Lecture 21: Gaussian Mixture Models and EM Algorithm

BIOS635

04/07/2020

Some slides taken/modified from other people's lectures online.

Mixture models

- Recall types of clustering methods
  - hard clustering: clusters do not overlap
    - element either belongs to cluster or it does not
  - soft clustering: clusters may overlap
    - stength of association between clusters and instances

Mixture models

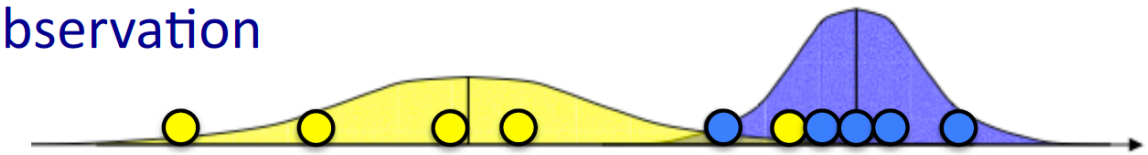
- Recall types of clustering methods
  - hard clustering: clusters do not overlap
    - element either belongs to cluster or it does not
  - soft clustering: clusters may overlap
    - stength of association between clusters and instances
- Mixture models
  - probabilistically-grounded way of doing soft clustering
  - each source: a generative model (Gaussian or multinomial)
  - parameters (e.g. mean/covariance are unknown)

Mixture models in 1-d

- Observations $x_1 \ldots x_n$
  - K=2 Gaussians with unknown μ, σ^2
  - estimation trivial if we know the source of each observation

$$\mu_b = \frac{x_1 + x_2 + \ldots + x_{n_b}}{n_b}$$

$$\sigma_b^2 = \frac{(x_1 - \mu_1)^2 + \ldots + (x_n - \mu_n)^2}{n_b}$$

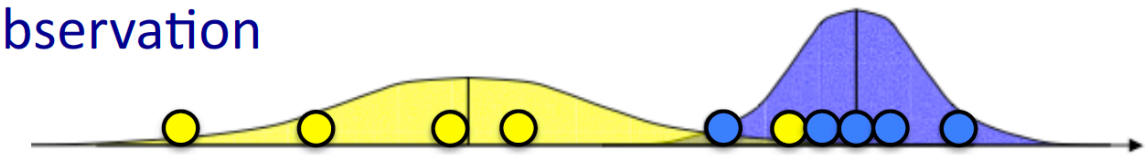


Mixture models in 1-d

- Observations $x_1 \ldots x_n$
  - K=2 Gaussians with unknown μ, σ^2
  - estimation trivial if we know the source of each observation

$$\mu_b = \frac{x_1 + x_2 + \ldots + x_{n_b}}{n_b}$$

$$\sigma_b^2 = \frac{(x_1 - \mu_1)^2 + \ldots + (x_n - \mu_n)^2}{n_b}$$



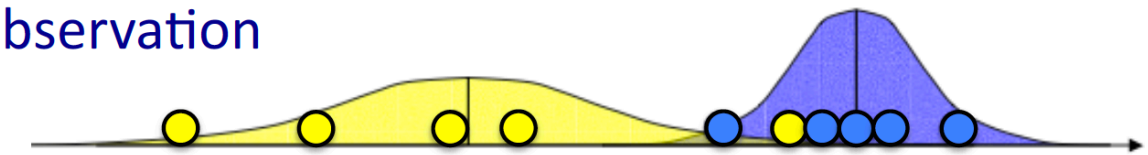
- What if we don't know the source?

Mixture models in 1-d

- Observations $x_1 \ldots x_n$
  - K=2 Gaussians with unknown μ, σ^2
  - estimation trivial if we know the source of each observation

$$\mu_b = \frac{x_1 + x_2 + \ldots + x_{n_b}}{n_b}$$

$$\sigma_b^2 = \frac{(x_1 - \mu_1)^2 + \ldots + (x_n - \mu_n)^2}{n_b}$$



- What if we don't know the source?
- If we knew parameters of the Gaussians (μ, σ^2)
  - can guess whether point is more likely to be a or b

$$P(b \mid x_i) = \frac{P(x_i \mid b)P(b)}{P(x_i \mid b)P(b) + P(x_i \mid a)P(a)}$$

$$P(x_i \mid b) = \frac{1}{\sqrt{2\pi\sigma_b^2}} \exp\left(-\frac{(x_i - \mu_b)^2}{2\sigma_b^2}\right)$$

Expectation Maximization (EM)

- Chicken and egg problem
  - need (μ_a, σ_a^2) and (μ_b, σ_b^2) to guess source of points
  - need to know source to estimate (μ_a, σ_a^2) and (μ_b, σ_b^2)

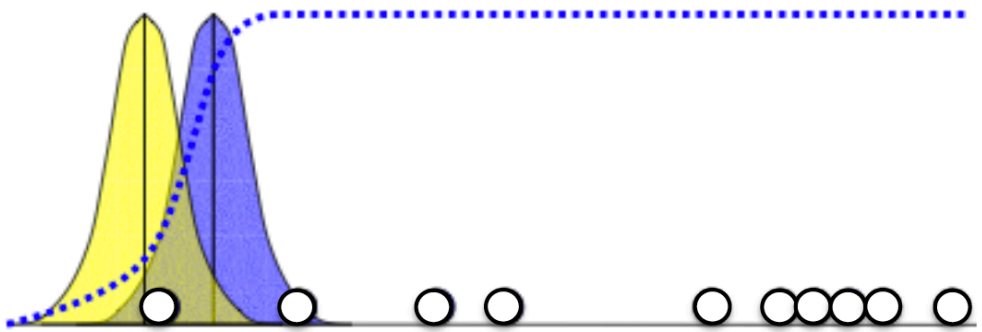
Expectation Maximization (EM)

- Chicken and egg problem
  - need (μ_a, σ_a^2) and (μ_b, σ_b^2) to guess source of points
  - need to know source to estimate (μ_a, σ_a^2) and (μ_b, σ_b^2)
- EM algorithm
  - start with two randomly placed Gaussians (μ_a, σ_a^2), (μ_b, σ_b^2)

E-step:
  - for each point: $P(b \mid x_i)$ = does it look like it came from b?

