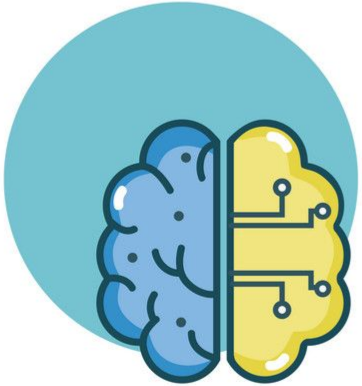
INTRODUCTION TO MACHINE LEARNING

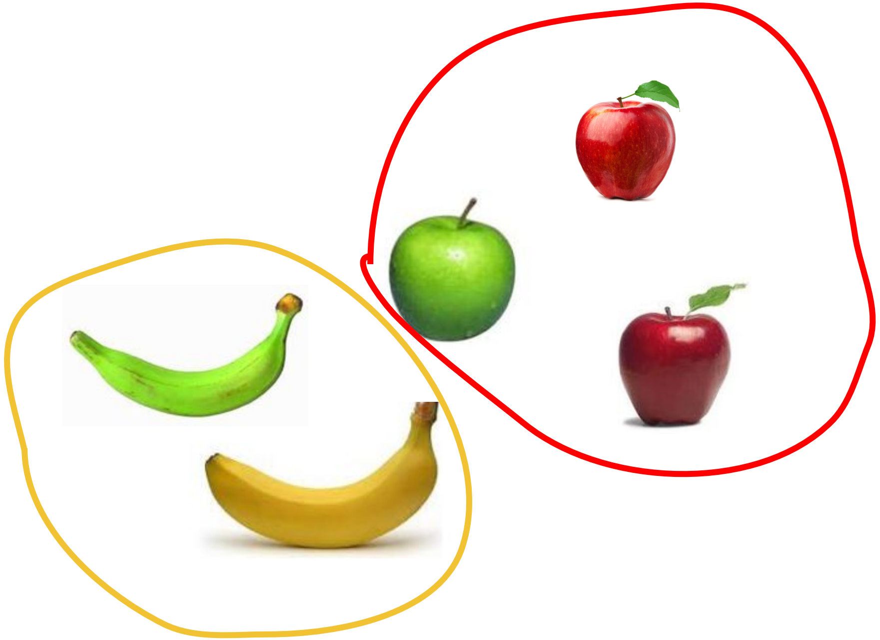
Clustering

Elisa Ricci

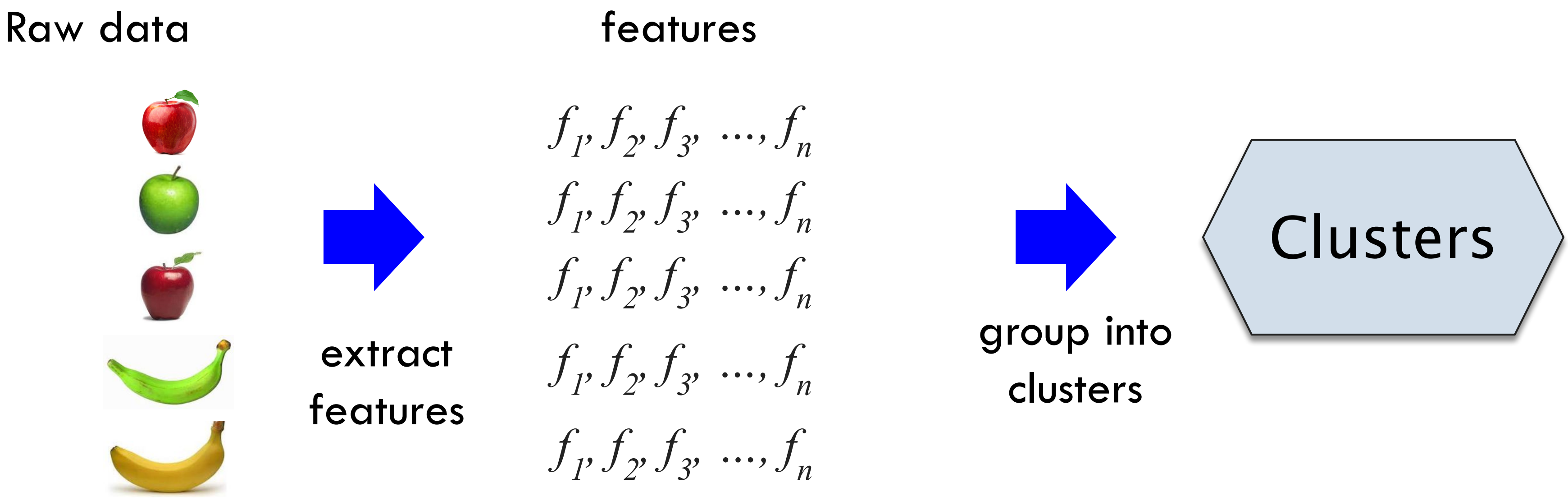


Unsupervised Learning: Clustering

Unsupervised learning setting: we are given data, i.e. examples, but no labels



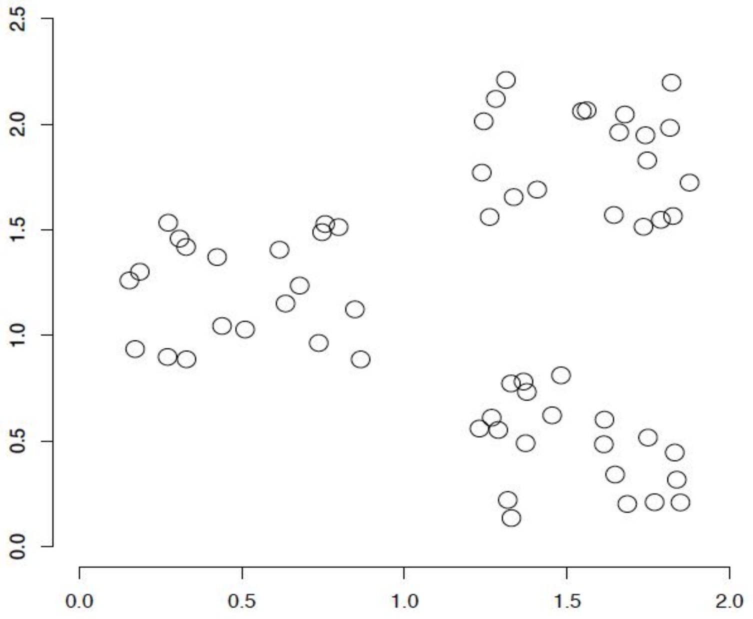
Unsupervised Learning: Clustering



No supervision: we are only given data and want to find groupings

Unsupervised Learning: Clustering

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

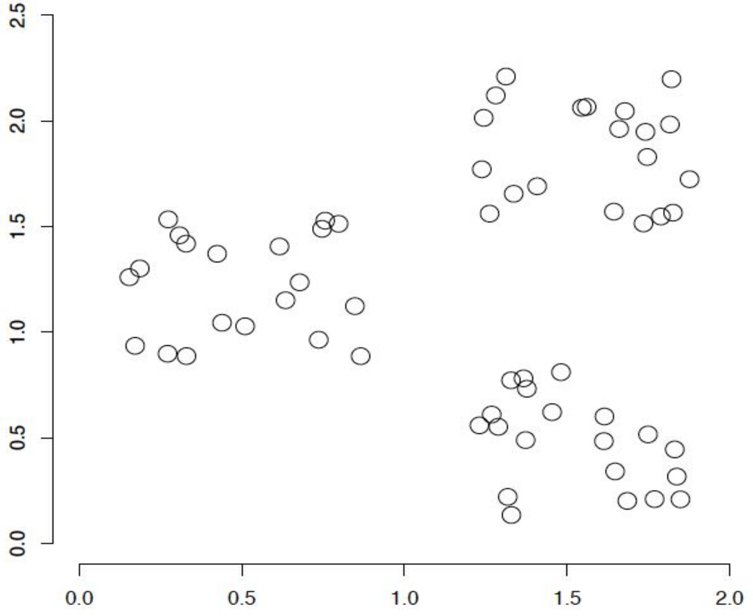


Which clustering algorithms can we use?

What are some of the issues for clustering?

