Lecture 15: Expectation Maximization

TTIC 31020: Introduction to Machine Learning

Instructor: Kevin Gimpel

TTI-Chicago

November 19, 2019

Administrivia

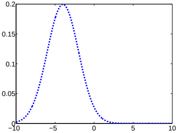
- Problem set 4 due tomorrow, Nov 20 11:59pm
- Quiz 3
  - Thursday Nov 21
  - Topic: material from Nov 14 lecture
- Recitations this week: EM

Review: Parametric mixtures

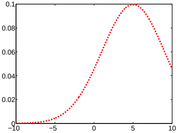
- Suppose that the parameters of the c-th component are θ_c. Then we can denote $\boldsymbol{\theta} = [\boldsymbol{\theta}_1, \ldots, \boldsymbol{\theta}_k]$ and write

$$p(\mathbf{x}; \boldsymbol{\theta}, \boldsymbol{\pi}) \;=\; \sum_{c=1}^{k} \pi_c \, p(\mathbf{x};\, \boldsymbol{\theta}_c)$$

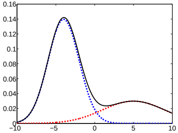
- Any valid setting of $\boldsymbol{\theta}$ and $\boldsymbol{\pi}$, subject to $\sum_{c=1}^{k} \pi_c = 1$, produces a valid pdf
- Example: mixture of Gaussians.



$\times 0.7 \quad +$



$\times 0.3 \quad =$



Marginal log-likelihood of a mixture model

- The **marginal log-likelihood** of $\boldsymbol{\pi}, \boldsymbol{\theta}$:

$$\log p(X;\, \boldsymbol{\pi}, \theta) \;=\; \sum_{i=1}^{n} \log \sum_{c=1}^{k} \pi_c \mathcal{N}\left(\mathbf{x}_i;\, \boldsymbol{\mu}_c, \boldsymbol{\Sigma}_c\right)$$

  "marginal" because we are marginalizing over y_i for each $\mathbf{x}_i$

- Learning mixture models = finding parameters that maximize marginal log-likelihood given the observed data

Marginal log-likelihood of a mixture model

- The **marginal log-likelihood** of $\boldsymbol{\pi}, \boldsymbol{\theta}$:

$$\log p(X;\, \boldsymbol{\pi}, \theta) \;=\; \sum_{i=1}^{n} \log \sum_{c=1}^{k} \pi_c \mathcal{N}\left(\mathbf{x}_i;\, \boldsymbol{\mu}_c, \boldsymbol{\Sigma}_c\right)$$

  "marginal" because we are marginalizing over y_i for each $\mathbf{x}_i$

- Learning mixture models = finding parameters that maximize marginal log-likelihood given the observed data
- No closed-form solution because of the sum inside log
  - We could use gradient ascent to maximize marginal log-likelihood, but today we will see an alternative approach

Mixture density estimation

- Suppose that we do observe $y_i \in \{1, \ldots, k\}$ for each $i = 1, \ldots, n$.
- Let us introduce a set of binary **indicator variables** $\mathbf{z}_i = [z_{i1}, \ldots, z_{ik}]$ where

$$z_{ic} = \begin{cases} 1 & \text{if } y_i = c \\ 0 & \text{otherwise} \end{cases}$$

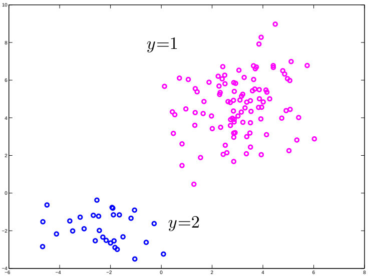
- Number of examples from c-th component:

$$n_c = \sum_{i=1}^{n} z_{ic}$$

Mixture density estimation: known "labels"

$\mathbf{z}_i = [z_{i1}, \ldots, z_{ik}]$ where

$$z_{ic} = \begin{cases} 1 & \text{if } y_i = c \\ 0 & \text{otherwise} \end{cases}$$



