

Expectation-Maximization

Nuno Vasconcelos

ECE Department, UCSD

Recall

- ▶ last class, we will have “Cheetah Day”
- ▶ what:
 - 4 teams, average of 6 people
 - each team will write a report on the 4 cheetah problems
 - each team will give a presentation on one of the problems
- ▶ I am waiting to hear on the teams



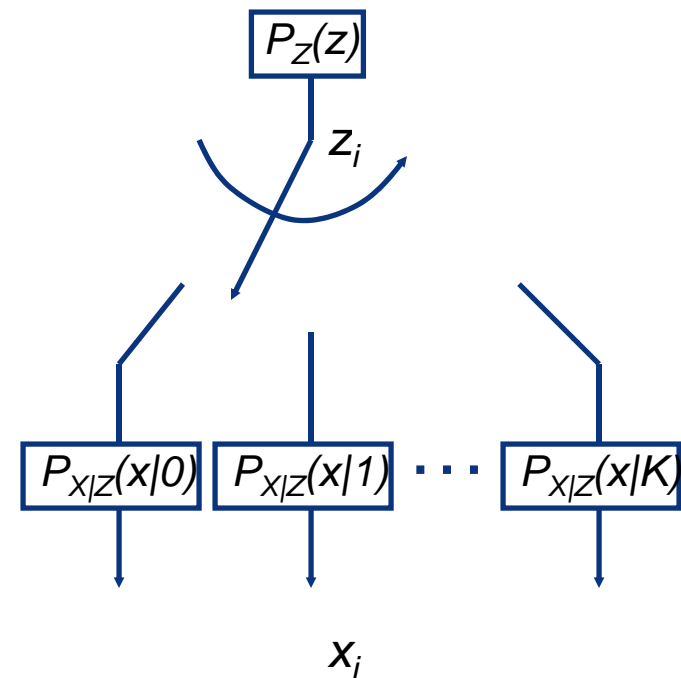
Plan for today

- ▶ we have been talking about mixture models
 - ▶ last time we introduced the basics of EM
 - ▶ today we study the application of EM for ML estimation of mixture parameters
-
- ▶ next class:
 - proof that EM maximizes likelihood of incomplete data

mixture model

- ▶ two types of random variables
 - Z – hidden state variable
 - X – observed variable
- ▶ observations sampled with a two-step procedure
 - a **state** (class) is sampled from the distribution of the hidden variable

$$P_Z(z) \rightarrow z_i$$



- an **observation** is drawn from the class conditional density for the selected state

$$P_{X|Z}(x|z_i) \rightarrow x_i$$

mixture model

- ▶ the sample consists of pairs (x_i, z_i)

$$D = \{(x_1, z_1), \dots, (x_n, z_n)\}$$

but we never get to see the z_i

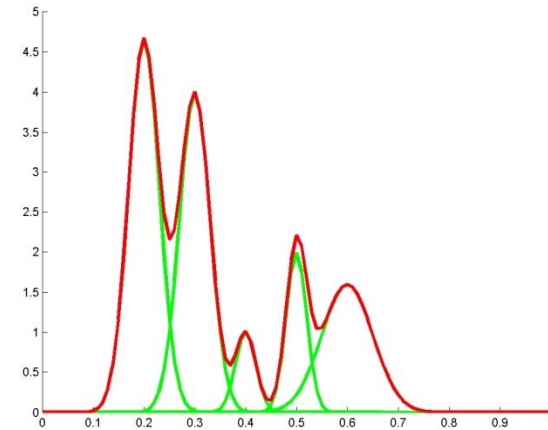
- ▶ the pdf of the observed data is

$$\begin{aligned} P_X(x) &= \sum_{c=1}^C P_{X|Z}(x|c) P_Z(c) \\ &= \sum_{c=1}^C P_{X|Z}(x|c) \pi_c \end{aligned}$$

of mixture components

component “weight”

c^{th} “mixture component”



The basics of EM

- ▶ as usual, we start from an iid sample $D = \{x_1, \dots, x_N\}$
- ▶ goal is to find parameters Ψ^* that maximize likelihood with respect to D

$$\begin{aligned}\Psi^* &= \arg \max_{\Psi} P_{\mathbf{X}}(\mathcal{D}; \Psi) \\ &= \arg \max_{\Psi} \int P_{\mathbf{X}|Z}(\mathcal{D}|z; \Psi) P_Z(z; \Psi) dz\end{aligned}$$

