Bayesian Models in Machine Learning

Lukáš Burget

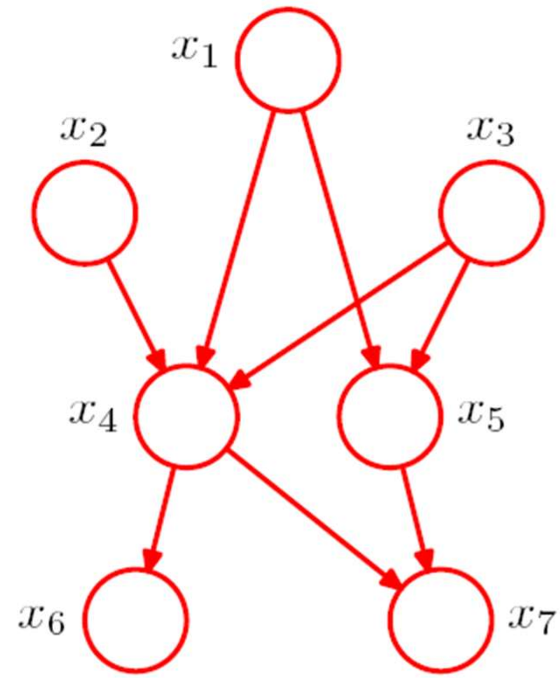
BRNO UNIVERSITY OF TECHNOLOGY FACULTY OF INFORMATION TECHNOLOGY

Escuela de Ciencias Informáticas 2017

Buenos Aires, July 24-29 2017

Bayesian Networks

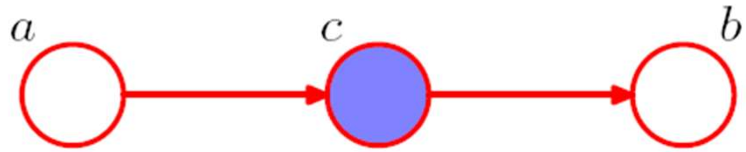
- The graph corresponds to a particular factorization of a joint probability distribution over a set of random variables
- Nodes are random variables, but the graph does not say what are the distributions of the variables
- The graph represents a set of distributions that conform to the factorization
- It is recipe for building more complex models out of simpler probability distributions
- **Describes the generative process**
- **Generally no closed form solutions for inferences in such models**



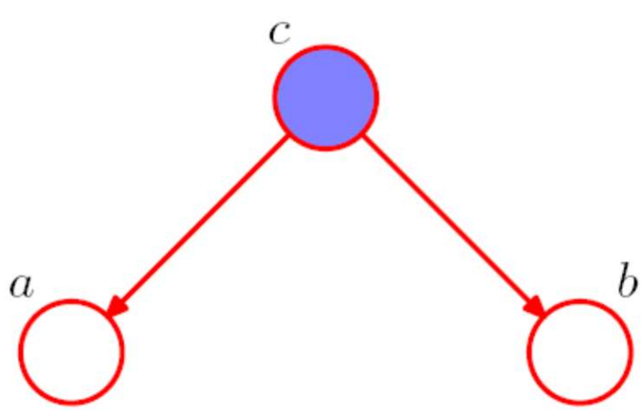
$$p(x_1, \ldots, x_7) = p(x_1)p(x_2)p(x_3)p(x_4|x_1, x_2, x_3)p(x_5|x_1, x_3)p(x_6|x_4)p(x_7|x_4, x_5)$$

Conditional independence

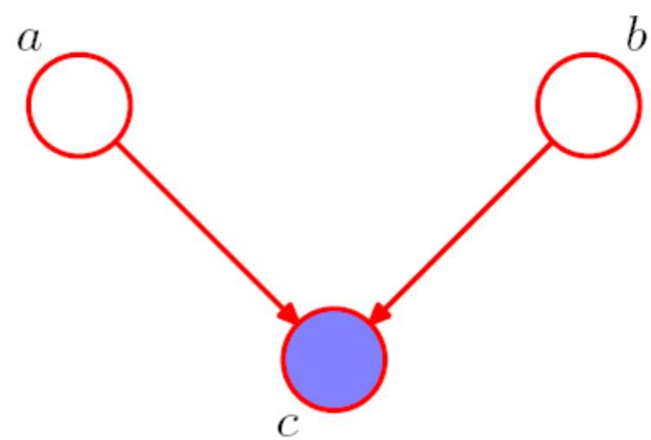
- Bayesian Networks allow us to see conditional independence properties.
- Blue nodes corresponds to observed random variables and empty nodes to latent (or hidden) random variables



$$p(a, b) \neq p(a)p(b)$$
$$p(a, b|c) = p(a|c)p(b|c)$$

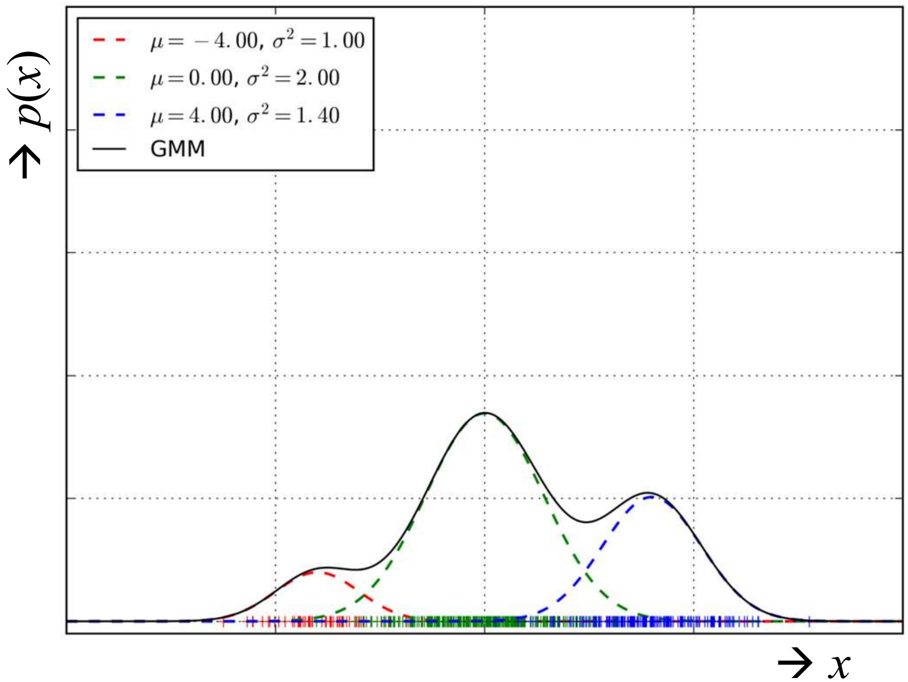


But the opposite is true for:



