Unsupervised Learning and Clustering

Lingqiao Liu
University of Adelaide

Outlines

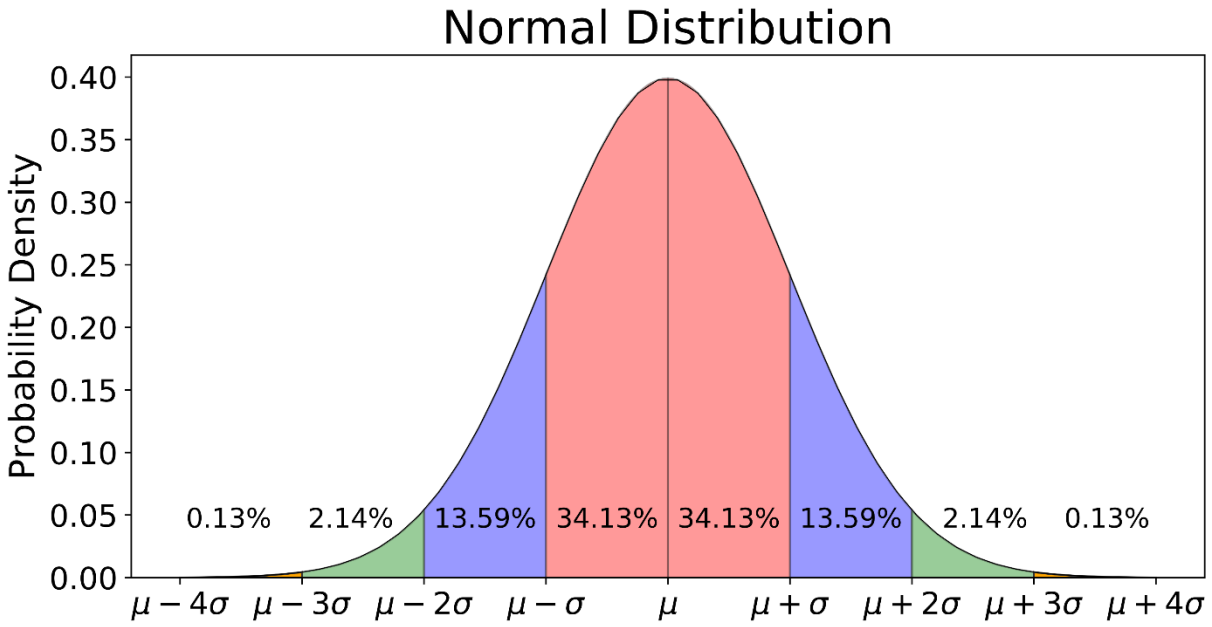
- Overview of Unsupervised Learning
- K-means clustering
- Gaussian mixture model
  - GMM
  - Latent variable
  - EM algorithm
- From GMM to other unsupervised learning approaches

Unsupervised learning

- Learning without supervision
  - Find the distribution of data
  - Learning to generate samples
  - Clustering
  - Anomaly detection
  - Feature learning

Distribution Modeling

- Distribution Modeling
  - Find a way to characterize the distribution of data
  - Modelling $P(x|\lambda)$
  - Example:



Distribution Modeling

- What is the form of $P(x|\lambda)$?
  - The unknow model parameters
- The function family
  - Based on assumptions
  - Based on prior knowledge about the data
  - Model it as a general function: parametric or nonparametric
- Once the function form is known, distribution modeling boils down to a parameter estimation problem
  - MLE: maximal likelihood estimation

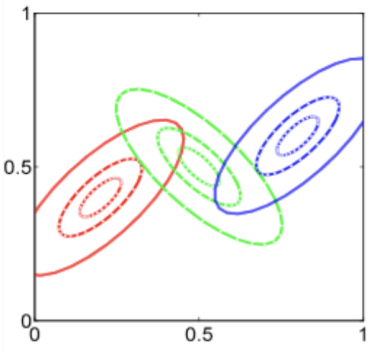
Commonly used assumption

- Multivariate Gaussian distribution

$$Pr(\mathbf{x}) = \mathcal{N}(\mathbf{x}|\mu_k, \Sigma_k) = \frac{\exp\left(-\frac{1}{2}(\mathbf{x}-\mu)^T\Sigma^{-1}(\mathbf{x}-\mu)\right)}{\sqrt{(2\pi)^k|\Sigma|}}$$

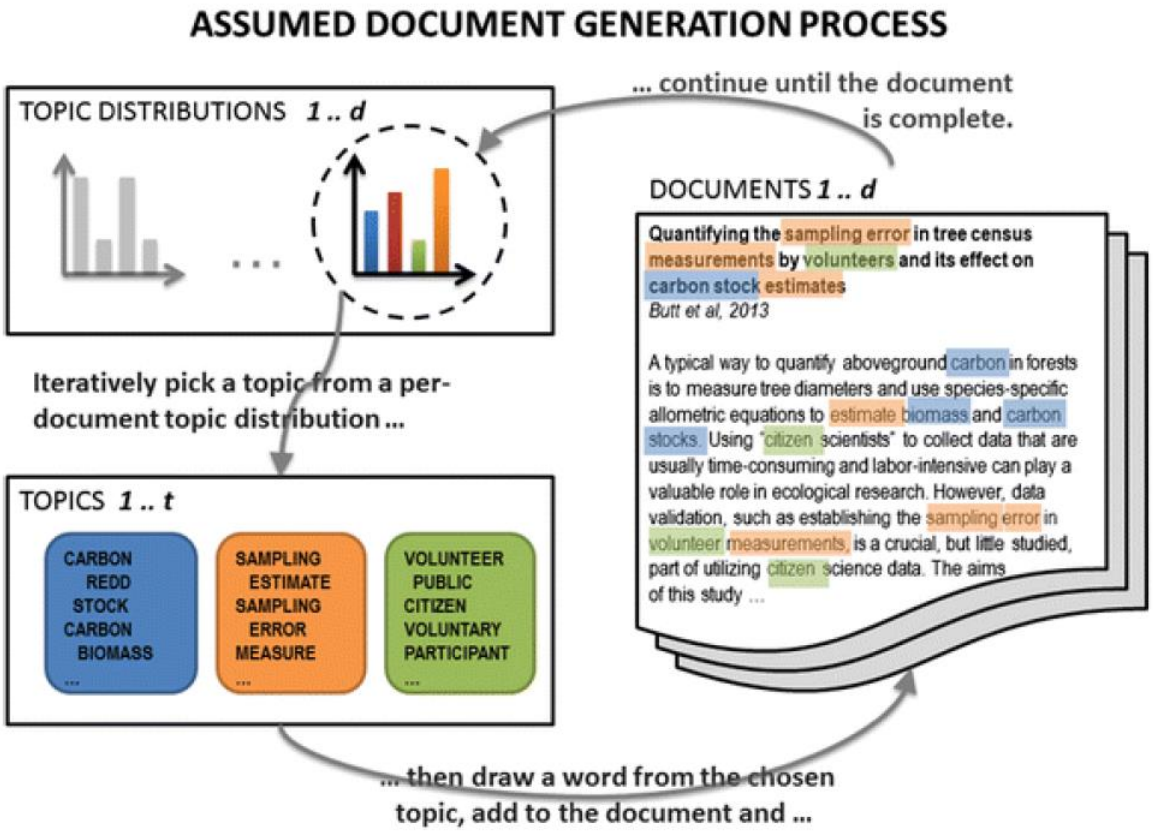
- Gaussian Mixture model

$$Pr(\mathbf{x}) = \sum_{k=1}^{K} \pi_k \mathcal{N}(\mathbf{x}|\mu_k, \Sigma_k)$$



Prior knowledge about the problem

- Topic model



General function

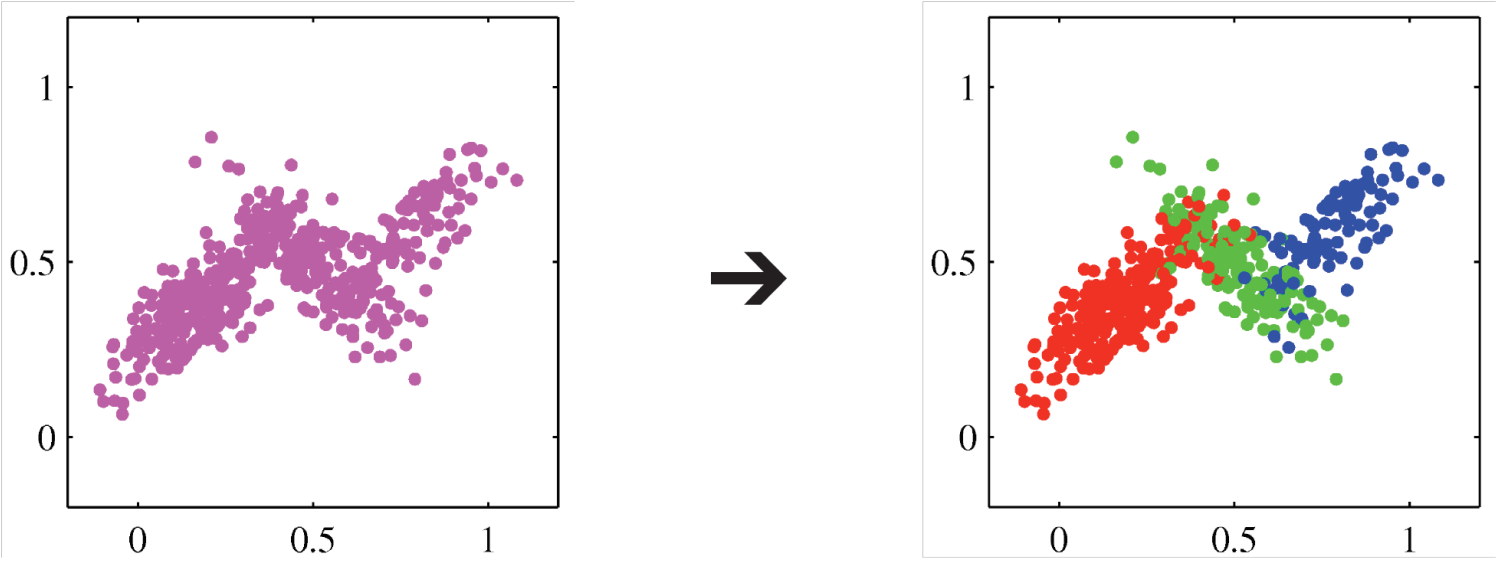
- Modeling PDF as a general function
  - Neural network, Kernel density estimation, Gaussian process
- Modeling the generative process