Unsupervised Learning: Clustering

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



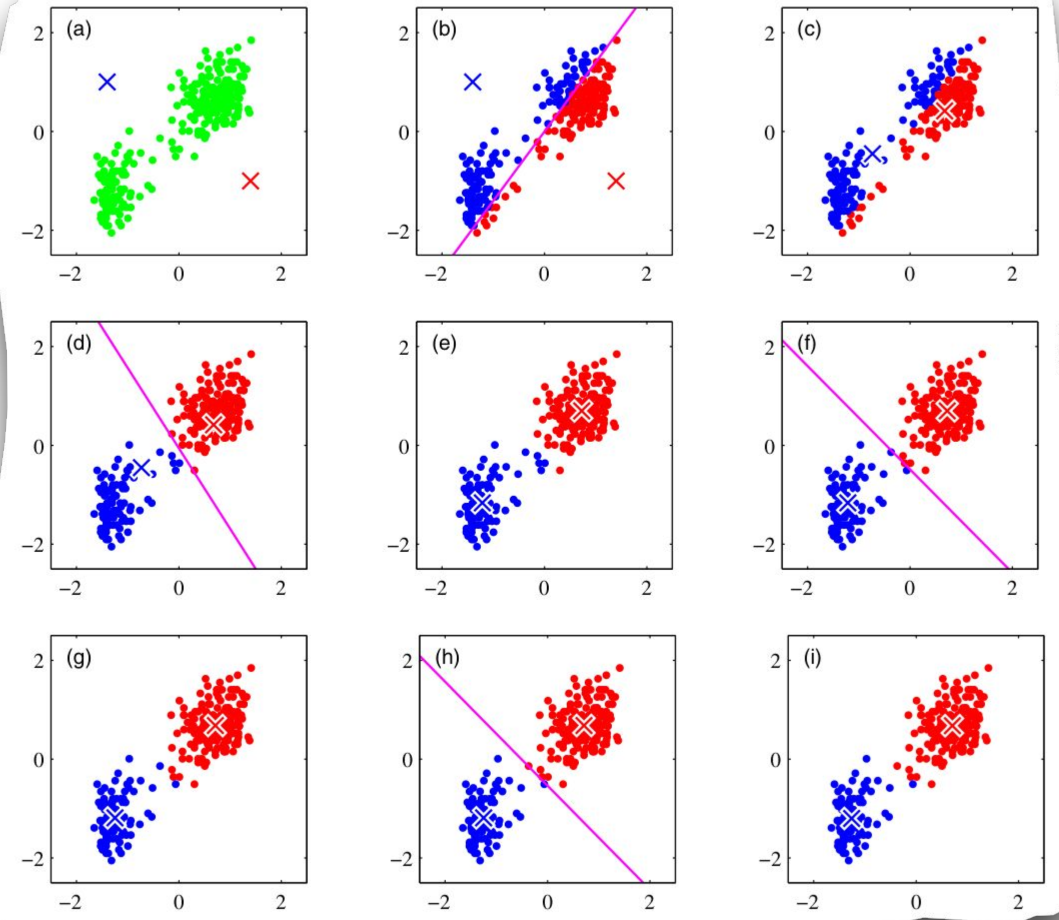
Which clustering algorithms can we use?

K-Means

Most well-known and popular clustering algorithm.

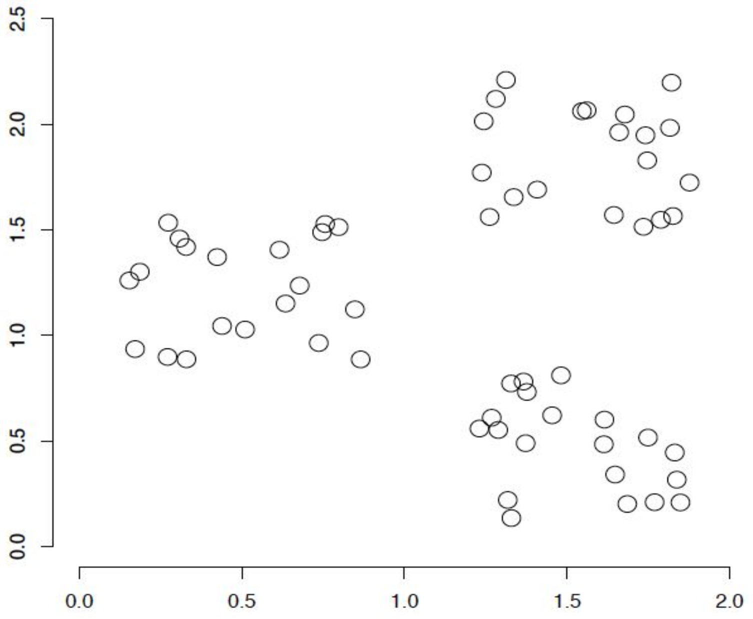
- Start with some initial cluster centroids
- Iterate:
  - Assign each example to closest centroid
  - Recalculate centers as the mean of the points in a cluster

K-Means



Unsupervised Learning: Clustering

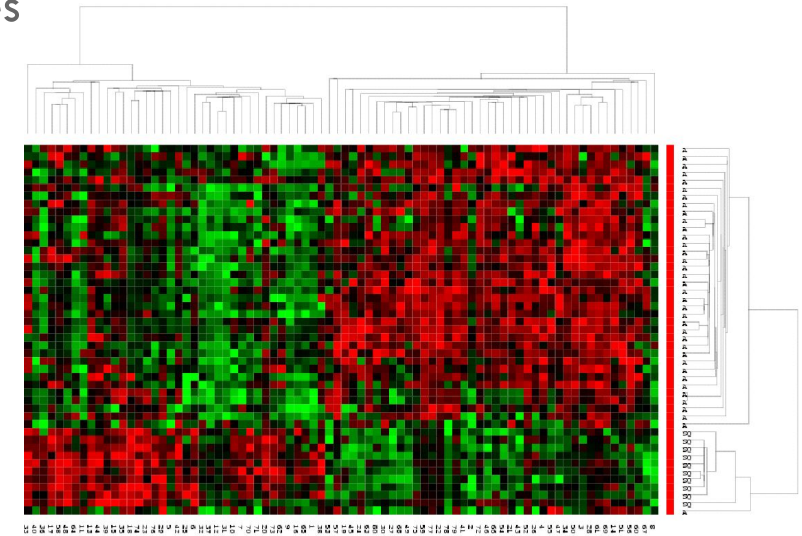
Clustering: the process of grouping a set of objects into classes of similar objects



What are some of the issues for clustering?

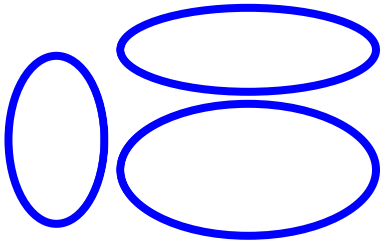
Issues for Clustering

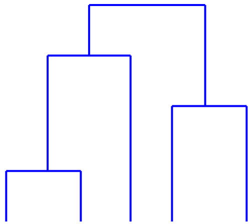
- Representation: how do we represent examples?
  - features?
- Similarity/distance between examples
- Flat clustering or hierarchical
- Number of clusters
  - Fixed a priori
  - Data driven



Clustering Algorithms

- **Flat algorithms**
  - K-means clustering (usually start with a random partitioning and refine it iteratively)
  - Spectral clustering
- **Hierarchical algorithms**
  - Bottom-up, agglomerative
  - Top-down, divisive