- ▶ the set

$$D_c = \{(x_1, z_1), \dots, (x_N, z_N)\}$$

is called the complete data

- ▶ the set

$$D = \{x_1, \dots, x_N\}$$

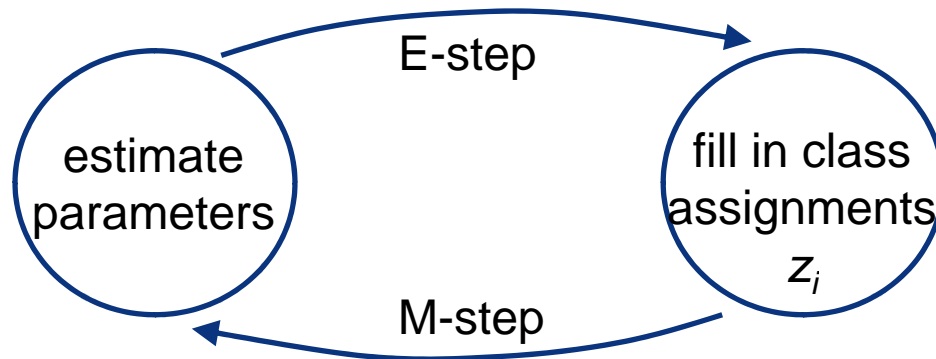
is called the incomplete data

Learning with incomplete data (EM)

► the basic idea is quite simple

1. start with an initial parameter estimate $\Psi^{(0)}$
2. **E-step:** given current parameters $\Psi^{(i)}$ and observations in D , “guess” what the values of the z_i are
3. **M-step:** with the new z_i , we have a complete data problem, solve this problem for the parameters, i.e. compute $\Psi^{(i+1)}$
4. go to 2.

► this can be summarized as



Classification-maximization

► C-step:

- given estimates $\Psi^{(i)} = \{\Psi^{(i)}_1, \dots, \Psi^{(i)}_C\}$
- determine z_l by the BDR

$$z_l = \arg \max_c P_{\mathbf{X}|Z} \left(\mathbf{x}_l | c; \Psi_c^{(i)} \right) \pi_c^{(i)}, l \in \{1, \dots, n\}$$

- split the training set according to the labels z_l

$$D^1 = \{\mathbf{x}_l | z_l = 1\}, \quad D^2 = \{\mathbf{x}_l | z_l = 2\}, \quad \dots, \quad D^C = \{\mathbf{x}_l | z_l = C\}$$

► M-step:

- as before, determine the parameters of each class independently

$$\Psi_c^{(i+1)} = \arg \max_{\Psi, \pi} P_{\mathbf{X}|Z}(D^c | c, \Psi) \pi$$

For Gaussian mixtures

► C-step:

- $$z_l = \arg \max_c \left\{ -\frac{1}{2} \left(\mathbf{x}_l - \mu_c^{(i)} \right)^T \left(\Sigma_c^{(i)} \right)^{-1} \left(\mathbf{x}_l - \mu_c^{(i)} \right) - \frac{1}{2} \log \left| \Sigma_c^{(i)} \right| + \log \pi_c^{(i)} \right\}, l \in \{1, \dots, n\}$$

- split the training set according to the labels z_i

$$D^1 = \{\mathbf{x}_i | z_i = 1\}, \quad D^2 = \{\mathbf{x}_i | z_i = 2\}, \quad \dots, \quad D^C = \{\mathbf{x}_i | z_i = C\}$$

► M-step:

- $$\pi_c^{(i+1)} = \frac{|\{\mathbf{x}_i \in \mathcal{D}^c\}|}{n} \qquad \mu_c^{(i+1)} = \frac{1}{|\{\mathbf{x}_i \in \mathcal{D}^c\}|} \sum_{i | \mathbf{x}_i \in \mathcal{D}^c} \mathbf{x}_i$$
$$\Sigma_c^{(i+1)} = \frac{1}{|\{\mathbf{x}_i \in \mathcal{D}^c\}|} \sum_{i | \mathbf{x}_i \in \mathcal{D}^c} \left(\mathbf{x}_i - \mu_c^{(i+1)} \right) \left(\mathbf{x}_i - \mu_c^{(i+1)} \right)^T$$