Unsupervised learning

- Focus of this lecture: Clustering
- Clustering:
  - Clustering is the process of identifying groups, or clusters, of data points in a (usually) multidimensional space.
  - Connection to distribution modeling: related to mixture model

Clustering

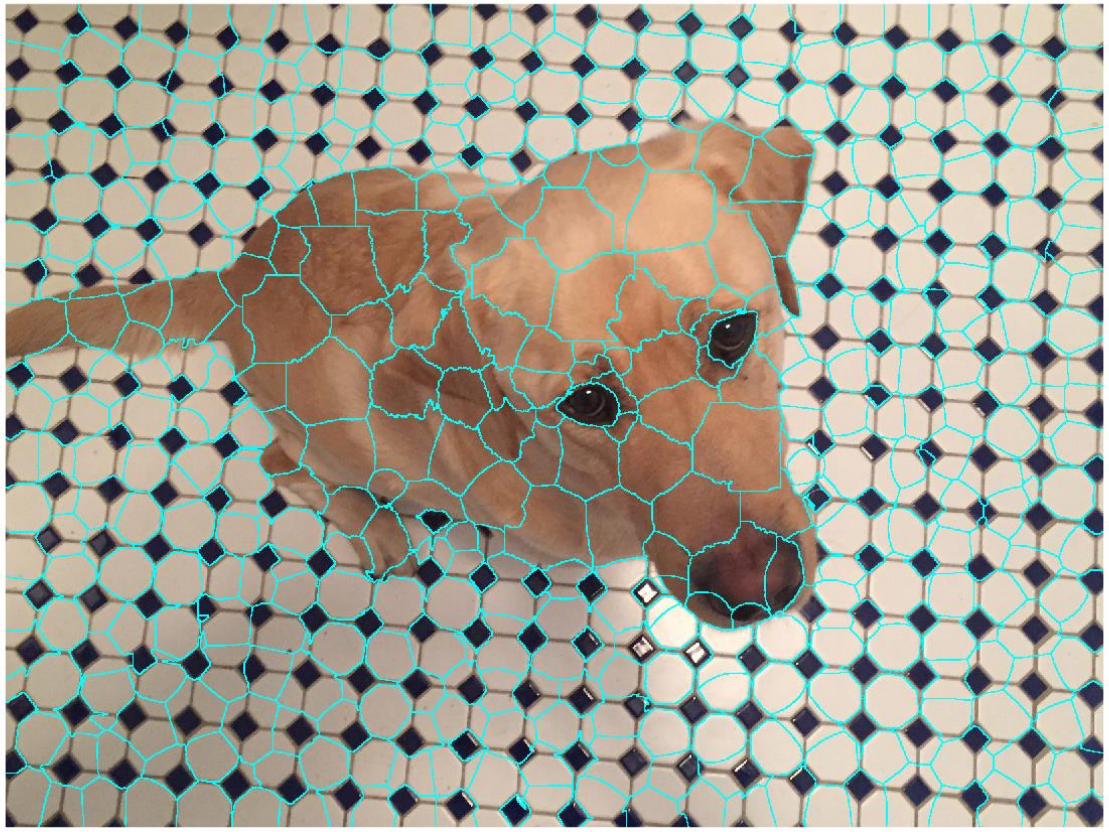
Discover groups such that samples within a group are more similar to each other than samples across groups.



Application of clustering: Segmentation

http://people.cs.uchicago.edu/ pff/segment

Superpixel

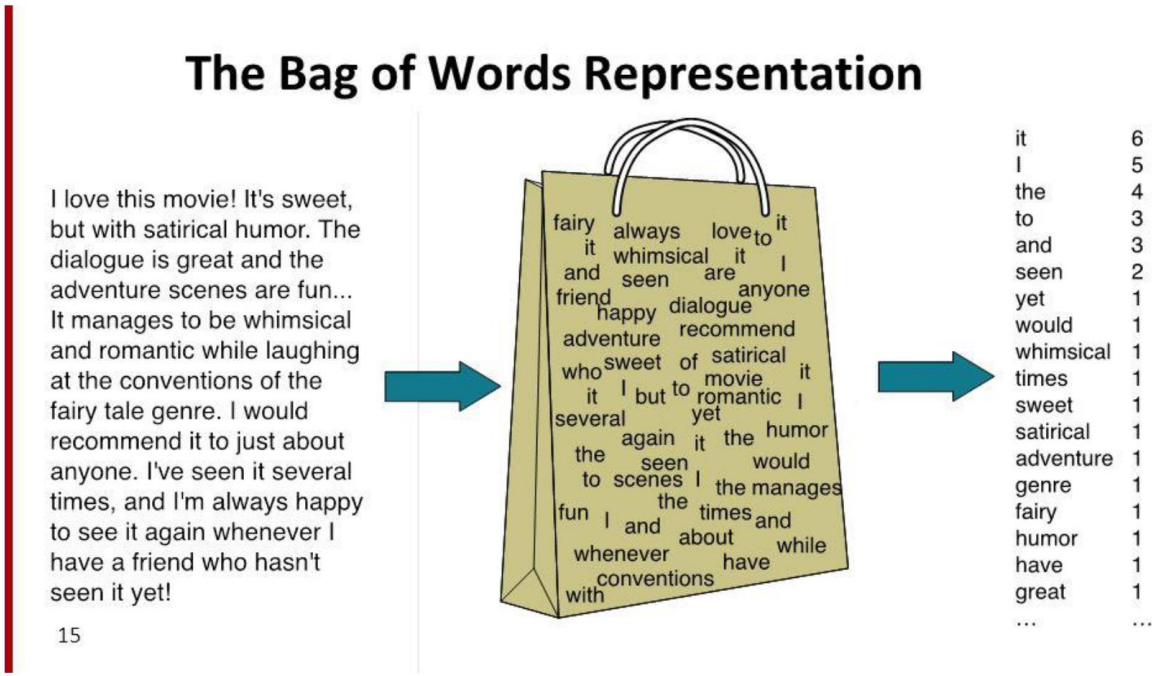


X = (r,g,b,x,y)

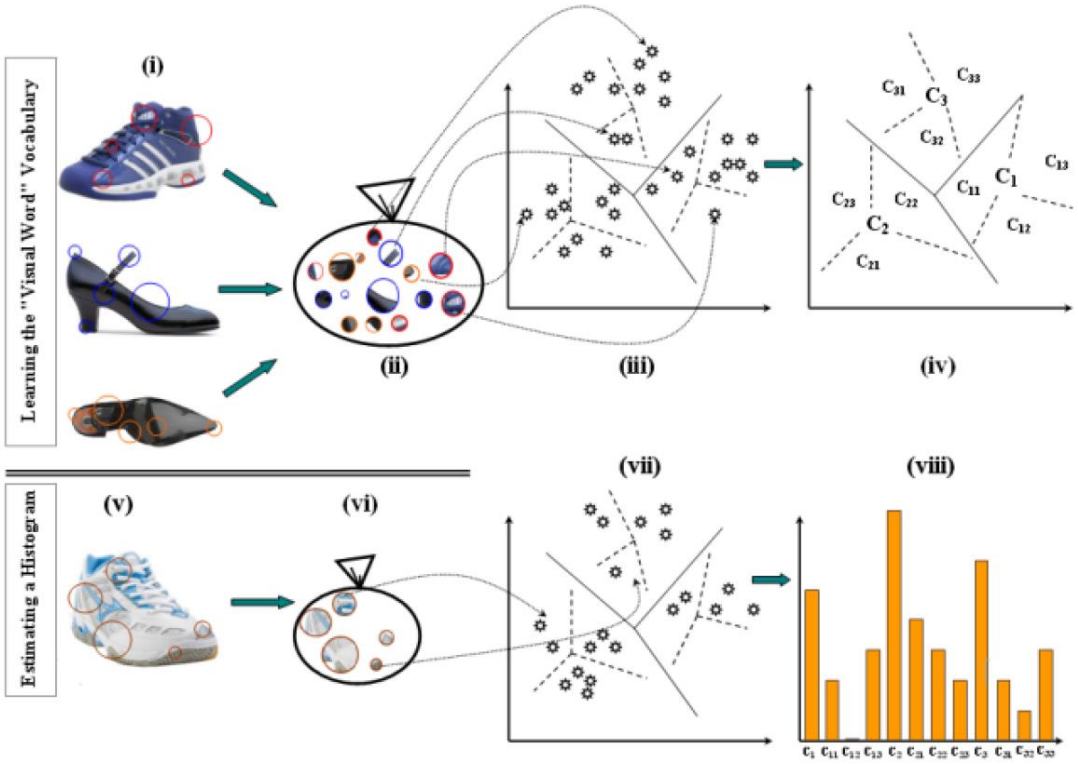
Application of clustering: Vector quantization in Bag-of-feature model

- Bag-of-features model
  - Extended from bag-of-words model

Yuo cna porbalby raed tihs esaliy desptie teh msispeillgns.