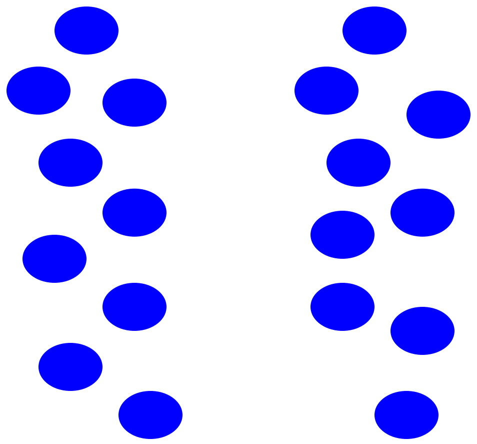




Hard vs. soft clustering

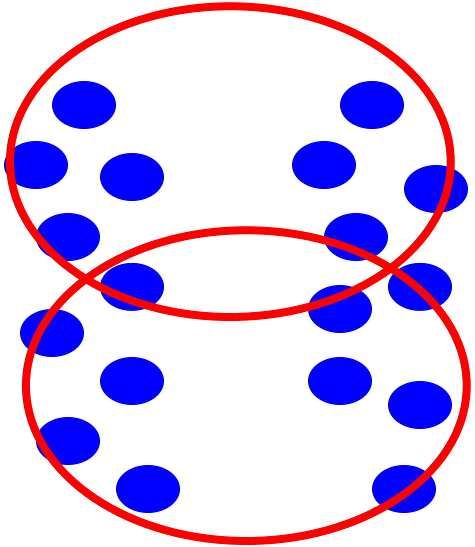
- **Hard clustering:** Each example belongs to exactly one cluster
  - K-Means
- **Soft clustering:** An example can belong to more than one cluster (probabilistic)
  - Makes more sense for applications like creating browsable hierarchies, e.g. you may want to put a pair of sneakers in two clusters: (i) sports apparel and (ii) shoes

Problems with K-means



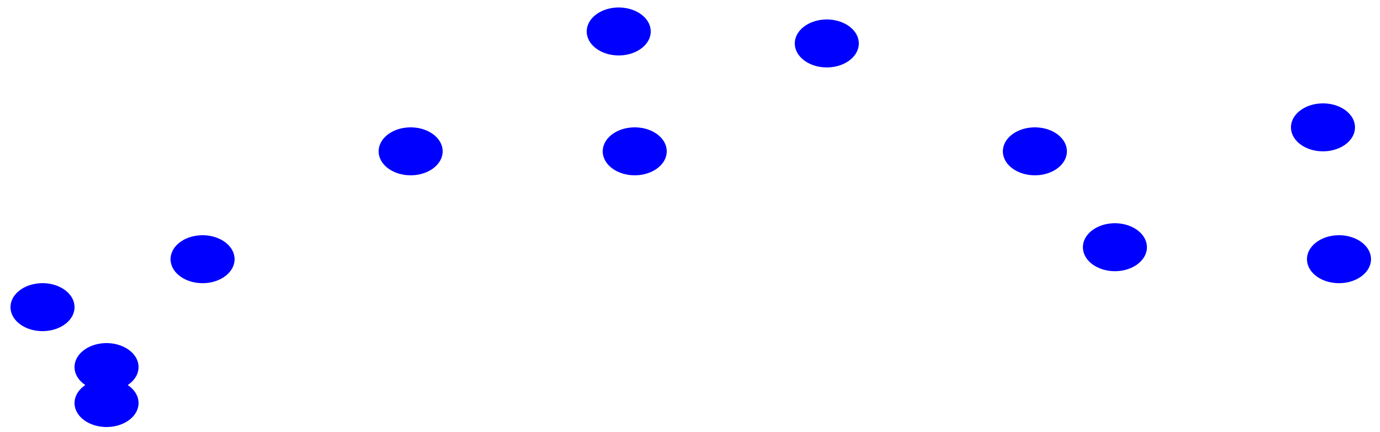
What would K-means give us here?

Problems with K-means

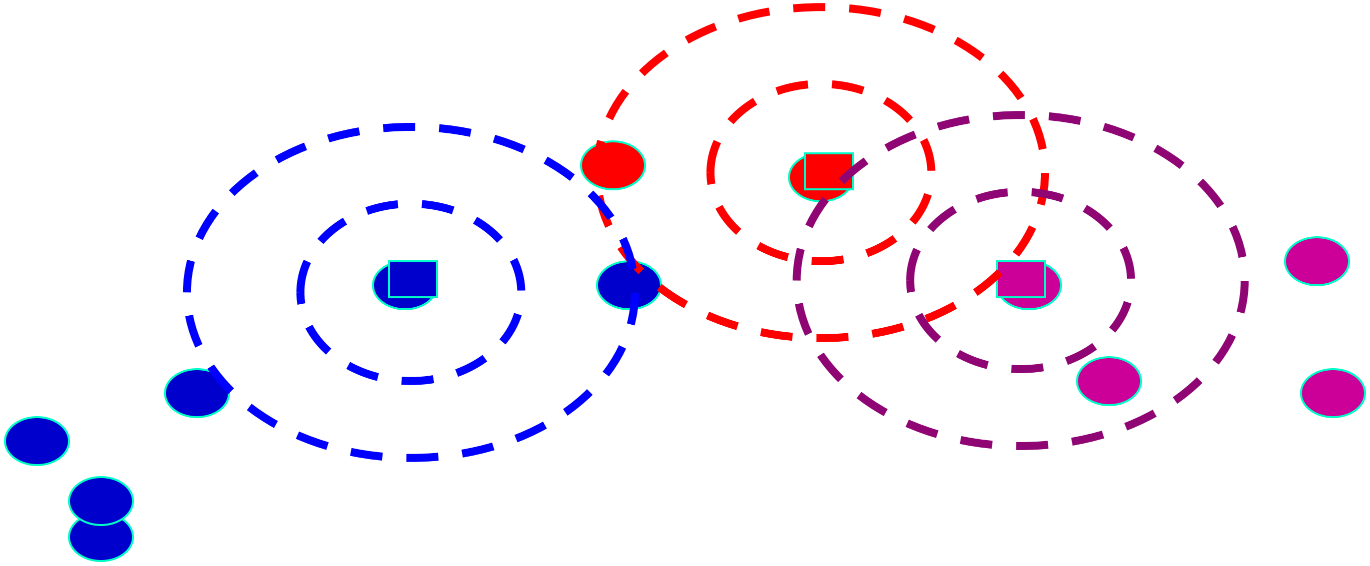


k-means assumes spherical clusters!