M-step:
  - adjust (μ_a, σ_a^2) and (μ_b, σ_b^2) to fit points assigned to them
  - iterate until convergence

EM: illustration on 1-d example

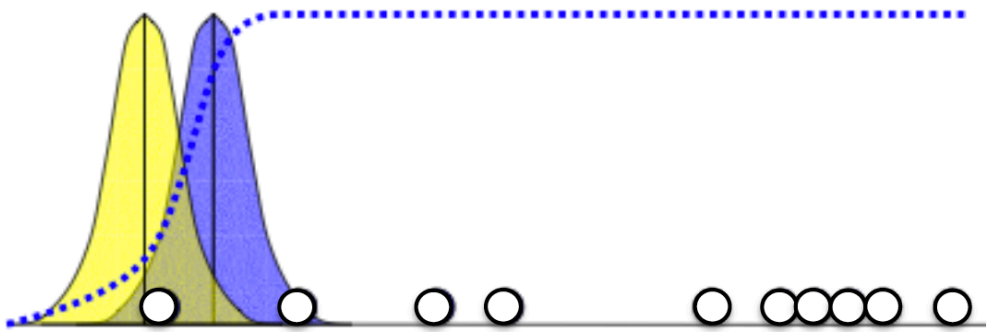


$$P(x_i \mid b) = \frac{1}{\sqrt{2\pi\sigma_b^2}} \exp\left(-\frac{(x_i - \mu_b)^2}{2\sigma_b^2}\right)$$

$$b_i = P(b \mid x_i) = \frac{P(x_i \mid b)P(b)}{P(x_i \mid b)P(b) + P(x_i \mid a)P(a)}$$

$$a_i = P(a \mid x_i) = 1 - b_i$$

EM: illustration on 1-d example



$$P(x_i \mid b) = \frac{1}{\sqrt{2\pi\sigma_b^2}} \exp\left(-\frac{(x_i - \mu_b)^2}{2\sigma_b^2}\right)$$

$$b_i = P(b \mid x_i) = \frac{P(x_i \mid b)P(b)}{P(x_i \mid b)P(b) + P(x_i \mid a)P(a)}$$

$$a_i = P(a \mid x_i) = 1 - b_i$$

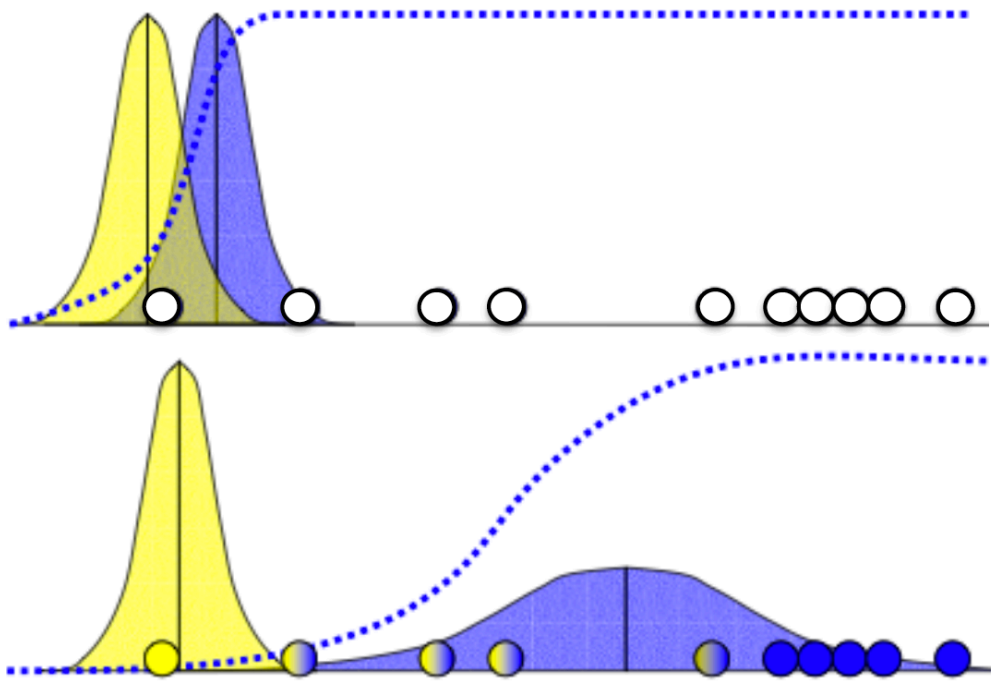
$$\mu_b = \frac{b_1 x_1 + b_2 x_2 + \ldots + b_n x_{n_b}}{b_1 + b_2 + \ldots + b_n}$$

$$\sigma_b^2 = \frac{b_1 (x_1 - \mu_1)^2 + \ldots + b_n (x_n - \mu_n)^2}{b_1 + b_2 + \ldots + b_n}$$

$$\mu_a = \frac{a_1 x_1 + a_2 x_2 + \ldots + a_n x_{n_b}}{a_1 + a_2 + \ldots + a_n}$$

$$\sigma_a^2 = \frac{a_1 (x_1 - \mu_1)^2 + \ldots + a_n (x_n - \mu_n)^2}{a_1 + a_2 + \ldots + a_n}$$