Application of clustering:



Vector Quantization: Build a dictionary with k centres.
For a vector, find its closet centre and use its ID as its “visual word”

Ingredient for Clustering

- A dissimilarity function between samples.
- A loss function to evaluate clusters.
- Algorithm that optimizes this loss function.

Dissimilar function

- Choice of dissimilarity function is application dependent.
- Need to consider the type of features.
  - Categorical, ordinal or quantitative.
- Possible to learn dissimilarity from data (later).

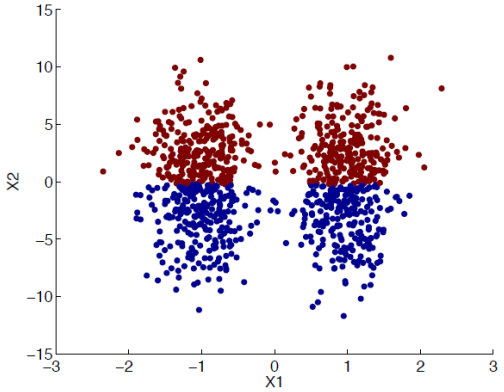
Dissimilar function

- Data point x_i has features $x_{ij}, j = 1, \ldots, p$.
- One choice of dissimilarity function is the Euclidean distance

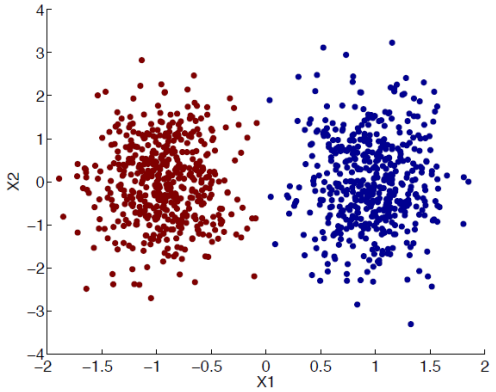
$$D(x_i, x_{i'}) = \sqrt{\sum_{j=1}^{p} (x_{ij} - x_{i'j})^2}$$

- Resulting clusters invariant to rotation and translation of features but not to scaling.
- If the features have different scales, standardize the data.

Standardization

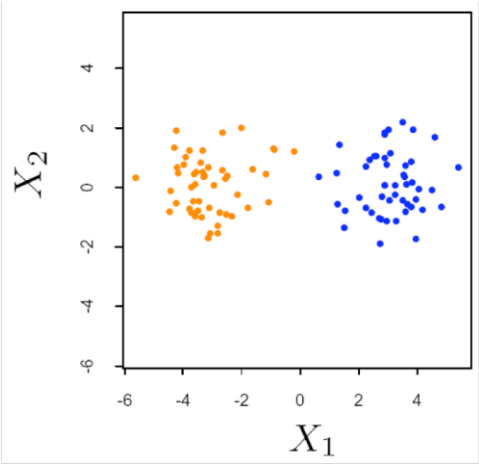


Without standardization

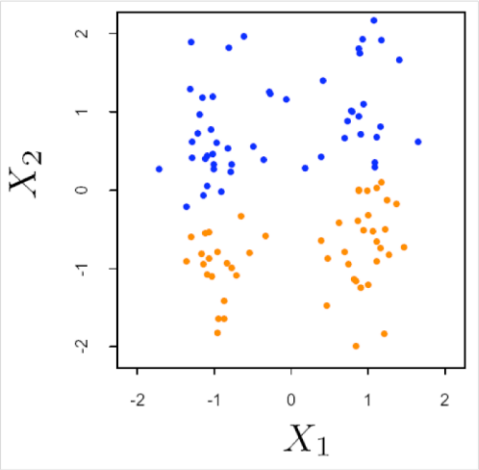


With standardization

Standardization is not always helpful



Without standardization



With standardization

K-means algorithm

- K clusters each summarized by a prototype μ_k.
- Assignment of data x_i to a cluster represented by responsibilities $r_{ik} \in \{0, 1\}$ with $\sum_{k=1}^{K} r_{ik} = 1$.
- An example with 4 data points and 3 clusters.

$$(r_{ik}) = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 0 & 1 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix}$$

- Loss function $J = \sum_{i=1}^{n} \sum_{k=1}^{K} r_{ik} \|x_i - \mu_k\|_2^2$.

K-means algorithm

- How do we minimize J w.r.t (r_{ik}, μ_k)?
- Chicken and egg problem
  - If prototypes known, can assign responsibilities.
  - If responsibilities known, can compute prototypes.

K-means algorithm

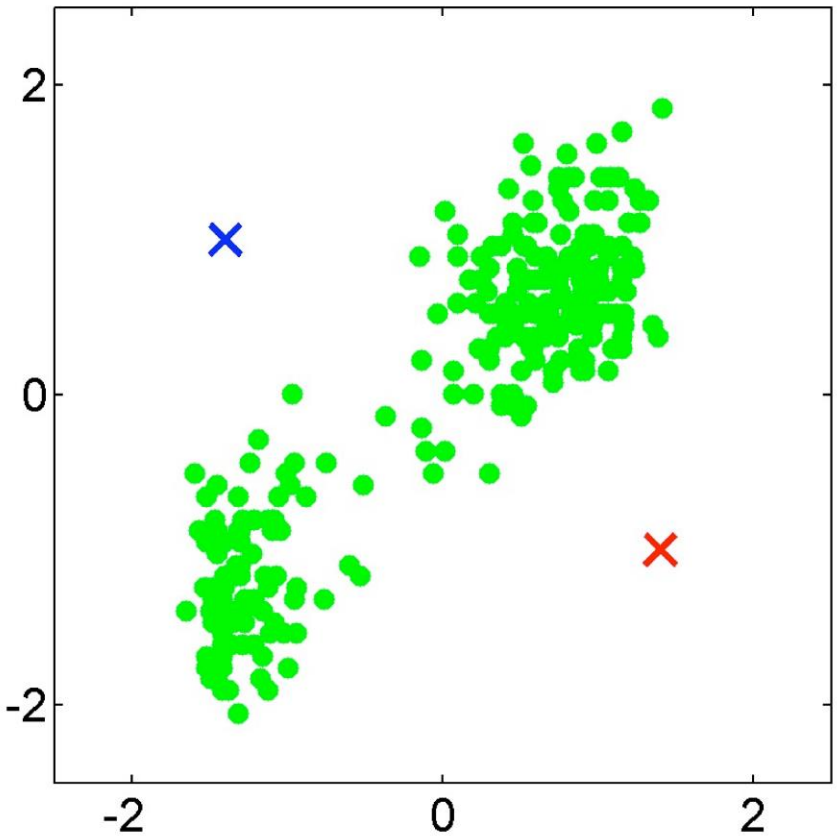
- How do we minimize J w.r.t (r_{ik}, μ_k)?
- Chicken and egg problem
  - If prototypes known, can assign responsibilities.
  - If responsibilities known, can compute prototypes.
- We use an iterative procedure.