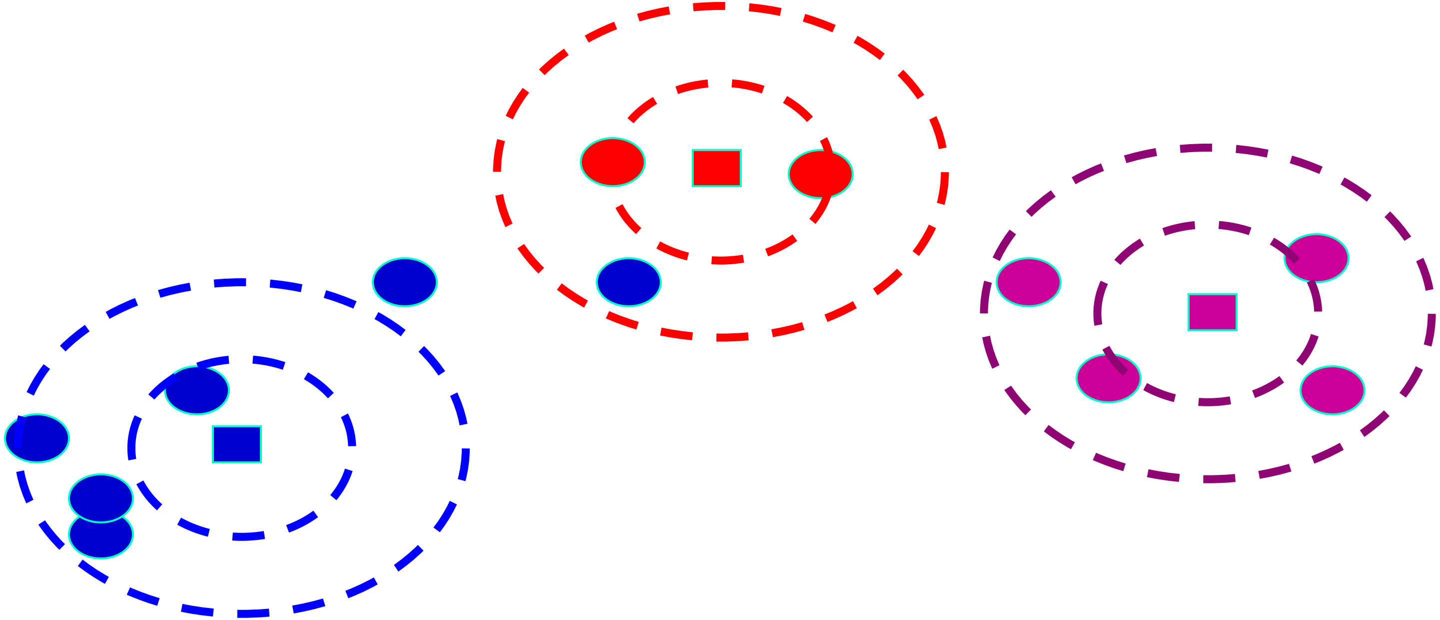
Spherical Clusters



Spherical Clusters



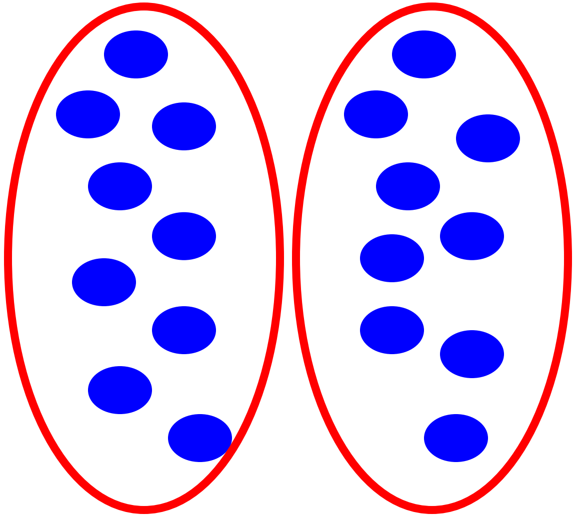
Spherical Clusters



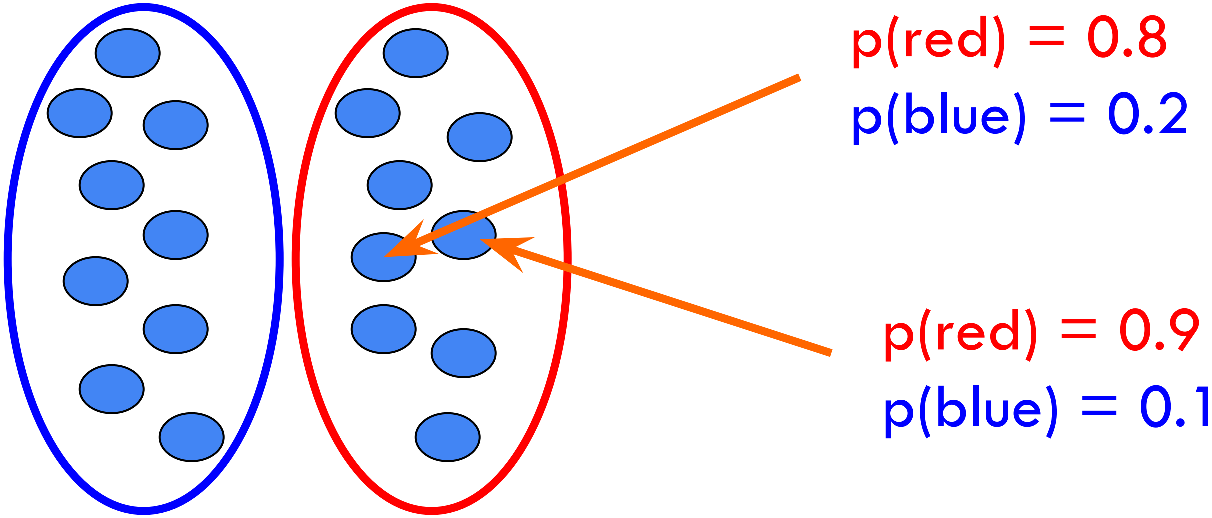
Iteratively learning a collection of spherical clusters

A Better Algorithm: EM Clustering

Expectation Maximization (EM): assume data came from a **mixture of Gaussians** (elliptical data), assign data to cluster with a certain probability (**soft clustering**)



Soft Clustering



EM Clustering

- Very similar at a high-level to K-means
  - Iterate between assigning points and recalculating cluster centers

- Two main differences between K-means and EM clustering
  - We assume **elliptical clusters** (instead of spherical)
  - It is a **soft clustering** algorithm

EM Clustering

- Start with some initial cluster centers
- Iterate:
  - Soft assign points to each cluster

    Calculate: $p(\theta_c | x)$

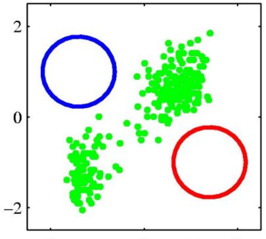
    the probability of each point belonging to each cluster

  - Recalculate the cluster centers

    Calculate new cluster parameters θ_c

    maximum likelihood cluster centers given the current soft clustering

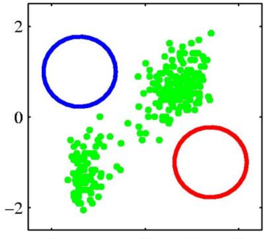
Example



(a)

Start with some initial cluster centers

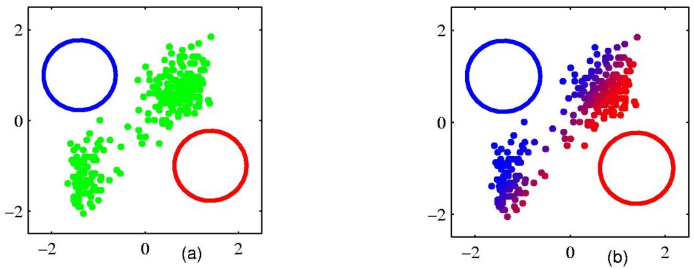
Example



(a)

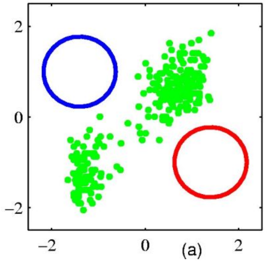
Which points belong to which clusters (soft)?

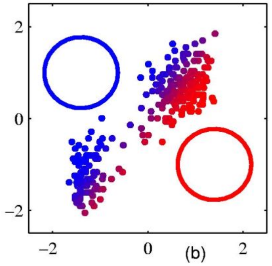
Example



It is a soft (probabilistic) assignment

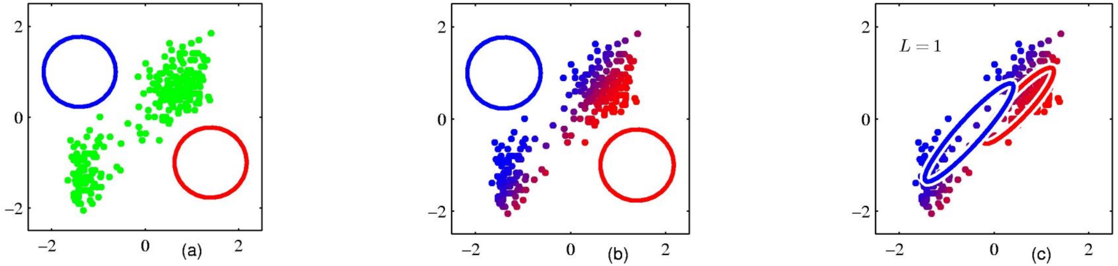
Example





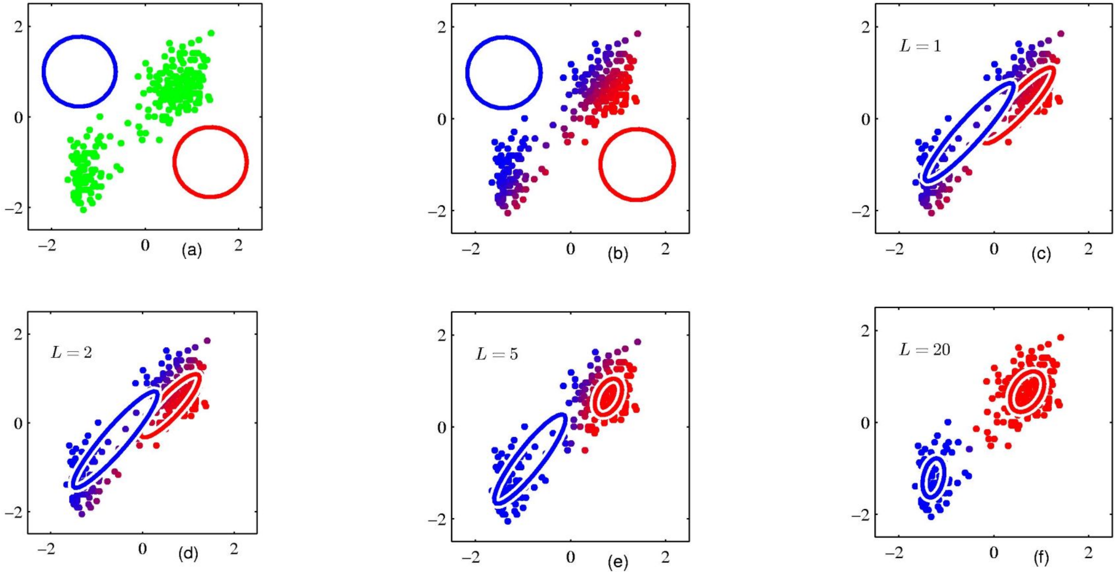
What do the new centers look like?