K-means

- ▶ when covariances are identity and priors uniform

- ▶ C-step:

- $z_l = \arg \min_c ||\mathbf{x}_l - \mu_c^{(i)}||^2, \quad l \in \{1, \dots, n\}$
- split the training set according to the labels z_i

$$D^1 = \{\mathbf{x}_i | z_i=1\}, \quad D^2 = \{\mathbf{x}_i | z_i=2\}, \quad \dots, \quad D^C = \{\mathbf{x}_i | z_i=C\}$$

- ▶ M-step:

- $\mu_c^{(i+1)} = \frac{1}{|\{\mathbf{x}_i \in \mathcal{D}^c\}|} \sum_{i|\mathbf{x}_i \in \mathcal{D}^c} \mathbf{x}_i$

- ▶ this is the **K-means** algorithm, aka **generalized Lloyd algorithm**, aka **LBG algorithm** in the vector quantization literature:

- “assign points to the closest mean; recompute the means”

The Q function

► is defined as

$$Q(\Psi; \Psi^{(n)}) = E_{Z|X; \Psi^{(n)}} \left[\log P_{X,Z}(\mathcal{D}, \{z_1, \dots, z_N\}; \Psi) | \mathcal{D} \right]$$

► and is a bit tricky:

- it is the expected value of likelihood with respect to complete data (joint X and Z)
- given that we observed incomplete data ($X=\mathcal{D}$)
- note that the likelihood is a function of Ψ (the parameters that we want to determine)
- but to compute the expected value we need to use the parameter values from the previous iteration (because we need a distribution for $Z|X$)

► the EM algorithm is, therefore, as follows

Expectation-maximization

► E-step:

- given estimates $\Psi^{(n)} = \{\Psi^{(n)}_1, \dots, \Psi^{(n)}_C\}$
- compute expected log-likelihood of complete data

$$Q(\Psi; \Psi^{(n)}) = E_{Z|\mathbf{X}; \Psi^{(n)}} \left[\log P_{\mathbf{X}, Z}(\mathcal{D}, \{z_1, \dots, z_N\}; \Psi) | \mathcal{D} \right]$$

► M-step:

- find parameter set that maximizes this expected log-likelihood

$$\Psi^{(n+1)} = \arg \max_{\Psi} Q(\Psi; \Psi^{(n)})$$

- let's make this more concrete by looking at the mixture case

Expectation-maximization

- ▶ to derive an EM algorithm you need to do the following
- 1. write down the likelihood of the COMPLETE data
- 2. E-step: write down the Q function, i.e. its expectation given the observed data
- 3. M-step: solve the maximization, deriving a closed-form solution if there is one

EM for mixtures (step 1)

- ▶ the first thing we always do in a EM problem is
 - compute the likelihood of the COMPLETE data
- ▶ very neat trick to use when z is discrete (classes)
 - instead of using z in $\{1, 2, \dots, C\}$
 - use a binary vector of size equal to the # of classes

$$\mathbf{z} \in \left\{ \begin{bmatrix} 1 \\ 0 \\ 0 \\ \vdots \\ 0 \end{bmatrix}, \begin{bmatrix} 0 \\ 1 \\ 0 \\ \vdots \\ 0 \end{bmatrix}, \begin{bmatrix} 0 \\ 0 \\ 1 \\ \vdots \\ 0 \end{bmatrix}, \dots, \begin{bmatrix} 0 \\ 0 \\ 0 \\ \vdots \\ 1 \end{bmatrix} \right\}$$

- where $z = j$ in the z in $\{1, 2, \dots, C\}$ notation, now becomes

$$\mathbf{z} = \mathbf{e}_j = \begin{bmatrix} 0 \\ \vdots \\ 1 \\ \vdots \\ 0 \end{bmatrix} \quad (j^{th} position)$$

EM for mixtures (step 1)

- ▶ we can now write the **complete data likelihood** as

$$\begin{aligned} P_{\mathbf{X}, \mathbf{Z}}(\mathbf{x}, \mathbf{z}; \Psi) &= P_{\mathbf{X}|\mathbf{Z}}(\mathbf{x}|\mathbf{z}; \Psi) P_{\mathbf{Z}}(\mathbf{z}; \Psi) \\ &= \prod_{j=1}^C \left[P_{\mathbf{X}|\mathbf{Z}}(\mathbf{x}|\mathbf{e}_j, \Psi) \pi_j \right]^{z_j} \end{aligned}$$