EM: illustration on 1-d example



$$P(x_i \mid b) = \frac{1}{\sqrt{2\pi\sigma_b^2}} \exp\left(-\frac{(x_i - \mu_b)^2}{2\sigma_b^2}\right)$$

$$b_i = P(b \mid x_i) = \frac{P(x_i \mid b)P(b)}{P(x_i \mid b)P(b) + P(x_i \mid a)P(a)}$$

$$a_i = P(a \mid x_i) = 1 - b_i$$

$$\mu_b = \frac{b_1 x_1 + b_2 x_2 + \ldots + b_n x_{n_b}}{b_1 + b_2 + \ldots + b_n}$$

$$\sigma_b^2 = \frac{b_1 (x_1 - \mu_1)^2 + \ldots + b_n (x_n - \mu_n)^2}{b_1 + b_2 + \ldots + b_n}$$

$$\mu_a = \frac{a_1 x_1 + a_2 x_2 + \ldots + a_n x_{n_b}}{a_1 + a_2 + \ldots + a_n}$$

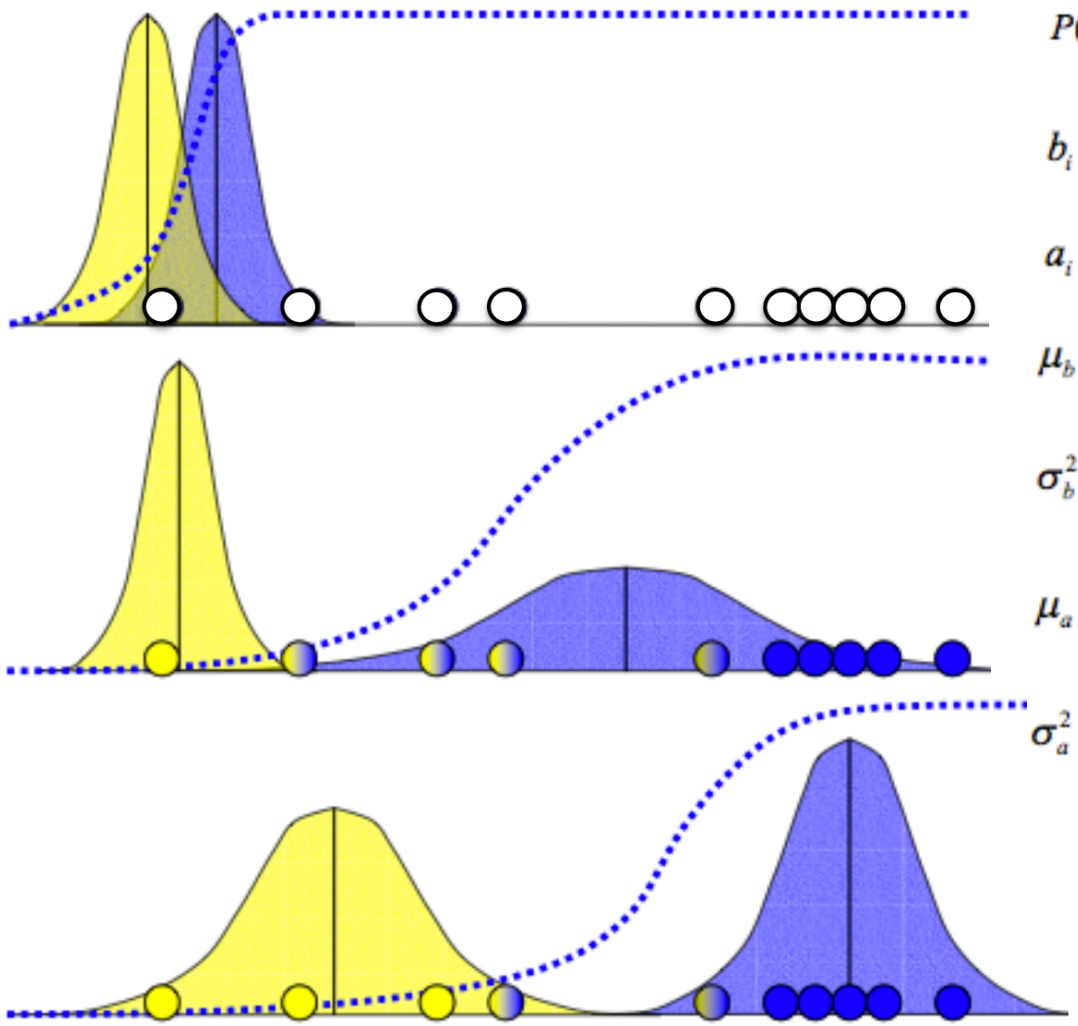
$$\sigma_a^2 = \frac{a_1 (x_1 - \mu_1)^2 + \ldots + a_n (x_n - \mu_n)^2}{a_1 + a_2 + \ldots + a_n}$$

could also estimate priors:

$P(b) = (b_1 + b_2 + \ldots b_n) / n$

$P(a) = 1 - P(b)$

EM: illustration on 1-d example



$$P(x_i \mid b) = \frac{1}{\sqrt{2\pi\sigma_b^2}} \exp\left(-\frac{(x_i - \mu_b)^2}{2\sigma_b^2}\right)$$

$$b_i = P(b \mid x_i) = \frac{P(x_i \mid b)P(b)}{P(x_i \mid b)P(b) + P(x_i \mid a)P(a)}$$

$$a_i = P(a \mid x_i) = 1 - b_i$$

$$\mu_b = \frac{b_1 x_1 + b_2 x_2 + \ldots + b_n x_{n_b}}{b_1 + b_2 + \ldots + b_n}$$

$$\sigma_b^2 = \frac{b_1 (x_1 - \mu_1)^2 + \ldots + b_n (x_n - \mu_n)^2}{b_1 + b_2 + \ldots + b_n}$$

$$\mu_a = \frac{a_1 x_1 + a_2 x_2 + \ldots + a_n x_{n_b}}{a_1 + a_2 + \ldots + a_n}$$

$$\sigma_a^2 = \frac{a_1 (x_1 - \mu_1)^2 + \ldots + a_n (x_n - \mu_n)^2}{a_1 + a_2 + \ldots + a_n}$$

could also estimate priors:

$P(b) = (b_1 + b_2 + \ldots b_n) / n$

$P(a) = 1 - P(b)$

1-d Gaussian mixture

$$
\begin{aligned}
Y_1 &\sim N(\mu_1, \sigma_1^2), \\
Y_2 &\sim N(\mu_2, \sigma_2^2), \\
Y &= (1-\Delta) \cdot Y_1 + \Delta \cdot Y_2,
\end{aligned}
$$

where $\Delta \in \{0, 1\}$ with $\Pr(\Delta = 1) = \pi$.

