IAML: Mixture models and EM

Victor Lavrenko and Nigel Goddard
School of Informatics

Semester 1

Mixture models

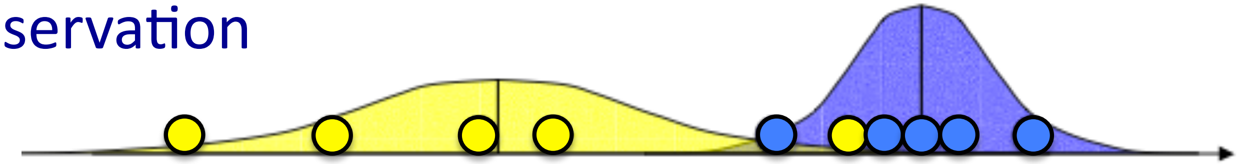
- Recall types of clustering methods
  - hard clustering: clusters do not overlap
    - element either belongs to cluster or it does not
  - soft clustering: clusters may overlap
    - stength of association between clusters and instances
- Mixture models
  - probabilistically-grounded way of doing soft clustering
  - each cluster: a generative model (Gaussian or multinomial)
  - parameters (e.g. mean/covariance are unknown)
- Expectation Maximization (EM) algorithm
  - automatically discover all parameters for the K "sources"

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

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

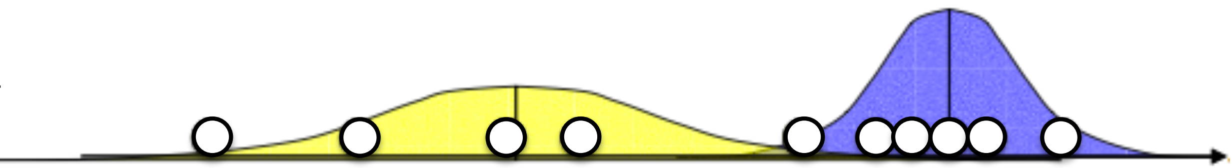
$$\sigma_b^2 = \frac{(x_1 - \mu_b)^2 + \ldots + (x_n - \mu_b)^2}{n_b}$$



- 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

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

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



Expectation Maximization (EM)