- If we know $\mathbf{z}_i$, the ML estimates of the Gaussian components, just like in class-conditional model, are

$$\widehat{\pi}_c = \frac{n_c}{n}, \text{ where } n_c \text{ is number of examples from component } c$$

$$\widehat{\boldsymbol{\mu}}_c = \frac{1}{n_c}\sum_{i=1}^{n} z_{ic}\mathbf{x}_i$$

$$\widehat{\boldsymbol{\Sigma}_c} = \frac{1}{n_c}\sum_{i=1}^{n} z_{ic}(\mathbf{x}_i - \widehat{\boldsymbol{\mu}}_c)(\mathbf{x}_i - \widehat{\boldsymbol{\mu}}_c)^\top$$

multiple Gaussian
weight sums to 1
pi is the prior
distribution

Credit assignment

- When we don't know $\mathbf{z}_i$, we face a **credit assignment** problem: which component is responsible for $\mathbf{x}_i$?
- Suppose for a moment that we do know component parameters $\boldsymbol{\mu}_1, \boldsymbol{\Sigma}_1, \ldots, \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k$ and mixing probabilities $\boldsymbol{\pi} = [\pi_1, \ldots, \pi_k]$
- Then, the posterior probability of component c using Bayes' rule:

$$\gamma_{ic} \;=\; \widehat{p}(y_i = c \,|\, \mathbf{x}_i;\, \boldsymbol{\pi}, \boldsymbol{\mu}_1, \boldsymbol{\Sigma}_1, \ldots) \;=\; \frac{\pi_c\, p(\mathbf{x}_i;\, \boldsymbol{\mu}_c, \boldsymbol{\Sigma}_c)}{\sum_{l=1}^{k} \pi_l\, p(\mathbf{x}_i;\, \boldsymbol{\mu}_l, \boldsymbol{\Sigma}_l)}$$

Credit assignment

- When we don't know $\mathbf{z}_i$, we face a **credit assignment** problem: which component is responsible for $\mathbf{x}_i$?
- Suppose for a moment that we do know component parameters $\boldsymbol{\mu}_1, \boldsymbol{\Sigma}_1, \ldots, \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k$ and mixing probabilities $\boldsymbol{\pi} = [\pi_1, \ldots, \pi_k]$
- Then, the posterior probability of component c using Bayes' rule:

$$\gamma_{ic} \;=\; \widehat{p}(y_i = c \,|\, \mathbf{x}_i;\, \boldsymbol{\pi}, \boldsymbol{\mu}_1, \boldsymbol{\Sigma}_1, \ldots) \;=\; \frac{\pi_c\, p(\mathbf{x}_i;\, \boldsymbol{\mu}_c, \boldsymbol{\Sigma}_c)}{\sum_{l=1}^{k} \pi_l\, p(\mathbf{x}_i;\, \boldsymbol{\mu}_l, \boldsymbol{\Sigma}_l)}$$

- We will call γ_{ic} the **responsibility** of the c-th component for $\mathbf{x}_i$
  - Note: $\sum_{c=1}^{k} \gamma_{ic} = 1$ for each i
  - Intuition: "competition" among components for explaining $\mathbf{x}_i$

Credit assignment

- When we don't know $\mathbf{z}_i$, we face a **credit assignment** problem: which component is responsible for $\mathbf{x}_i$?
- Suppose for a moment that we do know component parameters $\boldsymbol{\mu}_1, \boldsymbol{\Sigma}_1, \ldots, \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k$ and mixing probabilities $\boldsymbol{\pi} = [\pi_1, \ldots, \pi_k]$
- Then, the posterior probability of component c using Bayes' rule:

$$\gamma_{ic} \;=\; \widehat{p}(y_i = c \,|\, \mathbf{x}_i;\, \boldsymbol{\pi}, \boldsymbol{\mu}_1, \boldsymbol{\Sigma}_1, \ldots) \;=\; \frac{\pi_c\, p(\mathbf{x}_i;\, \boldsymbol{\mu}_c, \boldsymbol{\Sigma}_c)}{\sum_{l=1}^{k} \pi_l\, p(\mathbf{x}_i;\, \boldsymbol{\mu}_l, \boldsymbol{\Sigma}_l)}$$