Let $\phi_\theta(x)$ denote the normal density with parameters $\theta = (\mu, \sigma^2)$. Then the density of Y is

$$g_Y(y) = (1-\pi)\phi_{\theta_1}(y) + \pi\phi_{\theta_2}(y).$$

The log-likelihood based on the N training cases is

$$\ell(\theta; \mathbf{z}) = \sum_{i=1}^{N} \log[(1-\pi)\phi_{\theta_1}(y_i) + \pi\phi_{\theta_2}(y_i)].$$

1-d Gaussian mixture, cont'd

Direct maximization of $\ell(\theta; \mathbf{z})$ is quite difficult numerically, because of the sum of terms inside the logarithm. There is, however, a simpler approach. We consider unobserved latent variables Δ_i taking values 0 or 1: if $\Delta_i = 1$ then Y_i comes from model 2, otherwise it comes from model 1. Suppose we knew the values of the Δ_i's. Then the log-likelihood would be

$$\begin{aligned}\ell_0(\theta; \mathbf{z}, \mathbf{\Delta}) &= \sum_{i=1}^{N} [(1-\Delta_i) \log \phi_{\theta_1}(y_i) + \Delta_i \log \phi_{\theta_2}(y_i)] \\ &\quad + \sum_{i=1}^{N} [(1-\Delta_i) \log \pi + \Delta_i \log(1-\pi)]\end{aligned}$$

1-d Gaussian mixture, cont'd

Since the values of the Δ_i's are actually unknown, we proceed in an iterative fashion, substituting for each Δ_i its expected value

$$\gamma_i(\theta) = \mathrm{E}\,(\Delta_i|\theta, \mathbf{z}) = \Pr(\Delta_i = 1|\theta, \mathbf{z}),$$

also called the responsibility of model 2 for observation i. We use a procedure called the EM algorithm.

EM algorithm for two-component Gaussian mixture

- Take initial guesses for the parameters $\hat{\mu}_1, \hat{\sigma}_1^2, \hat{\mu}_2, \hat{\sigma}_2^2, \hat{\pi}$ (see text).
- *Expectation Step*: compute the responsibilities

$$\hat{\gamma}_i = \frac{\hat{\pi}\phi_{\hat{\theta}_2}(y_i)}{(1-\hat{\pi})\phi_{\hat{\theta}_1}(y_i) + \hat{\pi}\phi_{\hat{\theta}_2}.(y_i)}, \ i = 1, 2, \ldots, N.$$

- *Maximization Step*: compute the weighted means and variances:

$$\hat{\mu}_1 = \frac{\sum_{i=1}^N (1-\hat{\gamma}_i) y_i}{\sum_{i=1}^N (1-\hat{\gamma}_i)}, \qquad \hat{\sigma}_1^2 = \frac{\sum_{i=1}^N (1-\hat{\gamma}_i)(y_i - \hat{\mu}_1)^2}{\sum_{i=1}^N (1-\hat{\gamma}_i)},$$

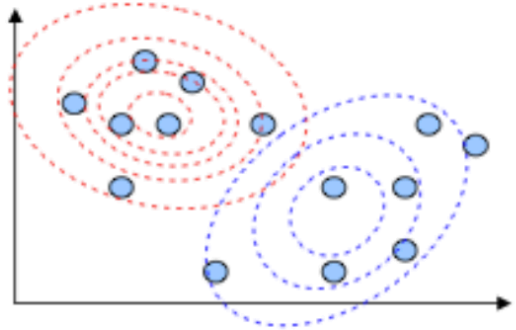
$$\hat{\mu}_2 = \frac{\sum_{i=1}^N \hat{\gamma}_i y_i}{\sum_{i=1}^N \hat{\gamma}_i}, \qquad \hat{\sigma}_2^2 = \frac{\sum_{i=1}^N \hat{\gamma}_i (y_i - \hat{\mu}_1)^2}{\sum_{i=1}^N \hat{\gamma}_i},$$

  and the mixing probability $\hat{\pi} = \sum_{i=1}^N \hat{\gamma}_i / N$.

- Iterate these steps until convergence.

Multivariate Gaussian mixtures with d > 1

- Data with d attributes, from k sources
- Each source c is a Gaussian
- Iteratively estimate parameters:
  - prior: what % of instances came from source c?
  - mean: expected value of attribute j from source c
  - covariance: how correlated are attributes j and k in source c?
  - based on: our guess of the source for each instance