Example



Cluster centers get a **weighted** contribution from points

Example



... after several iterations

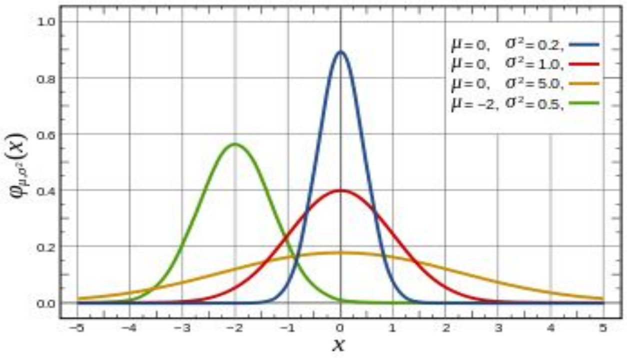
Recap: Mixture of Gaussians

How do you define a Gaussian (i.e. ellipse)?

In 1D

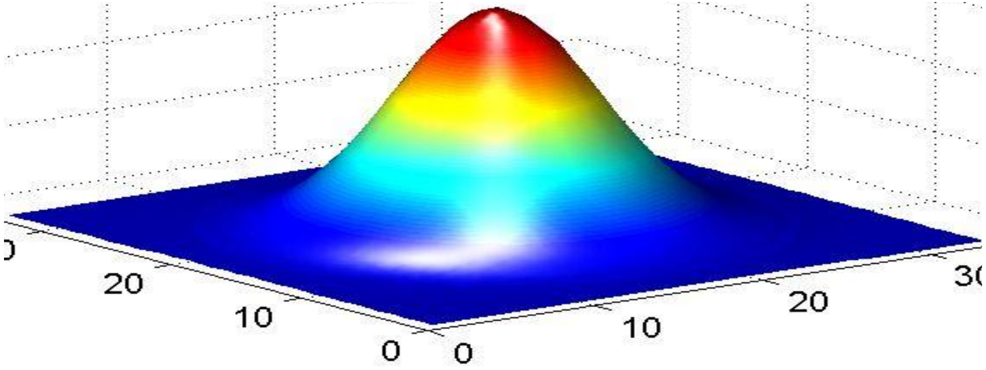
$$f(x) = \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{(x-\mu)^2}{2\sigma^2}}$$

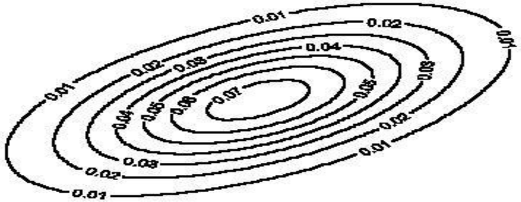
parameterized by the mean and the standard deviation



Recap: Mixture of Gaussians

In 2D





Covariance determines
the shape of these contours

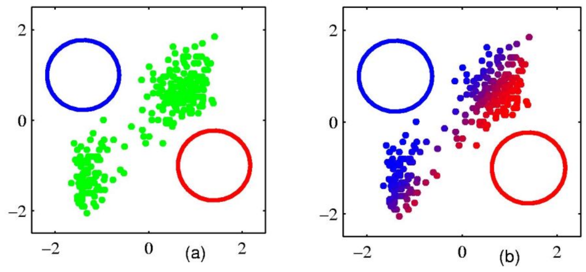
Recap: Mixture of Gaussians

In general…

$$N[x;\mu,\Sigma] = \frac{1}{(2\pi)^{d/2}\sqrt{\det(\Sigma)}}\exp[-\frac{1}{2}(x-\mu)^T \Sigma^{-1}(x-\mu)]$$

We learn the means of each cluster (i.e. the center) and the covariance matrix (i.e. how spread out it is in any given direction)

Step 1: soft cluster points

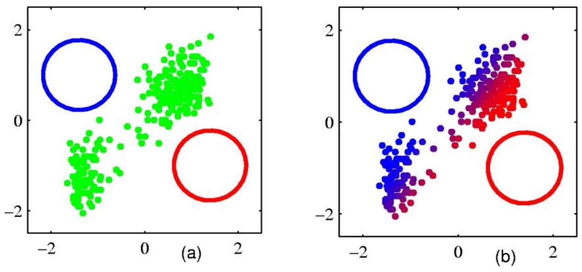


Soft assign points to each cluster

- Calculate: $p(\theta_c|x)$

  (the probability of each point belonging to each cluster)

Step 1: soft cluster points



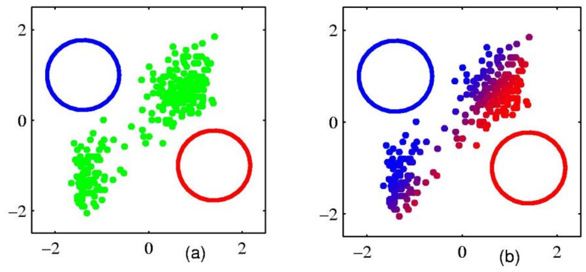
Soft assign points to each cluster

- Calculate: $p(\theta_c|x)$

  (the probability of each point belonging to each cluster)

How do we calculate these probabilities?

Step 1: soft cluster points



Soft assign points to each cluster

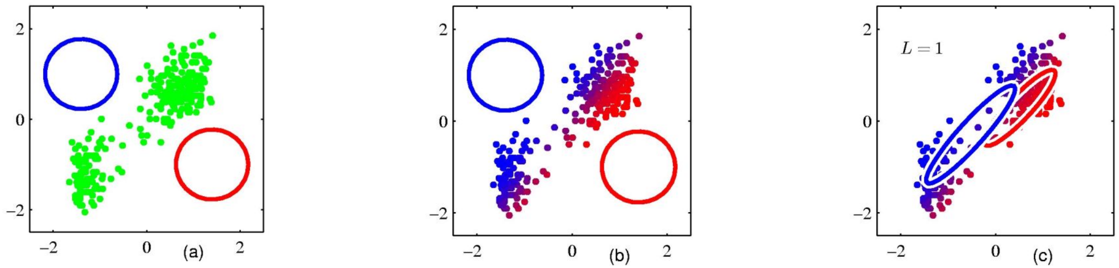
- Calculate: $p(\theta_c|x)$

  (the probability of each point belonging to each cluster)

Just plug into the probability the Gaussian equation for each cluster (normalized to make a probability)

Step 2: recalculate centers

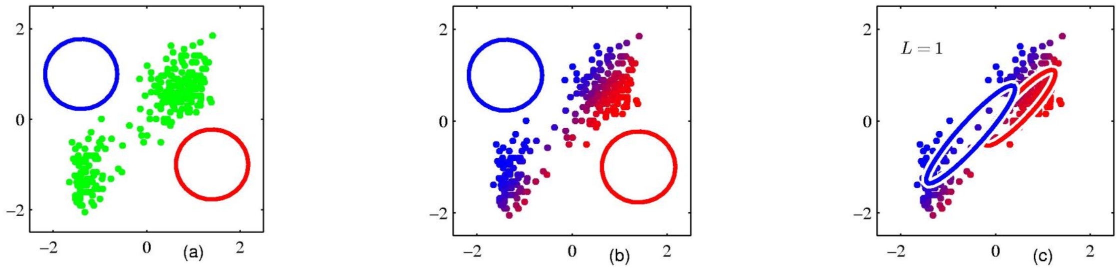
- Recalculate centers:
  - Calculate new cluster parameters θ_c
  - Maximum likelihood cluster centers given the current soft clustering



Step 2: recalculate centers

- Recalculate centers:
  - Calculate new cluster parameters θ_c
  - Maximum likelihood cluster centers given the current soft clustering

How do we do that?



