Computational considerations

- Computing all pairs of distances is **expensive**
 - Brute force algorithm is $O(N^2 \log(N))$
 # datapoints
- Smart implementations use triangle inequality to **rule out candidate pairs**
- Best known algorithm is $O(N^2)$

Statistical issues

Chaining: Distant points clustered together if there is a chain of pairwise close points between



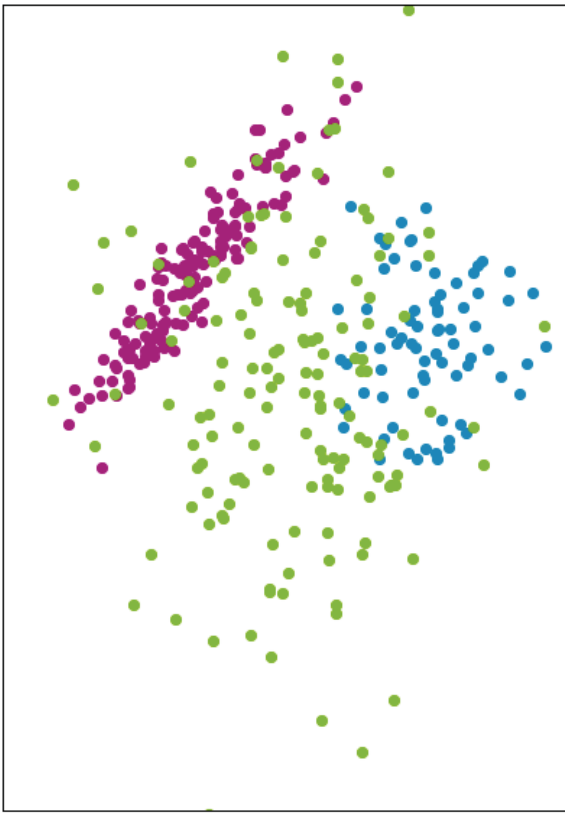
Other **linkage functions** can be more robust, but **restrict the shapes** of clusters that can be found

- **Complete linkage:**
max pairwise distance between clusters
- **Ward criterion:**
min within-cluster variance at each merge



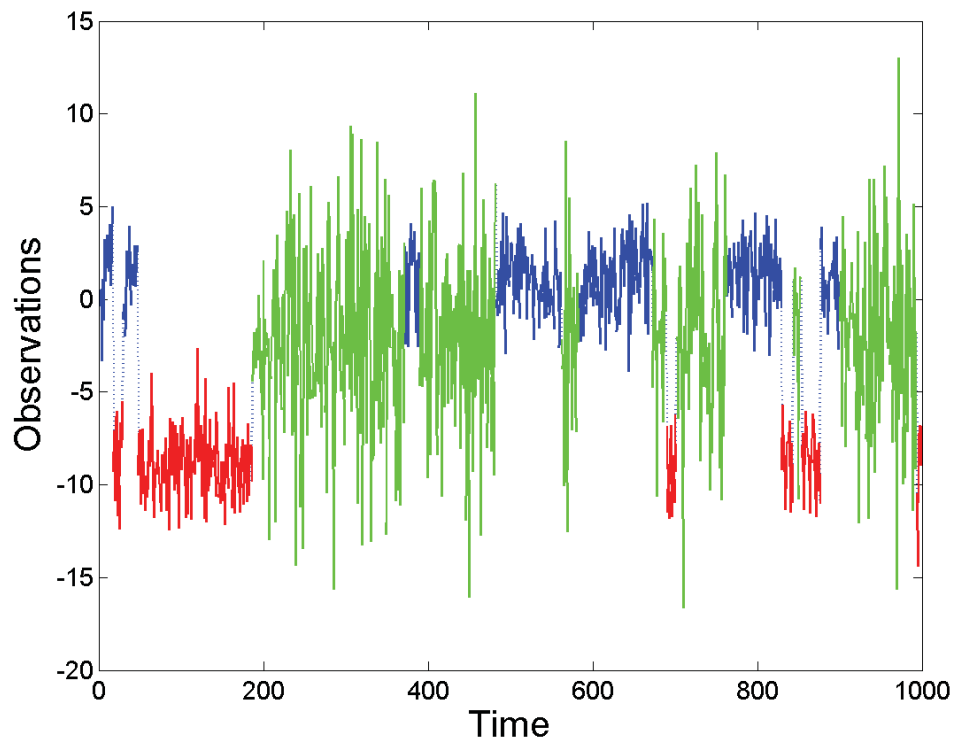
Hidden Markov models (HMMs): Another notion of “clustering”