$$P(c) = \frac{1}{n}\sum_{i=1}^{n} P(c \mid \vec{x}_i)$$

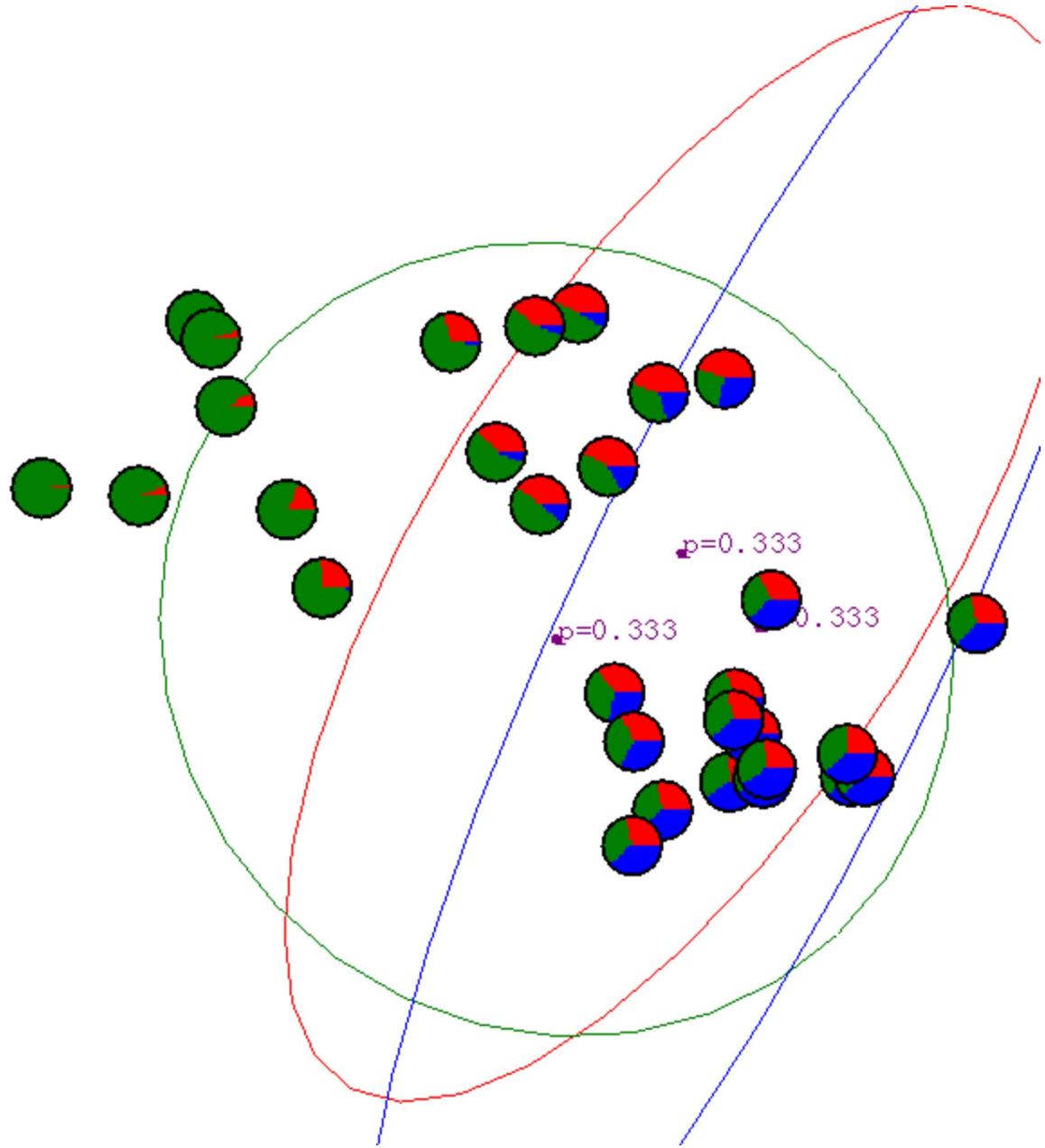
$$\mu_{c,j} = \sum_{i=1}^{n} \left(\frac{P(c|\vec{x}_i)}{nP(c)}\right) x_{i,j}$$

$$(\Sigma_c)_{j,k} = \sum_{i=1}^{n} \left(\frac{P(c|\vec{x}_i)}{nP(c)}\right)(x_{i,j} - \mu_{c,j})(x_{i,k} - \mu_{c,k})$$

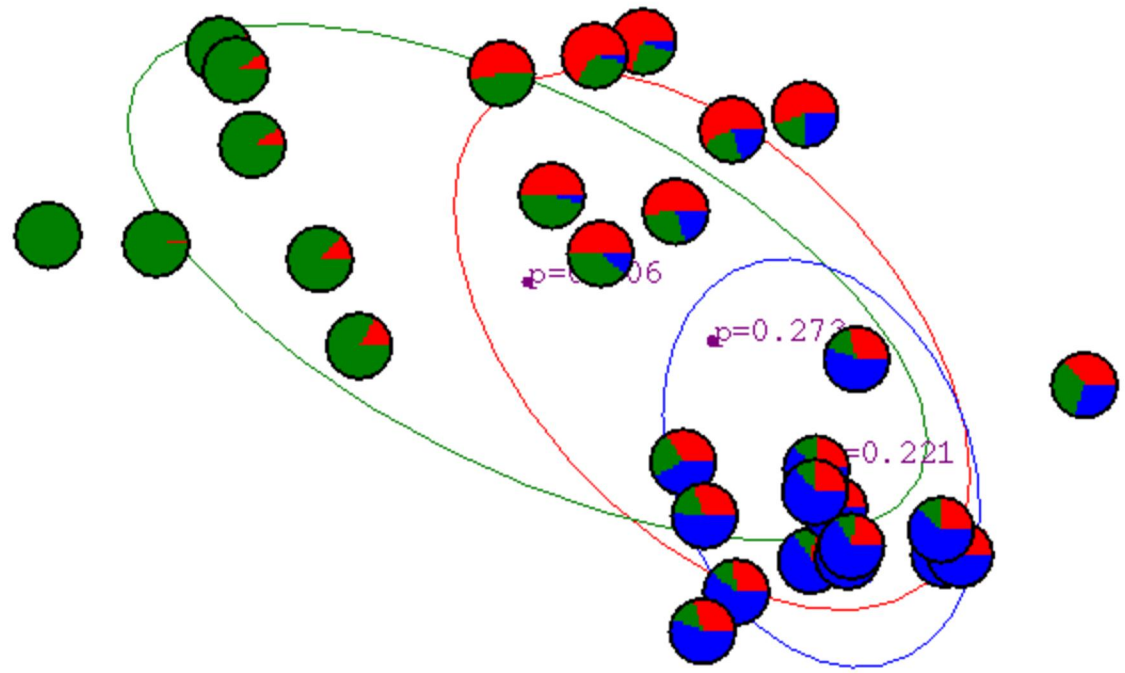
$$P(c \mid \vec{x}_i) = \frac{P(\vec{x}_i \mid c)P(c)}{\sum_{c'=1}^{k} P(\vec{x}_i \mid c')P(c')}$$