K-means algorithm

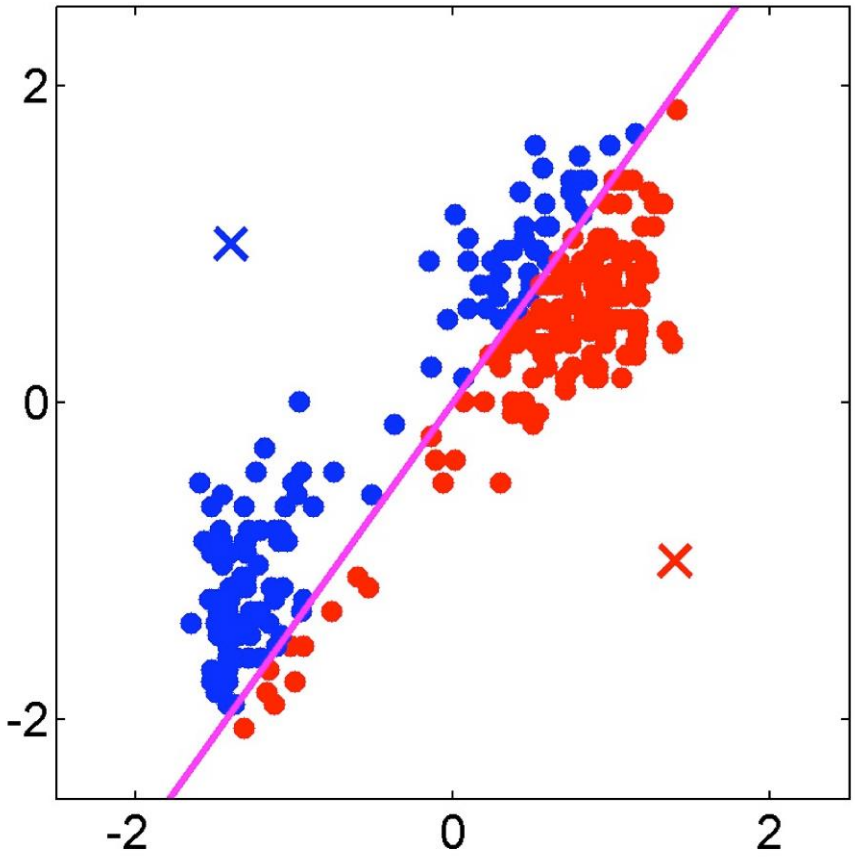
- E-step: Fix μ_k, minimize J w.r.t. r_{ik}.
  - Assign each data point to its nearest prototype.
- M-step: Fix r_{ik}, minimize J w.r.t. μ_k.
  - Set each prototype to the mean of the points in that cluster, i.e., $\mu_k = \frac{\sum_i r_{ik} x_i}{\sum_i r_{ik}}$.
- This procedure is guaranteed to converge.
- Converges to a local minimum.

- Loss function $J = \sum_{i=1}^{n} \sum_{k=1}^{K} r_{ik} \| x_i - \mu_k \|_2^2$.