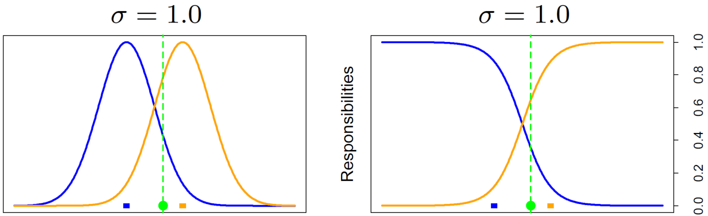
- We will call γ_{ic} the **responsibility** of the c-th component for $\mathbf{x}_i$
    - Note: $\sum_{c=1}^{k} \gamma_{ic} = 1$ for each i
    - Intuition: "competition" among components for explaining $\mathbf{x}_i$
- We will refer to the full posterior distribution as γ_i:

$$\gamma_i \;=\; \widehat{p}(y_i \mid \mathbf{x}_i;\, \boldsymbol{\pi}, \boldsymbol{\mu}_1, \boldsymbol{\Sigma}_1, \ldots)$$

Intuition: responsibilities

- The responsibilities represent "soft assignments" of credit

$$\gamma_{ic} = \widehat{p}(y_i = c \mid \mathbf{x}_i; \boldsymbol{\pi}, \boldsymbol{\mu}_1, \boldsymbol{\Sigma}_1, \ldots) = \frac{\pi_c \, p(\mathbf{x}_i; \boldsymbol{\mu}_c, \boldsymbol{\Sigma}_c)}{\sum_{l=1}^{k} \pi_l \, p(\mathbf{x}_i; \boldsymbol{\mu}_l, \boldsymbol{\Sigma}_l)}$$



posterior of the green point on blue or yellow

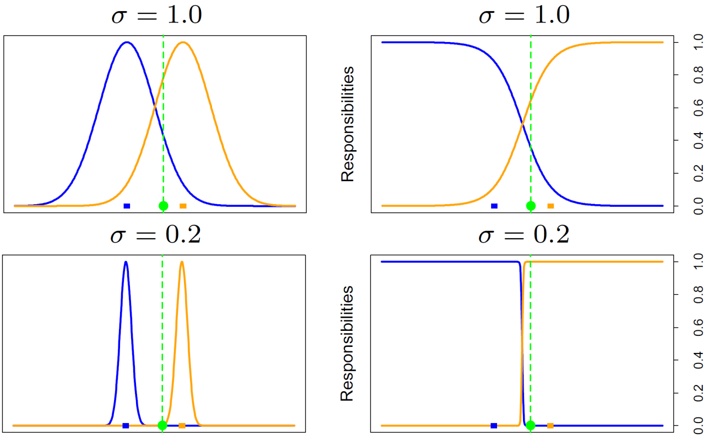
left: it is more likely that the green from the yellow point

[from Hastie & Tibshirani]

Intuition: responsibilities

- The responsibilities represent "soft assignments" of credit

$$\gamma_{ic} \;=\; \widehat{p}(y_i = c \,|\, \mathbf{x}_i;\, \boldsymbol{\pi}, \boldsymbol{\mu}_1, \boldsymbol{\Sigma}_1, \ldots) \;=\; \frac{\pi_c\, p(\mathbf{x}_i;\, \boldsymbol{\mu}_c, \boldsymbol{\Sigma}_c)}{\sum_{l=1}^{k} \pi_l\, p(\mathbf{x}_i;\, \boldsymbol{\mu}_l, \boldsymbol{\Sigma}_l)}$$



[from Hastie & Tibshirani]

Complete data log-likelihood

- The "complete data" or "complete" likelihood (when $\mathbf{z}$ are known):

$$p(X, Z;\, \boldsymbol{\pi}, \boldsymbol{\mu}_1, \ldots) \;=\; \prod_{i=1}^{n} \prod_{c=1}^{k} \left(\pi_c \mathcal{N}\left(\mathbf{x}_i;\, \boldsymbol{\mu}_c, \boldsymbol{\Sigma}_c\right)\right)^{z_{ic}}$$

  and taking the log gives us the **complete data log-likelihood**:

$$\log p(X, Z;\, \boldsymbol{\pi}, \boldsymbol{\mu}_1, \ldots) \;=\; \sum_{i=1}^{n} \sum_{c=1}^{k} z_{ic} \left(\log \pi_c \,+\, \log \mathcal{N}\left(\mathbf{x}_i;\, \boldsymbol{\mu}_c, \boldsymbol{\Sigma}_c\right)\right)$$

