Unsupervised Learning

Shane Shang

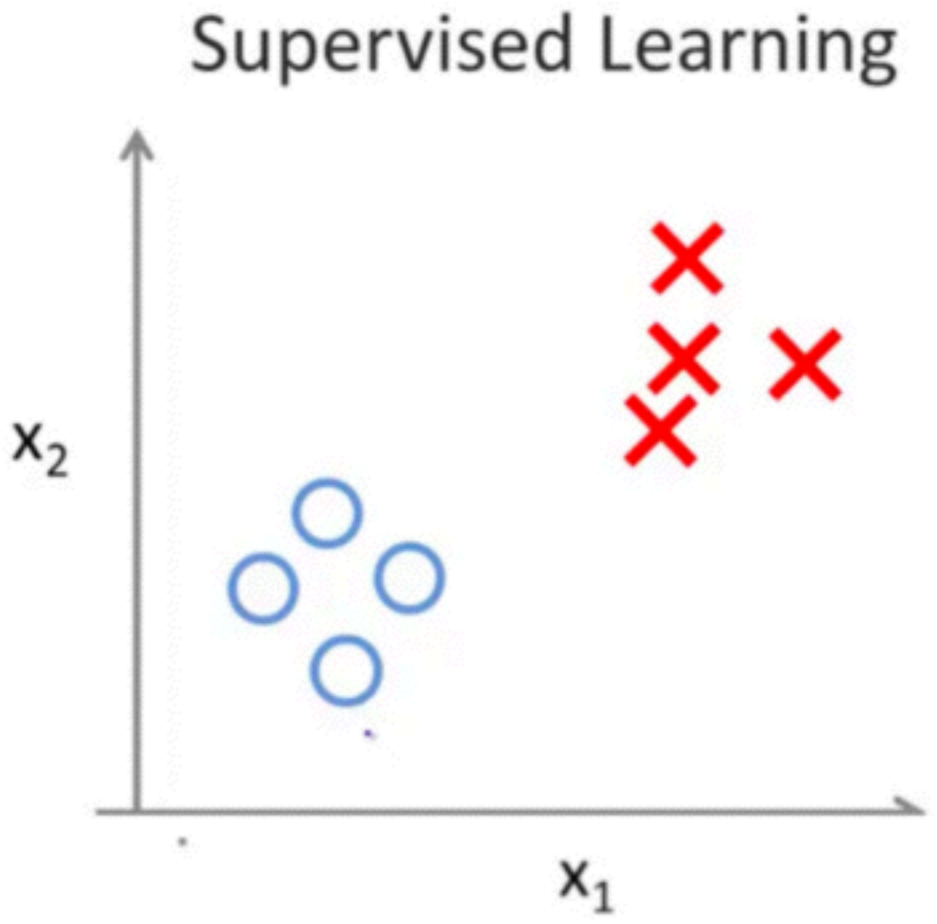
Outline

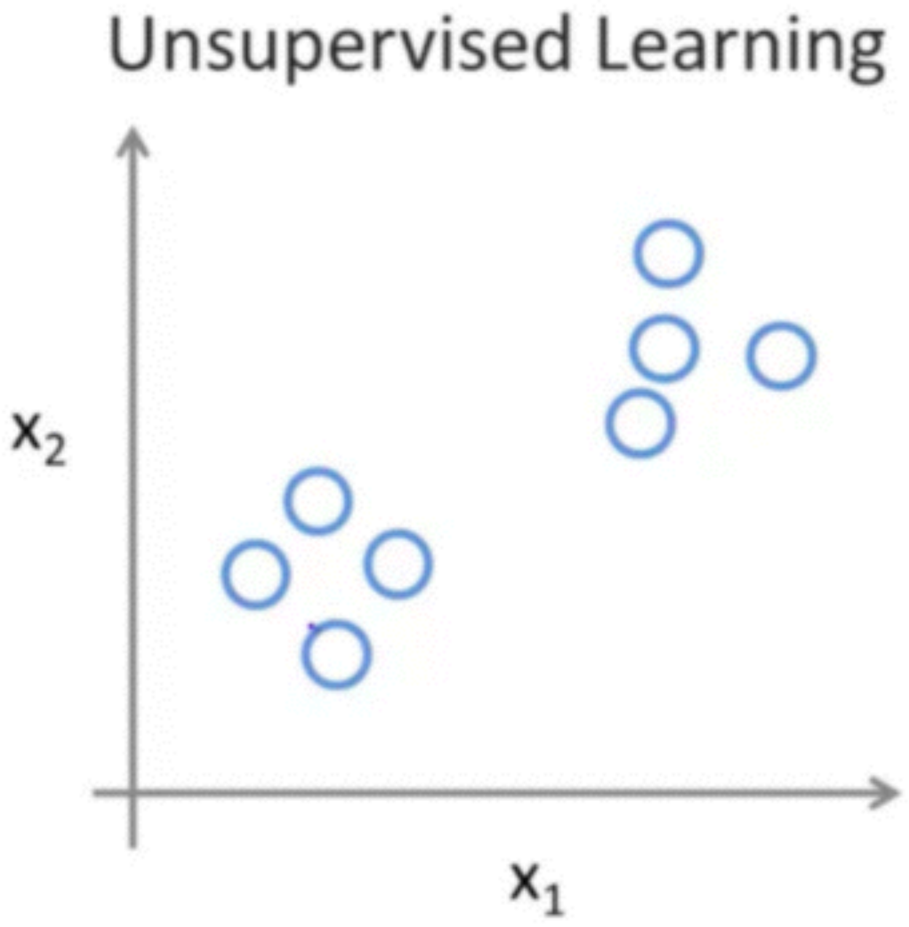
- **Introduction**
- Clustering
- Dimension reduction
- Extensions & Summary

What is unsupervised learning:

Learn patterns from “unlabeled data”.

(classification is not included in the observation)





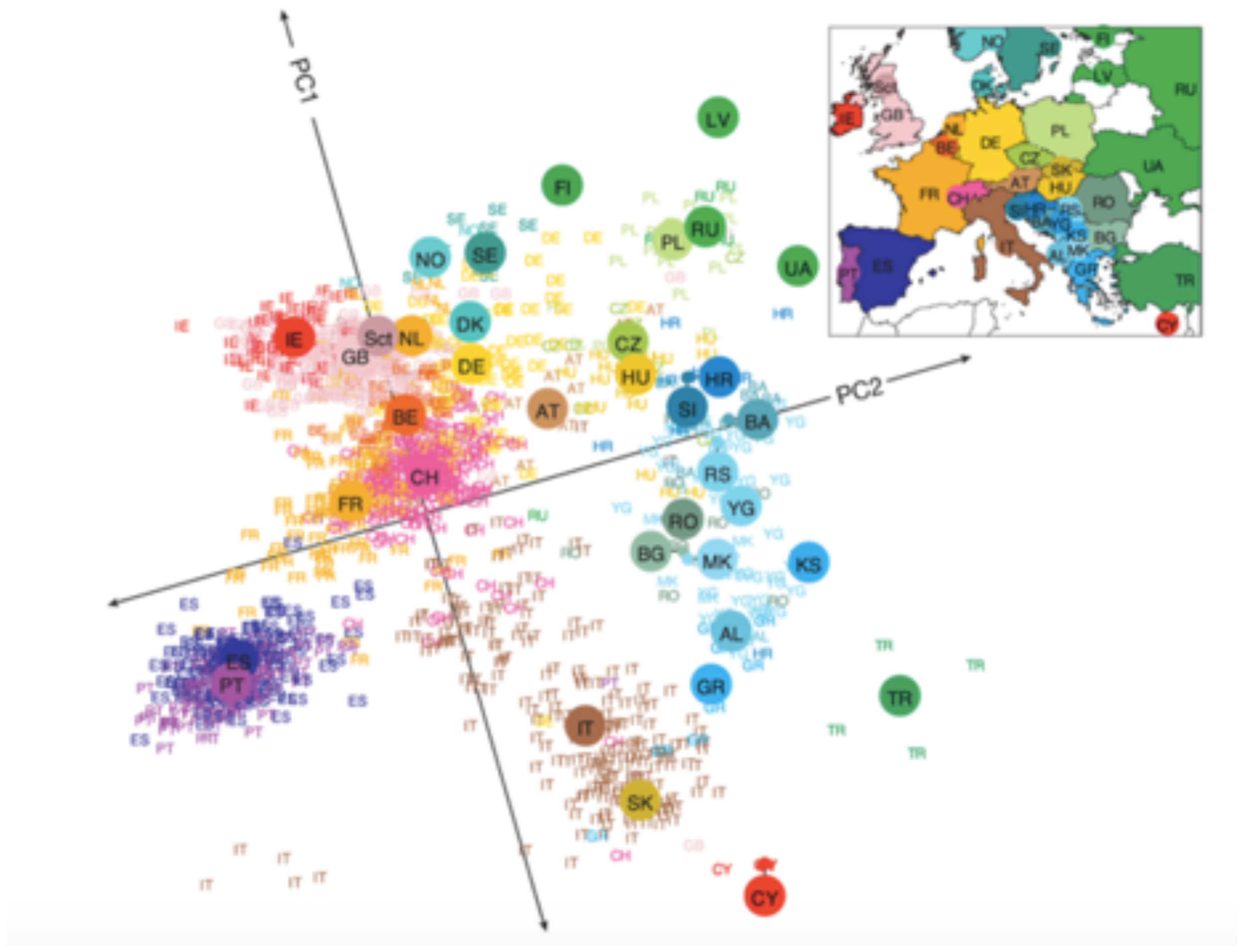
Finance and marketing

GBPUSD,Daily 1.58604 1.58736 1.58591 1.58725
Buy
Stoch(5,3,3)