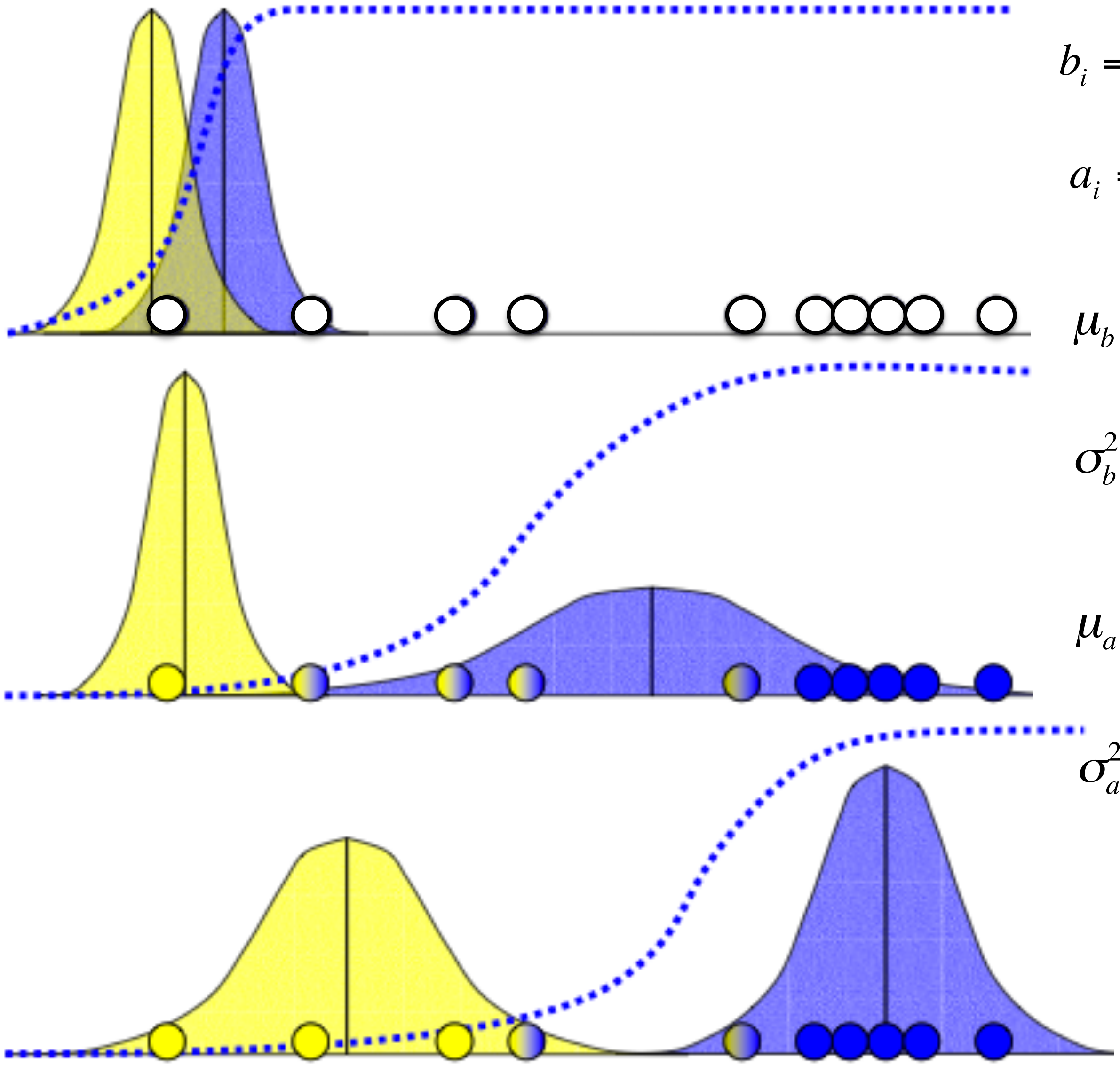
- Chicken and egg problem
  - need (μ_a, σ_a^2) and (μ_b, σ_b^2) to guess source of points
  - need to know source to estimate (μ_a, σ_a^2) and (μ_b, σ_b^2)
- EM algorithm
  - start with two randomly placed Gaussians (μ_a, σ_a^2), (μ_b, σ_b^2)
  - E-step: for each point: $P(b|x_i)$ = does it look like 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: 1-d example

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

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

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



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

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

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

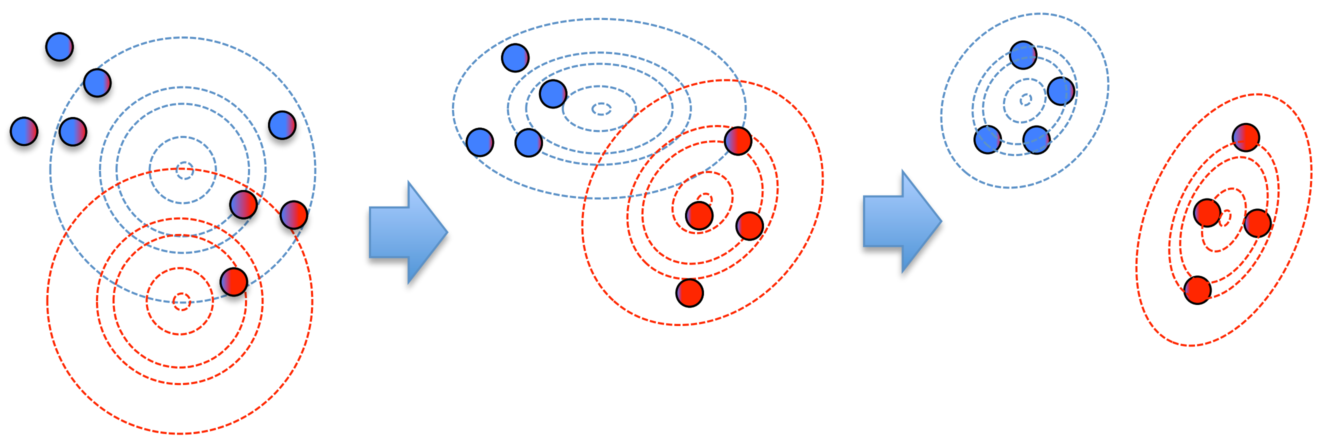
$$\sigma_a^2 = \frac{a_1(x_1 - \mu_a)^2 + \ldots + a_n(x_n - \mu_a)^2}{a_1 + a_2 + \ldots + a_n}$$