$$P(\vec{x}_i \mid c) = \frac{1}{\sqrt{2\pi|\Sigma_c|}} \exp\left(-\frac{1}{2}\underbrace{(\vec{x}_i - \vec{\mu}_c)^T \Sigma_c^{-1} (\vec{x}_i - \vec{\mu}_c)}_{\sum_{j=1}^{d}\sum_{k=1}^{d}(x_{i,j} - \mu_{c,j})(\Sigma_c^{-1})_{j,k}(x_{i,k} - \mu_{c,k})}\right)$$

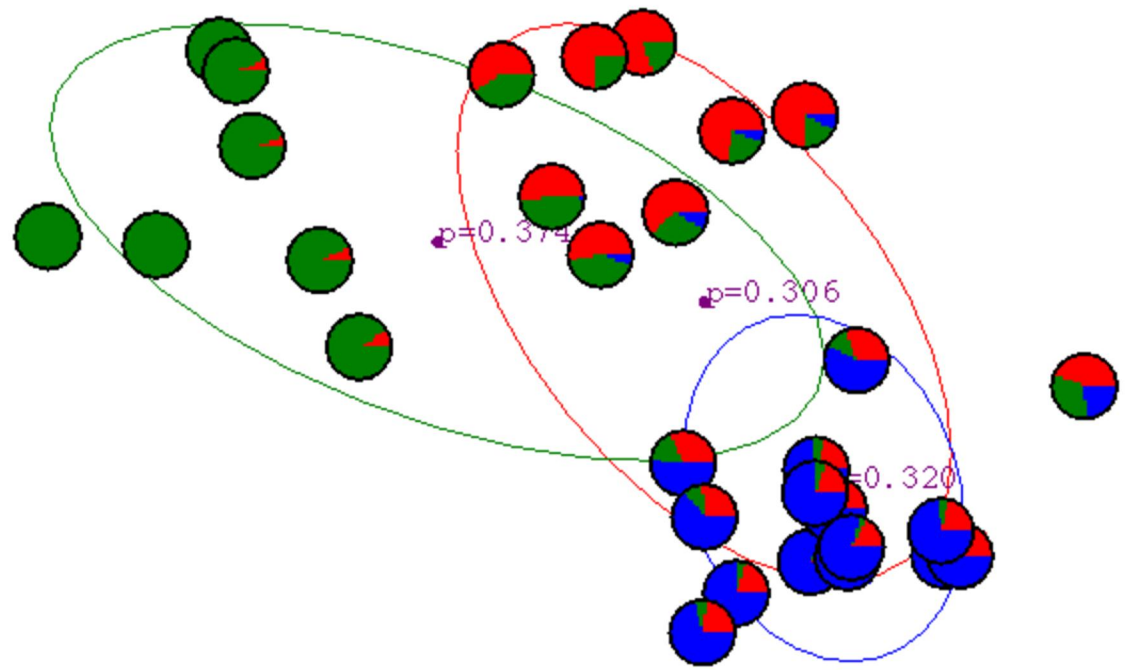
Gaussian Mixture Example: Start



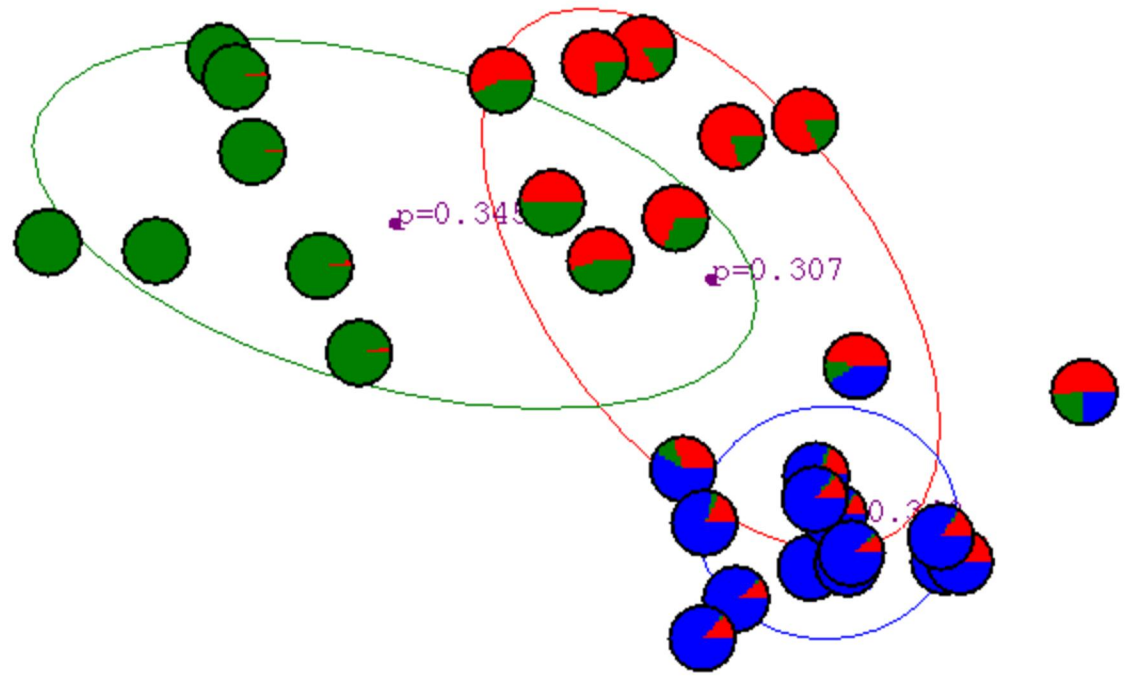
After first iteration



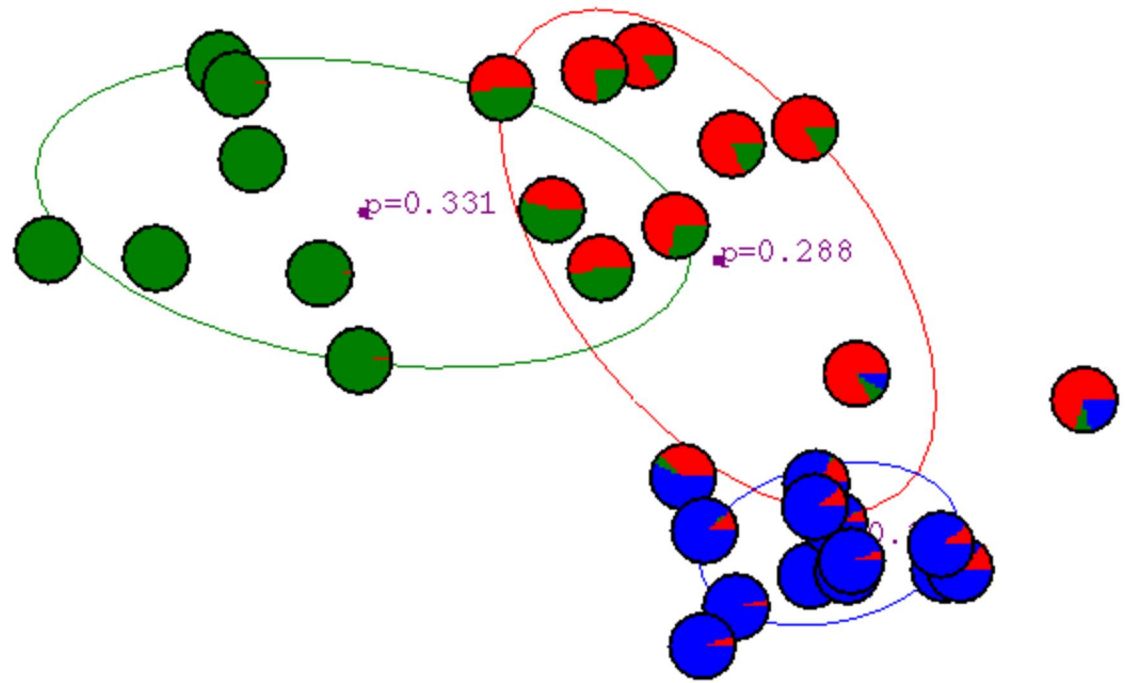
After 2nd iteration



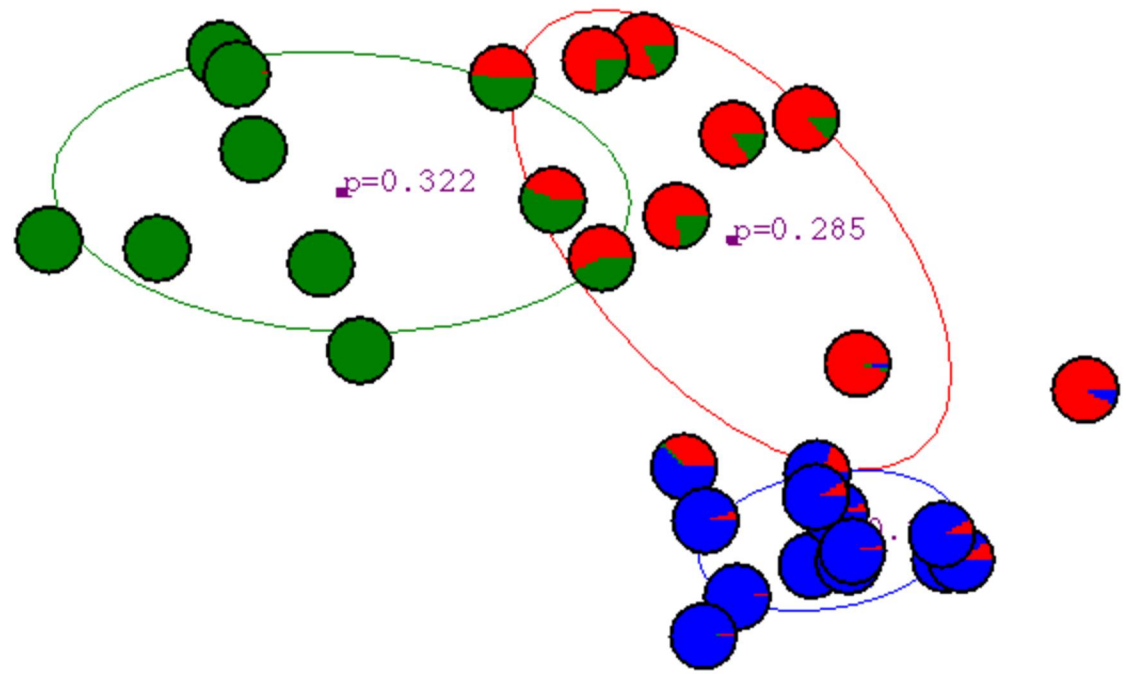
After 3rd iteration



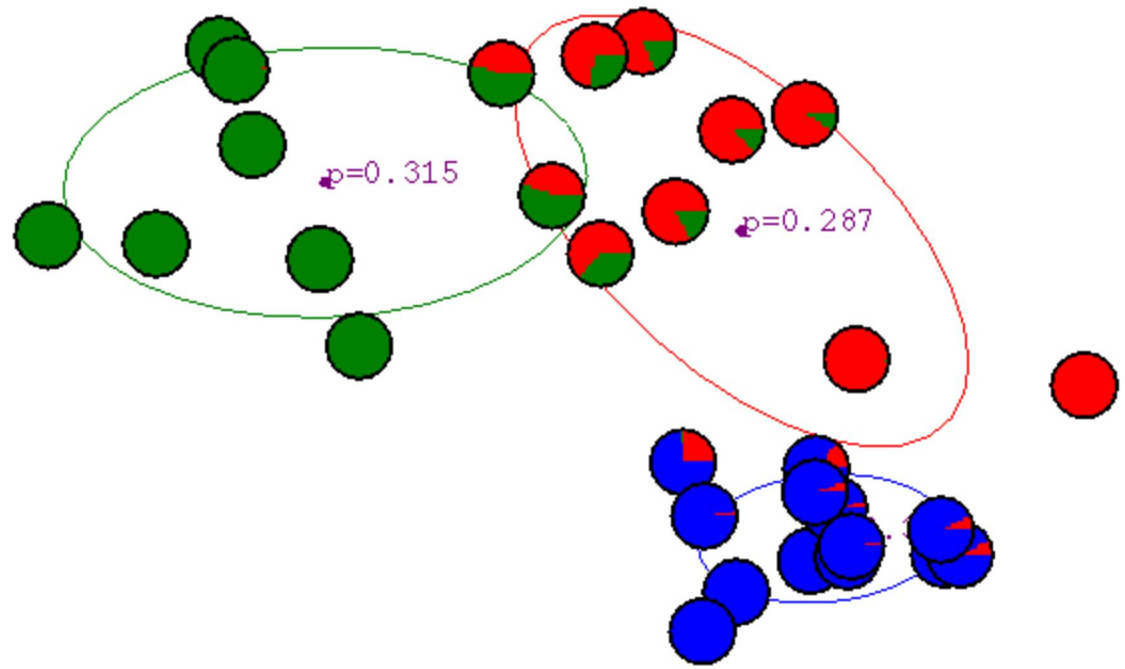
After 4th iteration



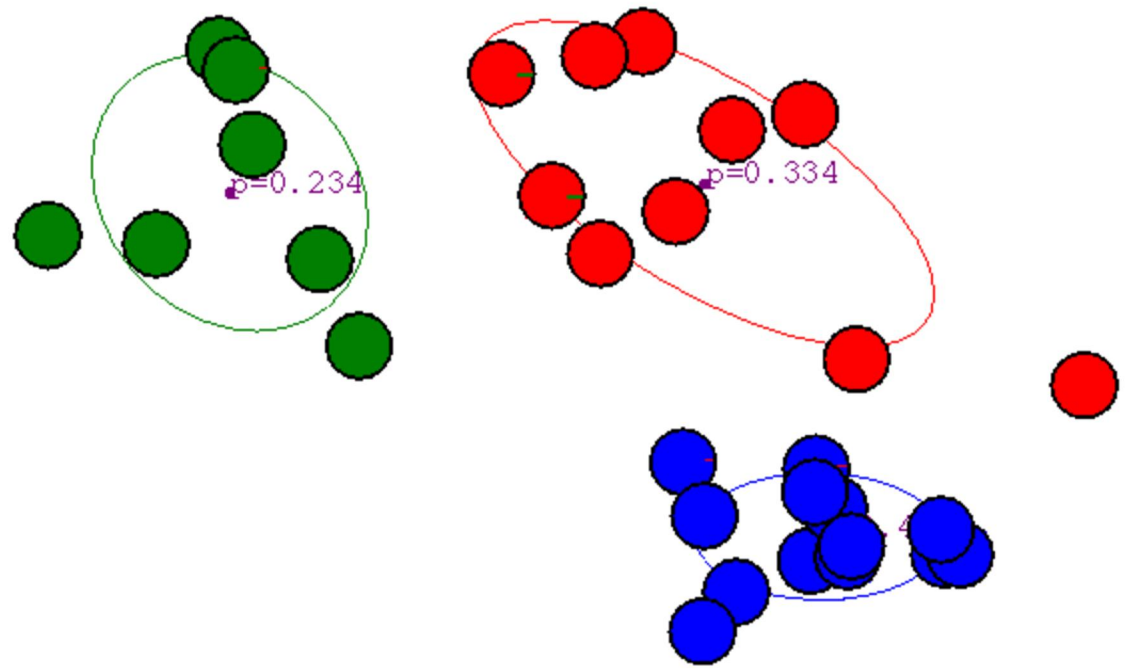
After 5th iteration