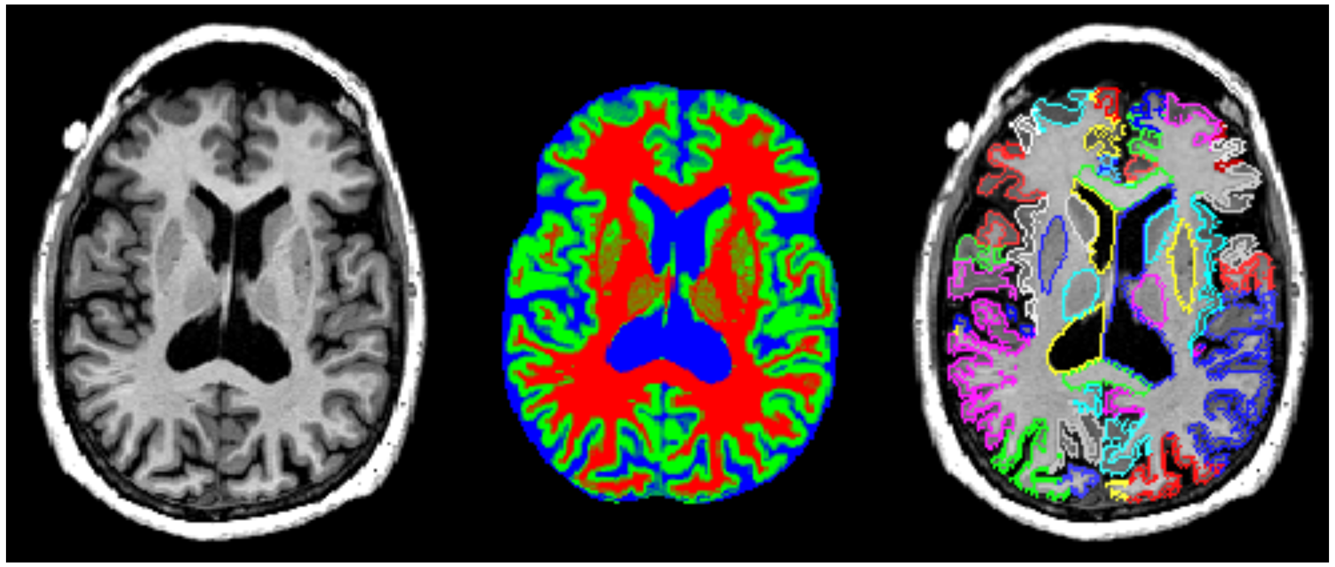
Stock market / forex trading

Costumer segmentation

Bioinformatic



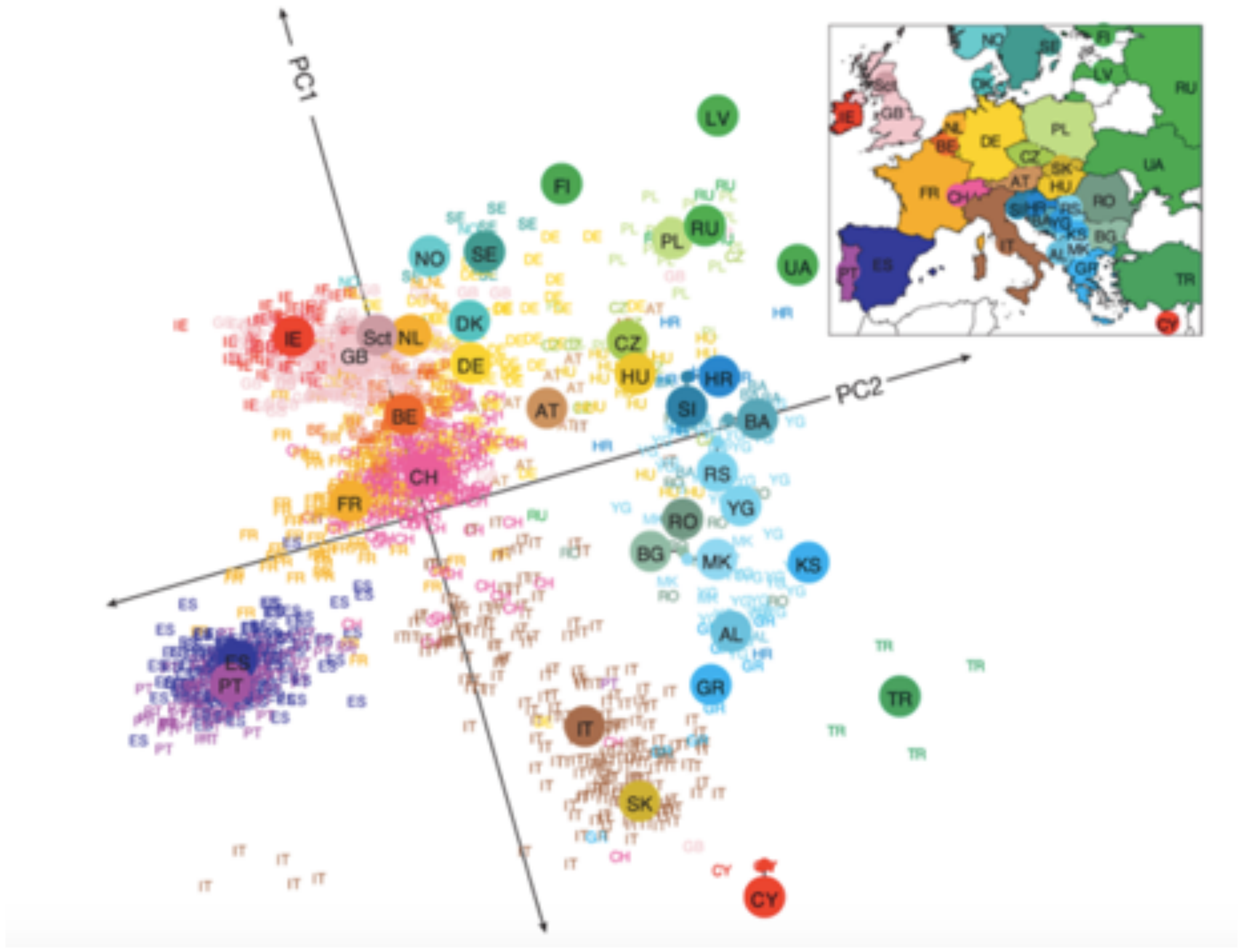
Genetic clustering of the Europeans by Principal Component Analysis



Brain MR image segmentation

Outline

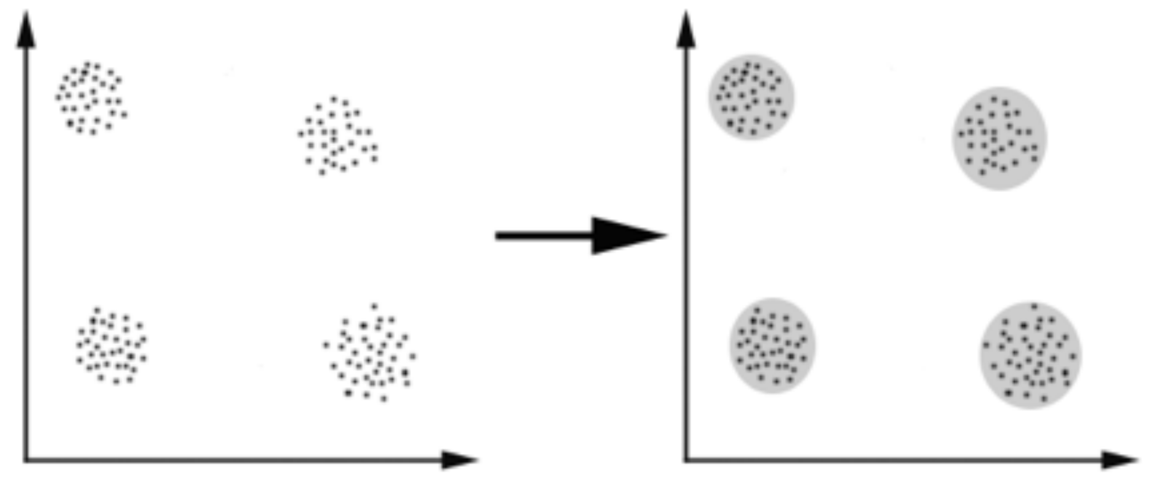
- Introduction
- **Clustering**
- Dimension reduction
- Extensions & Summary


PC1
PC2

Frequent Buyers
Seasonal Buyers
Top Spenders
Never Bought
First time coupon users

Clustering

Task: group a set of objects in such a way that objects in the same group (called a cluster) are more similar (in some sense or another) to each other than to those in other groups.



A more formal definition:

Given a data set {x1, x2, .., xN} consisting of N observations of a random D-dimensional variable x, partition the data set into k number of clusters.

Clustering — typical models

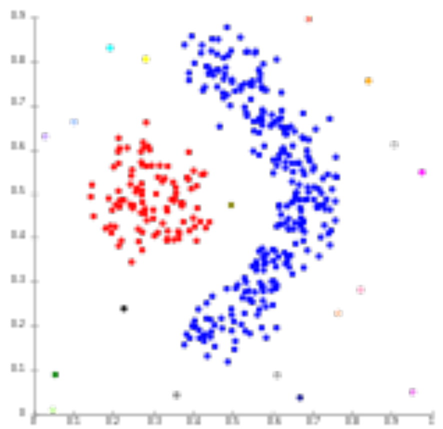
Depending on how a cluster is defined, there are different types of clustering models.

connectivity

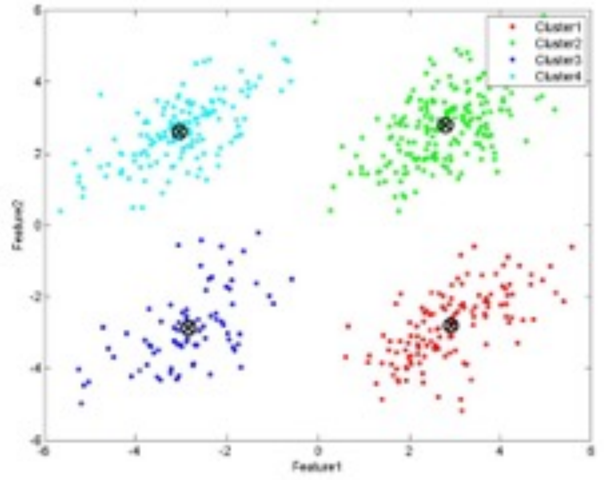
centroid

distribution

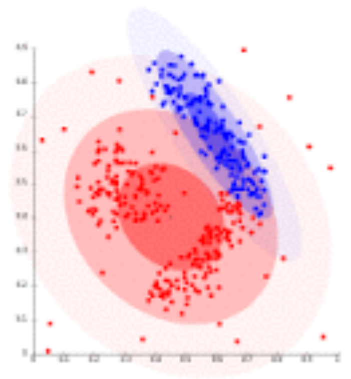
density



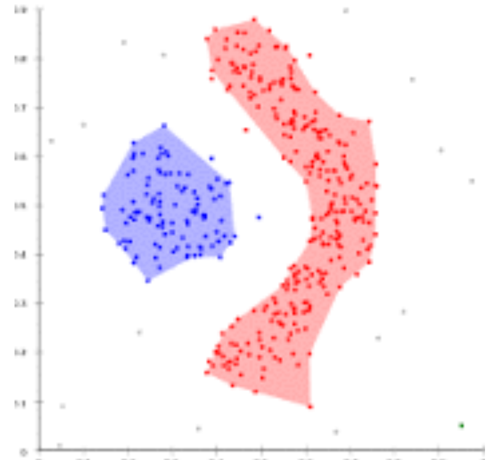
connectivity models



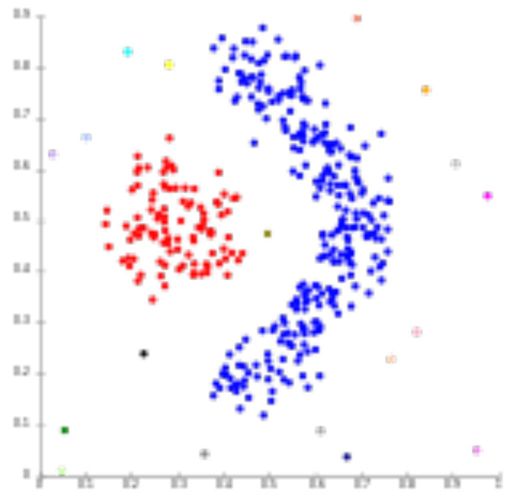
centroid models



distribution models



density models



connectivity

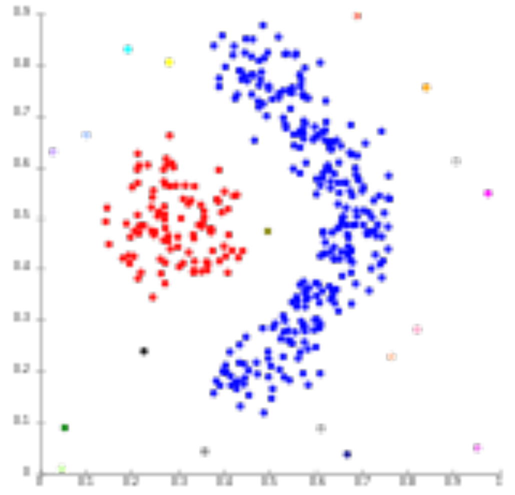
centroid

distribution

density

Connectivity based clustering (also known as hierarchical clustering) is based on the core idea of objects being more related to nearby objects than to objects farther away.

core idea: distance



What constitutes a connectivity clustering algorithm:

1. distance function: distance between two objects (Ex: Euclidean distance, Manhattan distance …)

2. the linkage criterion: distance between clusters

   1. single-linkage clustering (the minimum of object distances)

   2. complete linkage clustering (the maximum of object distances)

   3. average linkage clustering (the average of object distances)

3. agglomerative (starting with single elements and aggregate them into clusters)

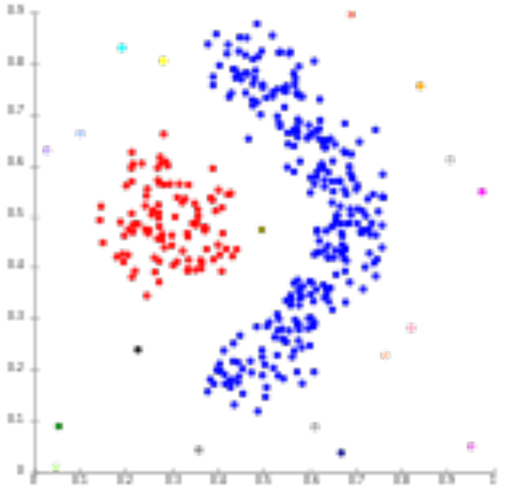
   divisive (starting with the complete data set and dividing them into partitions)

connectivity

centroid

distribution

density



Example: single linkage, agglomerative

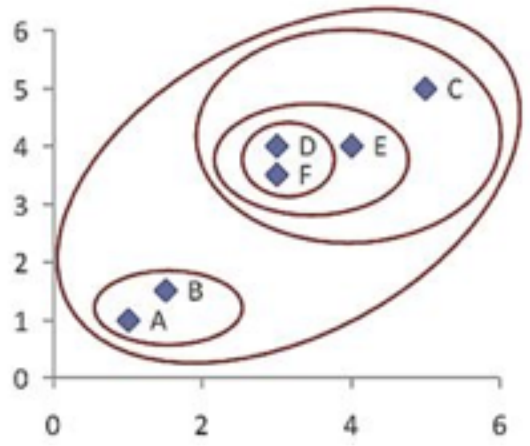
1. Start by assigning each item to its own cluster
2. Find the closest pair of clusters and merge them into a single cluster
3. Compute the distance between the new cluster and old clusters
4. Repeat 2 and 3 until all items are clustered into a single cluster of size N