Fitting a gaussian: Idea

- What is the "best"-fit Gaussian for this data?

  10, 10, 10, 9, 9, 8, 11, 7, 6, ...

- Recall this is the 1-D Gaussian equation:

$$f(x) = \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{(x-\mu)^2}{2\sigma^2}}$$

The Maximum Likelihood Estimation (MLE) is just the mean and variance of the data!

EM Clustering

- EM stands for Expectation Maximization
  - **Expectation:** Given the current model, figure out the expected probabilities of the data points to each cluster

$$p(\theta_c|x)$$

What is the probability of each point belonging to each cluster?

  - **Maximization:** Given the probabilistic assignment of all the points, estimate a new model θ_c

Maximum likelihood estimation

EM Clustering

- Similar to K-Means
  - **Assign/cluster each point to closest center**

    $p(\theta_c|x)$ Soft assignment

  - **Recalculate centers as the mean of the points in a cluster**

    Estimate a new model θ_c

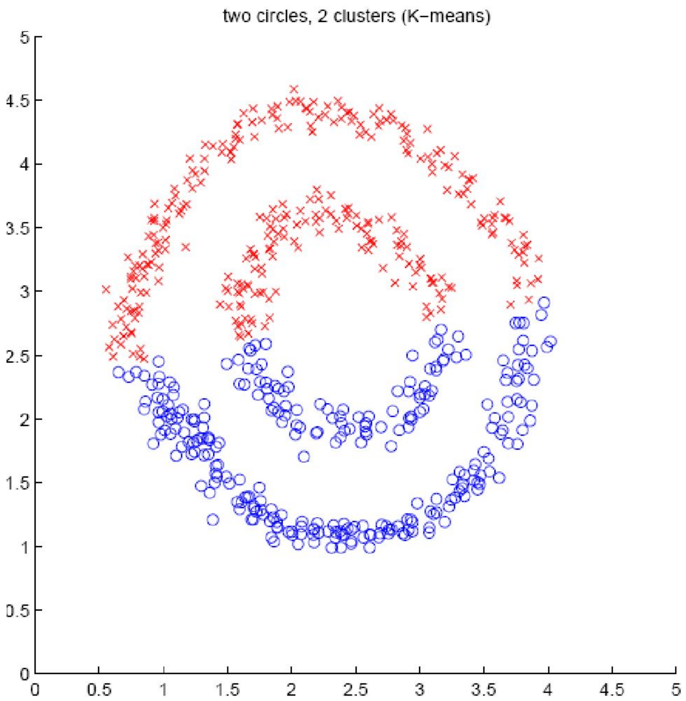
Expectation Maximization: Summary

- Similar to K-Means
- Each iterations increases the likelihood of the data and is guaranteed to converge (though to a local optimum)
- Not just for clustering while K-means is just for clustering
- EM is a general purpose approach for training a model when you do not have labels

Other Clustering Algorithms

- K-means and EM-clustering are by far the most popular for clustering
- However, they cannot handle all clustering tasks
- What types of clustering problems cannot they handle?

Non-Gaussian data



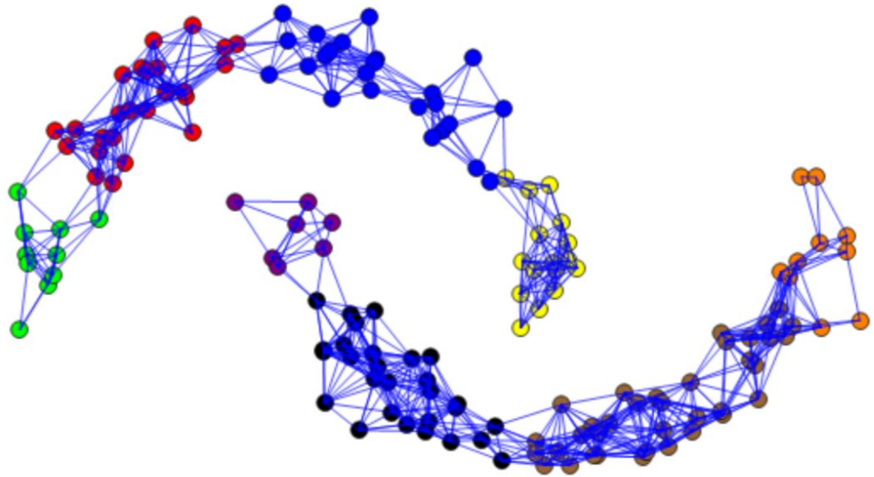
What is the problem?

Similar to classification:
global decision (linear model)
vs. local decision (K-NN)

Spectral clustering

Spectral Clustering

Group points based on links in a graph



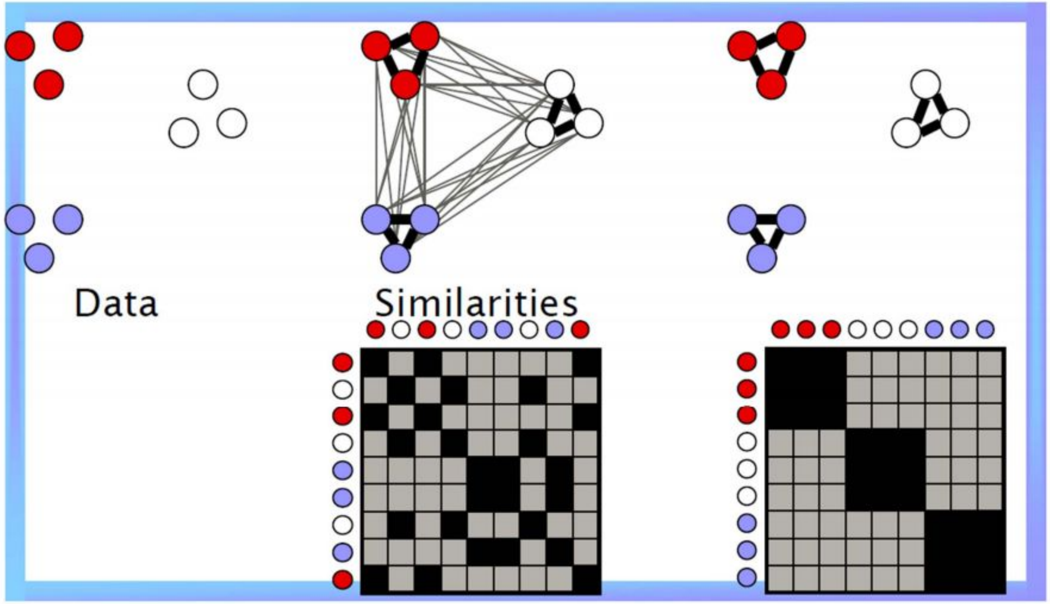
Create a graph

- We can create a fully connected graph or a K-nearest neighbor graph (each node is only connected to its K closest neighbors)
- We can create a graph with some notion of similarity among nodes. It is common to use a Gaussian Kernel

$$W(i,j) = \exp \frac{-\left|x_i - x_j\right|^2}{\sigma^2}$$

Graph partitioning

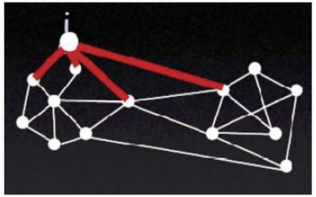
Define a similarity matrix and cut the graph