After 6th iteration



After 20th iteration



How to pick K?

- Probabilistic model $L = \log P(x_1 \ldots x_n) = \sum_{i=1}^{n} \log \sum_{k=1}^{K} P(x_i \mid k) P(k)$
  - tries to "fit" the data (maximize likelihood)
- Pick K that makes *L* as large as possible?
  - *K* = *n*: each data point has its own "source"
- Occam's razor: pick "simplest" of all models that fit
  - Bayes Inf. Criterion (BIC): $\max_p \{ L - \frac{1}{2} p \log n \}$
  - Akaike Inf. Criterion (AIC): $\min_p \{ 2p - L \}$

L … **likelihood**, how well our model fits the data
p … **number of parameters** how "simple" is the model

Summary

- Walked through 1-d version
  - works for higher dimensions
    - d-dimensional Gaussians, can be non-spherical
  - works for discrete data (text)
    - d-dimensional multinomial distributions (pLSI)
- Maximizes likelihood of the data: $P(x_1 \ldots x_n) = \prod_{i=1}^{n} \sum_{k=1}^{K} P(x_i \mid k) P(k)$
- Very similar to K-means
  - sensitive to starting point, converges to a local maximum
  - convergence: when change in $P(x_1 \ldots x_n)$ is sufficiently small
  - cannot discover K (likelihood keeps growing with K)
- Different from K-means

  -- Soft clustering: instance can come from multiple clusters

  -- Covariance: notion of "distance" changes over time