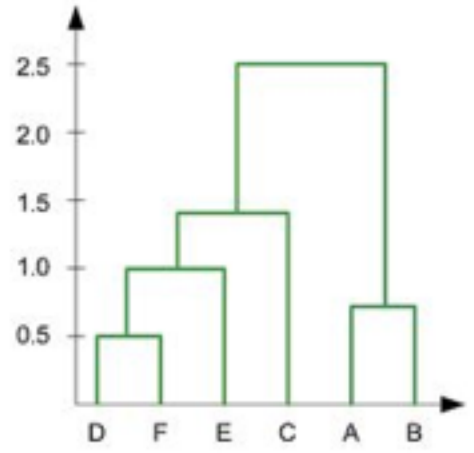
connectivity

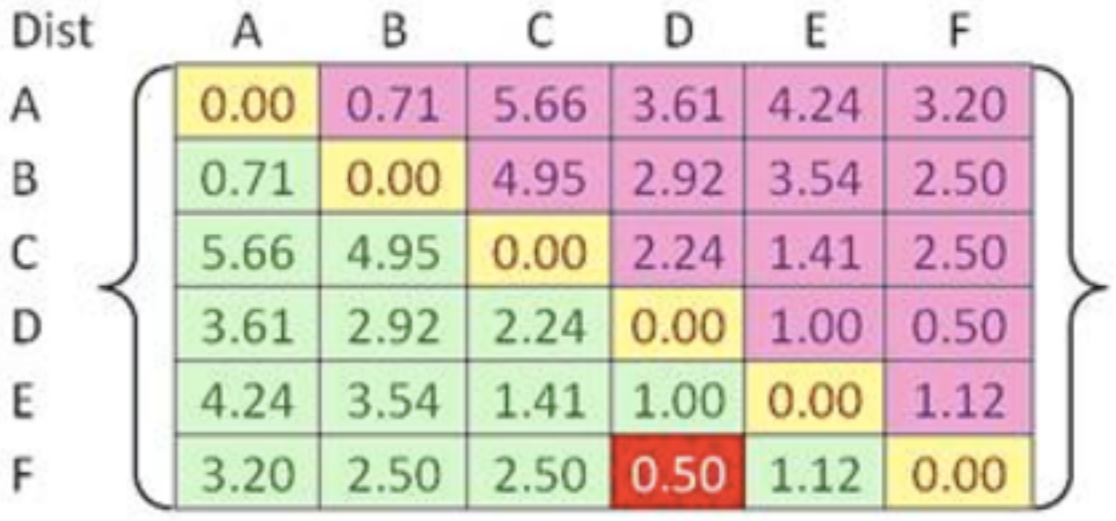
centroid

distribution

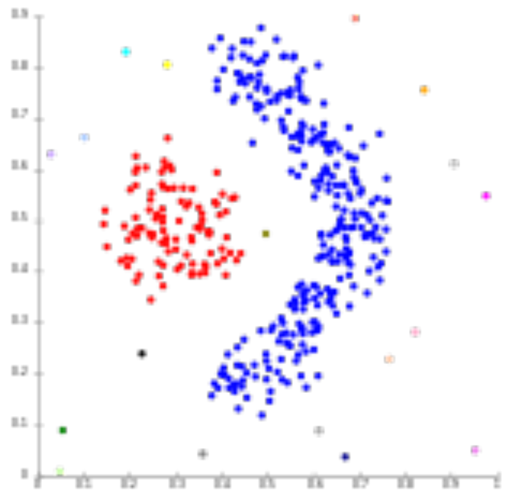
density







Dist	A	B	C	D	E	F
A	0.00	0.71	5.66	3.61	4.24	3.20
B	0.71	0.00	4.95	2.92	3.54	2.50
C	5.66	4.95	0.00	2.24	1.41	2.50
D	3.61	2.92	2.24	0.00	1.00	0.50
E	4.24	3.54	1.41	1.00	0.00	1.12
F	3.20	2.50	2.50	0.50	1.12	0.00



connectivity

centroid

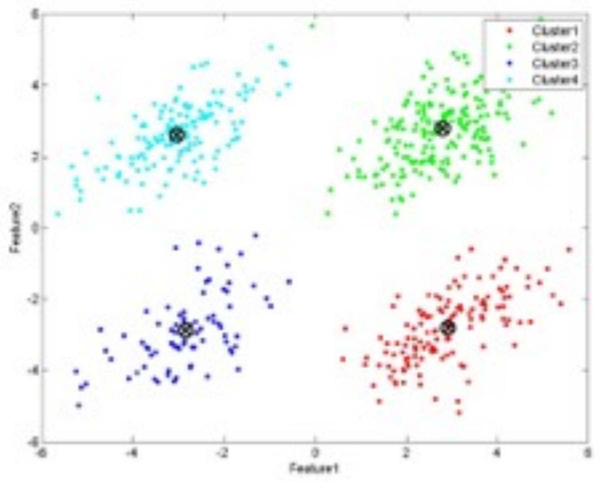
distribution

density

Remark:

- Not robust towards outliers, which will either show up as additional clusters or even cause other clusters to merge. (does not have a notion of "noise")
- Too slow for large data set.

In the data mining community, these methods are recognised as a theoretical foundation of cluster analysis, but often considered obsolete.



connectivity

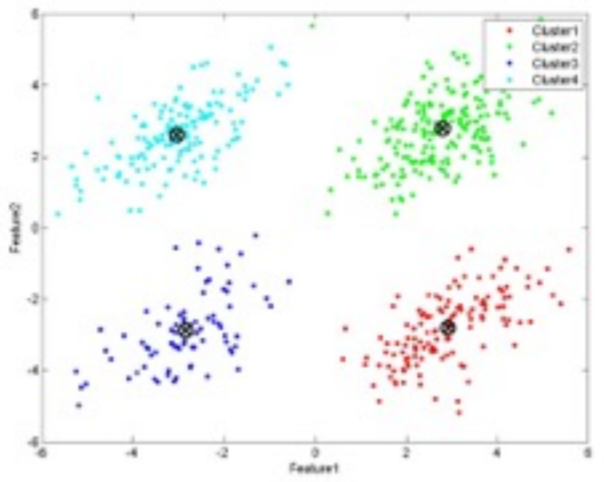
centroid

distribution

density

Clusters are represented by a central vector, which may not necessarily be a member of the data set.

The similarity of two clusters is defined as the similarity of their centroids



Example: k-mean (the number of clusters is fixed to k)

Given a set of observations (x1, x2, …, xn), partition the observations into k sets S = {S1, S2, …, Sk} so as to minimise the within-cluster sum of squares (**WCSS**).

connectivity

centroid

distribution

density

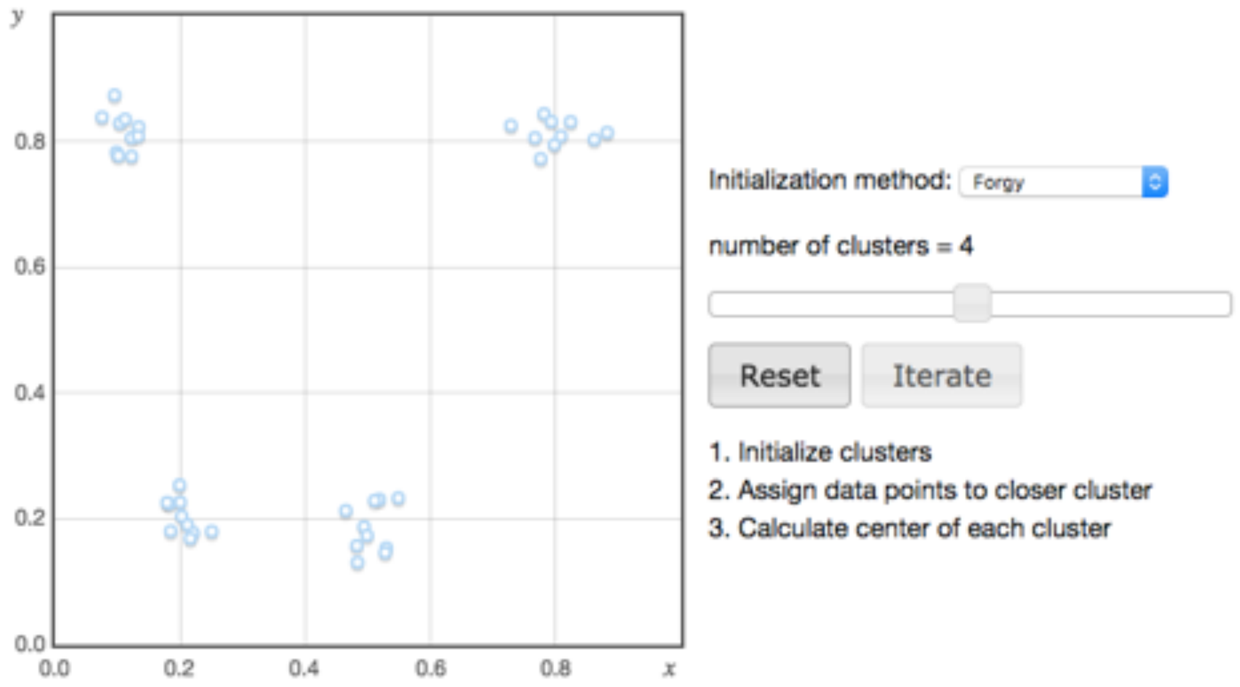
$$\underset{\mathbf{S}}{\arg\min} \sum_{i=1}^{k} \sum_{\mathbf{x} \in S_i} \left\| \mathbf{x} - \boldsymbol{\mu}_i \right\|^2$$

where μi is the mean of points in Si.

Algorithm (Lloyd's algorithm)

1. Initialize the center of the clusters	$\mu_i =$ some value $, i = 1, \ldots, k$
2. Attribute the closest cluster to each data point	$\mathbf{c}_i = \{j : d(\mathbf{x}_j, \mu_i) \leq d(\mathbf{x}_j, \mu_l), l \neq i, j = 1, \ldots, n\}$
3. Set the position of each cluster to the mean of all data points belonging to that cluster	$\mu_i = \frac{1}{\|c_i\|} \sum_{j \in c_i} \mathbf{x}_j, \forall i$
4. Repeat steps 2-3 until convergence	

Good example



Initialization method: Forgy

number of clusters = 4

1. Initialize clusters
2. Assign data points to closer cluster
3. Calculate center of each cluster

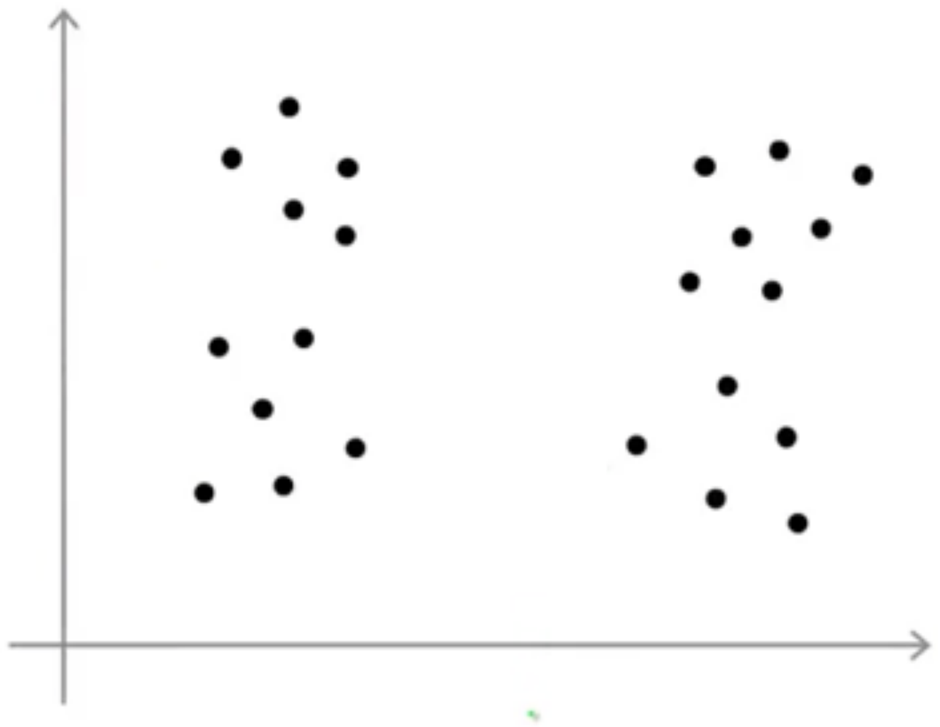
http://www.onmyphd.com/?p=k-means.clustering

Issues:

1. Deciding k

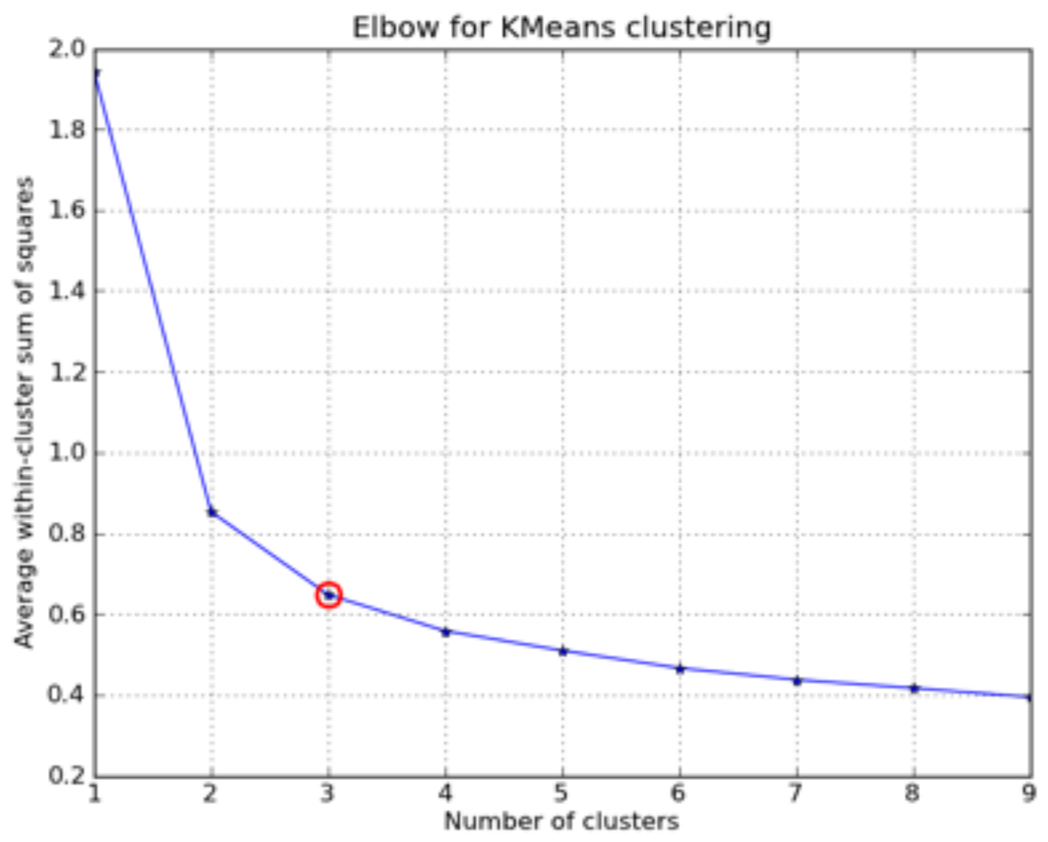
2. Initialisation

It is often generally ambiguous how many clusters there are in the data.



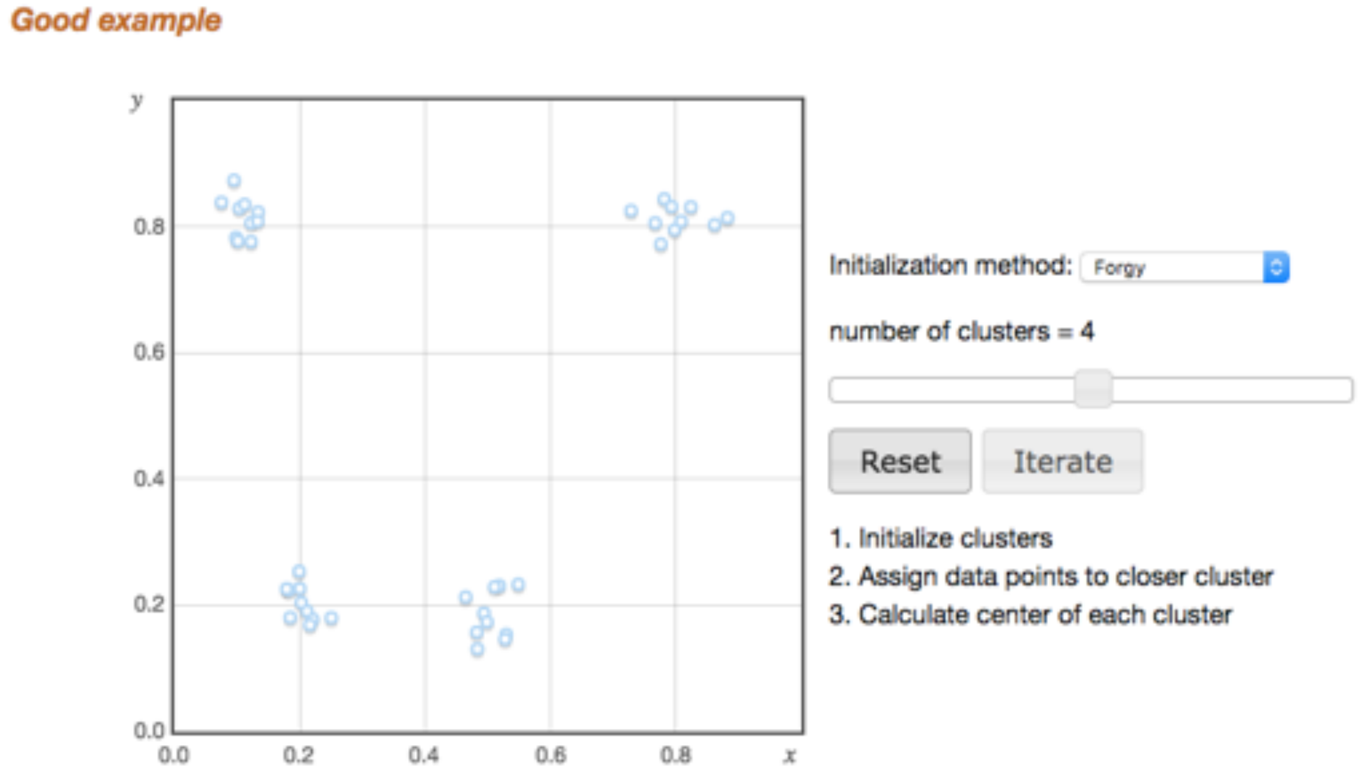
Most common way: manually

The elbow method: Try K-means clustering with different k and measure the resulting sum of squares.



Forgy: set the positions of the k clusters to k observations chosen randomly from the dataset.

Random partition: assign a cluster randomly to each observation and compute means.



Initialization method: Forgy

number of clusters = 4

1. Initialize clusters
2. Assign data points to closer cluster
3. Calculate center of each cluster

http://www.onmyphd.com/?p=k-means.clustering

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

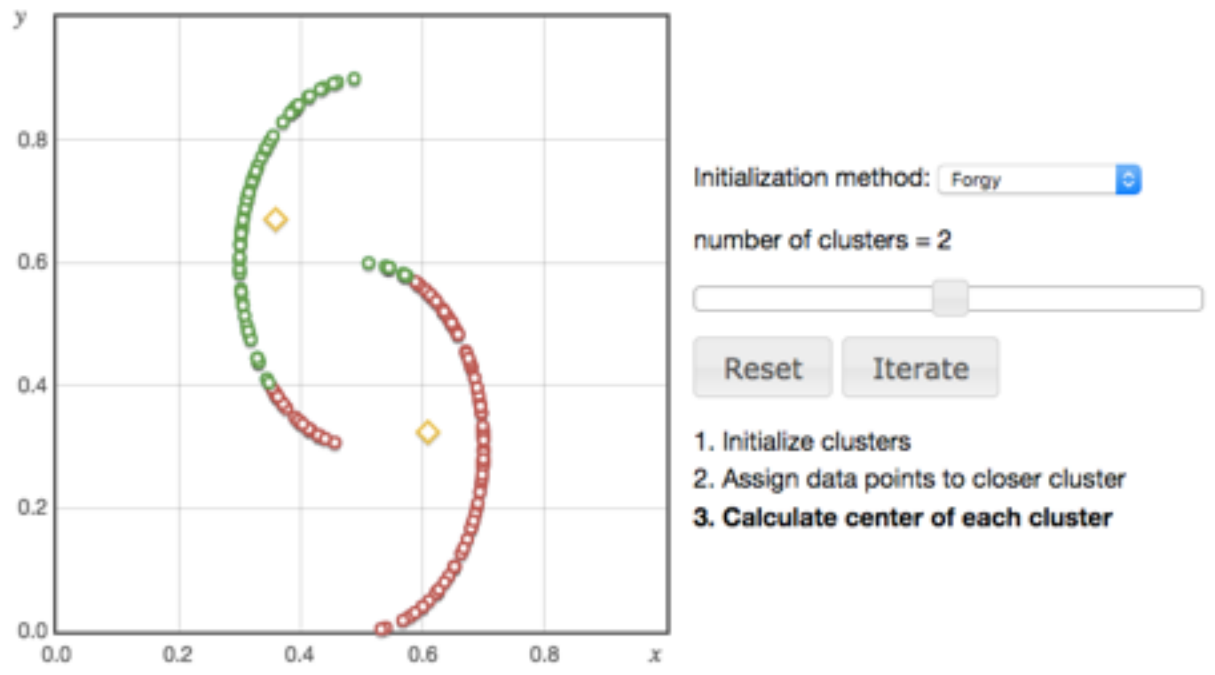
Remark:

- The algorithm only finds a local optimum. Typically run multiple times with different random initialisations.
- Optimises cluster centres, not cluster borders, which often leads to incorrectly cut borders in between clusters.

The k-means algorithm works reasonably well when the data fits the cluster model:

- The clusters are spherical (the data points in a cluster are centered around that cluster)
- The spread/variance of the clusters is similar (Each data point belongs to the closest cluster)

If any one these principles does not hold, the result will be counter-intuitive:

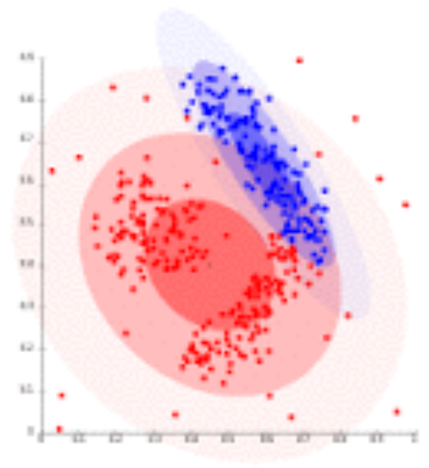


Initialization method: Forgy

number of clusters = 2

1. Initialize clusters
2. Assign data points to closer cluster
3. **Calculate center of each cluster**

http://www.onmyphd.com/?p=k-means.clustering



connectivity

centroid

distribution

density

Clusters are defined as objects belonging most likely to the same distribution.

Distribution based models suffer from one key problem known as overfitting, unless constraints are put on the model complexity.

Gaussian mixture models (GMM)

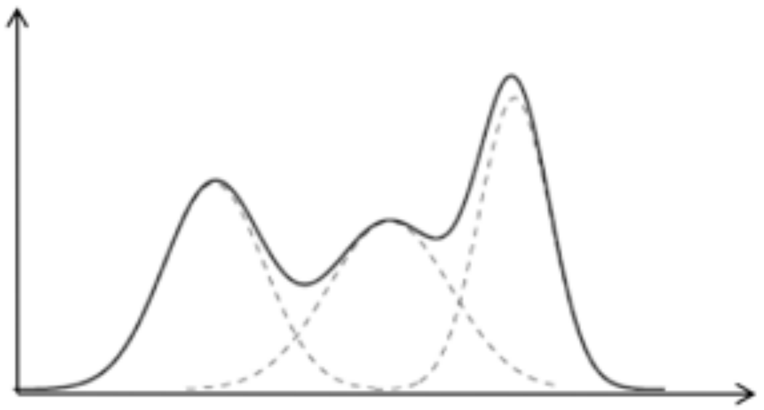
The probability given in a mixture of *K* Gaussians is:

$$p(x) = \sum_{j=1}^{K} w_j \cdot N(x \mid \mu_j, \Sigma_j)$$

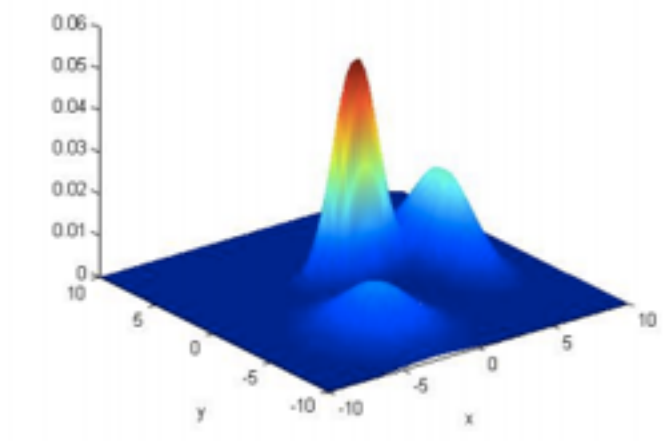
where w_j is the prior probability (weight) of the jth Gaussian.

$$\sum_{j=1}^{K} w_j = 1 \quad \text{and} \quad 0 \le w_j \le 1$$

d=1:



d=2:



- **Problem:**

  Given a set of data $X = \{x_1, x_2, ..., x_N\}$ drawn from an unknown distribution (probably a GMM), estimate the parameters θ of the GMM model that fits the data.