So far, looked at clustering unordered data



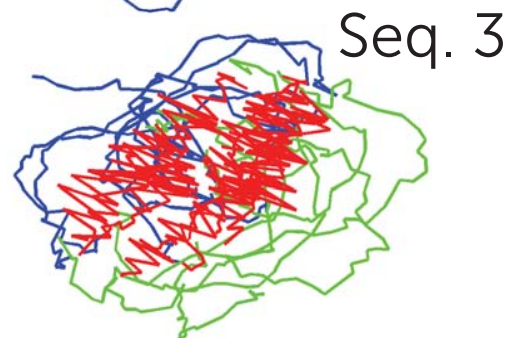
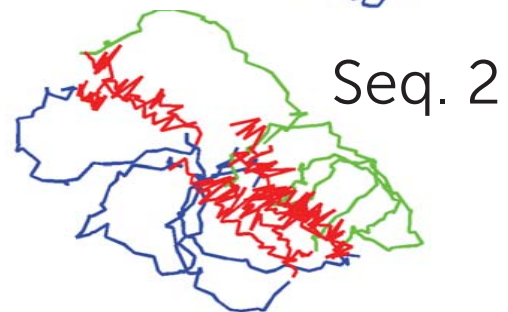
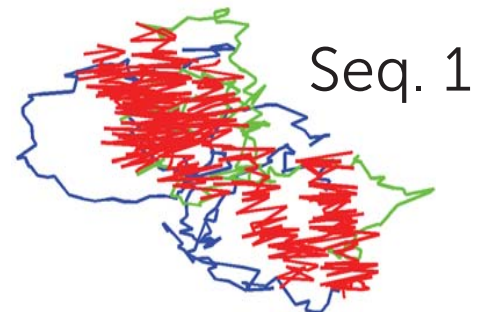
Data index (i.e., when observation was recorded) does not influence clustering

What if we have time series data?



Would be hard to distinguish **red**, **blue**, and **green** clusters if we ignored order of data

Example: Honey bee dances



Repeated patterns of dance transitions

Sequence 1



Sequence 2



Sequence 3



Cluster labels over time



waggle
dance

turn
right

turn
left

Similar ideas appear in many applications

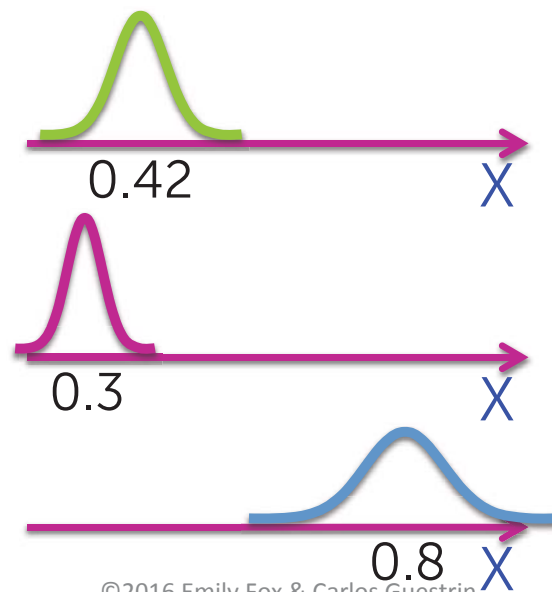


Hidden Markov model (HMM)

As in mixture model...

Every observation x_t is associated with cluster assignment variable z_t

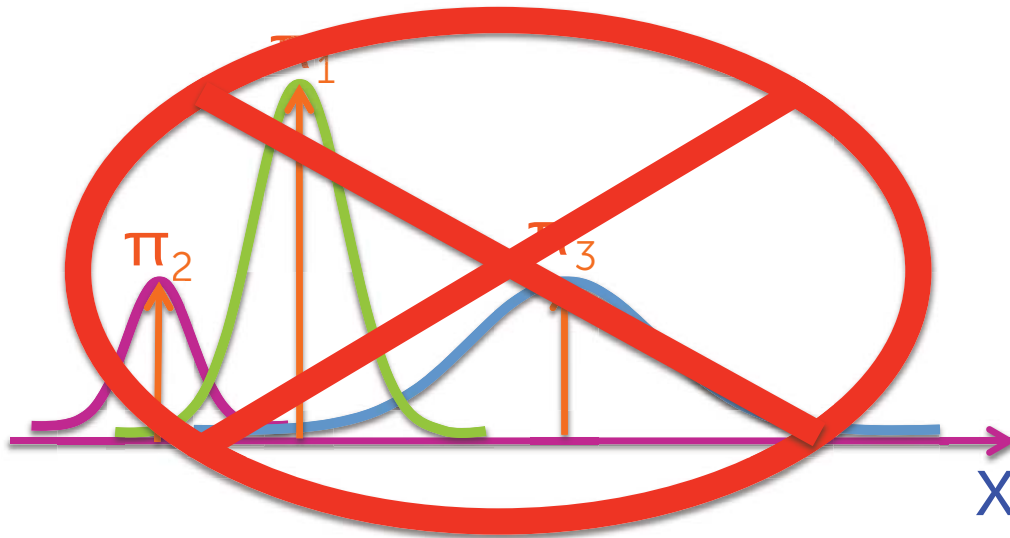
Each cluster has a distribution over observed values



