Mixture Models and EM

If our data is unlabeled?

- If labeled data (training set) is given:
  - We can extract one class data.
  - We assume the parametric form of the distribution for the class of data. Eg: Gaussian.
  - We can employ maximum likelihood parameter estimation.
- This is what we saw in maximum likelihood parametric density estimation.

Two classes: *a* and *b*

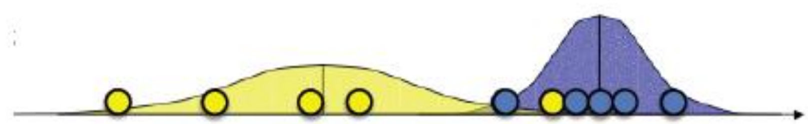
- Observations $x_1 \ldots x_n$
  - K=2 Gaussians with unknown μ, σ^2
  - estimation trivial if we know the source of each observation

Two classes: *a* and *b*

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

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

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



If our data is unlabeled?

- If we know the probability distribution from which the data is drawn,
  - We can label the data ..
  - By employing the Bayes classifier

If our data is unlabeled?

- Distributions are available.

That is, $P(a), P(b), p(x_i|a)$ and $p(x_i|b)$ are given.

Let $p(x_i|a) = \frac{1}{\sqrt{2\pi\sigma_a^2}} exp\left(-\frac{(x_i-\mu_a)^2}{2\sigma_a^2}\right)$, then

the posterior $P(a|x_i) = \frac{p(x_i|a)P(a)}{p(x_i)}$, and the posterior $P(b|x_i)$ can be used in finding the class label for x_i

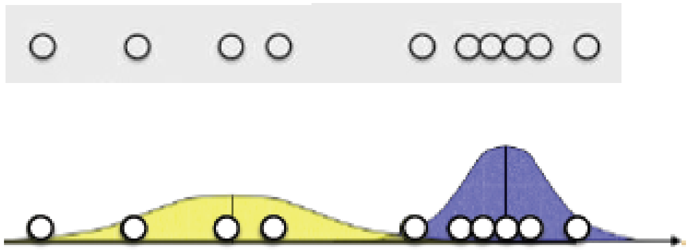
If our data is unlabeled?

- Distributions are available.

That is, $P(a), P(b), p(x_i|a)$ and $p(x_i|b)$ are given.

Let $p(x_i|a) = \frac{1}{\sqrt{2\pi\sigma_a^2}} exp\left(-\frac{(x_i-\mu_a)^2}{2\sigma_a^2}\right)$, then

the posterior $P(a|x_i) = \frac{p(x_i|a)P(a)}{p(x_i)}$, and the posterior $P(b|x_i)$ can be used in finding the class label for x_i



- Labels are needed to get distributions
- Distributions are needed to get labels.
-

- Labels are needed to get distributions
- Distributions are needed to get labels.
- Chicken and egg problem.

How the nature solved this chicken and egg problem?

- Neither chicken, nor egg was first!
- Both evolved over time.
- Initially very hazy distinction between them, but as time progressed it became two clear distinct things.
- So, we too employ this, but we call this solution **the EM algorithm**.
- Later, we learn that K-means clustering algorithm is a grandson of this algorithm.

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)
- Expectation Maximization (EM) algorithm
  - automatically discover all parameters for the K "sources"

EM with Gaussian assumptions becomes GMM.

Further, GMM, with more assumptions can become K-means ☺

GAUSSIAN MIXTURE MODEL (GMM)

Expectation Maximization (EM)

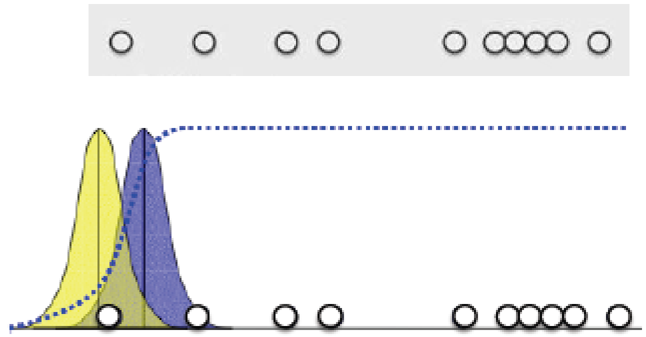
- Chicken and egg problem
  - need (μ_a, σ_a^2) and (μ_b, σ_b^2) to guess source of points
  - need to know source to estimate (μ_a, σ_a^2) and (μ_b, σ_b^2)