Gaussian Mixture Model (GMM)

$$p(x|\boldsymbol{\eta}) = \sum_c \mathcal{N}(x; \mu_c, \sigma_c^2)\pi_c$$



where

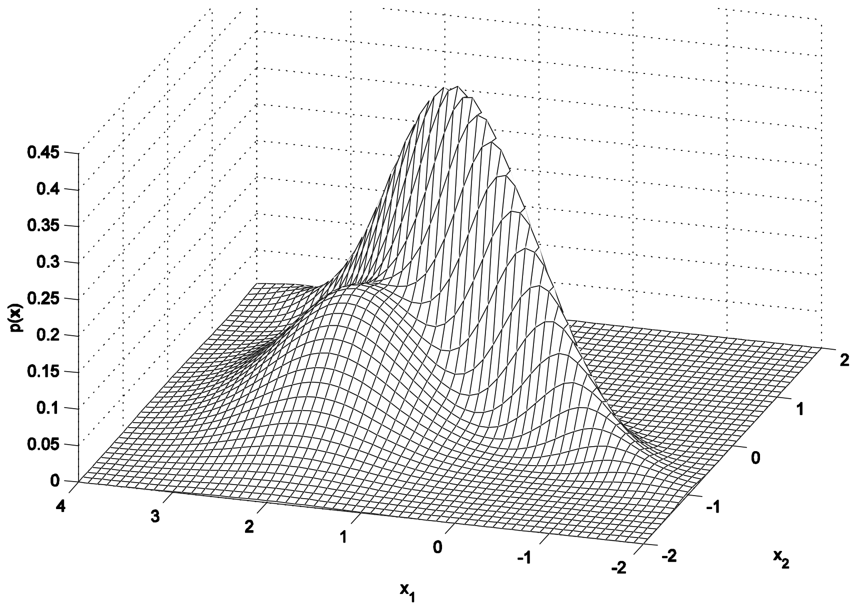
$$\boldsymbol{\eta} = \{\pi_c, \mu_c, \sigma_c^2\}$$

$$\sum_c \pi_c = 1$$

- We can see the sum above just as a function defining the shape of the probability density function
- or …

Multivariate GMM

$$p(\mathbf{x}|\boldsymbol{\eta}) = \sum_c \mathcal{N}(\mathbf{x}; \boldsymbol{\mu}_c, \boldsymbol{\Sigma}_c)\pi_c$$



where

$$\boldsymbol{\eta} = \{\pi_c, \boldsymbol{\mu}_c, \boldsymbol{\Sigma}_c\}$$

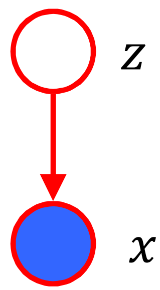
$$\sum_c \pi_c = 1$$

- We can see the sum above just as a function defining the shape of the probability density function
- or …

Gaussian Mixture Model

$$p(x) = \sum_z p(x|z)P(z) = \sum_c \mathcal{N}(x; \mu_c, \sigma_c^2)\mathrm{Cat}(z = c|\boldsymbol{\pi})$$

- or we can see it as a generative probabilistic model described by Bayesian network with **Categorical** latent random variable z identifying **Gaussian** distribution generating the observation x

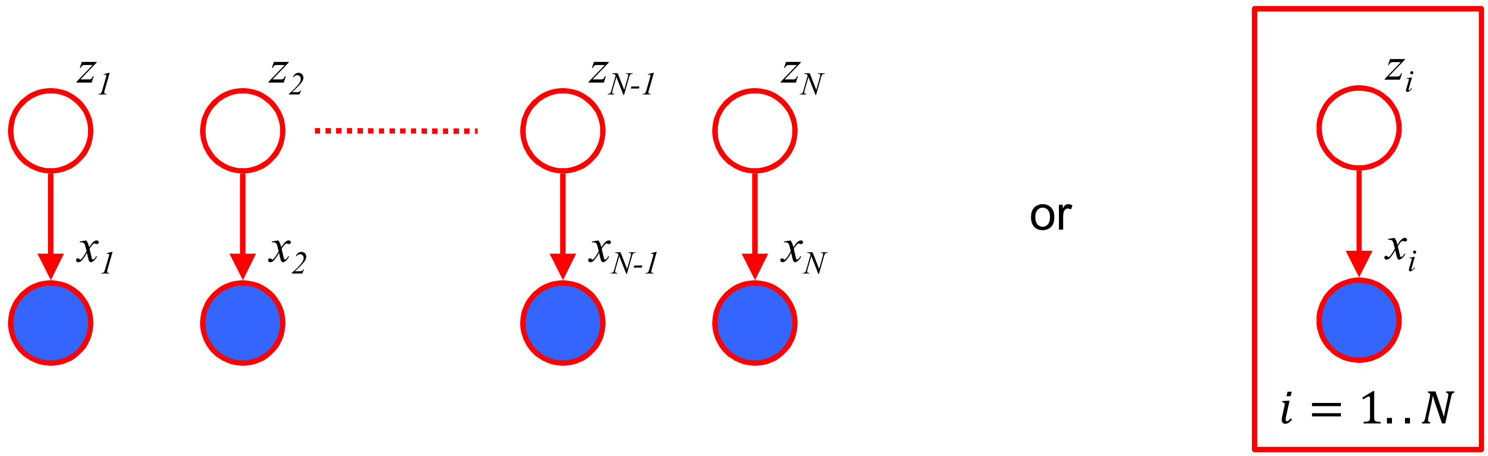


$$p(x, z) = p(x|z)P(z)$$

- Observations are assumed to be generated as follows:
  - randomly select Gaussian component according probabilities $P(z)$
  - generate observation x form the selected Gaussian distribution
- To evaluate $p(x)$, we have to marginalize out z
- No close form solution for training

Bayesian Networks for GMM

- Multiple observations:

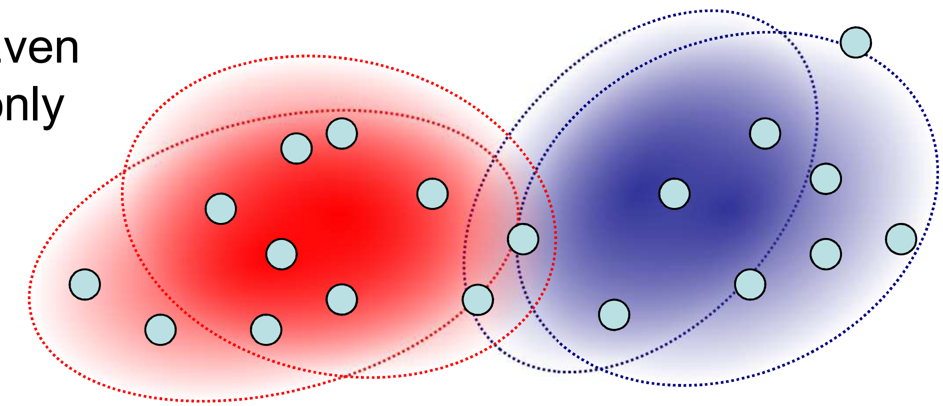


$$p(x_1, x_2, \ldots, x_\mathrm{N}, z_1, z_2, \ldots z_\mathrm{N}) = \prod_{i=1}^{N} p(x_\mathrm{i}|z_\mathrm{i}) P(z_i)$$

Training GMM –Viterbi training

- Intuitive and Approximate iterative algorithm for training GMM parameters.

- Using current model parameters, let Gaussians classify data as if the Gaussians were different classes (Even though all the data corresponds to only one class modeled by the GMM)



- Re-estimate parameters of Gaussians using the data assigned to them in the previous step. New weights will be proportional to the number of data points assigned to the Gaussians.

- Repeat the previous two steps until the algorithm converges.

Training GMM – EM algorithm

- **Expectation Maximization** is a general tool applicable do different generative models with latent (hidden) variables.
- Here, we only see the result of its application to the problem of re-estimating GMM parameters.
- It guarantees to increase the likelihood of training data in every iteration, however, it does not guarantee to find the global optimum.
- The algorithm is very similar to the Viterbi training presented above. However, instead of hard alignments of frames to Gaussian components, the posterior probabilities $P(c|x_i)$ (calculated given the old model) are used as soft weights. Parameters μ_c,σ_c^2 are then calculated using a weighted average.