- ▶ for example, if $\mathbf{z} = \mathbf{e}_k$ in the \mathbf{z} in $\{1, 2, \dots, C\}$ notation,

$$\begin{aligned} P_{\mathbf{X}, \mathbf{Z}}(\mathbf{x}, \mathbf{e}_k; \Psi) &= P_{\mathbf{X}, \mathbf{Z}}(\mathbf{x}, \mathbf{e}_k; \Psi) \\ &= \left[P_{\mathbf{X}|\mathbf{Z}}(\mathbf{x}|\mathbf{e}_k, \Psi) \pi_k \right]^1 \prod_{j \neq k} \left[P_{\mathbf{X}|\mathbf{Z}}(\mathbf{x}|\mathbf{e}_j, \Psi) \pi_j \right]^0 \end{aligned}$$

- ▶ the **advantage** is that

$$\log P_{\mathbf{X}, \mathbf{Z}}(\mathbf{x}, \mathbf{z}; \Psi) = \sum_{j=1}^C z_j \log \left[P_{\mathbf{X}|\mathbf{Z}}(\mathbf{x}|\mathbf{e}_j, \Psi) \pi_j \right]$$

- ▶ becomes **LINEAR** in the components z_j !!!

The assignment vector trick

- ▶ this is similar to something that we used already
- ▶ Bernoulli random variable

$$P_Z(z) = \begin{cases} p & z = 1 \\ 1 - p & z = 0 \end{cases}$$

- ▶ can be written as

$$P_Z(z) = p^z (1 - p)^{1-z}$$

- ▶ or, using $z \in \left\{ \begin{bmatrix} 0 \\ 1 \end{bmatrix}, \begin{bmatrix} 1 \\ 0 \end{bmatrix} \right\}$ instead of $z \in \{0,1\}$, as

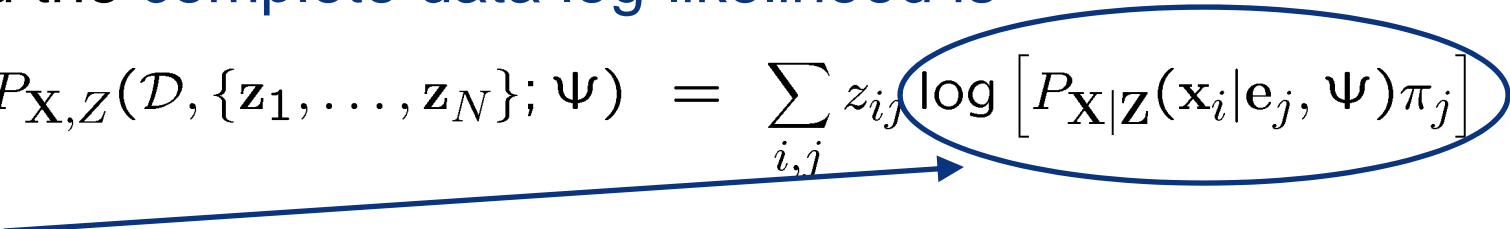
$$P_Z(z) = p^{z_1} (1 - p)^{z_2}$$

EM for mixtures (step 1)

- ▶ for the complete iid dataset $D_c = \{(x_1, z_1), \dots, (x_N, z_N)\}$

$$\begin{aligned} P_{\mathbf{X}, \mathbf{Z}}(\mathcal{D}, \{\mathbf{z}_1, \dots, \mathbf{z}_N\}; \Psi) &= \prod_{i=1}^N P_{\mathbf{X}, \mathbf{Z}}(\mathbf{x}_i, \mathbf{z}_i; \Psi) \\ &= \prod_{i=1}^N \prod_{j=1}^C [P_{\mathbf{X}|\mathbf{Z}}(\mathbf{x}_i | \mathbf{e}_j, \Psi) \pi_j]^{z_{ij}} \end{aligned}$$

- ▶ and the complete data log-likelihood is

$$\log P_{\mathbf{X}, \mathbf{Z}}(\mathcal{D}, \{\mathbf{z}_1, \dots, \mathbf{z}_N\}; \Psi) = \sum_{i,j} z_{ij} \log [P_{\mathbf{X}|\mathbf{Z}}(\mathbf{x}_i | \mathbf{e}_j, \Psi) \pi_j]$$