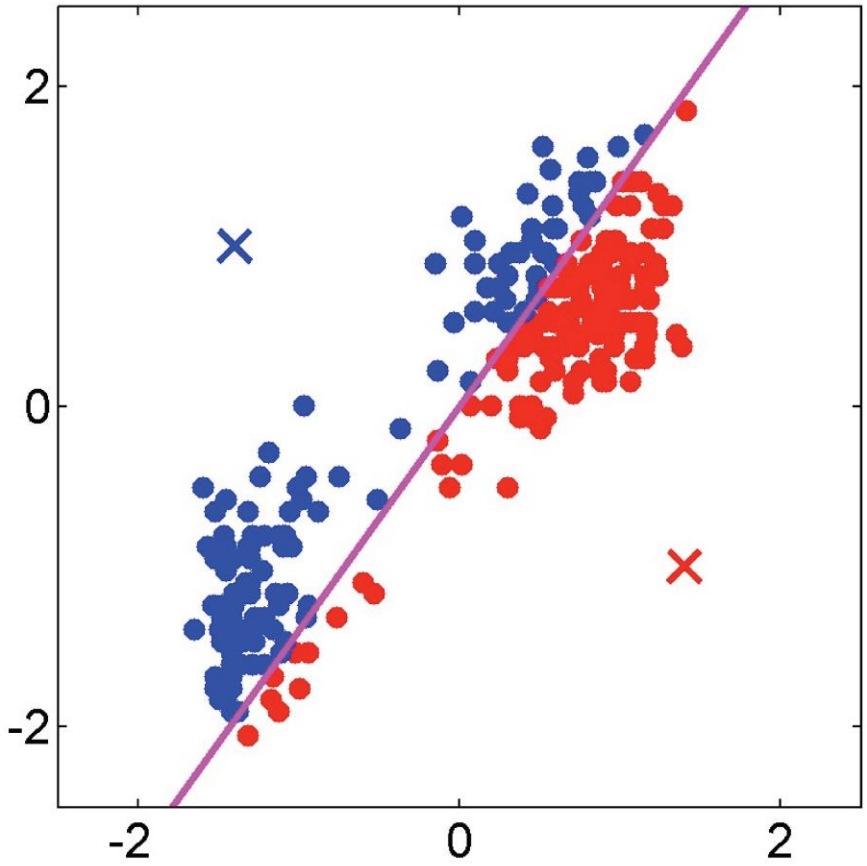
K-means example



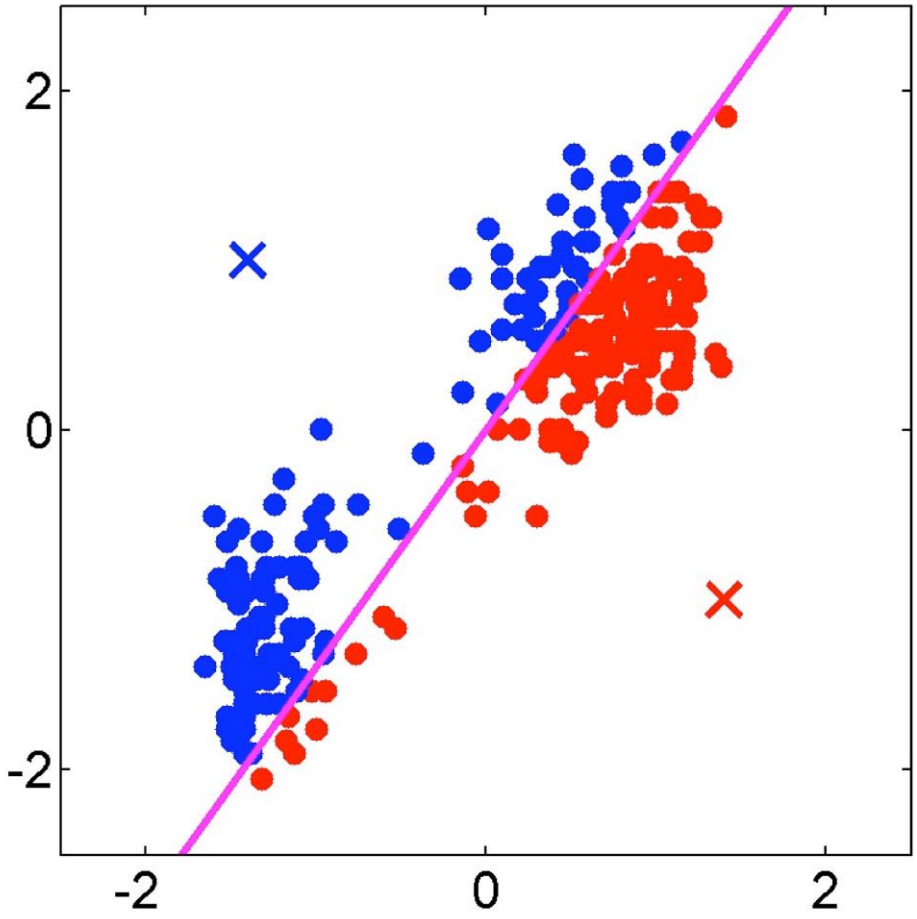
K-means example



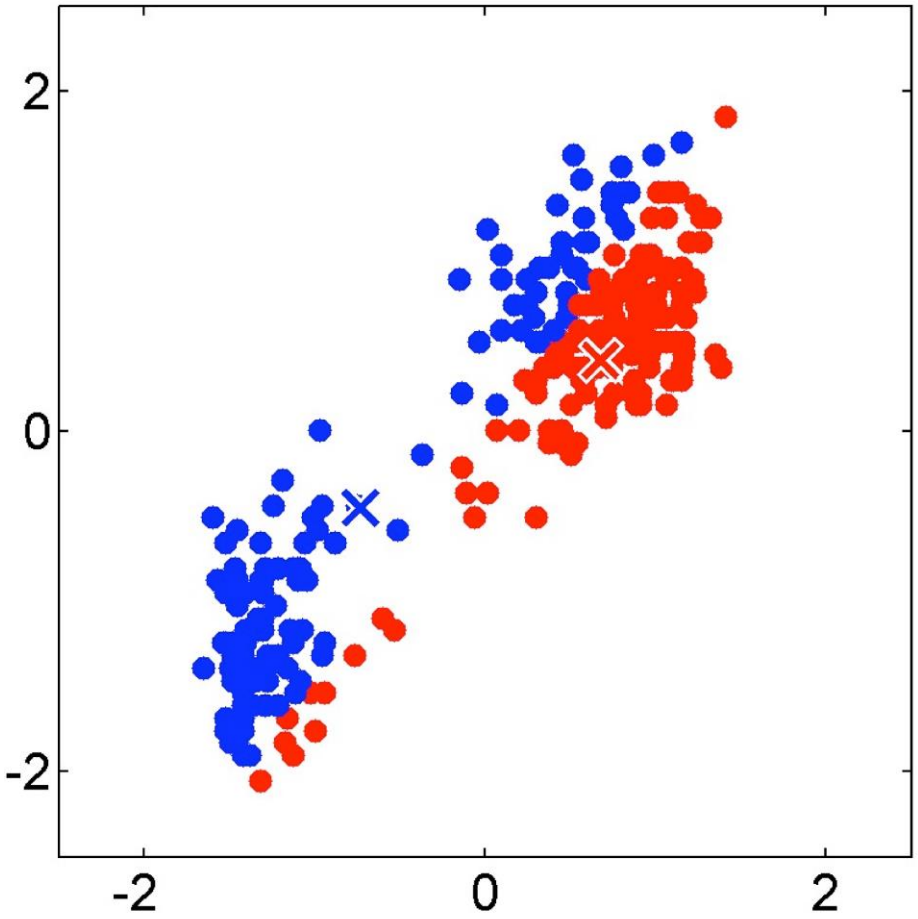
K-means example



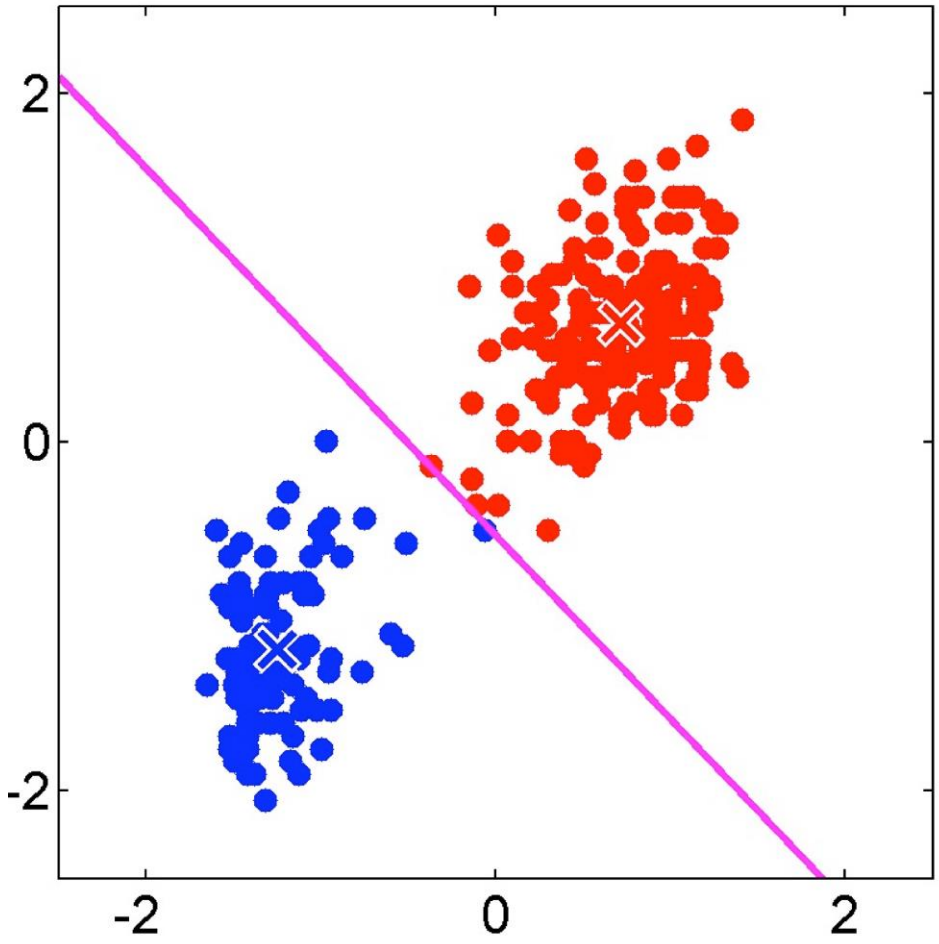
K-means example



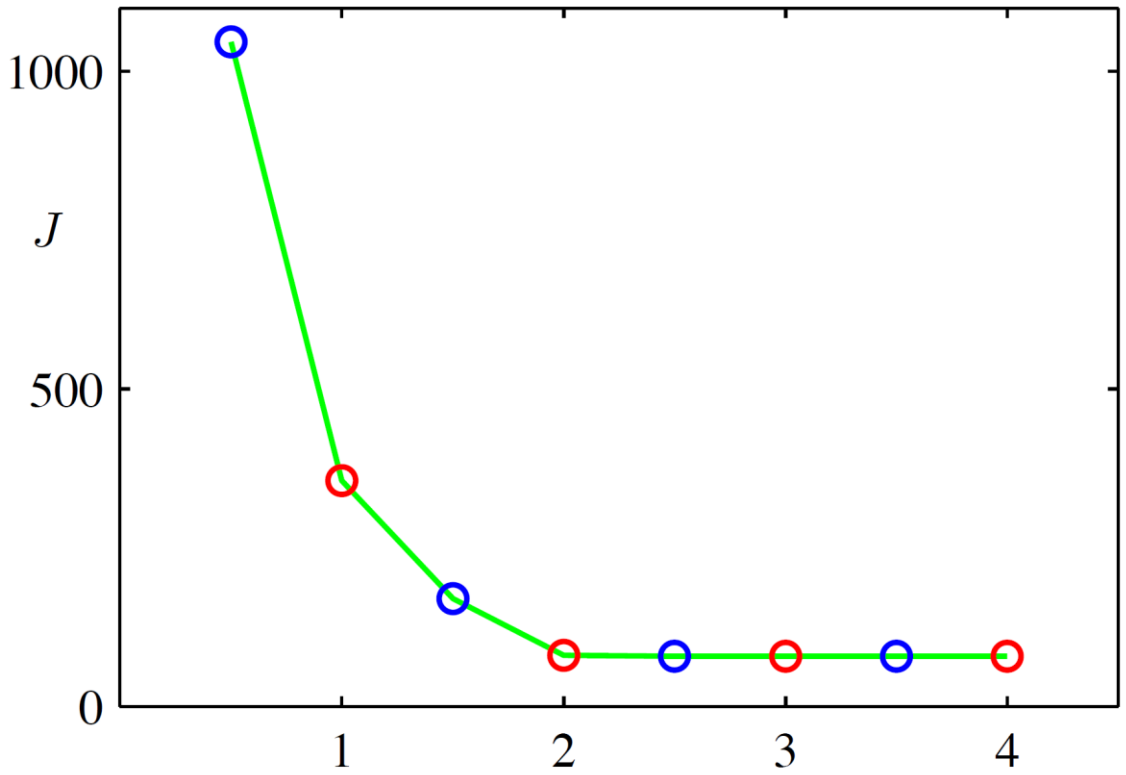
K-means example



K-means example



K-means example: loss function



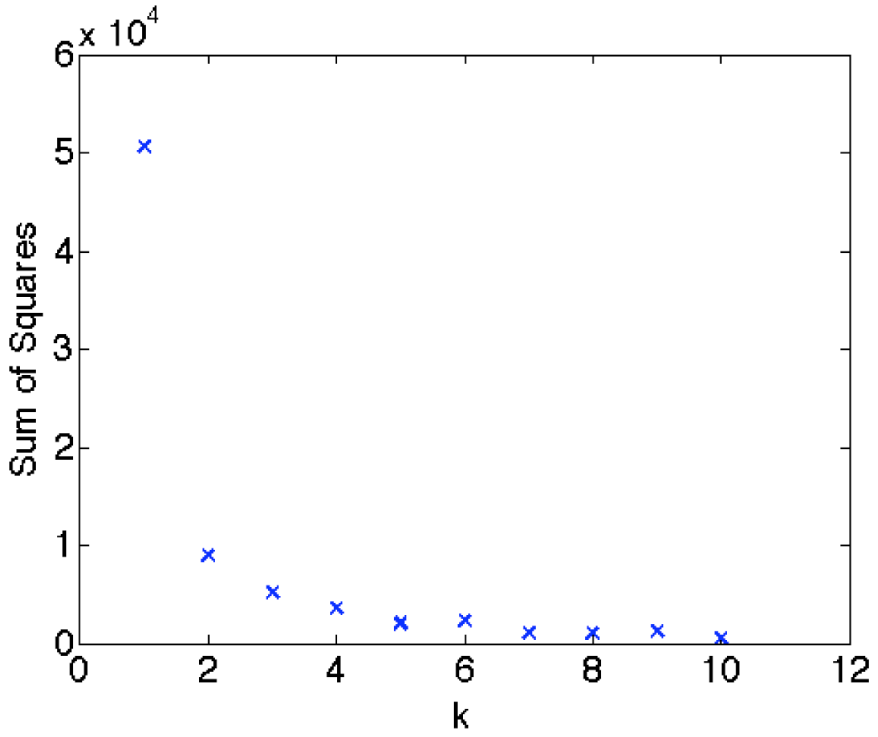
K-means algorithm

- Converges to a local minimum.
  - Use different initializations and pick the best solution.
  - May still be insufficient for large search spaces.
  - Other ways include a split-merge approach.

- Some heuristics
  - Randomly pick K data points as prototypes.
  - Pick prototype $i+1$ to be farthest from prototypes $\{1, \ldots, i\}$.

How to choose K?

- The loss function J generally decreases with K.



How to choose K?