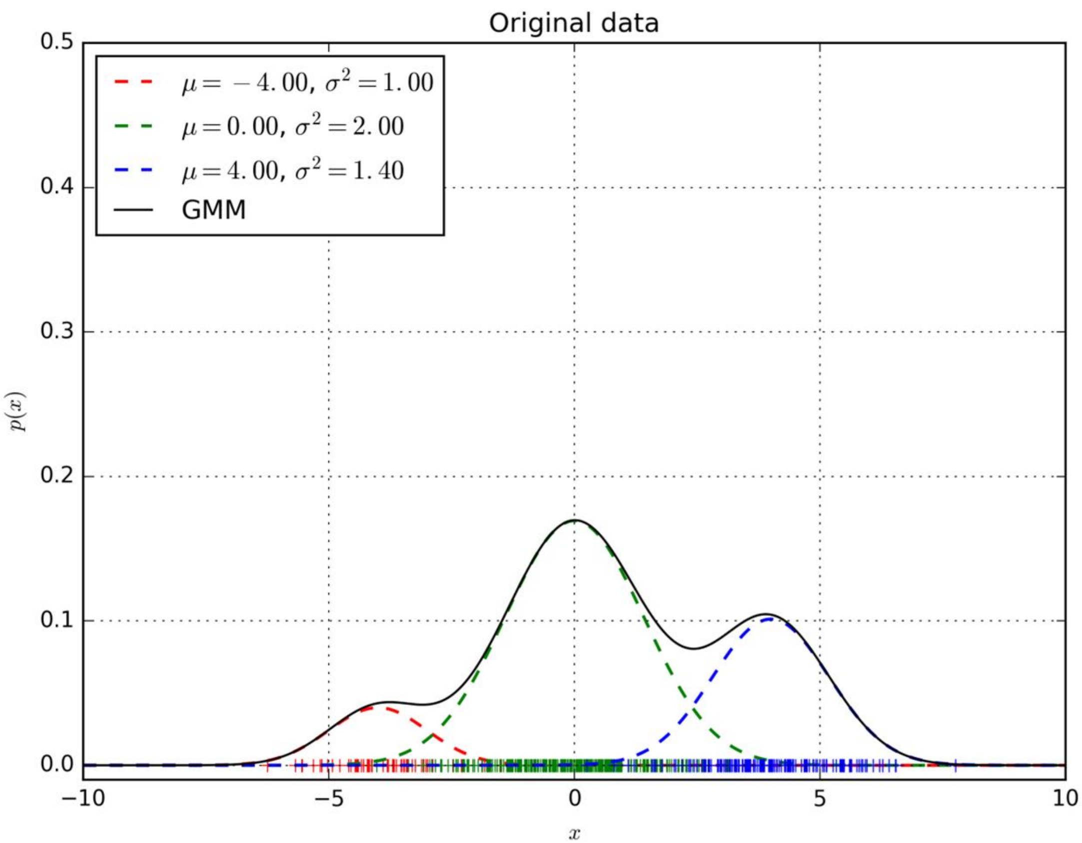
$$\gamma_{zi} = \frac{\mathcal{N}\left(x_i | \mu_{z_i}^{(old)}, {\sigma^2}_{z_i}^{(old)}\right) \pi_{z_i}^{(old)}}{\sum_k \mathcal{N}\left(\mathrm{x}_i | \mu_k^{(old)}, {\sigma^2}_k^{(old)}\right) \pi_k^{(old)}} = \frac{p(\mathrm{x}_i|z_i)P(z_i)}{\sum_k p(x_i|z_i = k)P(z_i = k)} = P(z_i|\mathrm{x}_i)$$

$$\mu_k^{(new)} = \frac{1}{\sum_i \gamma_{ki}} \sum\nolimits_i \gamma_{ki} \mathrm{x}_i$$

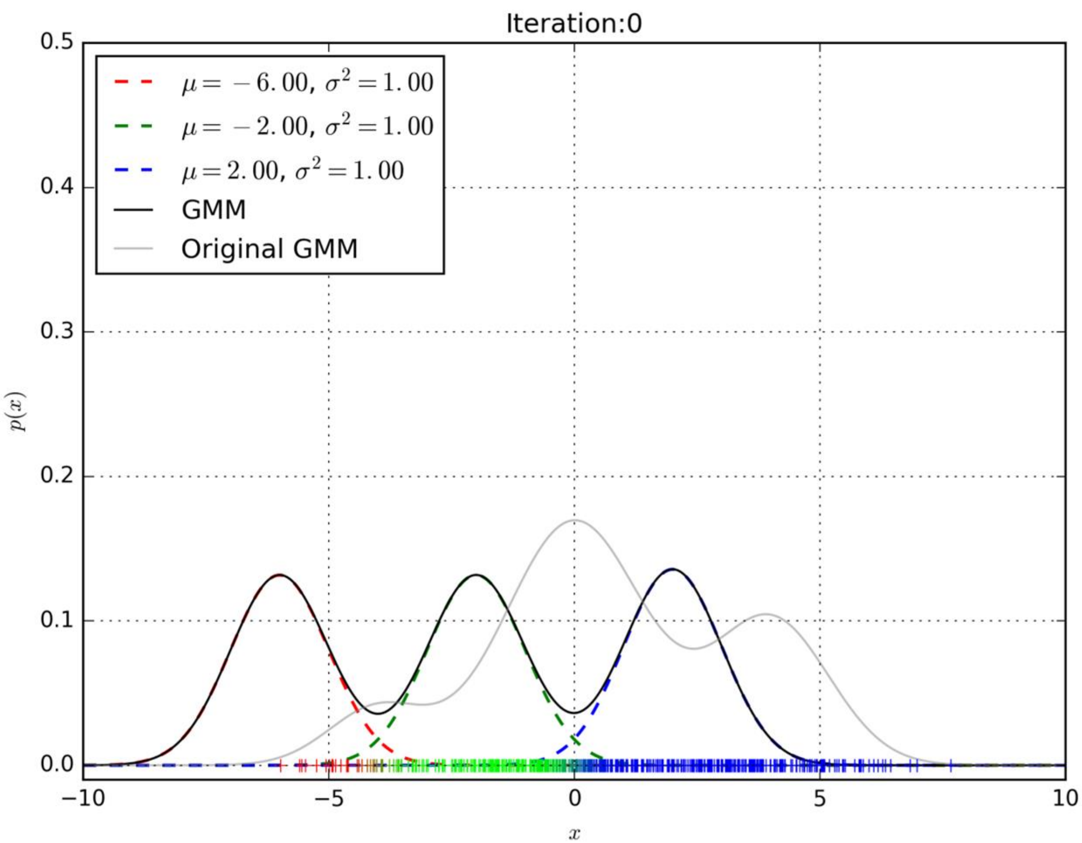
$$\pi_k^{(new)} = \frac{\sum_i \gamma_{ki}}{\sum_k \sum_i \gamma_{ki}} = \frac{\sum_i \gamma_{ki}}{N}$$

$${\sigma^2}_{\mathrm{z}}^{(new)} = \frac{1}{\sum_i \gamma_{ki}} \sum\nolimits_i \gamma_{ki} (\mathrm{x_i} - \mu_k)^2$$