- But we can't compute this because we don't know $\mathbf{z}$

Expected complete log-likelihood

- Complete data log-likelihood:

$$\log p(X, Z; \boldsymbol{\pi}, \boldsymbol{\mu}_1, \ldots) = \sum_{i=1}^{n} \sum_{c=1}^{k} z_{ic} \left(\log \pi_c + \log \mathcal{N}\left(\mathbf{x}_i; \boldsymbol{\mu}_c, \boldsymbol{\Sigma}_c\right)\right)$$

- While we can't compute this (Z is unknown), we can take the *expectation* over the z_{ic} w.r.t. the posteriors over the z_{ic}
  - The result is the “expected complete” log-likelihood

Expected complete log-likelihood

- Complete data log-likelihood:

$$\log p(X, Z;\, \boldsymbol{\pi}, \boldsymbol{\mu}_1, \ldots) \,=\, \sum_{i=1}^{n} \sum_{c=1}^{k} z_{ic} \left(\log \pi_c \,+\, \log \mathcal{N}\left(\mathbf{x}_i;\, \boldsymbol{\mu}_c, \boldsymbol{\Sigma}_c\right)\right)$$

- While we can't compute this (Z is unknown), we can take the *expectation* over the z_{ic} w.r.t. the posteriors over the z_{ic}
  - The result is the "expected complete" log-likelihood

- Posterior distribution over values for z_{ic}:

$$\widehat{p}(z_{ic} = 1 \mid \mathbf{x}_i; \boldsymbol{\pi}, \boldsymbol{\mu}_1, \ldots) = \widehat{p}(y_i = c \mid \mathbf{x}_i; \boldsymbol{\pi}, \boldsymbol{\mu}_1, \ldots) = \gamma_{ic}$$
$$\widehat{p}(z_{ic} = 0 \mid \mathbf{x}_i; \boldsymbol{\pi}, \boldsymbol{\mu}_1, \ldots) = 1 - \widehat{p}(y_i = c \mid \mathbf{x}_i; \boldsymbol{\pi}, \boldsymbol{\mu}_1, \ldots) = 1 - \gamma_{ic}$$

Zic transform to gamma

Expected complete log-likelihood

- Complete data log-likelihood:

$$\log p(X, Z;\, \boldsymbol{\pi}, \boldsymbol{\mu}_1, \ldots) \;=\; \sum_{i=1}^{n} \sum_{c=1}^{k} z_{ic} \left(\log \pi_c \,+\, \log \mathcal{N}\left(\mathbf{x}_i;\, \boldsymbol{\mu}_c, \boldsymbol{\Sigma}_c\right)\right)$$

- While we can't compute this (Z is unknown), we can take the *expectation* over the z_{ic} w.r.t. the posteriors over the z_{ic}
  - The result is the "expected complete" log-likelihood

- Posterior distribution over values for z_{ic}:

$$\widehat{p}(z_{ic} = 1 \mid \mathbf{x}_i; \boldsymbol{\pi}, \boldsymbol{\mu}_1, \ldots) = \widehat{p}(y_i = c \mid \mathbf{x}_i; \boldsymbol{\pi}, \boldsymbol{\mu}_1, \ldots) = \gamma_{ic}$$
$$\widehat{p}(z_{ic} = 0 \mid \mathbf{x}_i; \boldsymbol{\pi}, \boldsymbol{\mu}_1, \ldots) = 1 - \widehat{p}(y_i = c \mid \mathbf{x}_i; \boldsymbol{\pi}, \boldsymbol{\mu}_1, \ldots) = 1 - \gamma_{ic}$$

- Expectation of a single z_{ic} w.r.t. the posterior $\widehat{p}(z_{ic} \mid \mathbf{x}_i; \boldsymbol{\pi}, \boldsymbol{\mu}_1, \ldots)$:

$$\mathbb{E}_{z_{ic} \sim \widehat{p}(z_{ic} \mid \mathbf{x}_i; \boldsymbol{\pi}, \boldsymbol{\mu}_1, \ldots)} \left[z_{ic} \right] \;=\; 0 \cdot (1 - \gamma_{ic}) + 1 \cdot \gamma_{ic} = \gamma_{ic}$$

Complete and expected complete log-likelihood

- Complete log-likelihood:

$$\log p(X, Z;\, \boldsymbol{\pi}, \boldsymbol{\mu}_1, \ldots) = \sum_{i=1}^{n} \sum_{c=1}^{k} z_{ic} \left(\log \pi_c + \log \mathcal{N}\left(\mathbf{x}_i;\, \boldsymbol{\mu}_c, \boldsymbol{\Sigma}_c\right) \right)$$