- **Solution:**

  Maximize the likelihood $p(X \mid \theta)$ of the data with regard to the model parameters?

$$\boxed{\theta^* = \arg\max_{\theta} p(X \mid \theta) = \arg\max_{\theta} \prod_{i=1}^{N} p(x_i \mid \theta)}$$

Expectation - Maximization algorithm

An iterative method to find maximum likelihood or maximum a posteriori (MAP) estimates of parameters in statistical models.

The EM iteration alternates between:

Expectation (E) step: creates a likelihood function using the current parameters

Maximization (M) step: computes parameters maximizing the likelihood function from the E step.

EM for GMM

Expectation (E) Step

Calculate $\forall i, k$:

$$\hat{\gamma}_{ik} = \frac{\hat{\phi}_k \mathcal{N}(x_i \mid \hat{\mu}_k, \hat{\sigma}_k}{\sum_{j=1}^{K} \hat{\phi}_j \mathcal{N}(x_i \mid \hat{\mu}_j, \hat{\sigma}_j)})$$

$\hat{\gamma}_{ik}$ is the probability that x_i is generated by component C_k. Thus, $\hat{\gamma}_{ik} = p(C_k | x_i, \hat{\phi}, \hat{\mu}, \hat{\sigma})$

Maximization (M) Step

Using the $\hat{\gamma}_{ik}$ calculated in the Expectation step, calculate in the following order $\forall k$:

$$\hat{\phi}_k = \sum_{i=1}^{N} \frac{\hat{\gamma}_{ik}}{N}$$

$$\hat{\mu}_k = \frac{\sum_{i=1}^{N} \hat{\gamma}_{ik} x_i}{\sum_{i=1}^{N} \hat{\gamma}_{ik}}$$

$$\hat{\sigma}_k = \frac{\sum_{i=1}^{N} \hat{\gamma}_{ik} (x_i - \hat{\mu}_k)^2}{\sum_{i=1}^{N} \hat{\gamma}_{ik}}$$

The probability given in a mixture of *K* Gaussians is:

$$p(x) = \sum_{j=1}^{K} w_j \cdot N(x \mid \mu_j, \Sigma_j)$$

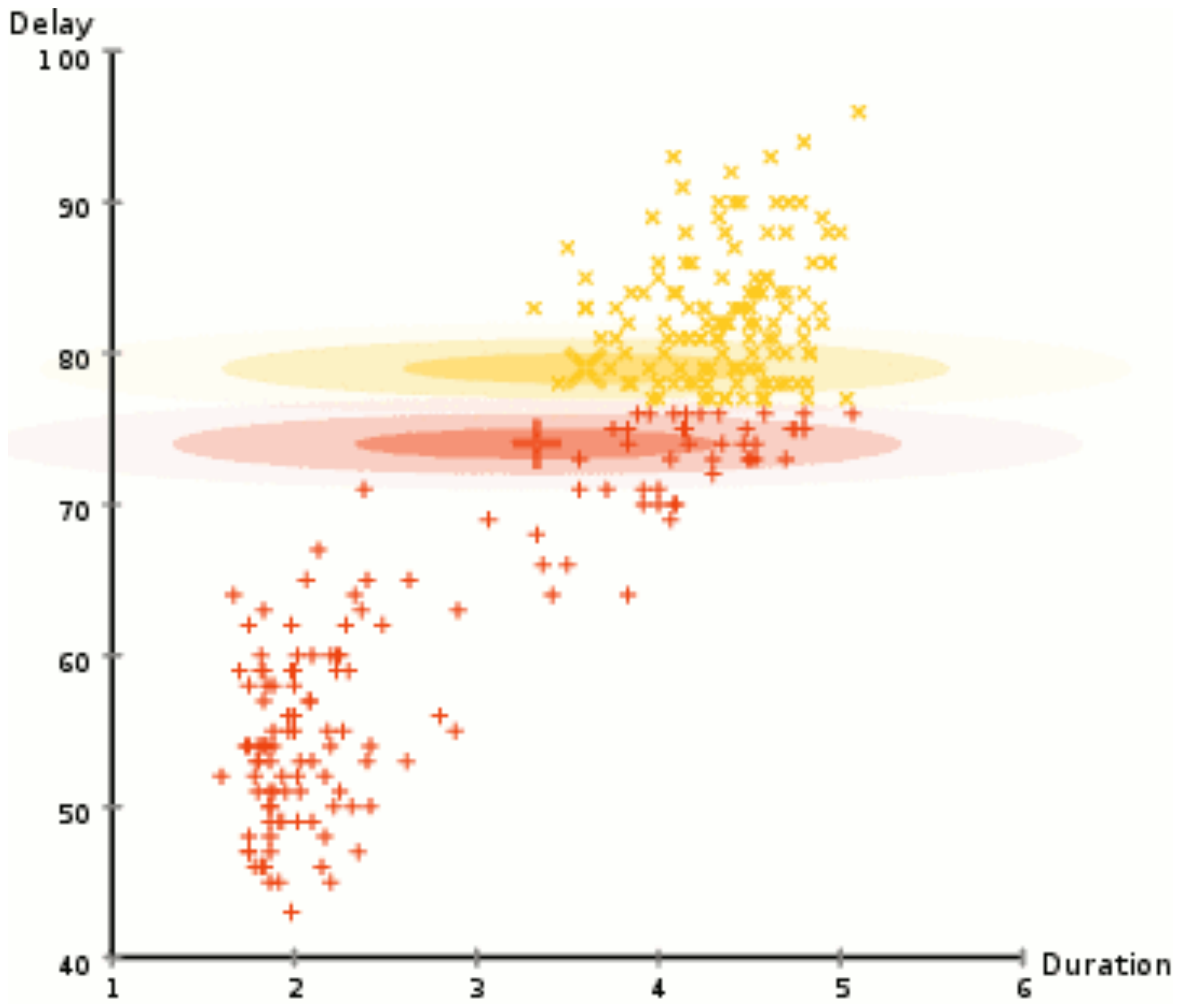
where w_j is the prior probability (weight) of the *j*th Gaussian.

$$\sum_{j=1}^{K} w_j = 1 \qquad \text{and} \qquad 0 \leq w_j \leq 1$$

.

Goal: maximize

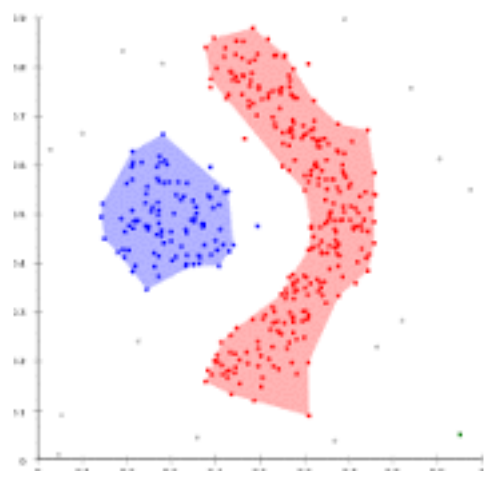
$$\ln p(\mathbf{X}|\boldsymbol{\mu}, \boldsymbol{\Sigma}, \boldsymbol{\pi}) = \sum_{n=1}^{N} \ln \left\{ \sum_{k=1}^{K} \underline{\pi_k} \mathcal{N}(\mathbf{x}_n | \underline{\boldsymbol{\mu}_k}, \underline{\boldsymbol{\Sigma}_k}) \right\}$$



The EM algorithm updating the parameters of a two component bivariate Gaussian mixture model.

Remark:

- Assuming gaussian distributions is a rather strong assumption on the data
- EM is very sensitive to initial conditions —> usually use the K-Means to get a good initialisation



connectivity

centroid

distribution

density

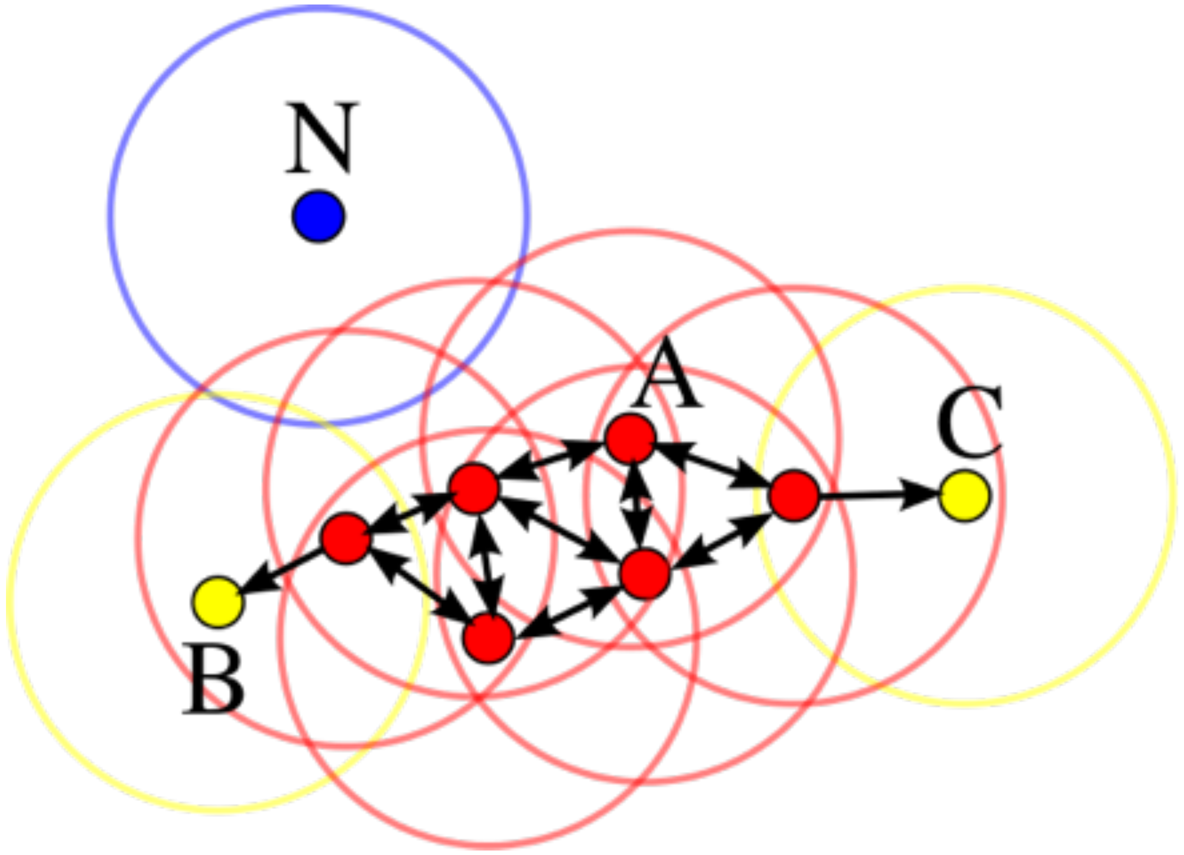
Clusters are defined as ares of higher density than the remainder of the data set.

Objects in the sparse areas are usually considered to be noise and border points.

Density-based spatial clustering of applications with noise (DBSCAN)

Connects points that satisfy a density criterion.