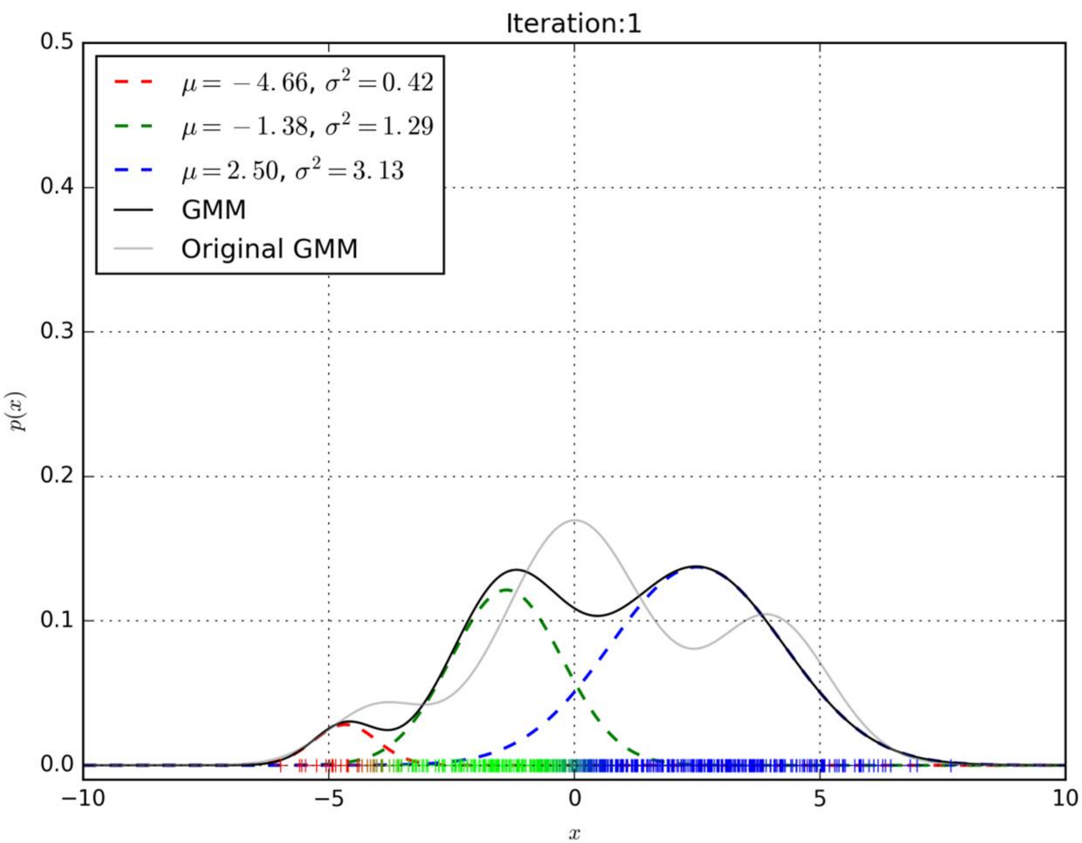
GMM to be learned



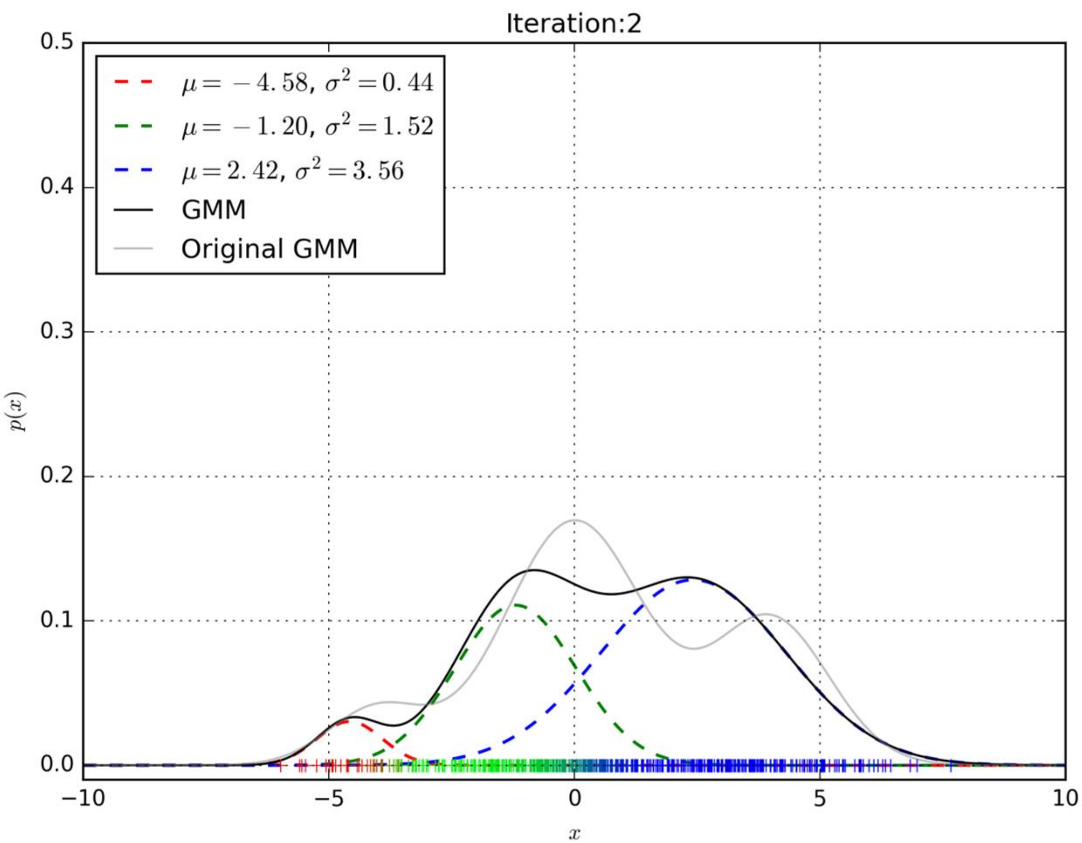
EM algorithm



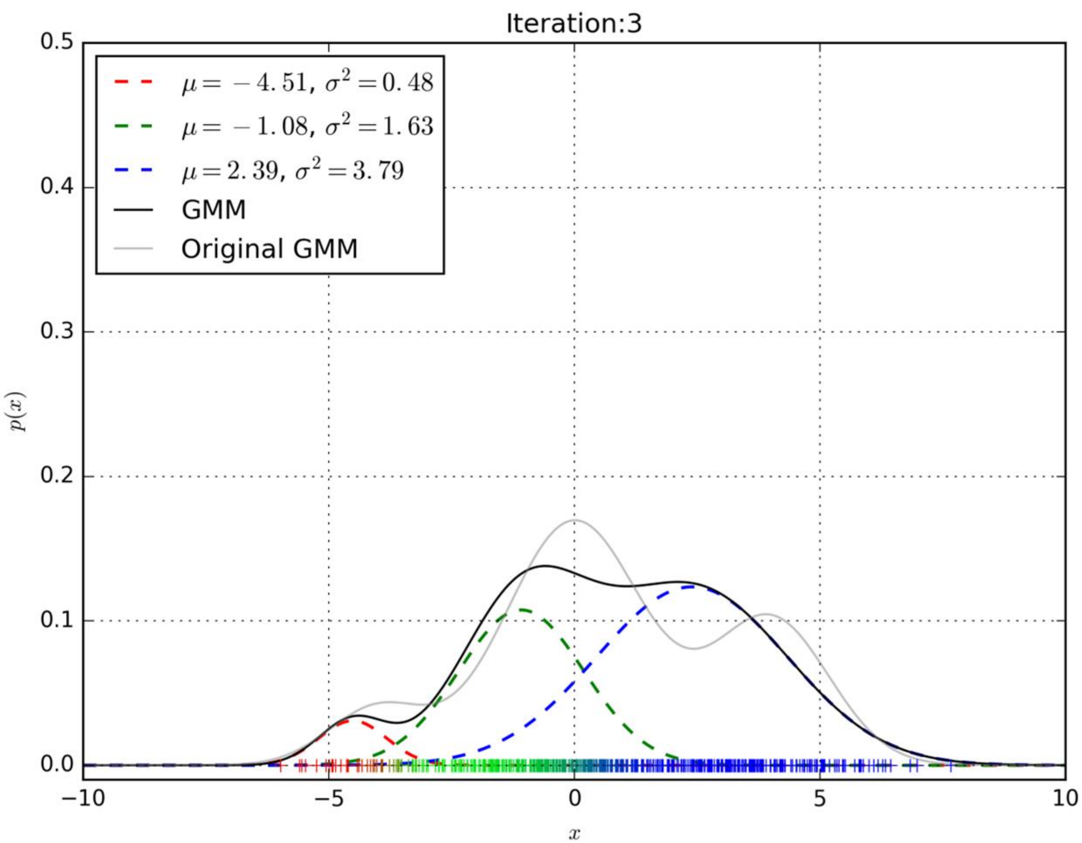
EM algorithm



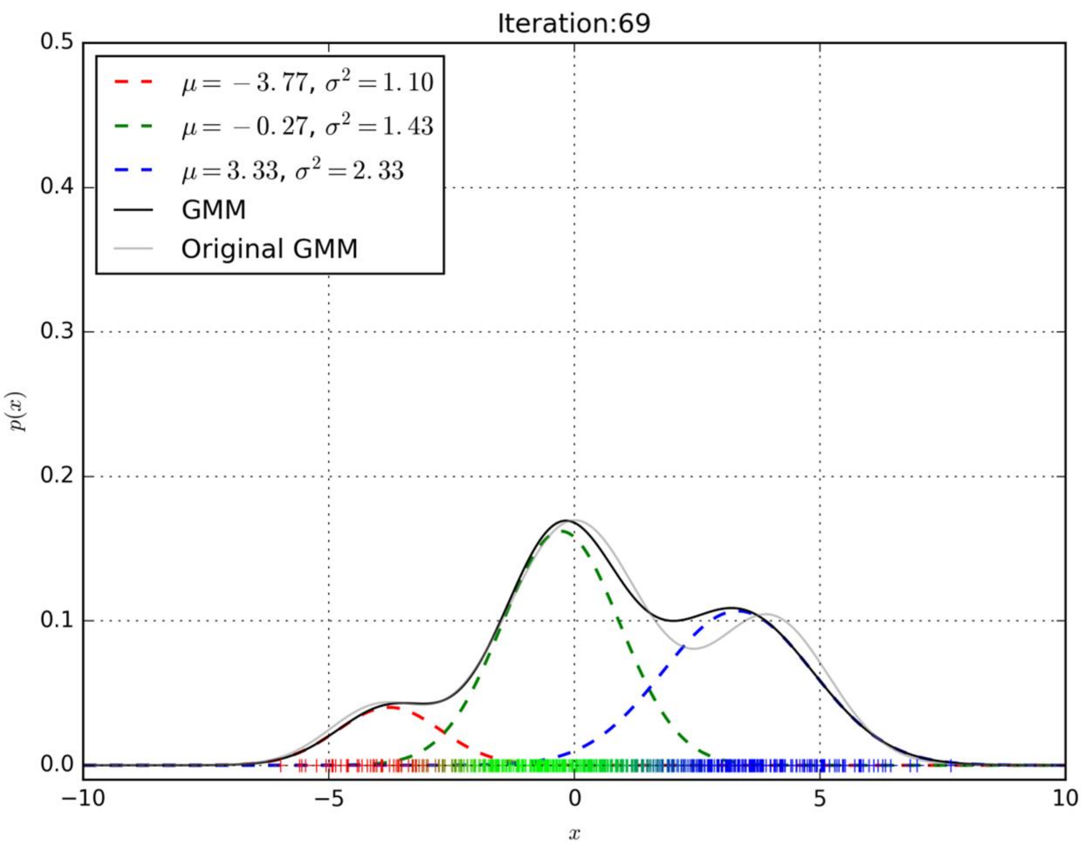
EM algorithm



EM algorithm



EM algorithm



Expectation maximization algorithm

$$\ln p(\mathbf{X}|\boldsymbol{\eta}) = \underbrace{\left(\sum_{\mathbf{Z}} q(\mathbf{Z})\right)}_{=1} \ln \underbrace{\frac{p(\mathbf{X},\mathbf{Z}|\boldsymbol{\eta})}{p(\mathbf{Z}|\mathbf{X},\boldsymbol{\eta})}}_{=p(\mathbf{X}),\forall \mathbf{Z}} \underbrace{\frac{q(\mathbf{Z})}{q(\mathbf{Z})}}_{=1} = \sum_{\mathbf{Z}} q(\mathbf{Z}) \ln \frac{p(\mathbf{X},\mathbf{Z}|\boldsymbol{\eta})q(\mathbf{Z})}{p(\mathbf{Z}|\mathbf{X},\boldsymbol{\eta})q(\mathbf{Z})}$$