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).
  - While (not converged) do
    - E-step: Find $P(a|x_i), P(b|x_i)$ for each data element. This gives label for x_i. Fishy: This label is a random variable !
    - M-step: Adjust (μ_a, σ_a^2) and (μ_b, σ_b^2) to fit points assigned to them.

EM: 1-d example



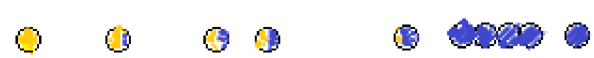
Source parameters are randomly fixed to begin with.

$$p(x_i|a) = \frac{1}{\sqrt{2\pi\sigma_a^2}} exp\left(-\frac{(x_i-\mu_a)^2}{2\sigma_a^2}\right)$$

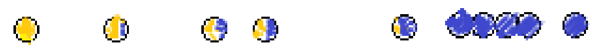
$$a_i = P(a|x_i) = \frac{p(x_i|a)P(a)}{p(x_i)}$$

$$b_i = 1 - a_i$$

(a_i, b_i) is the label for x_i



EM: 1-d example



$$\mu_a = \frac{a_1x_1 + a_2x_2 + \ldots + a_nx_{n_b}}{a_1 + a_2 + \ldots + a_n}$$

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

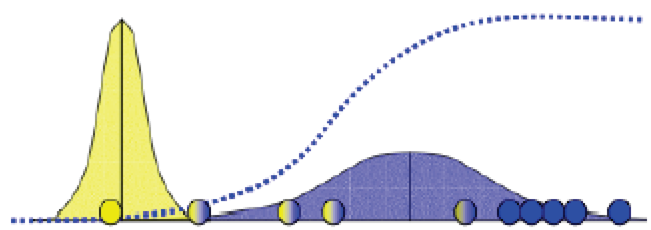
(a_i, b_i) is the label for x_i

Prior $P(a)$ can be estimated from $\frac{a_1 + a_2 + \cdots + a_n}{n}$

So, $P(b) = \frac{b_1 + b_2 + \cdots + b_n}{n}$

$$\mu_b = \frac{b_1x_1 + b_2x_2 + \ldots + b_nx_{n_b}}{b_1 + b_2 + \ldots + b_n}$$

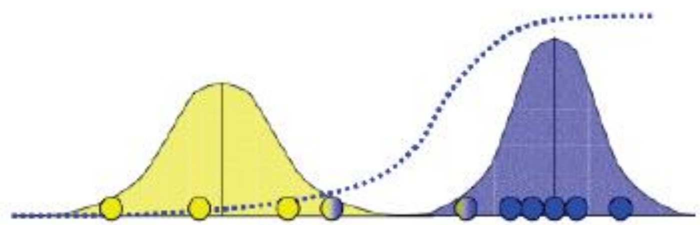
$$\sigma_b^2 = \frac{b_1(x_1 - \mu_1)^2 + \ldots + b_n(x_n - \mu_n)^2}{b_1 + b_2 + \ldots + b_n}$$



Now we are with a new estimation of the source.
A better estimate. Repeat till convergence.