Hidden Markov model (HMM)

Difference from mixture model:

Probability of ($z_t = k$) depends on previous cluster assignment z_{t-1}



Inference in HMMs

- Learn MLE of HMM parameters using EM algorithm = **Baum Welch**
- Infer MLE of state sequence given fixed model parameters using dynamic programming = **Viterbi algorithm**
- Infer soft assignments of state sequence using dynamic programming = **forward-backward algorithm**



What we didn't cover



Other clustering + retrieval topics

Retrieval:

- Other distance metrics
- Distance metric learning

Clustering:

- Nonparametric clustering
- Spectral clustering

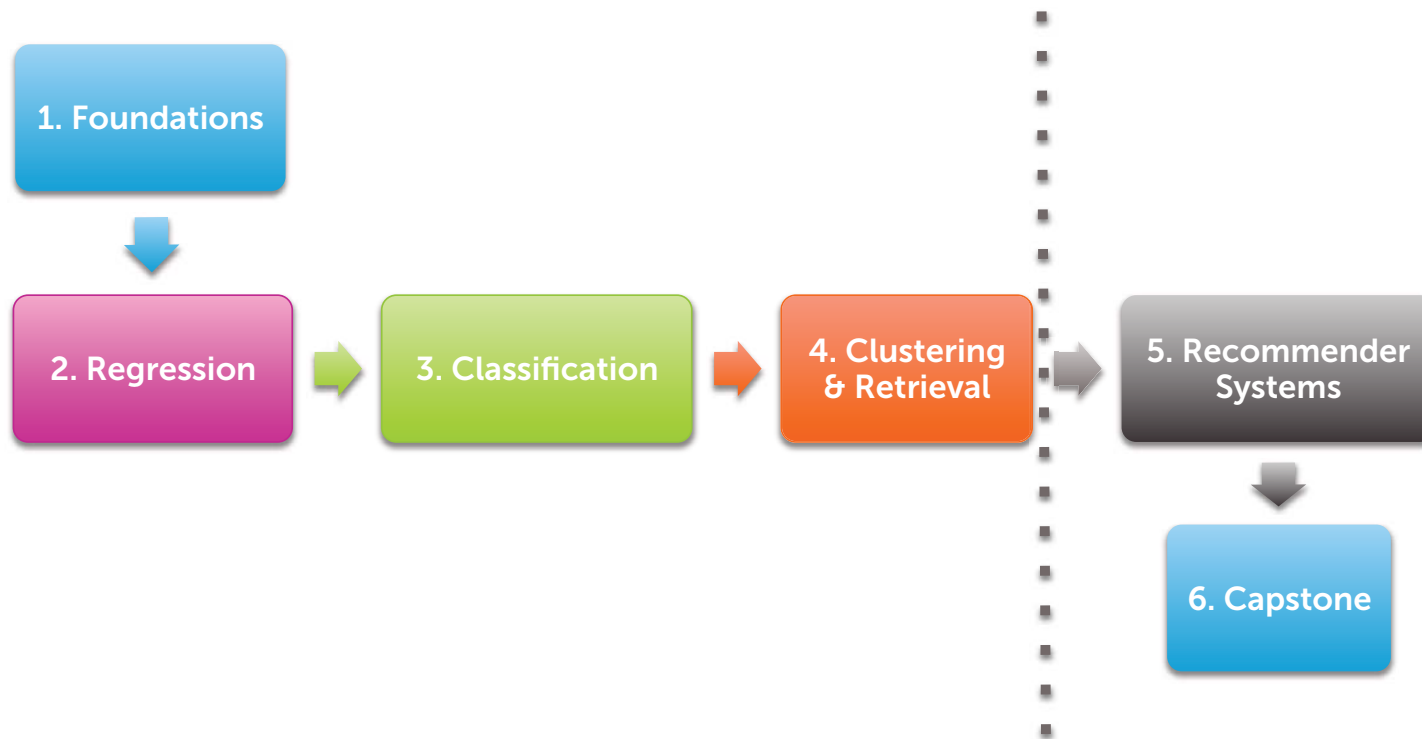
Related ideas:

- Density estimation
- Anomaly detection



What's ahead in this specialization

This course is a part of the Machine Learning Specialization



5. Recommender Systems & Dimensionality Reduction

Case study: Recommending Products

Models

- Collaborative filtering
- Matrix factorization
- PCA

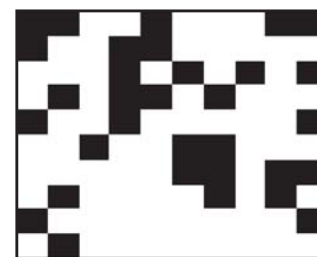
Algorithms

- Coordinate descent
- Eigen decomposition
- SVD

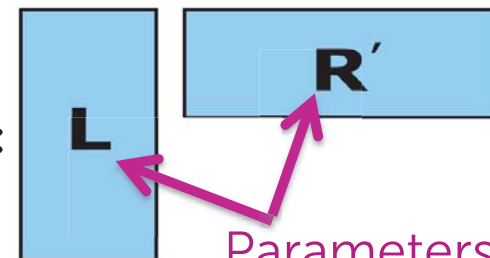
Concepts

- Matrix completion, eigenvalues, cold-start problem, diversity, scaling up

Rating =



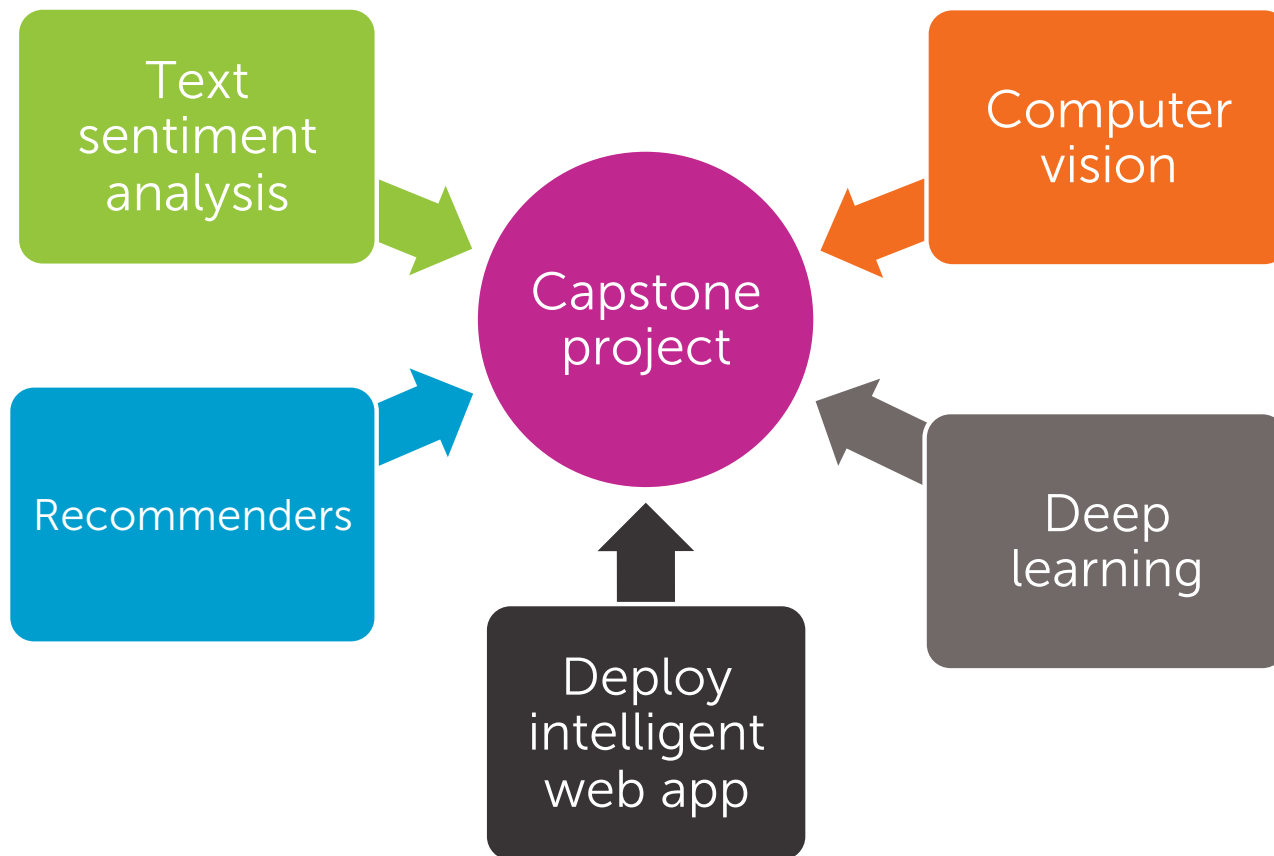
\approx



Parameters of model

Machine Learning Specialization

6. Capstone: *Build and deploy an intelligent application with deep learning*





Thank you...