- core point: at least minPts points are within distance ε
- reachable: there is a path p1, ..., pn with p1 = p and pn = q, where each pi+1 is directly reachable from pi
- outliers: points not reachable from any other point

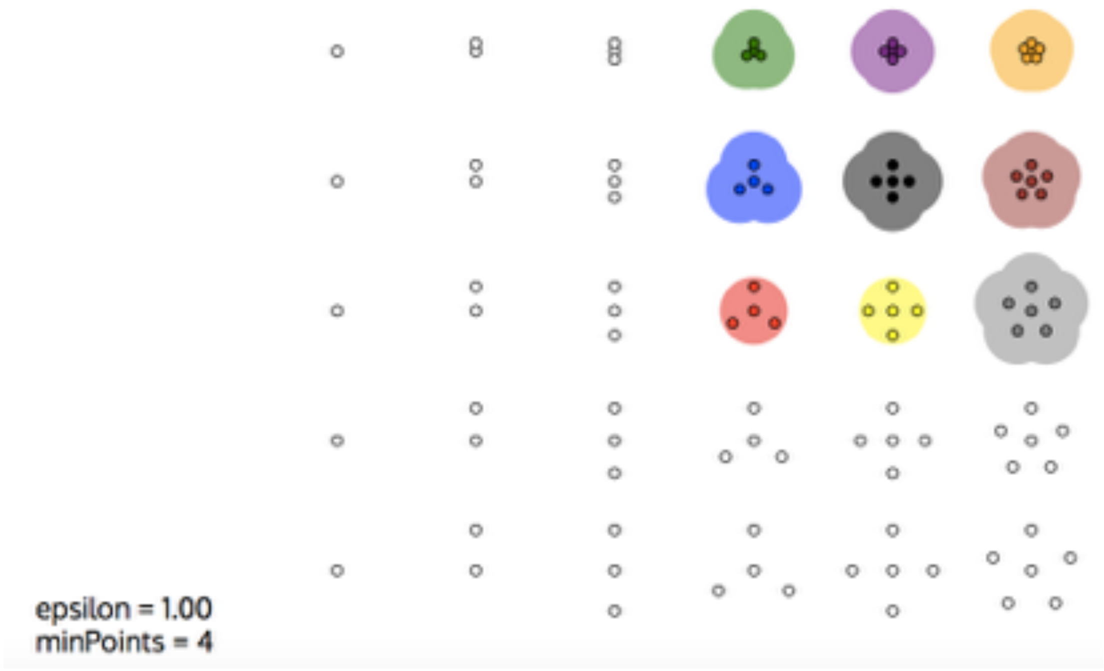


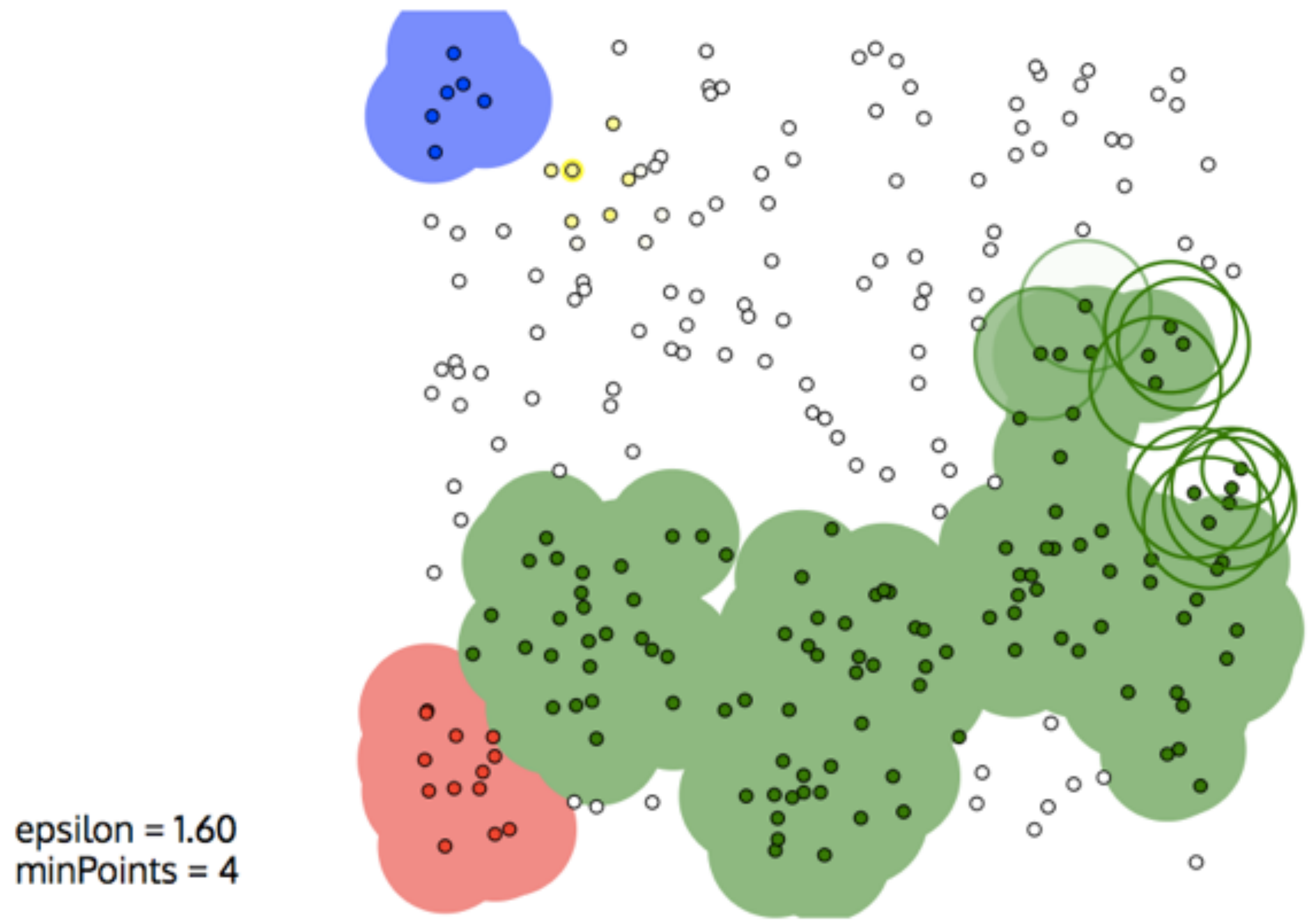
minPts = 4

```
DBSCAN(D, eps, MinPts) {
   C = 0
   for each point P in dataset D {
      if P is visited
         continue next point
      mark P as visited
      NeighborPts = regionQuery(P, eps)
      if sizeof(NeighborPts) < MinPts
         mark P as NOISE
      else {
         C = next cluster
         expandCluster(P, NeighborPts, C, eps, MinPts)
      }
   }
}

expandCluster(P, NeighborPts, C, eps, MinPts) {
   add P to cluster C
   for each point P' in NeighborPts {
      if P' is not visited {
         mark P' as visited
         NeighborPts' = regionQuery(P', eps)
         if sizeof(NeighborPts') >= MinPts
            NeighborPts = NeighborPts joined with NeighborPts'
      }
      if P' is not yet member of any cluster
         add P' to cluster C
   }
}

regionQuery(P, eps)
   return all points within P's eps-neighborhood (including P)
```





https://www.naftaliharris.com/blog/visualizing-dbscan-clustering/

Remarks

Advantages:

- The number of clusters is not required, as opposed to k-means.
- Can find arbitrarily shaped clusters.
- Has a notion of noise —> robust to outliers
- Discover essentially the same results in each run (no need to run it multiple times).

Disadvantages:

- Not entirely deterministic: border points that are reachable from more than one cluster can be part of either cluster, depending on the order the data is processed.

Clustering Evaluation

- internal evaluation: the clustering result is evaluated based on the data that was clustered itself. The best score is assigned to the algorithm that produces clusters with high similarity within a cluster and low similarity between clusters.

- external evaluation: the clustering is compared to an existing “ground truth” classification (in reality do not have this “ground truth” labels)

- manual evaluation: by a human expert

- indirect evaluation: evaluate the utility of the clustering in its intended application

Clustering — typical models

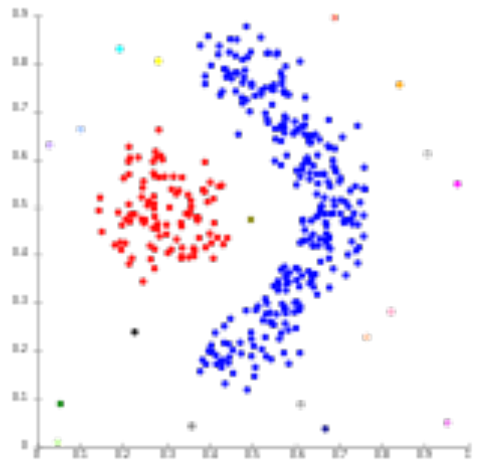
Depending on how a cluster is defined, there are different types of clustering models.