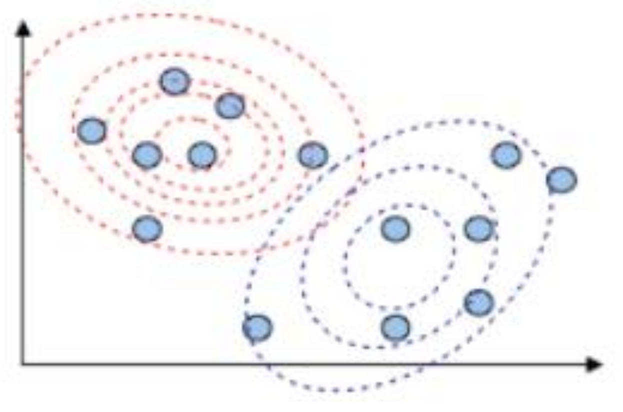
EM: 1-d example

after convergence, the output:



Extension to d > 1, c > 2

- Assume c component mixture (c classes).
- Start with randomly chosen means (randomly choose k distinct data elements).
- Similarly randomly chosen covariance matrix for each class. Usually we begin with identity matrix.



E-step: Find label for each $\boldsymbol{x_i}$. Let this be the random variable given by $(\boldsymbol{P_{1i}}, \boldsymbol{P_{2i}}, \dots, \boldsymbol{P_{ci}})$. This is done for each $\boldsymbol{x_i}, \boldsymbol{1 \leq i \leq n}$.

E-step: Find label for each $\boldsymbol{x_i}$. Let this be the random variable given by $(\boldsymbol{P_{1i}}, \boldsymbol{P_{2i}}, \dots, \boldsymbol{P_{ci}})$. This is done for each $\boldsymbol{x_i}, 1 \leq \boldsymbol{i} \leq \boldsymbol{n}$.

M-step: Let $\boldsymbol{\mu}^{(1)}$ be the mean of class 1. Then, $\boldsymbol{\mu}^{(1)} = \frac{\sum_{i=1}^{n} \boldsymbol{P_{1i} x_i}}{\sum_{i=1}^{n} \boldsymbol{P_{1i}}}$. Similarly mean vector for other classes, $\boldsymbol{\mu}^{(2)}, \dots, \boldsymbol{\mu}^{(c)}$ can be found.

E-step: Find label for each $\boldsymbol{x}_i$. Let this be the random variable given by $(\boldsymbol{P}_{1i}, \boldsymbol{P}_{2i}, \dots, \boldsymbol{P}_{ci})$. This is done for each $\boldsymbol{x}_i, 1 \le \boldsymbol{i} \le \boldsymbol{n}$.

M-step: Let $\boldsymbol{\mu}^{(1)}$ be the mean of class 1. Then, $\boldsymbol{\mu}^{(1)} = \frac{\sum_{i=1}^{n} \boldsymbol{P}_{1i} \boldsymbol{x}_i}{\sum_{i=1}^{n} \boldsymbol{P}_{1i}}$. Similarly mean vector for other classes, $\boldsymbol{\mu}^{(2)}, \dots, \boldsymbol{\mu}^{(c)}$ can be found.

Covariance Matrix for class 1, $\boldsymbol{\Sigma}^{(1)} = \frac{\sum_{i=1}^{n} \boldsymbol{P}_{1i} \left(\boldsymbol{x}_i - \boldsymbol{\mu}^{(1)}\right) \left(\boldsymbol{x}_i - \boldsymbol{\mu}^{(1)}\right)^{t}}{\sum_{i=1}^{n} \boldsymbol{P}_{1i}}$. Similarly covariance matrix for other classes, $\boldsymbol{\Sigma}^{(2)}, \dots, \boldsymbol{\Sigma}^{(c)}$ can be found.

- We stop EM algorithm here.
- In exams, I can ask some numeric problem for 1D two class case. {Do not worry about multidimensional problem (as of now)}.

- Theoretically, the iterative process can get stuck in a local maximum.

K-means is an approximation of GMM

- Initially pick k distinct random seed points (in GMM: the set of initial mean vectors)
- We assume that

$$\boldsymbol{\Sigma}^{(1)} = \boldsymbol{\Sigma}^{(2)} = \cdots = \boldsymbol{\Sigma}^{(c)} = \boldsymbol{I}$$

- The Bayes classifier becomes "the minimum distance classifier".
- Let the label be deterministic (not a random variable). Choose the nearest's mean's label (this is what the minimum distance classifier will do).
- GMM becomes k-means clustering algorithm.


K-Means: Grandson
GMM: Grandfather