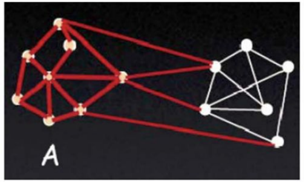
Graph Terminology

Degree of a node and volume of a set

$$d_i = \sum_j w_{i,j}$$

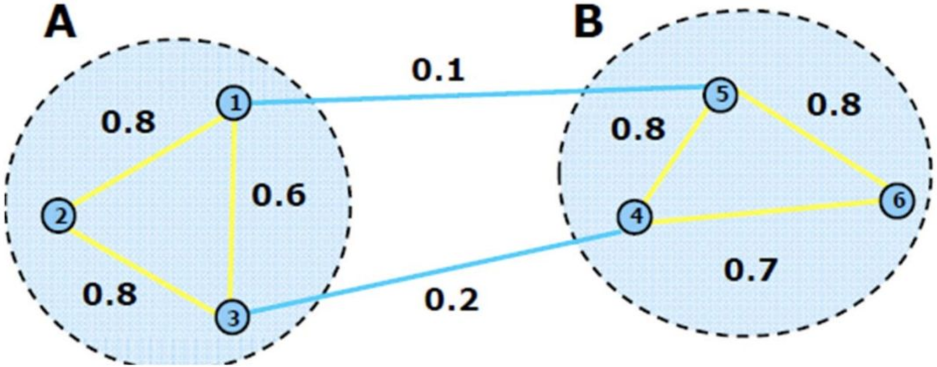


$$vol(A) = \sum_{i \in A} d_i, A \subseteq V$$



Graph cut

- Consider a partition of the graph into two parts A and B
- $Cut(A, B)$: sum of the weights of the set of edges that connect the two groups
- An intuitive goal is **find the partition that minimizes the cut**



$Cut(A, B) = 0.3$

Normalized cut

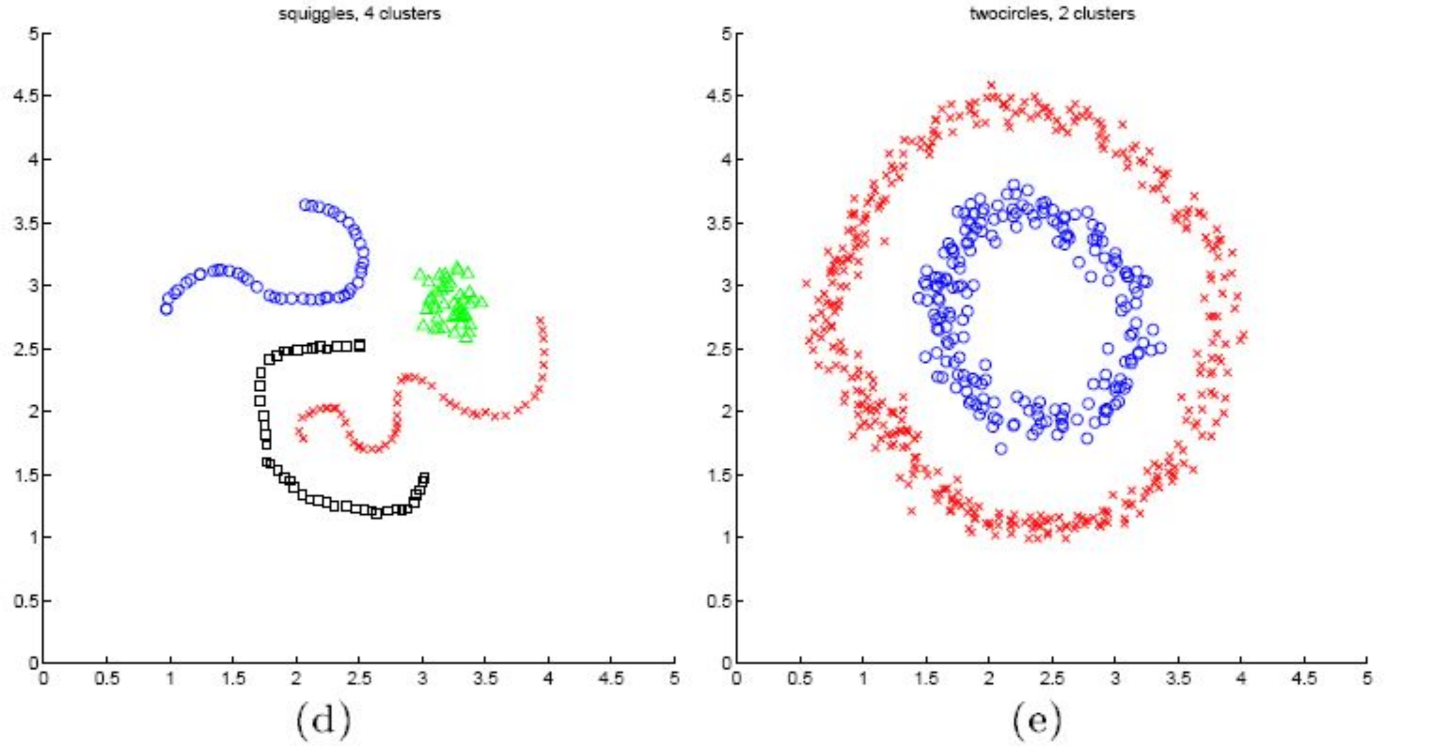
- Minimize

  $NCut(A, B) = Cut(A, B)/Vol(A) + Cut(A, B)/Vol(B)$

- This problem can formalized as a discrete optimization problem and then relaxed in the continuous domain and become a generalized eigenvalue problem

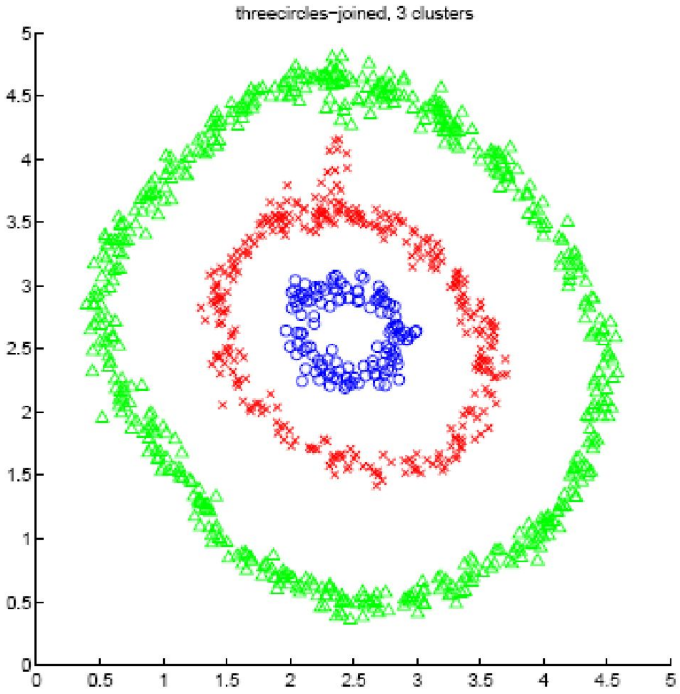
http://www.sci.utah.edu/~gerig/CS7960-S2010/handouts/Normalized%20Graph%20cuts.pdf