connectivity

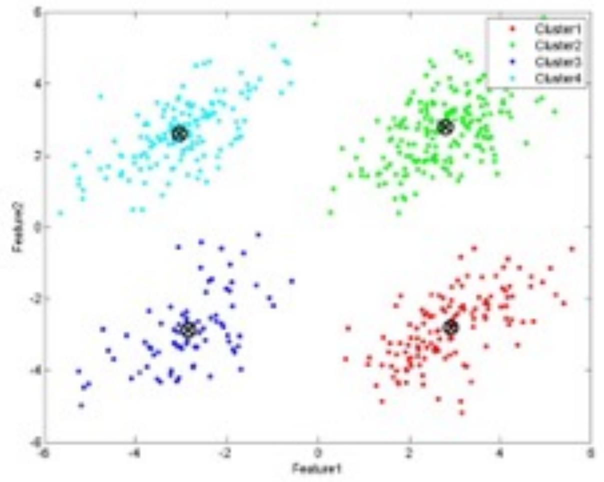
centroid

distribution

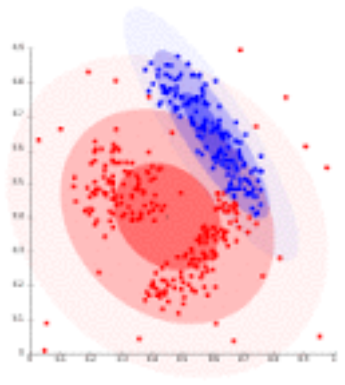
density



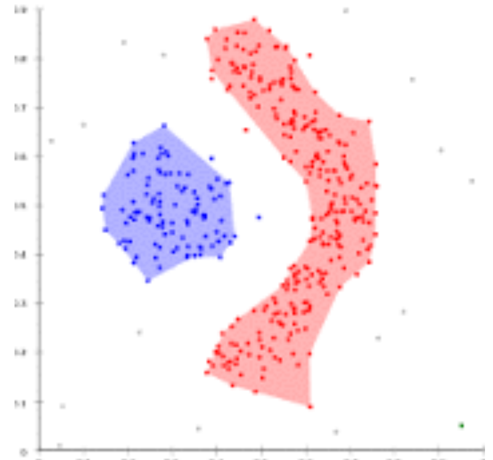
connectivity models



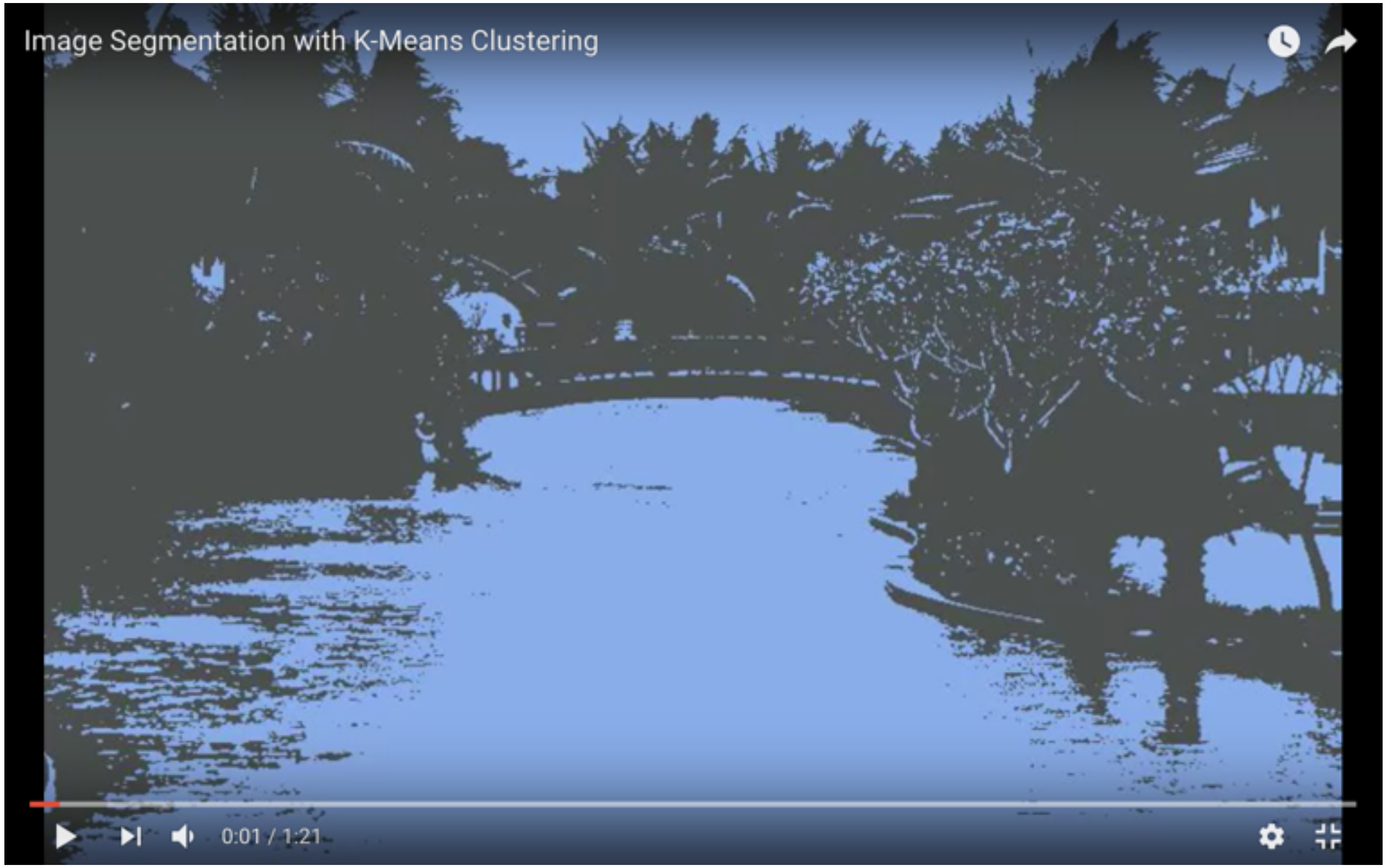
centroid models



distribution models



density models



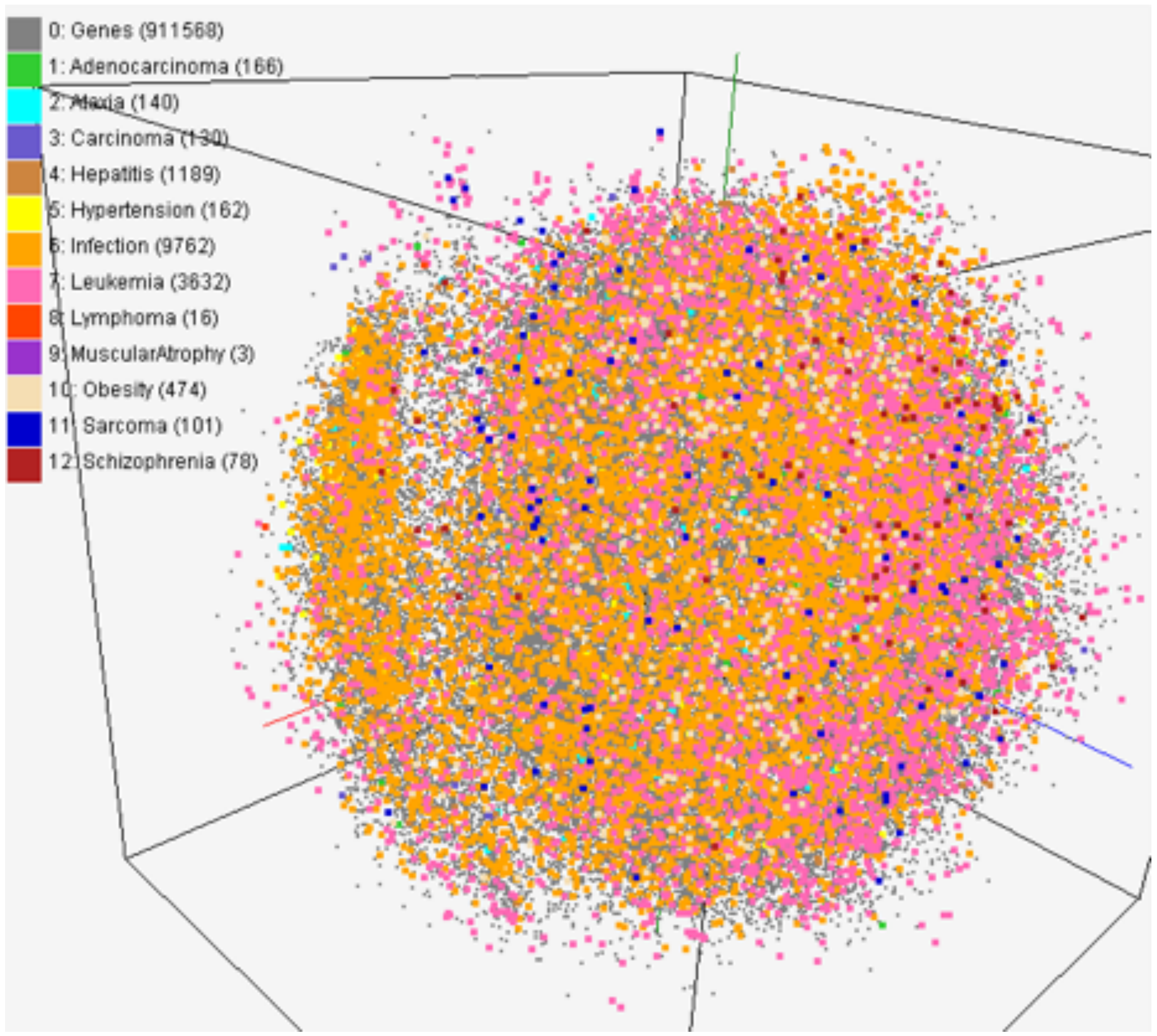
iterating from K=1 to K=80 clusters, with the last 3 frames being the original image.

https://www.youtube.com/watch?v=qstdP1DSUKQ

High-dimensional data

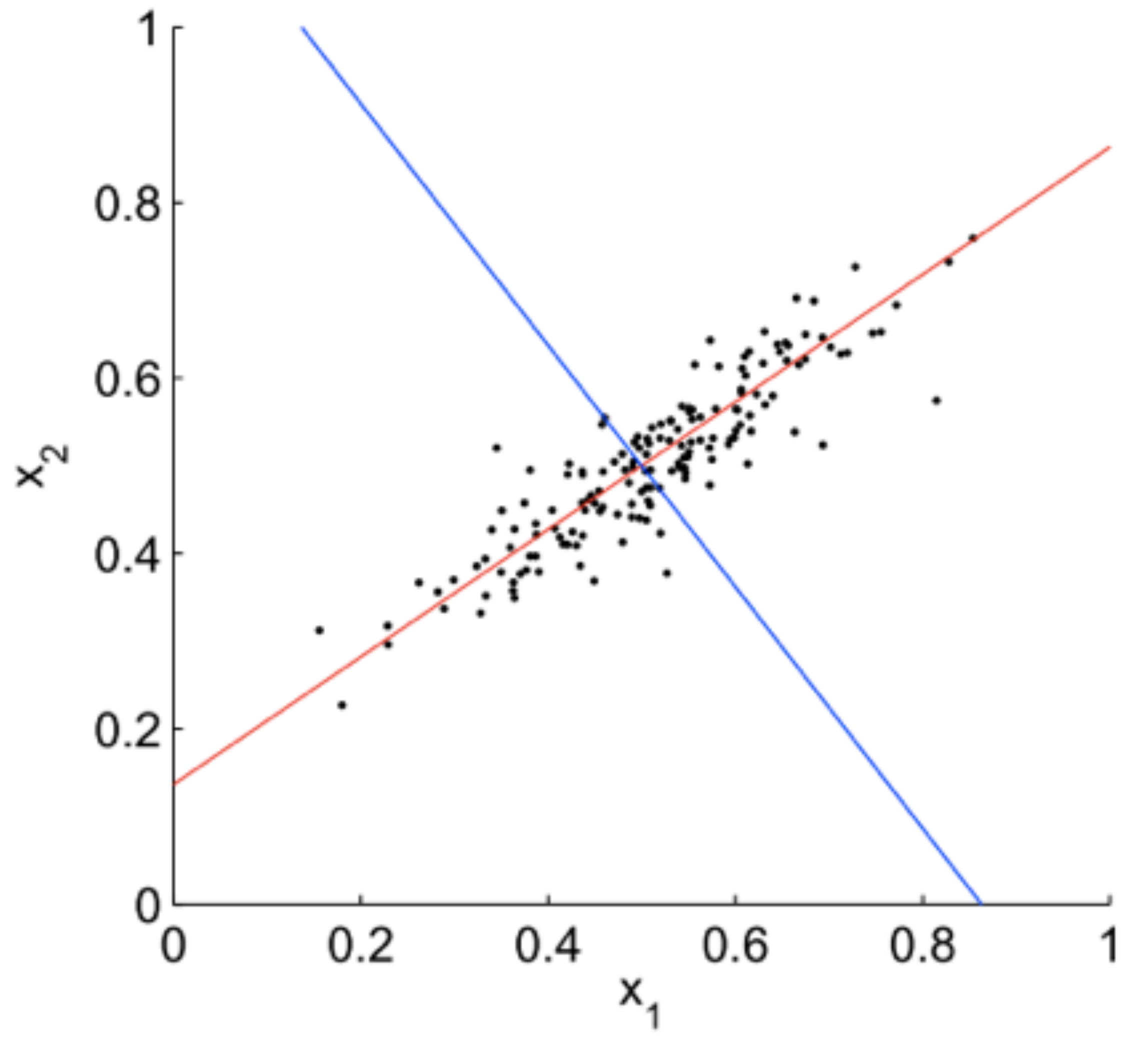
For example, Data on health status of patients can be high-dimensional

- blood analysis
- immune system status
- genetic background
- nutrition
- …



PCA

Many of these dimensions are, however, not important because they are highly correlated.



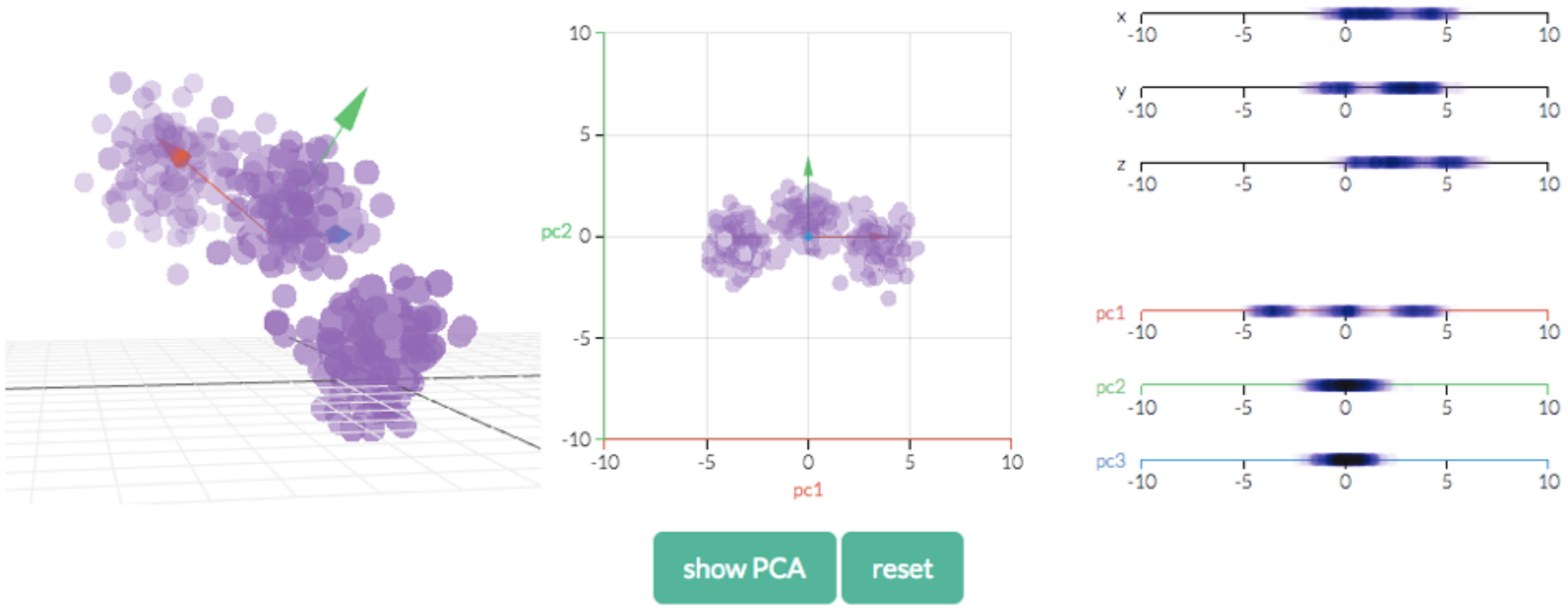
Reduce the dimension of a data set such that only "important dimensions" (red axis) are taken into account for further analysis.

PCA

Approach: d —> k dimensions

1. Compute the covariance matrix of the whole data set
2. compute eigenvectors and corresponding eigenvalues
3. choose k eigenvectors with the largest eigenvalues to form a d*k dimensional matrix (where every column represents an eigenvector)
4. Use this eigenvector matrix to transform the samples onto the new subspace.