- ▶ this does not depend on \mathbf{z} and simply becomes a constant for the expectation that we have to compute in the E-step

Expectation-maximization

- ▶ to derive an EM algorithm you need to do the following
 1. write down the likelihood of the COMPLETE data
 - 2. E-step: write down the Q function, i.e. its expectation given the observed data
 3. M-step: solve the maximization, deriving a closed-form solution if there is one
- ▶ important E-step advice:
 - do not compute terms that you do not need
 - at the end of the day we only care about the parameters
 - terms of Q that do not depend on the parameters are useless, e.g. in

$$Q = f(z, \Psi) + \log(\sin z)$$

the expected value of $\log(\sin z)$ appears to be difficult and is completely unnecessary, since it is dropped in the M-step

EM for mixtures (step 2)

- ▶ once we have the complete data likelihood

$$\begin{aligned} Q(\Psi; \Psi^{(n)}) &= E_{Z|\mathbf{X}; \Psi^{(n)}} [\log P_{\mathbf{X}, Z}(\mathcal{D}, \{z_1, \dots, z_N\}; \Psi) | \mathcal{D}] \\ &= \sum_{i,j} E_{Z|\mathbf{X}; \Psi^{(n)}} [z_{ij} | \mathcal{D}] \log [P_{\mathbf{X}|Z}(\mathbf{x}_i | \mathbf{e}_j, \Psi) \pi_j] \end{aligned}$$

- ▶ i.e. to compute the Q function we only need to compute

$$E_{Z|\mathbf{X}; \Psi^{(n)}} [z_{ij} | \mathcal{D}], \quad \forall i, j$$

- ▶ note that this expectation can only be computed because we use $\Psi^{(n)}$
- ▶ note that the Q function will be a function of both Ψ and $\Psi^{(n)}$

EM for mixtures (step 2)

- ▶ since z_{ij} is binary and only depends on x_i

$$E_{\mathbf{Z}|\mathbf{X};\Psi^{(n)}}[z_{ij}|\mathcal{D}] = P_{\mathbf{Z}|\mathbf{X}}(z_{ij} = 1|\mathbf{x}_i; \Psi^{(n)}) = P_{\mathbf{Z}|\mathbf{X}}(\mathbf{e}_j|\mathbf{x}_i; \Psi^{(n)})$$

- ▶ the E-step reduces to computing the posterior probability of each point under each class!

- ▶ defining

$$h_{ij} = P_{\mathbf{Z}|\mathbf{X}}(\mathbf{e}_j|\mathbf{x}_i; \Psi^{(n)})$$

- ▶ the Q function is

$$Q(\Psi; \Psi^{(n)}) = \sum_{i,j} h_{ij} \log [P_{\mathbf{X}|\mathbf{Z}}(\mathbf{x}_i|\mathbf{e}_j, \Psi)\pi_j]$$

Expectation-maximization

► to derive an EM algorithm you need to do the following

1. write down the likelihood of the COMPLETE data

$$\log P_{\mathbf{X}, \mathbf{Z}}(\mathcal{D}, \{\mathbf{z}_1, \dots, \mathbf{z}_N\}; \Psi) = \sum_{i,j} z_{ij} \log [P_{\mathbf{X}|\mathbf{Z}}(\mathbf{x}_i | \mathbf{e}_j, \Psi) \pi_j]$$

2. E-step: write down the Q function, i.e. its expectation given the observed data

$$h_{ij} = P_{\mathbf{Z}|\mathbf{X}}(\mathbf{e}_j | \mathbf{x}_i; \Psi^{(n)})$$

$$Q(\Psi; \Psi^{(n)}) = \sum_{i,j} h_{ij} \log [P_{\mathbf{X}|\mathbf{Z}}(\mathbf{x}_i | \mathbf{e}_j, \Psi) \pi_j]$$

- 3. M-step: solve the maximization, deriving a closed-form solution if there is one

$$\Psi^{(n+1)} = \arg \max_{\Psi} \sum_{i,j} h_{ij} \log [P_{\mathbf{X}|\mathbf{Z}}(\mathbf{x}_i | \mathbf{e}_j, \Psi) \pi_j]$$

EM vs CM

► let's compare this with the CM algorithm

- the C-step

$$\mathbf{z}_i = \arg \max_j P_{\mathbf{Z}|\mathbf{X}}(\mathbf{e}_j|\mathbf{x}_i; \psi^{(n)})$$

assigns each point to the class of largest posterior

- the E-step

$$h_{ij} = P_{\mathbf{Z}|\mathbf{X}}(\mathbf{e}_j|\mathbf{x}_i)$$

assigns the point to all classes with weight given by the posterior

► for this, EM is said to make “soft-assignments”

- it does not commit to any of the classes (unless the posterior is one for that class), i.e. it is less greedy
- no longer partition space into rigid cells, but now the boundaries are soft

EM vs CM

► what about the M-steps?