$$= \underbrace{\underbrace{\sum_{\mathbf{Z}} q(\mathbf{Z}) \ln p(\mathbf{X},\mathbf{Z}|\boldsymbol{\eta})}_{\mathcal{Q}(q(\mathbf{Z}),\boldsymbol{\eta})} \underbrace{- \sum_{\mathbf{Z}} q(\mathbf{Z}) \ln q(\mathbf{Z})}_{H(q(\mathbf{Z}))}}_{\mathcal{L}(q(\mathbf{Z}),\boldsymbol{\eta})} \underbrace{- \sum_{\mathbf{Z}} q(\mathbf{Z}) \ln \frac{p(\mathbf{Z}|\mathbf{X},\boldsymbol{\eta})}{q(\mathbf{Z})}}_{D_{KL}(q(\mathbf{Z})||p(\mathbf{Z}|\mathbf{X},\boldsymbol{\eta}))}$$

- where $q(\mathbf{Z})$ is any distribution over the latent variable
- Kullback-Leibler divergence $\mathrm{D}_{\mathrm{KL}}(q||p)$ measures “unsimilarity” between two distributions q, p
- $D_{KL}(q||p) \geq 0$ and $D_{\mathrm{KL}}(q||p) = 0 \Leftrightarrow q = p$
- $\Rightarrow$ Evidence lower bound (**ELBO**) $\mathcal{L}(q(\mathbf{Z}),\boldsymbol{\eta}) \leq p(\mathbf{X}|\boldsymbol{\eta})$
- $H(q(\mathbf{Z}))$ is (non-negative) Entropy of distribution $q(\mathbf{Z})$
- $\mathcal{Q}(q(\mathbf{Z}),\boldsymbol{\eta})$ is called auxiliary function.