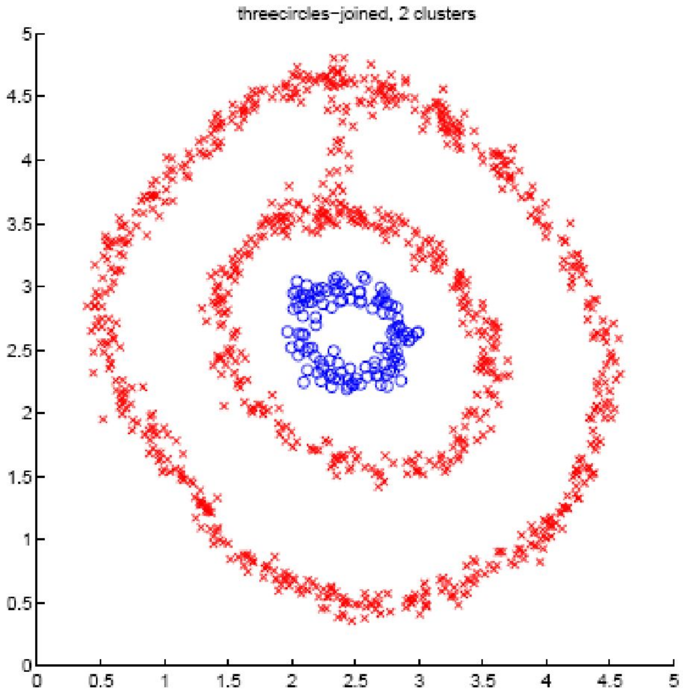
Spectral Clustering



Ng et al. On Spectral clustering: analysis and algorithm

Spectral Clustering

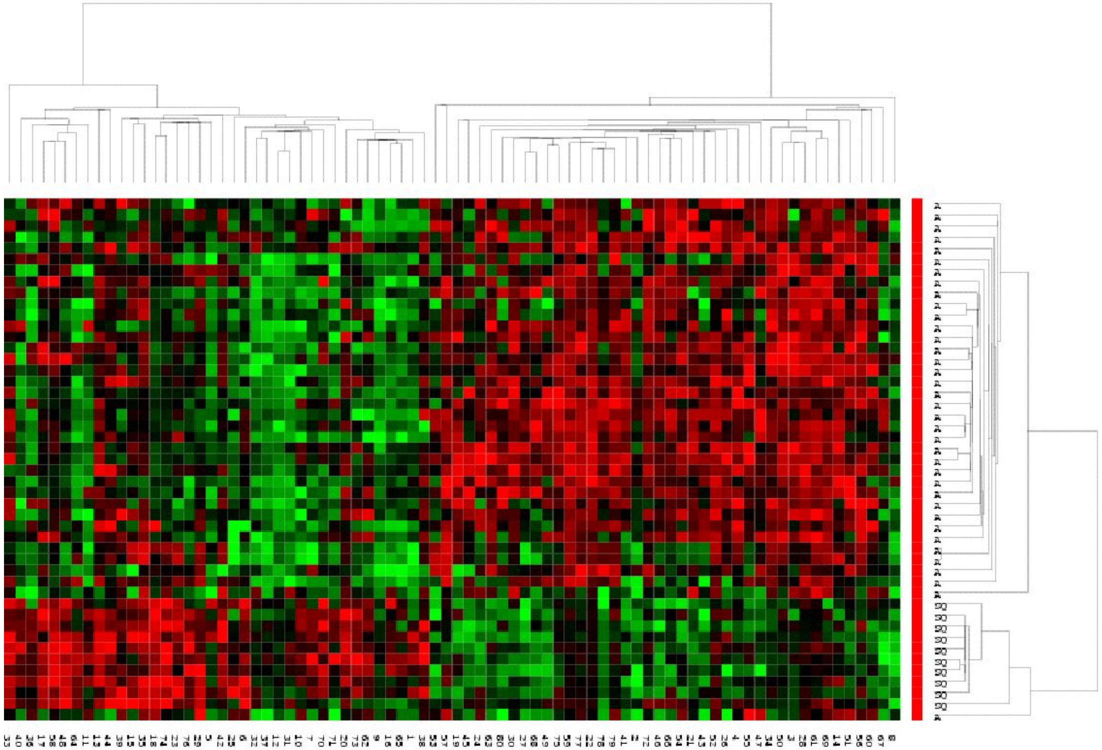




Ng et al. On Spectral clustering: analysis and algorithm

Hierarchical Clustering

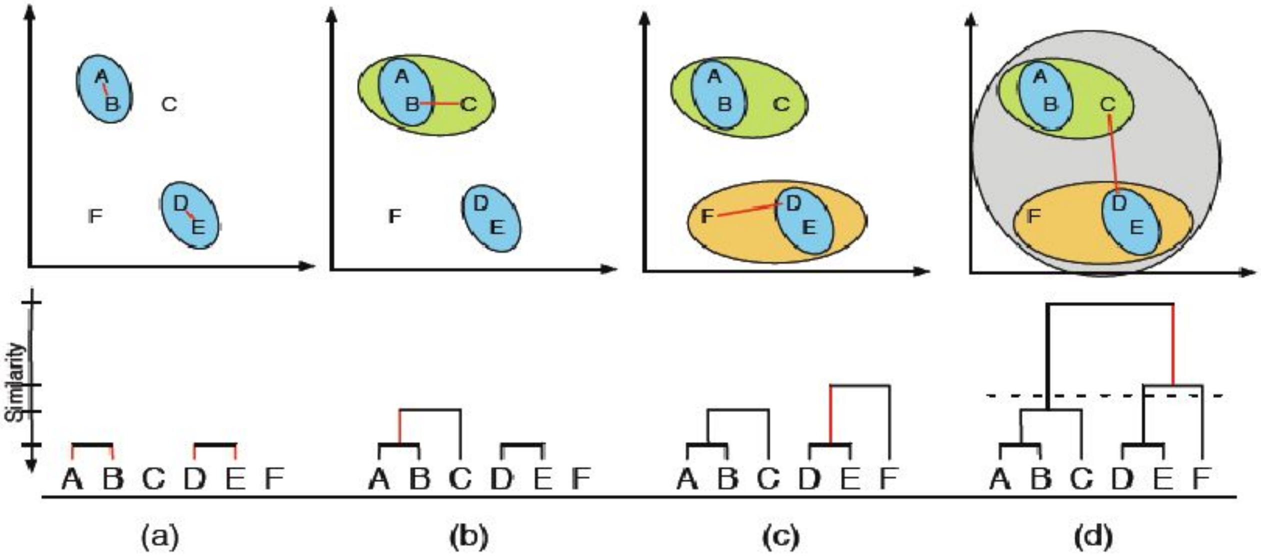
Gene profiles



Hierarchical Clustering

Produces a set of nested clusters organized as a hierarchical tree

The tree is called **dendrogram**

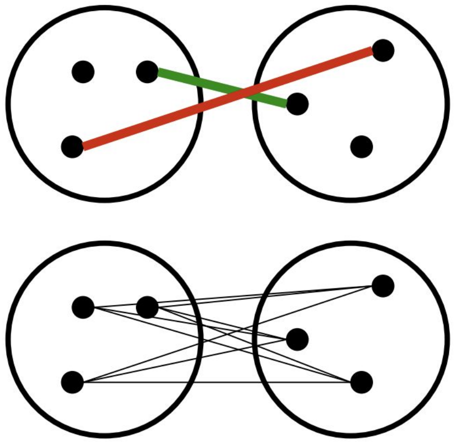


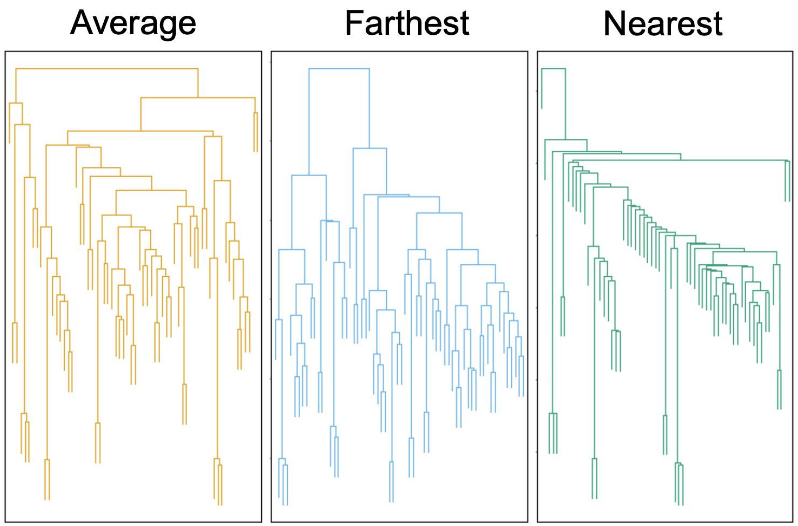
Agglomerative clustering

- **Idea:** first merge very similar instances and then incrementally build larger clusters out of smaller clusters
- **Algorithm**:
  - Initially, each instance in its own cluster
  - Repeat:
    - Pick the two closest clusters
    - Merge them into a new cluster
    - Stop when there is only one cluster left
- Produces not one clustering, but **a family of clusterings** represented by a dendrogram

Agglomerative Clustering

- How should we define "closest" for clusters with multiple elements?
- Different choices (closest pair, farest pair, average among all distances, etc.) creates different solutions





QUESTIONS?