- for CM

$$\begin{aligned}\psi_j^{(n+1)} &= \arg \max_{\psi} P_{\mathbf{X}|\mathbf{Z}}(\mathcal{D}^j | \mathbf{e}_j, \psi) \pi \\ &= \arg \max_{\psi} \sum_{i|z_i=j} \log [P_{\mathbf{X}|\mathbf{Z}}(\mathbf{x}_i | \mathbf{e}_j, \psi) \pi] \\ &= \arg \max_{\psi} \sum_i \delta_{z_i=j} \log [P_{\mathbf{X}|\mathbf{Z}}(\mathbf{x}_i | \mathbf{e}_j, \psi) \pi]\end{aligned}$$

- for EM

$$\psi^{(n+1)} = \arg \max_{\psi} \sum_{ij} h_{ij} \log [P_{\mathbf{X}|\mathbf{Z}}(\mathbf{x}_i | \mathbf{e}_j, \psi) \pi_j]$$

► these are the same if we threshold the h_{ij} to make, for each i , $\max_j h_{ij} = 1$ and all other $h_{ij} = 0$

► M-steps the same up to the difference of assignments

EM for Gaussian mixtures

► in summary:

- CM = EM + hard assignments
- CM special case, cannot be better

► let's look at the special case of Gaussian mixtures

► **E-step:**

$$\begin{aligned} h_{ij} &= P_{\mathbf{Z}|\mathbf{X}}(\mathbf{e}_j|\mathbf{x}_i; \boldsymbol{\psi}^{(n)}) \\ &= \frac{\mathcal{G}\left(\mathbf{x}_i, \mu_j^{(n)}, \sigma_j^{(n)}\right) \pi_j^{(n)}}{\sum_{k=1}^C \mathcal{G}\left(\mathbf{x}_i, \mu_k^{(n)}, \sigma_k^{(n)}\right) \pi_k^{(n)}} \end{aligned}$$

M-step for Gaussian mixtures

► M-step:

$$\begin{aligned}\psi^{(n+1)} &= \arg \max_{\psi} \sum_{ij} h_{ij} \log [\mathcal{G}(\mathbf{x}_i, \mu_j, \sigma_j) \pi_j] \\ &= \arg \min_{\psi} \sum_{ij} \frac{h_{ij}(\mathbf{x}_i - \mu_j)^2}{2\sigma_j^2} + \frac{h_{ij}}{2} \log \sigma_j^2 - h_{ij} \log \pi_j\end{aligned}$$

► important note:

- in the M-step, the optimization must be subject to whatever constraint may hold
- in particular, we **always have the constraint** $\sum_j \pi_j = 1$
- as usual we introduce a Lagrangian

$$L = \sum_{ij} \left[\frac{h_{ij}(\mathbf{x}_i - \mu_j)^2}{2\sigma_j^2} + \frac{h_{ij}}{2} \log \sigma_j^2 - h_{ij} \log \pi_j \right] + \lambda \left(\sum_j \pi_j - 1 \right)$$

M-step for Gaussian mixtures

► Lagrangian

$$L = \sum_{ij} \left[\frac{h_{ij}(\mathbf{x}_i - \mu_j)^2}{2\sigma_j^2} + \frac{h_{ij}}{2} \log \sigma_j^2 - h_{ij} \log \pi_j \right] + \lambda \left(\sum_j \pi_j - 1 \right)$$