Expectation maximization algorithm

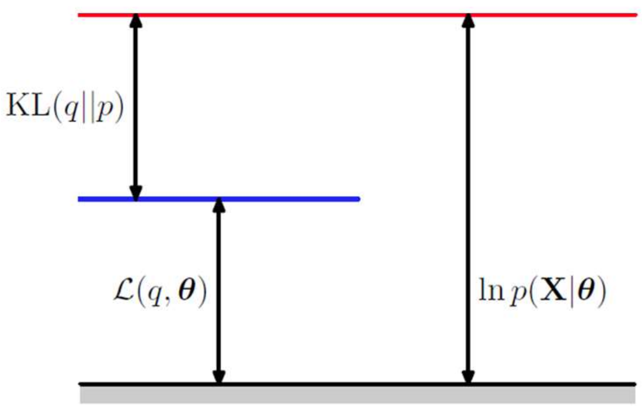
$$\ln p(\mathbf{X}|\boldsymbol{\eta}) = \underbrace{\mathcal{Q}(q(\mathbf{Z}), \boldsymbol{\eta}) + H(q(\mathbf{Z}))}_{\mathcal{L}(q(\mathbf{Z}),\boldsymbol{\eta})} + D_{KL}\left(q(\mathbf{Z})||p(\mathbf{Z}|\mathbf{X}, \boldsymbol{\eta})\right)$$

- We aim to find parameters $\boldsymbol{\eta}$ that maximize $\ln p(\mathbf{X}|\boldsymbol{\eta})$
- E-step: $q(\mathbf{Z}) \coloneqq P(\mathbf{Z}|\mathbf{X}, \boldsymbol{\eta}^{old})$
  – makes the $D_{\mathrm{KL}}(q||p)$ term 0
  – makes $\mathcal{L}(q(\mathbf{Z}), \boldsymbol{\eta}) = \ln p(\mathbf{X}|\boldsymbol{\eta})$

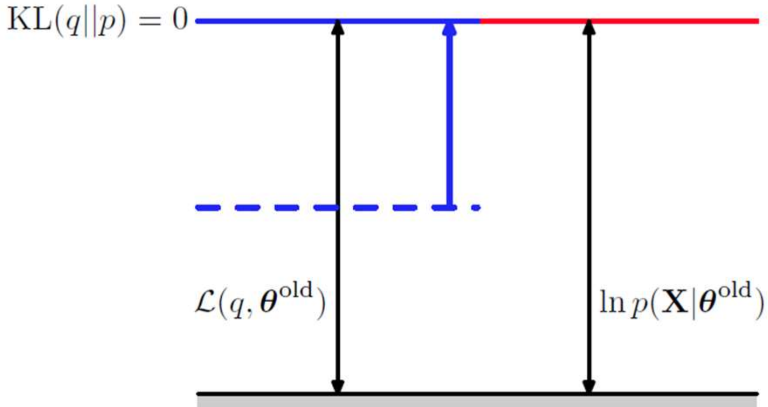
- M-step: $\boldsymbol{\eta}^{new} = \arg\max_{\boldsymbol{\eta}} \mathcal{Q}(q(\mathbf{Z}), \boldsymbol{\eta})$
  – $D_{KL}(q||p)$ increases as $P(\mathbf{X}|\mathbf{Z}, \boldsymbol{\eta})$deviates from $q(\mathbf{Z})$
  – $H\big(q(\mathbf{Z})\big)$ does not change for fixed $q(\mathbf{Z})$
  – $\mathcal{L}(q(\mathbf{Z}), \boldsymbol{\eta})$ increases like $\mathcal{Q}(q(\mathbf{Z}), \boldsymbol{\eta})$
  – $\ln p(\mathbf{X}|\boldsymbol{\eta})$ increases more than $\mathcal{Q}(q(\mathbf{Z}), \boldsymbol{\eta})$

Expectation maximization algorithm

$$\ln p(\mathbf{X}|\boldsymbol{\eta}) = \underbrace{\mathcal{Q}(q(\mathbf{Z}),\boldsymbol{\eta}) + H(q(\mathbf{Z}))}_{\mathcal{L}(q(\mathbf{Z}),\boldsymbol{\eta})} + D_{KL}\left(q(\mathbf{Z})||p(\mathbf{Z}|\mathbf{X},\boldsymbol{\eta})\right)$$

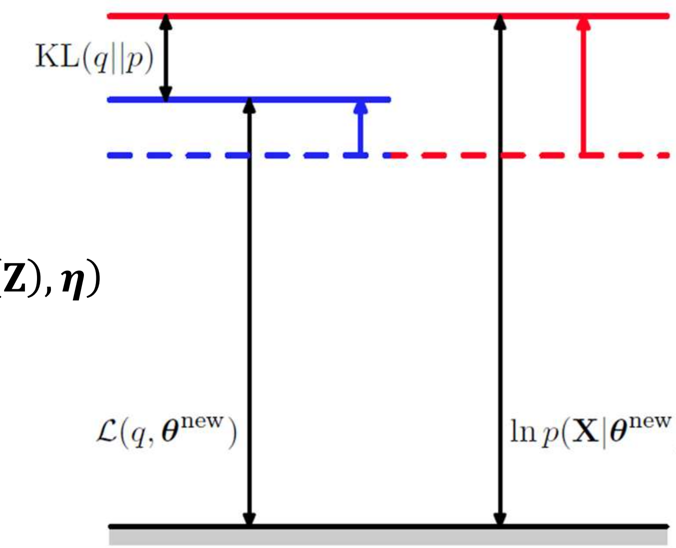


⇩ E-step: $q(\mathbf{Z}) \coloneqq P(\mathbf{Z}|\mathbf{X},\boldsymbol{\eta}^{old})$



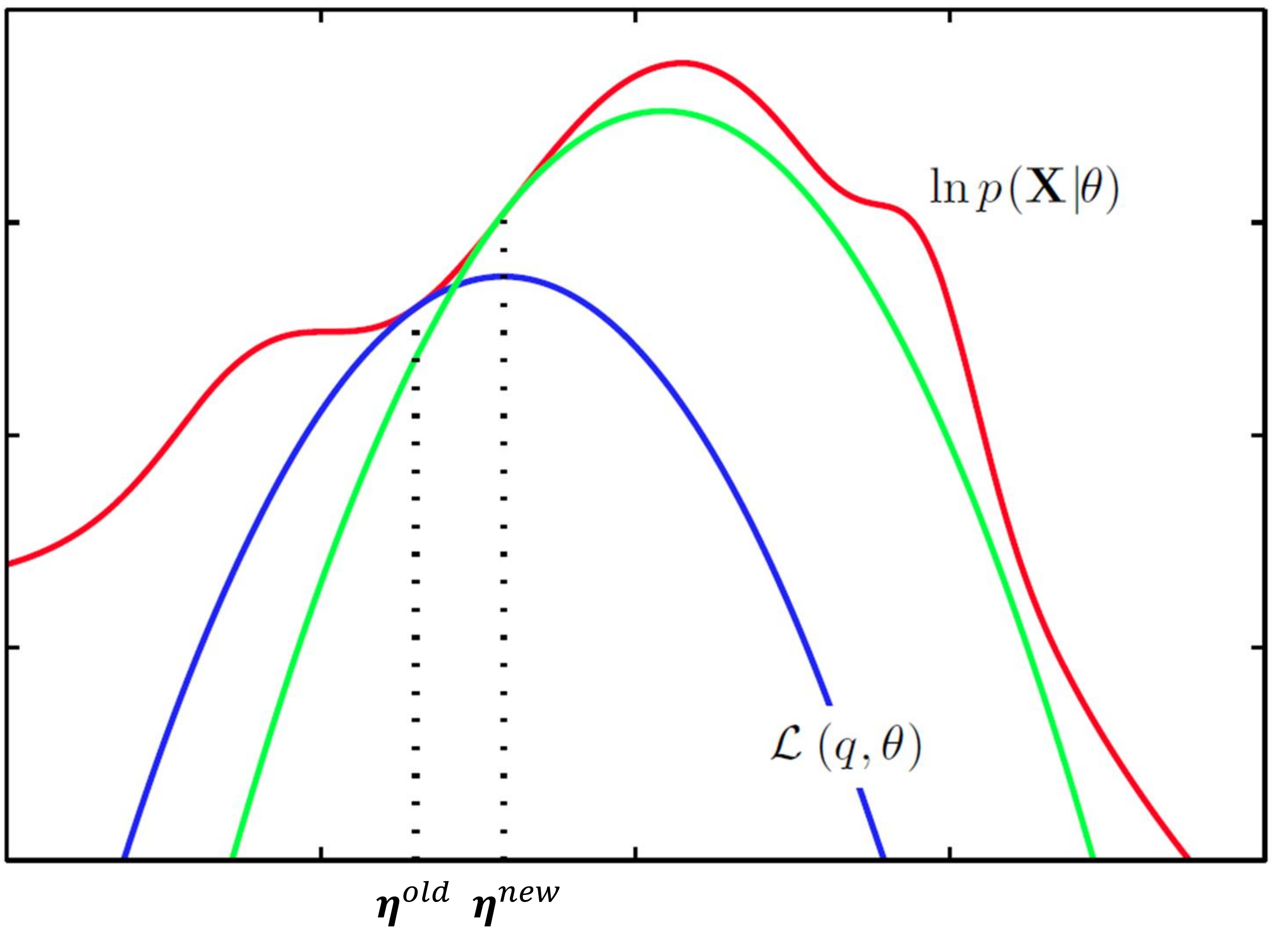
M-step: ⇨

$$\boldsymbol{\eta}^{new} = \arg\max_{\boldsymbol{\eta}} \mathcal{Q}(q(\mathbf{Z}),\boldsymbol{\eta})$$