- Cross-validation: Partition data into two sets. Estimate prototypes on one and use these to compute the loss function on the other.
- Stability of clusters: Measure the change in the clusters obtained by resampling or splitting the data

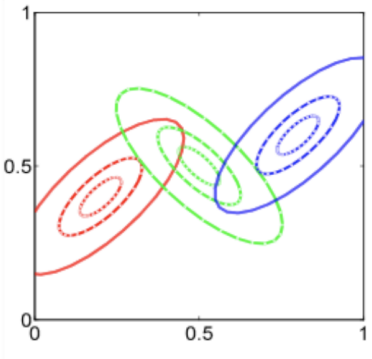
Limitations of k-means

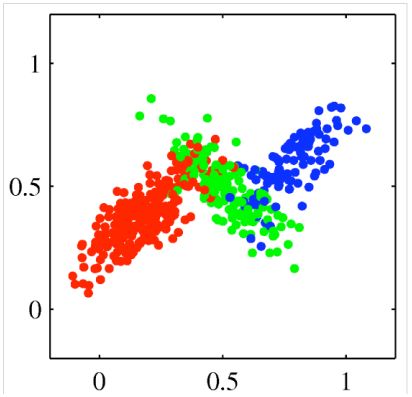
- Hard assignments of data points to clusters can cause a small perturbation to a data point to flip it to another cluster
- Assumes spherical clusters and equal probabilities for each cluster
- Those limitations can be solved by GMM based clustering

Gaussian Mixture Model (GMM)

- Likelihood $\Pr(x) = \sum_{k=1}^{K} \pi_k \mathcal{N}(x|\mu_k, \Sigma_k)$ where

$$\sum_{k=1}^{K} \pi_k = 1, 0 \leq \pi_k \leq 1.$$





Gaussian Mixture Model (GMM)

- Generative process:

$\{\pi_k, \mu_k, \Sigma_k\}$

1. Randomly select the k-th Gaussian distribution according to π_k
2. Randomly sample x from the selected Gaussian distribution

Imagine this process as a piece of program. Once written, it can generate a sample when we run it. The output x will be a random variable.

Gaussian Mixture Model (GMM)

- Generative process:

The distribution of x?

1. Randomly select the k-th Gaussian distribution according to π_k
2. Randomly sample x from the selected Gaussian distribution

$$P(\mathbf{x}) = P(\mathbf{x}|c=1)P(c=1) + P(\mathbf{x}|c=2)P(c=2) + \cdots$$

Let $P(c=k) = \pi_k$

$P(\mathbf{x}|c=k) = \mathcal{N}(\mathbf{x}|\mu_k, \Sigma_k)$

$$\Rightarrow \quad Pr(\mathbf{x}) = \sum_{k=1}^{K} \pi_k \mathcal{N}(\mathbf{x}|\mu_k, \Sigma_k)$$

Gaussian Mixture Model (GMM)

- Generative process:

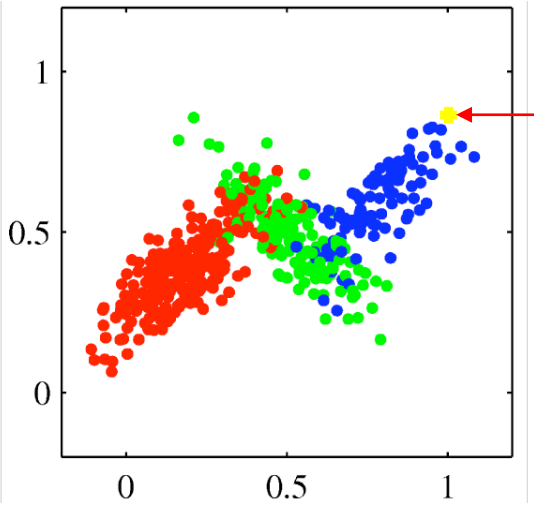
$\{\pi_k, \mu_k, \Sigma_k\}$

1. Randomly select the k-th Gaussian distribution according to π_k
2. Randomly sample x from the selected Gaussian distribution

The selection made inside the generative process, c is an internal variable, hidden from us. We call it latent variable

Latent variable

- Intermediate results inside a generative process
- Each sample is associated with a latent variable
- We do not know its exact value
- But we can infer how likely it can be
- Example



We may know this point is more likely to belong to the blue cluster

Latent variable and Inference

- Given an observation, we can estimate the likelihood of the latent variable. In the case of GMM, we try to estimate $P(c|\mathbf{x})$
- Because the latent variable in GMM indicates the membership of a sample belonging to a cluster. $P(c|\mathbf{x})$ can be seen as a soft-membership (soft-clustering)

Latent variable and Inference

- Given an observation, we can estimate the likelihood of the latent variable. In the case of GMM, we try to estimate $P(c|\mathbf{x})$

- The difference between $P(c|\mathbf{x})$ and $P(c) = \pi_k$

  $P(c|\mathbf{x})$ Posterior probability. The likelihood about c after x is observed

  $P(c)$ Prior probability. The likelihood before any observation

- The process of calculate $P(c|\mathbf{x})$ or the most likely c is called Inference

Inference in GMM

- Inference can be done by using Bayes theory

$$P(c|\mathbf{x}) = \frac{P(\mathbf{x}|c)P(c)}{P(\mathbf{x})}$$

$$P(c=k) = \pi_k$$
$$P(\mathbf{x}|c=k) = \mathcal{N}(\mathbf{x}|\mu_k, \Sigma_k)$$
$$P(\mathbf{x}) = \sum_{k=1}^{K} \pi_k \mathcal{N}(\mathbf{x}|\mu_k, \Sigma_k)$$

$$P(c=k|\mathbf{x}) = \frac{\mathcal{N}(\mathbf{x}|\mu_k, \Sigma_k)\pi_k}{\sum_{j=1}^{K} \mathcal{N}(\mathbf{x}|\mu_j, \Sigma_j)\pi_j}$$

Parameter estimation for GMM

- Use MLE (maximal likelihood estimation)

$$\begin{aligned}\mathcal{L} &= -\log P(X) \\ &= -\sum_{i=1}^{N} \log P(x_i) \\ &= -\sum_{i=1}^{N} \log\{\sum_{k=1}^{K} \pi_k \mathcal{N}(x_i|\mu_k, \Sigma_k)\}\end{aligned}$$

- Unfortunately, it is difficult to solve $\frac{\partial \mathcal{L}}{\partial \theta} = 0$

EM algorithm

- Actually, this difficulty occurs for many probabilistic models having latent variable inside its generative process.
- EM algorithm is a solution to those problems

EM algorithm iterates between the following two steps:

- **E-step:** calculate the posterior probability of latent variable $P(z|x,\theta)$ given current model parameter θ.
- **M-step:** Update the model parameter based on the expected log likelihood. This is done by maximizing

$$E_{z|x,\theta} \log P(x,z|\theta) = \int P(z|x,\theta) \log(P(x|\theta,z)P(z|\theta))dz$$

Iterating between those two steps will converge to a local minimum

EM algorithm

EM algorithm iterates between the following two steps:

- **E-step:** calculate the posterior probability of latent variable $P(z|x,\theta)$ given current model parameter θ.
- **M-step:** Update the model parameter based on the expected log likelihood. This is done by maximizing

$$E_{z|x,\theta} \log P(x,z|\theta) = \int P(z|x,\theta) \log \underbrace{\left(P(x|\theta,z)P(z|\theta)\right)}_{P(x,z).} dz$$

EM algorithm is useful because usually both E-step and M-step can be easily calculated,

EM algorithm: motivation explained

- Want to calculate $\hat{\theta} = argmax_\theta \log P(x|\theta)$ but it is difficult
- If latent variable is known, $\hat{\theta} = argmax_\theta \log P(x, z|\theta)$ is easy to calculate
- But we do not know z, however it is easy to know $P(z|x, \theta)$
- Let's enumerate all the possible z, and calculate the weighted sum

$$\int P(z|x,\theta)P(x,z|\theta)dz = \int P(z|x,\theta)\log\left(P(x|\theta,z)P(z|\theta)\right)dz$$

with the weight as $P(z|x, \theta)$

Apply EM to GMM

- E-step: calculate the posterior probability about the latent variable:

$$r_{ik} = P(c = k|\mathbf{x}_i) = \frac{\mathcal{N}(\mathbf{x}_i|\mu_k, \Sigma_k)\pi_k}{\sum_{j=1}^{K} \mathcal{N}(\mathbf{x}_i|\mu_j, \Sigma_j)\pi_j}$$

- M-step:

  Try to maximize

$$E[\log P(x, z|\pi, \mu, \Sigma)] = \sum_i \sum_k r_{ik} \left(\log \pi_k + \log \mathcal{N}(x_i|\mu_k, \Sigma_k)\right)$$