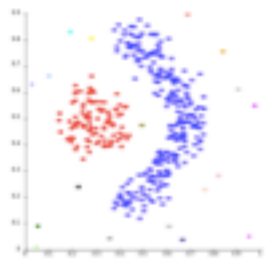
PCA — example

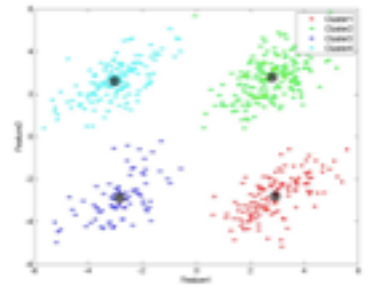


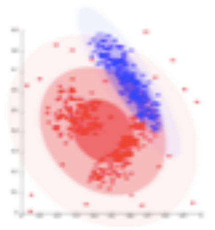
http://setosa.io/ev/principal-component-analysis/

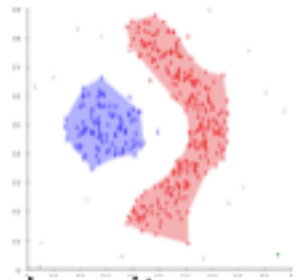
Outline

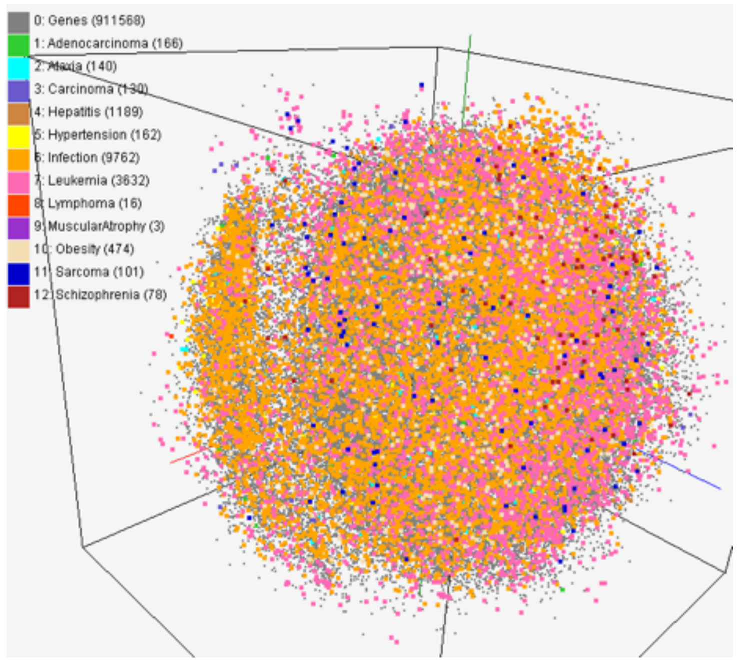
- Introduction
- Clustering
- Dimension reduction
- Extensions & Summary

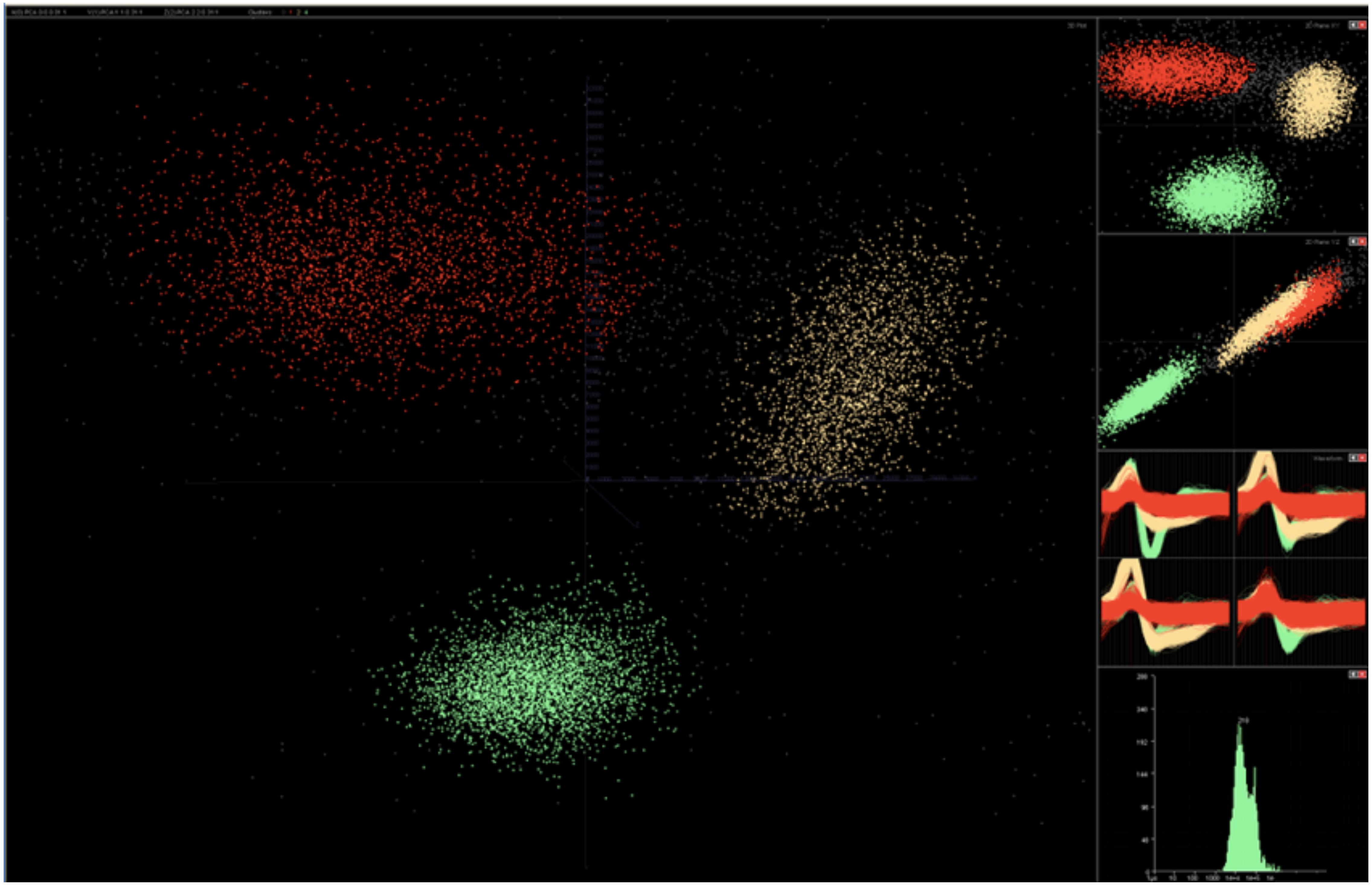

connectivity models


centroid models


distribution models

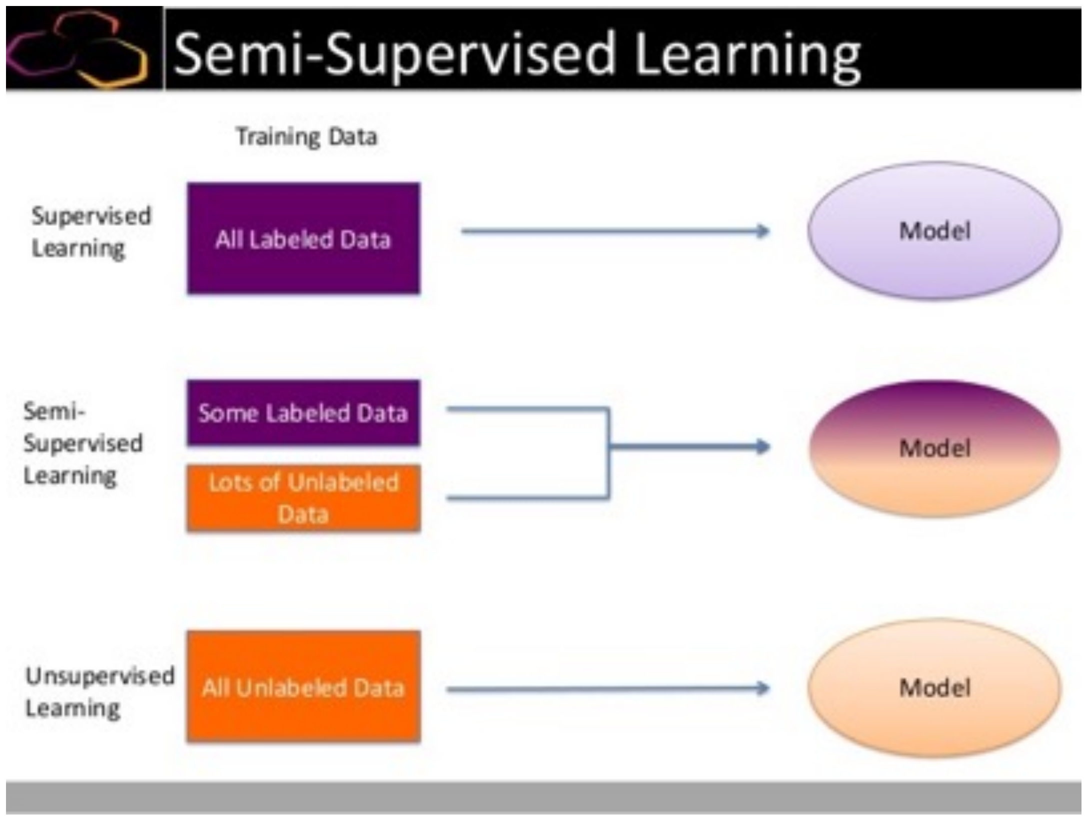

density models



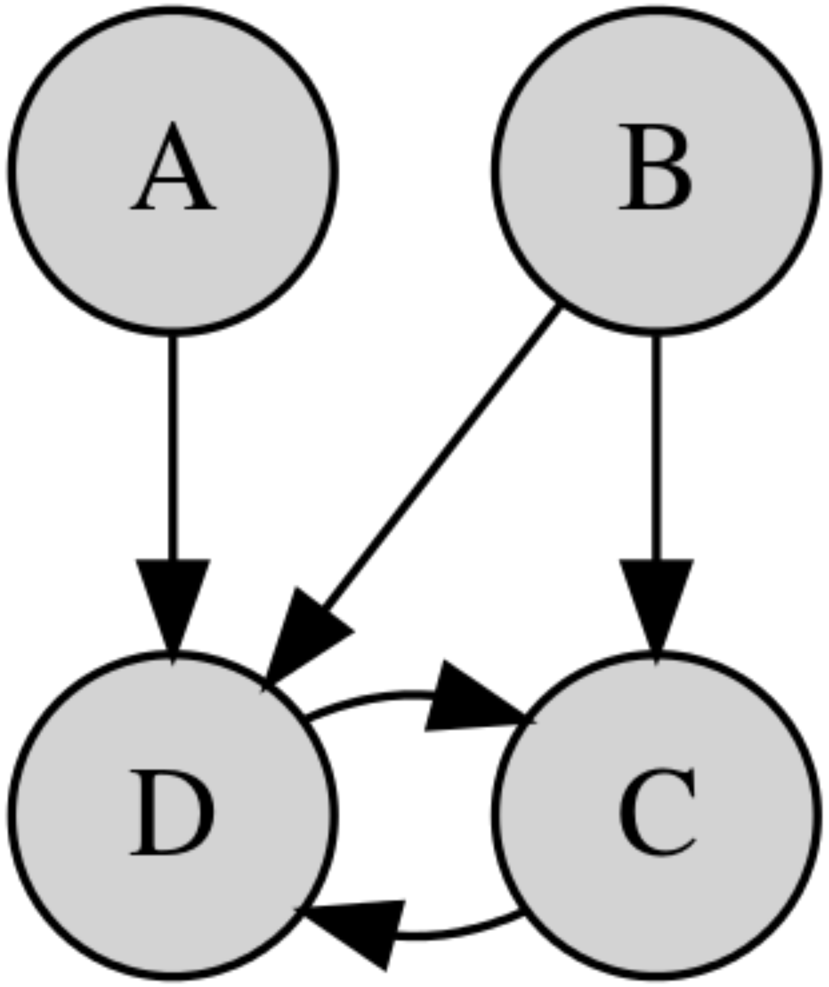


https://www.youtube.com/watch?v=IfNVv0A8QvI

Extensions


Semi-Supervised Learning
Training Data
Supervised Learning
All Labeled Data
Model
Semi-Supervised Learning
Some Labeled Data
Lots of Unlabeled Data
Model
Unsupervised Learning
All Unlabeled Data
Model

semi-supervised learning



$$\begin{tikzcd}
A \arrow[d] & B \arrow[dl] \arrow[d] \\
D \arrow[r, bend left] & C \arrow[l, bend left]
\end{tikzcd}$$

graphic models