could also estimate priors:

P(b) = (b_1 + b_2 + … b_n) / n

P(a) = 1 – P(b)

Gaussian Mixture Model



- Data with D attributes, from Gaussian sources $c_1 \dots c_k$
  - how typical is $\mathbf{x}_i$ under source **c**
    $$P(\vec{x}_i \mid c) = \frac{1}{\sqrt{2\pi|\Sigma_c|}} \exp\left\{-\tfrac{1}{2}\underbrace{(\vec{x}_i - \vec{\mu}_c)^T \Sigma_c^{-1} (\vec{x}_i - \vec{\mu}_c)}_{\sum_a \sum_b (x_{ia} - \mu_{ca})\left[\Sigma_c^{-1}\right]_{ab} (x_{ib} - \mu_{cb})}\right\}$$
  - how likely that $\mathbf{x}_i$ came from **c**
    $$P(c \mid \vec{x}_i) = \frac{P(\vec{x}_i \mid c) P(c)}{\sum_{c=1}^{k} P(\vec{x}_i \mid c) P(c)}$$
  - how important is $\mathbf{x}_i$ for source **c**: $w_{i,c} = P(c \mid \vec{x}_i) \big/ \left(P(c \mid \vec{x}_1) + \ldots + P(c \mid \vec{x}_n)\right)$
  - mean of attribute **a** in items assigned to **c**: $\mu_{ca} = w_{c1} x_{1a} + \ldots + w_{cn} x_{na}$
  - covariance of **a** and **b** in items from **c**: $\Sigma_{cab} = \sum_{i=1}^{n} w_{ci} (x_{ia} - \mu_{ca})(x_{ib} - \mu_{cb})$
  - prior: how many items assigned to **c**: $P(c) = \frac{1}{n}\left(P(c \mid \vec{x}_1) + \ldots + P(c \mid \vec{x}_n)\right)$

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"
  - may not work well for new data points
- Split points into training set T and validation set V
  - for each *K*: fit parameters of T, measure likelihood of V
  - sometimes still best when *K* = *n*
- Occam's razor: pick "simplest" of all models that fit
  - Bayes Inf. Criterion (BIC): $\max_p \{ L - ½\, p \log n \}$
  - 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: $P(x_1 \ldots x_n) = \prod_{i=1}^{n} \sum_{k=1}^{K} P(x_i \mid k) P(k)$
- 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"
  - co-variance: notion of "distance" changes over time
- How can you make GMM = K-means?

$$L = \log \prod_{i=1}^{N} P(x_i) = \sum_{i=1}^{N} \log \sum_{k=1}^{K} P(k) \tfrac{1}{\sqrt{2\pi\sigma^2}} \exp\left\{ -\frac{(x_i - \mu_k)^2}{2\sigma^2} \right\}$$

$$\frac{\partial L}{\partial \mu_j} = \sum_{i=1}^{N} \frac{p_j N(x_i; \mu_j, \sigma_j^2)}{\sum_{k=1}^{K} p_k N(x_i; \mu_k, \sigma_k^2)} \left\{ + \frac{2(x_i - \mu_k)}{2\sigma_j^2} \right\}$$