► setting derivatives to zero

$$\frac{\partial L}{\partial \mu_j} = - \sum_i \frac{h_{ij}(\mathbf{x}_i - \mu_j)}{\sigma_j^2} = 0$$

$$\frac{\partial L}{\partial \sigma_j^2} = - \sum_i \left[\frac{h_{ij}(\mathbf{x}_i - \mu_j)^2}{\sigma_j^4} - \frac{h_{ij}}{\sigma_j^2} \right] = 0$$

$$\frac{\partial L}{\partial \pi_j} = - \sum_i \frac{h_{ij}}{\pi_j} + \lambda = 0$$

$$\frac{\partial L}{\partial \lambda} = \sum_j \pi_j - 1 = 0$$

M-step for Gaussian mixtures

- ▶ leads to the update equations

$$\begin{aligned}\mu_j^{(n+1)} &= \frac{\sum_i h_{ij} \mathbf{x}_i}{\sum_i h_{ij}} & \pi_j^{(n+1)} &= \frac{1}{n} \sum_i h_{ij} \\ \sigma_j^{2(n+1)} &= \frac{\sum_i h_{ij} (\mathbf{x}_i - \mu_j)^2}{\sum_i h_{ij}}\end{aligned}$$

- ▶ comparing to those of CM

$$\begin{aligned}\pi_c^{(n+1)} &= \frac{|\{\mathbf{x}_i \in \mathcal{D}^c\}|}{N} & \mu_c^{(n+1)} &= \frac{1}{|\{\mathbf{x}_i \in \mathcal{D}^c\}|} \sum_{i|\mathbf{x}_i \in \mathcal{D}^c} \mathbf{x}_i \\ \Sigma_c^{(n+1)} &= \frac{1}{|\{\mathbf{x}_i \in \mathcal{D}^c\}|} \sum_{i|\mathbf{x}_i \in \mathcal{D}^c} \left(\mathbf{x}_i - \mu_c^{(n+1)} \right) \left(\mathbf{x}_i - \mu_c^{(n+1)} \right)^T\end{aligned}$$

- ▶ they are the same up to hard vs soft assignments.

Expectation-maximization

- note that the procedure is the same for all mixtures

1. write down the likelihood of the COMPLETE data

$$\log P_{\mathbf{X}, \mathbf{Z}}(\mathcal{D}, \{\mathbf{z}_1, \dots, \mathbf{z}_N\}; \Psi) = \sum_{i,j} z_{ij} \log [P_{\mathbf{X}|\mathbf{Z}}(\mathbf{x}_i | \mathbf{e}_j, \Psi) \pi_j]$$

2. E-step: write down the Q function, i.e. its expectation given the observed data

$$h_{ij} = P_{\mathbf{Z}|\mathbf{X}}(\mathbf{e}_j | \mathbf{x}_i; \Psi^{(n)})$$

$$Q(\Psi; \Psi^{(n)}) = \sum_{i,j} h_{ij} \log [P_{\mathbf{X}|\mathbf{Z}}(\mathbf{x}_i | \mathbf{e}_j, \Psi) \pi_j]$$

3. M-step: solve the maximization, deriving a closed-form solution if there is one

$$\Psi^{(n+1)} = \arg \max_{\Psi} \sum_{i,j} h_{ij} \log [P_{\mathbf{X}|\mathbf{Z}}(\mathbf{x}_i | \mathbf{e}_j, \Psi) \pi_j]$$

Expectation-maximization

- E.g. for a mixture of exponential distributions

$$P_X(x) = \sum_{i=1}^C \pi_i \lambda_i e^{-\lambda_i x}$$

1. E-step: write down the Q function, i.e. its expectation given the observed data

$$h_{ij} = P_{Z|X}(j | x_i) = \frac{\pi_j \lambda_j e^{-\lambda_j x_i}}{\sum_{c=1}^C \pi_c \lambda_c e^{-\lambda_c x_i}}$$

2. M-step: solve the maximization, deriving a closed-form solution if there is one

$$Q(\Psi; \Psi^{(n)}) = \sum_{i,j} h_{ij} \log [P_{X|Z}(x_i | e_j, \Psi) \pi_j]$$

M-step for exponential mixtures

► M-step:

$$\begin{aligned}\Psi^{(n+1)} &= \arg \max_{\Psi} \sum_{ij} h_{ij} \log [\pi_j \lambda_j e^{-\lambda_j x_i}] \\ &= \arg \min_{\Psi} \sum_{ij} h_{ij} (\lambda_j x_i - \log [\pi_j \lambda_j])\end{aligned}$$