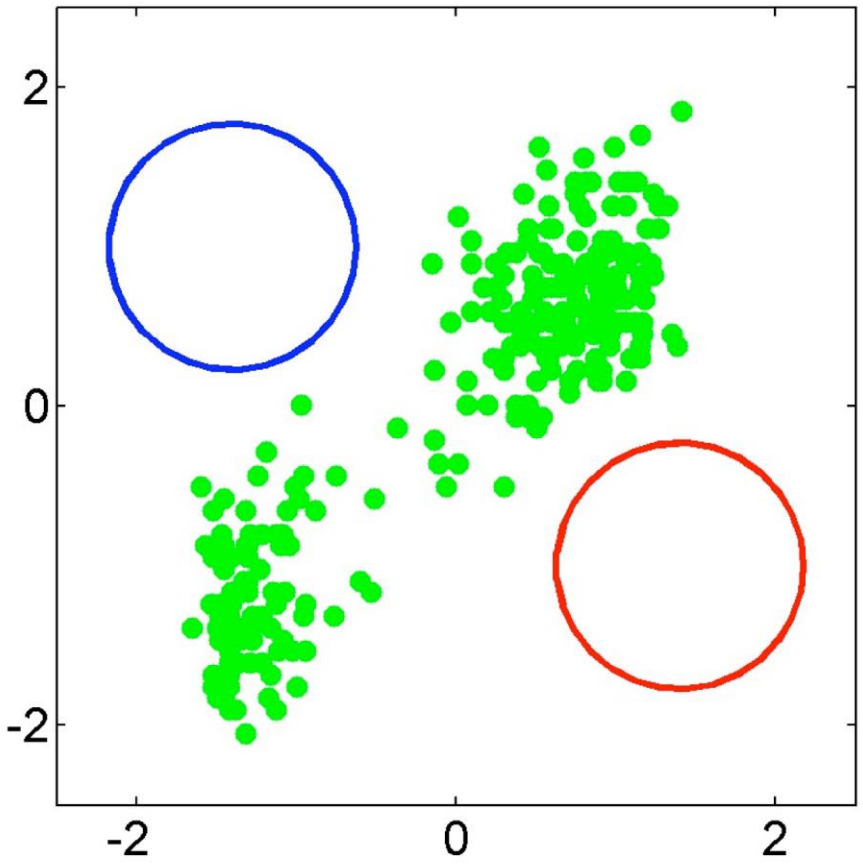
  This leads to the following solutions

$$\pi_k = \frac{\sum_i r_{ik}}{N}, \ \mu_k = \frac{\sum_i r_{ik} x_i}{\sum_i r_{ik}}, \ \Sigma_k = \frac{\sum_i r_{ik} (x_i - \mu_k)(x_i - \mu_k)^T}{\sum_i r_{ik}}$$

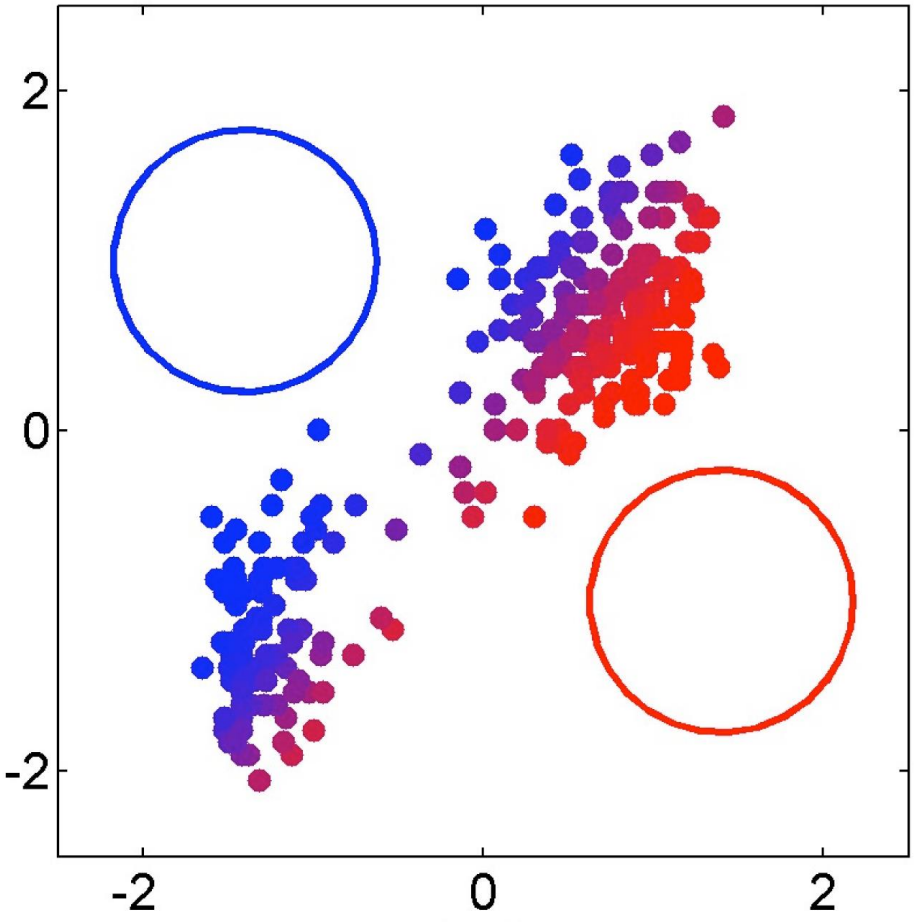
Connection to K-means

- E-step in GMM a soft version of K-means. $r_{ik} \in [0, 1]$ instead of $\{0, 1\}$.
- M-step in GMM estimates the probabilities and the covariance matrix of each cluster in addition to the means.
- All π_k are equal. $\Sigma_k = \delta^2 I$. As $\delta^2 \to 0$, $r_{ik} \to \{0, 1\}$, and the two methods coincide.