Expectation maximization algorithm

$\mathcal{Q}(q(\mathbf{Z}), \boldsymbol{\eta})$ and $\mathcal{L}(q(\mathbf{Z}), \boldsymbol{\eta})$ will be easy to optimize (e.g. quadratic function) compared to $\ln p(\mathbf{X}|\boldsymbol{\eta})$



EM for GMM

- Now, we aim to train parameters $\boldsymbol{\eta} = \{\mu_z, \sigma_z^2, \pi_z\}$ of Gaussian Mixture model

$$p(x) = \sum_z p(x|z)P(z) = \sum_c \mathcal{N}(x; \mu_c, \sigma_c^2)\mathrm{Cat}(z = c|\boldsymbol{\pi})$$

- Given training observations $\mathbf{x} = [x_1, x_2, \dots, x_N]$ we search for ML estimate of $\boldsymbol{\eta}$ that maximizes log likelihood of the training data.

$$\ln p(\mathbf{x}) = \sum_n \ln p(x_n) = \sum_n \left[\ln \sum_c \mathcal{N}(x_n; \mu_c, \sigma_c^2) + \ln \pi_c\right]$$

- Direct maximization of this objective function w.r.t. $\boldsymbol{\eta}$ is intractable.
- We will use EM algorithm, where we maximize the auxiliary function which is (for simplicity) sum of per-observation auxiliary functions

$$Q(q(\mathbf{z}), \boldsymbol{\eta}) = \sum_n Q_n(q(z_n), \boldsymbol{\eta})$$

- Again, in M-step $\sum_n \ln p(x_n)$ has to increase more than $\sum_n Q_n(q(z_n), \boldsymbol{\eta})$

EM for GMM – E-step

$$q(z_n) = P(z_n | x_n, \boldsymbol{\eta}^{old})$$
$$= \frac{p(x_n | z_n, \boldsymbol{\eta}^{old}) P(z_n | \boldsymbol{\eta}^{old})}{p(x_n | \boldsymbol{\eta}^{old})}$$

$$q(z_n = c) = \frac{\mathcal{N}(x_n | \mu_c^{old} \sigma_c^{2old}) \pi_c^{old}}{\sum_k \mathcal{N}(x_n | \mu_k^{old}, \sigma_k^{2old}) \pi_k^{old}}$$
$$= \gamma_{nc}$$

- γ_{nc} is the so called responsibility of Gaussian component z for observation n.
- It is the probability for an observation n being generated from component c

EM for GMM – M-step

$$
\begin{aligned}
Q(q(\mathbf{z}), \boldsymbol{\eta}) &= \sum_n Q_n(q(z_n), \boldsymbol{\eta}) \\
&= \sum_n \sum_{z_n} q(z_n) \ln p(x_n, z_n | \boldsymbol{\eta}) \\
&= \sum_n \sum_c \gamma_{nc} \left[\ln \mathcal{N}(x_n; \mu_c, \sigma_c) + \ln \pi_c\right]
\end{aligned}
$$

- In M-step, the auxiliary function is maximized w.r.t. all GMM parameters

EM for GMM –update of means

- Update for component mean means:

$$\frac{\partial}{\partial \mu_c} \sum_n Q_n(q(z_n), \boldsymbol{\eta}) = \frac{\partial}{\partial \mu_c} \sum_n \sum_k \gamma_{nk} \left[\ln \mathcal{N}(x_n; \mu_k, \sigma_k^2) + \ln \pi_k \right]$$

$$= \frac{\partial}{\partial \mu_c} \sum_n \gamma_{nc} \left[-\frac{(x_n - \mu_c)^2}{2\sigma_c^2} + K \right]$$

$$= \frac{1}{\sigma_c^2} \sum_n \gamma_{nc} (\mu_c - x_n) = 0$$

$$\Longrightarrow \mu_c = \frac{\sum_n \gamma_{nc} x_n}{\sum_n \gamma_{nc}}$$

- Update for variances can be derived similarly.

Flashback: ML estimate for Gaussian

$$\arg\max_{\mu,\sigma^2} p(\mathbf{x}|\mu,\sigma^2) = \arg\max_{\mu,\sigma^2} \log p(\mathbf{x}|\mu,\sigma^2) = \sum_i \log \mathcal{N}(x_i;\mu,\sigma^2)$$

$$= -\frac{1}{2\sigma^2}\sum_i x_i^2 + \frac{\mu}{\sigma^2}\sum_i x_i - N\frac{\mu^2}{2\sigma^2} - \frac{\log(2\pi)}{2}$$

$$\frac{\partial}{\partial\mu}\log p(\mathbf{x}|\mu,\sigma^2) = \frac{\partial}{\partial\mu}\left(-\frac{1}{2\sigma^2}\sum_i x_i^2 + \frac{\mu}{\sigma^2}\sum_i x_i - N\frac{\mu^2}{2\sigma^2} - \frac{\log(2\pi)}{2}\right)$$

$$= \frac{1}{\sigma^2}\left(\sum_i x_i - N\mu\right) = 0 \quad \Rightarrow \quad \hat{\mu}^{ML} = \frac{1}{N}\sum_i x_i$$

and similarly: $\widehat{\sigma^2}^{ML} = \frac{1}{N}\sum_i (x_i - \mu)^2$

EM for GMM –update of weights

- Weights π_c need to sum up to one. When updating weights, Lagrange multiplier λ is used to enforce this constraint.

$$\frac{\partial}{\partial \pi_c}\left(\sum_n Q_n(q(z_n),\boldsymbol{\eta}) - \lambda\left(\sum_k \pi_k - 1\right)\right) =$$

$$\frac{\partial}{\partial \pi_c}\left(\sum_n \sum_k \gamma_{nk} \ln \pi_k - \lambda\left(\sum_k \pi_k - 1\right)\right) =$$

$$\sum_n \frac{\gamma_{nc}}{\pi_c} - \lambda = 0$$

$$\Longrightarrow \pi_c = \frac{\sum_n \gamma_{nc}}{\lambda} = \frac{\sum_n \gamma_{nc}}{\sum_k \sum_n \gamma_{nk}}$$

Factorization of the auxiliary function more formally

- Before, we have introduced the per-observation auxiliary functions

$$Q(q(\mathbf{z}), \boldsymbol{\eta}) = \sum_n Q_n(q(z_n), \boldsymbol{\eta})$$
$$= \sum_n \sum_{z_n} q(z_n) \ln p(x_n, z_n | \boldsymbol{\eta})$$