► the Lagrangian is

$$L = \sum_{ij} h_{ij} (\lambda_j x_i - \log \lambda_j - \log \pi_j) + \kappa \left(\sum_j \pi_j - 1 \right)$$

M-step for exponential mixtures

►
$$L = \sum_{ij} h_{ij} (\lambda_j x_i - \log \lambda_j - \log \pi_j) + \kappa \left(\sum_j \pi_j - 1 \right)$$

and has minimum at

$$\frac{\partial L}{\partial \lambda_k} = \sum_i h_{ik} \left(x_i - \frac{1}{\lambda_k} \right) = 0$$

$$\frac{\partial L}{\partial \pi_k} = - \sum_i \frac{h_{ik}}{\pi_k} + \kappa = 0$$

$$\frac{\partial L}{\partial \kappa} = \sum_j \pi_j - 1 = 0$$



$$\frac{1}{\lambda_k} = \frac{\sum_i h_{ik} x_i}{\sum_i h_{ik}}$$

$$\kappa = \sum_{ij} h_{ik}$$

$$\pi_k = \frac{\sum_i h_{ik}}{\sum_{ij} h_{ik}}$$

EM algorithm

- ▶ note, however, that EM is much more general than this recipe for mixtures
- ▶ it can be applied for any problem where we have observed and hidden random variables
- ▶ here is a very simple example
 - X observed Gaussian variable, $X \sim N(\mu, 1)$,
 - Z hidden exponential variable
 - It is known that Z is independent of X
 - sample $D = \{x_1, \dots, x_n\}$ of iid observations from X
- ▶ note that the assumption of independence does not really make sense (why?)
- ▶ how does this affect EM?

Example

- **toy model:** X iid, Z iid, $X_i \sim N(\mu, 1)$, $Z_i \sim \lambda e^{-\lambda z}$,
 X independent of Z

$$\begin{aligned} \text{► } Q(\Psi; \Psi^{(n)}) &= E_{Z|\mathbf{X}; \Psi^{(n)}} \left[\log P_{\mathbf{X}, Z}(\mathcal{D}, \{z_1, \dots, z_N\}; \Psi) | \mathcal{D} \right] \\ &= E_{Z|\mathbf{X}; \Psi^{(n)}} \left[- \sum_k \frac{(x_k - \mu)^2}{2} - \frac{N}{2} \log 2\pi - \lambda \sum_k z_k + N \log \lambda | \mathcal{D} \right] \\ &= - \sum_k \frac{(x_k - \mu)^2}{2} - \frac{N}{2} \log 2\pi - \lambda \sum_k E_{Z|\mathbf{X}; \Psi^{(n)}}[z_k | x_k] + N \log \lambda \\ &= - \sum_k \frac{(x_k - \mu)^2}{2} - \frac{N}{2} \log 2\pi - \lambda \sum_k E_{Z_k; \Psi^{(n)}}[z_k] + N \log \lambda \\ &= - \sum_k \frac{(x_k - \mu)^2}{2} - \frac{N}{2} \log 2\pi - N \lambda E_{Z; \Psi^{(n)}}[z] + N \log \lambda \\ &= - \sum_k \frac{(x_k - \mu)^2}{2} - \frac{N}{2} \log 2\pi - N \frac{\lambda}{\lambda^{(n)}} + N \log \lambda \end{aligned}$$

Example

$$\blacktriangleright \psi^{(n+1)} = \arg \max_{\psi} Q(\psi; \psi^{(n)})$$

$$\blacktriangleright Q(\psi; \psi^{(n)}) = - \sum_k \frac{(x_k - \mu)^2}{2} - \frac{N}{2} \log 2\pi - N \frac{\lambda}{\lambda^{(n)}} + N \log \lambda$$

$$\frac{\partial Q}{\partial \mu} = 0 \Leftrightarrow \boxed{\mu^{(n+1)} = \frac{1}{n} \sum_k x_k} \qquad \frac{\partial Q}{\partial \lambda} = 0 \Leftrightarrow \boxed{\lambda^{(n+1)} = \lambda^{(n)}}$$

► this makes sense:

- since hidden variables Z are independent of observed X
- ML estimate of μ is always the same: the sample mean, no dependence on z_i
- ML estimate of λ is always the initial estimate $\lambda^{(0)}$: since the observations are independent of the z_i we have no information on what λ should be, other than initial guess.

► note that model does not make sense, not EM solution

Any Questions?