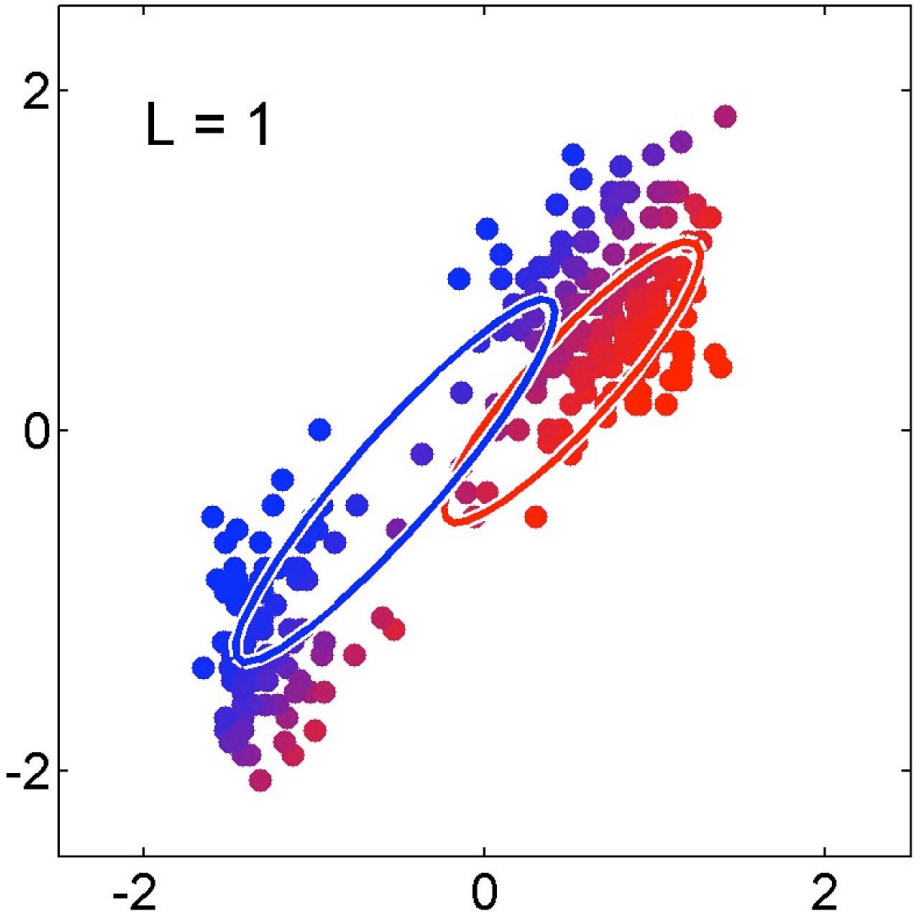
GMM example



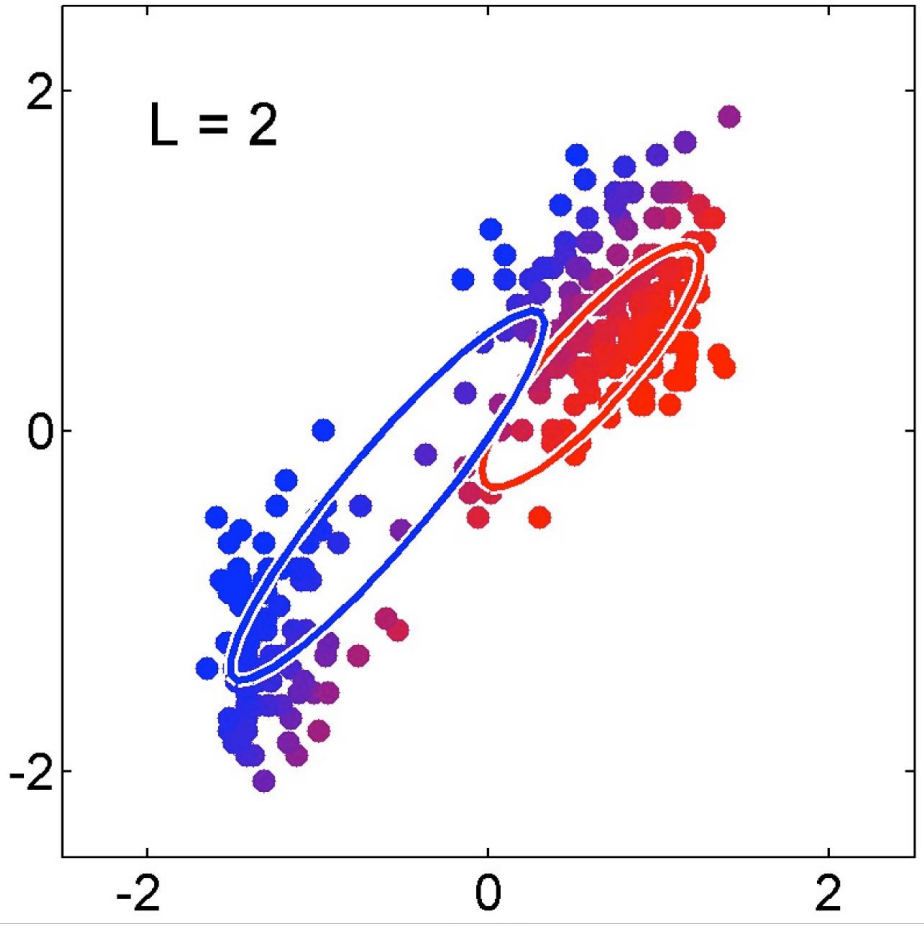
GMM example



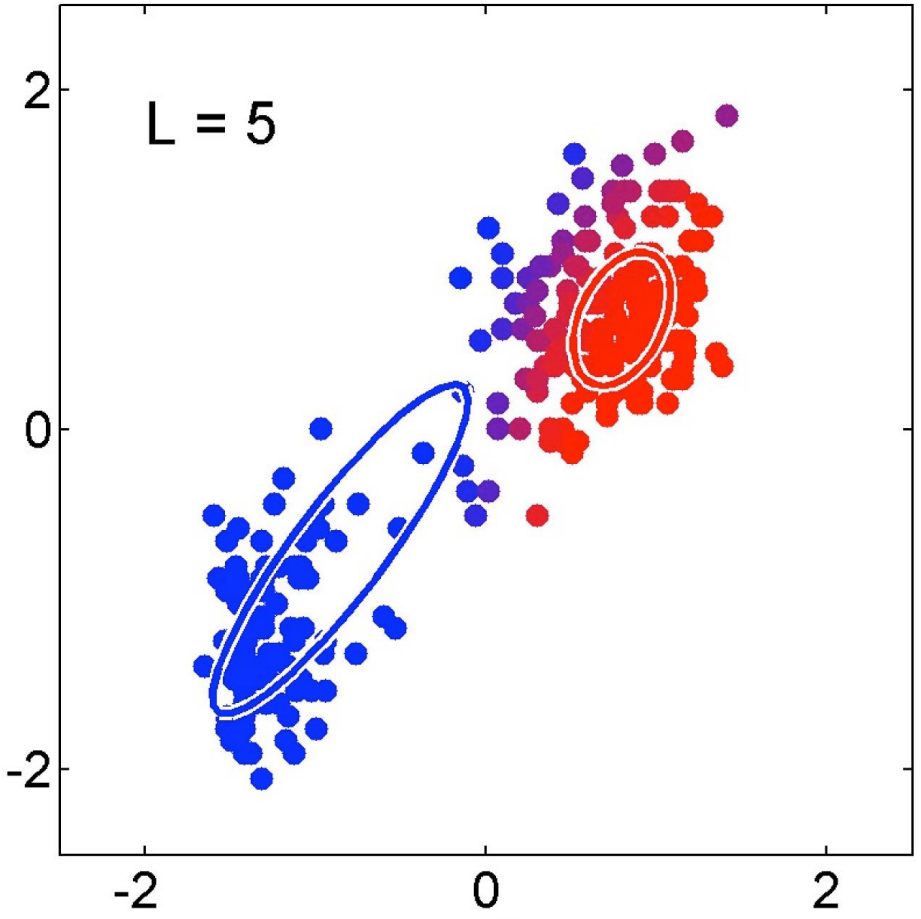
GMM example



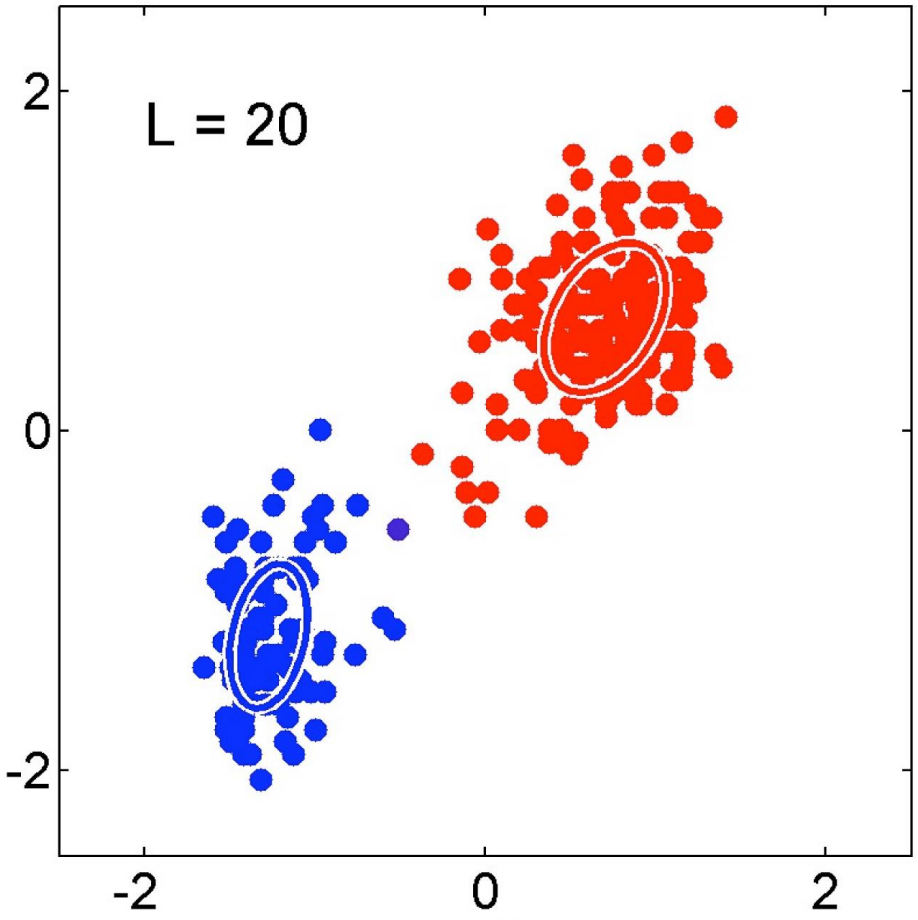
GMM example



GMM example



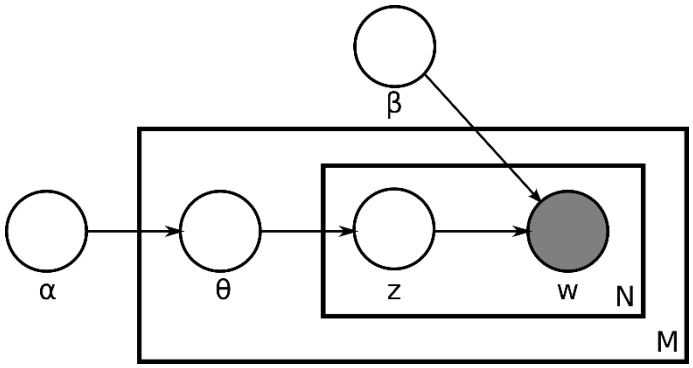
GMM example

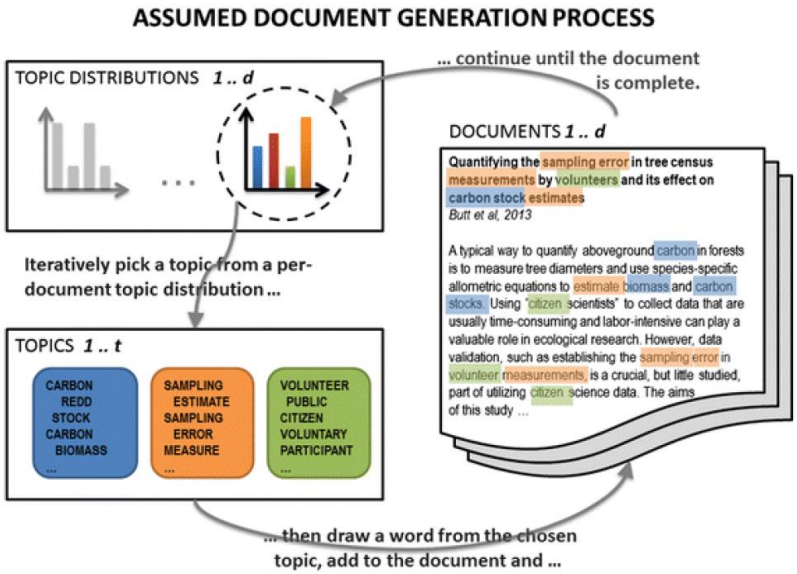


From GMM to other unsupervised learning methods

- Like GMM, many unsupervised learning problem turns into a parameter estimation problem for distribution models.
- Many distribution models involves latent variables. Those latent variables often have physic meaning and are something we want to estimate
- Training stage: estimate the parameters
- Testing stage: infer the latent variable

Example: Latent Dirichlet Allocation [optional]





Generating a document by using LDA

- 1. Sample a topic distribution
- 2. From the distribution, sample a topic
- 3. From the topic, sample a word
- 4. repeat 2-3 to sample N words in a document