- We can show that such factorization comes naturally even if we directly write the auxiliary function as defined for the EM algorithm:

$$\mathcal{Q}(q(\mathbf{Z}), \boldsymbol{\eta}) = \sum_{\mathbf{Z}} q(\mathbf{Z}) \ln p(\mathbf{X}, \mathbf{Z} | \boldsymbol{\eta})$$
$$= \sum_{\mathbf{Z}} \prod_n q(\mathbf{Z}) \sum_n p(x_n, z_n | \boldsymbol{\eta}) = \sum_c \sum_n q(\mathbf{Z}) p(x_n, z_n | \boldsymbol{\eta})$$

- See the next slide for proof

Factorization over components

Example with only 3 fames (i.e $\mathbf{z} = [z_1, z_2, z_3]$)

$$\sum_{\mathbf{z}} \prod_n q(z_n) \sum_n f(z_n) =$$

$$\sum_{z_1} \sum_{z_2} \sum_{z_3} q(z_1)q(z_2)q(z_3)f(z_1) + \sum_{z_1} \sum_{z_2} \sum_{z_3} q(z_1)q(z_2)q(z_3)f(z_2) + \sum_{z_1} \sum_{z_2} \sum_{z_3} q(z_1)q(z_2)q(z_3)f(z_3) =$$

$$\sum_{z_1} q(z_1)f(z_1) \underbrace{\sum_{z_2} q(z_2)}_{1} \underbrace{\sum_{z_3} q(z_3)}_{1} + \underbrace{\sum_{z_1} q(z_1)}_{1} \sum_{z_2} q(z_2)f(z_2) \underbrace{\sum_{z_3} q(z_3)}_{1} + \underbrace{\sum_{z_1} q(z_1)}_{1} \underbrace{\sum_{z_2} q(z_2)}_{1} \sum_{z_3} q(z_3)f(z_3) =$$

$$\sum_{z_1} q(z_1)f(z_1) + \sum_{z_2} q(z_2)f(z_2) + \sum_{z_3} q(z_3)f(z_3) =$$

$$\sum_{c=1}^{C} q(z_1 = c)f(z_1 = c) + \sum_{c=1}^{C} q(z_2 = c)f(z_2 = c) + \sum_{c=1}^{C} q(z_3 = c)f(z_3 = c) =$$

$$\sum_{c=1}^{C} \sum_n q(z_n = c)f(z_n = c)$$

EM for continuous latent variable

- Same equations, where sums over the latent variable $\mathbf{Z}$ are simply replaced by integrals

$$
\ln p(\mathbf{X}|\boldsymbol{\eta}) = \underbrace{\int q(\mathbf{Z}) \ln p(\mathbf{X}, \mathbf{Z}|\boldsymbol{\eta})\,\mathrm{d}\mathbf{Z}}_{\mathcal{Q}(q(\mathbf{Z}),\boldsymbol{\eta})} \underbrace{- \int q(\mathbf{Z}) \ln q(\mathbf{Z})\,\mathrm{d}\mathbf{Z}}_{H(q(\mathbf{Z}))} \underbrace{- \int q(\mathbf{Z}) \ln \frac{p(\mathbf{Z}|\mathbf{X}, \boldsymbol{\eta})}{q(\mathbf{Z})}\,\mathrm{d}\mathbf{Z}}_{D_{KL}(q(\mathbf{Z})||p(\mathbf{Z}|\mathbf{X},\boldsymbol{\eta}))}
$$

$$
= \underbrace{\mathcal{Q}(q(\mathbf{Z}), \boldsymbol{\eta}) + H(q(\mathbf{Z}))}_{\mathcal{L}(q(\mathbf{Z}),\boldsymbol{\eta})} + D_{KL}\left(q(\mathbf{Z})||p(\mathbf{Z}|\mathbf{X}, \boldsymbol{\eta})\right)
$$

PLDA model for speaker verification

$$p(\mathbf{r}) = \mathcal{N}(\mathbf{r}|\boldsymbol{\mu}, \boldsymbol{\Sigma}_{ac})$$ - distribution of speaker means

$$p(\mathbf{i}|\mathbf{r}) = \mathcal{N}(\mathbf{i}|\mathbf{r}, \boldsymbol{\Sigma}_{wc})$$ - within class (channel) variability

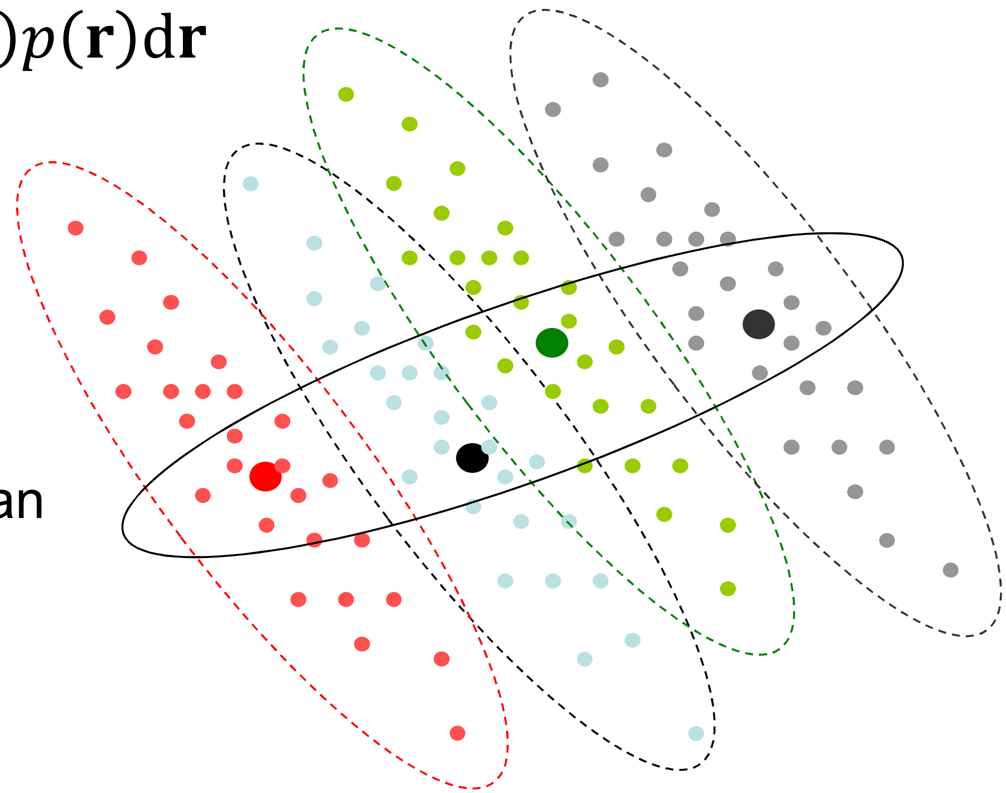
Same speaker hypothesis likelihood:

$$p(\mathbf{i}_1, \mathbf{i}_2|\mathcal{H}_s) = \int p(\mathbf{i}_1|\mathbf{r})p(\mathbf{i}_2|\mathbf{r})p(\mathbf{r})\mathrm{d}\mathbf{r}$$

Different speaker hyp. Likelihood:

$$p(\mathbf{i}_1, \mathbf{i}_2|\mathcal{H}_d) = p(\mathbf{i}_1)p(\mathbf{i}_2)$$

$$p(\mathbf{i}) = \int p(\mathbf{i}|\mathbf{r})p(\mathbf{r})\mathrm{d}\mathbf{r}$$



Verification score based on Bayesian model comparison:

$$s = \log\frac{p(\mathbf{i}_1, \mathbf{i}_2|\mathcal{H}_s)}{p(\mathbf{i}_1, \mathbf{i}_2|\mathcal{H}_d)}$$