- **Expected complete log-likelihood**:

$$\mathbb{E}_{z_{ic} \sim \widehat{p}(z_{ic} | \mathbf{x}_i; \boldsymbol{\pi}, \boldsymbol{\mu}_1, \ldots)} \left[\log p(X, Z;\, \boldsymbol{\pi}, \boldsymbol{\mu}_1, \ldots) \right] = \sum_{i=1}^{n} \sum_{c=1}^{k} \gamma_{ic} \left(\log \pi_c + \log \mathcal{N}\left(\mathbf{x}_i;\, \boldsymbol{\mu}_c, \boldsymbol{\Sigma}_c\right) \right)$$

Maximizing the expected complete log-likelihood

- Expected complete log-likelihood:

$$\mathbb{E}_{z_{ic} \sim \widehat{p}(z_{ic}|\ldots)} \left[\log p(X, Z; \boldsymbol{\pi}, \boldsymbol{\mu}_1, \ldots) \right] = \sum_{i=1}^{n} \sum_{c=1}^{k} \gamma_{ic} (\log \pi_c + \log \mathcal{N}(\mathbf{x}_i;\, \boldsymbol{\mu}_c, \boldsymbol{\Sigma}_c))$$

- We can find $\boldsymbol{\pi}$, $\boldsymbol{\mu}_1, \ldots, \boldsymbol{\Sigma}_k$ that maximize this expected likelihood by setting derivatives to zero and, for $\boldsymbol{\pi}$, using Lagrange multipliers to enforce $\sum_c \pi_c = 1$

Maximizing the expected complete log-likelihood

- Expected complete log-likelihood:

$$\mathbb{E}_{z_{ic}\sim\widehat{p}(z_{ic}|\ldots)}\left[\log p(X, Z; \boldsymbol{\pi}, \boldsymbol{\mu}_1, \ldots)\right] = \sum_{i=1}^{n}\sum_{c=1}^{k}\gamma_{ic}(\log \pi_c + \log \mathcal{N}\left(\mathbf{x}_i;\, \boldsymbol{\mu}_c, \boldsymbol{\Sigma}_c\right))$$

- We can find $\boldsymbol{\pi}$, $\boldsymbol{\mu}_1, \ldots, \boldsymbol{\Sigma}_k$ that maximize this expected likelihood by setting derivatives to zero and, for $\boldsymbol{\pi}$, using Lagrange multipliers to enforce $\sum_c \pi_c = 1$
- We get closed-form solutions!

$$\begin{aligned}
\hat{\pi}_c &= \frac{1}{n}\sum_{i=1}^{n}\gamma_{ic} \\
\hat{\boldsymbol{\mu}}_c &= \frac{1}{\sum_{i=1}^{n}\gamma_{ic}}\sum_{i=1}^{n}\gamma_{ic}\mathbf{x}_i \\
\widehat{\boldsymbol{\Sigma}}_c &= \frac{1}{\sum_{i=1}^{n}\gamma_{ic}}\sum_{i=1}^{n}\gamma_{ic}(\mathbf{x}_i - \hat{\boldsymbol{\mu}}_c)(\mathbf{x}_i - \hat{\boldsymbol{\mu}}_c)^\top
\end{aligned}$$

Are we done?

- We found values for $\boldsymbol{\pi}$, $\boldsymbol{\mu}_1, \ldots, \boldsymbol{\Sigma}_k$ that maximize the expected complete log-likelihood:

$$\mathbb{E}_{z_{ic} \sim \widehat{p}(z_{ic}|\ldots)} \left[\log p(X, Z; \boldsymbol{\pi}, \boldsymbol{\mu}_1, \ldots)\right] = \sum_{i=1}^{n} \sum_{c=1}^{k} \gamma_{ic} (\log \pi_c + \log \mathcal{N}\left(\mathbf{x}_i;\, \boldsymbol{\mu}_c, \boldsymbol{\Sigma}_c\right))$$

Are we done?

- We found values for $\boldsymbol{\pi}$, $\boldsymbol{\mu}_1, \ldots, \boldsymbol{\Sigma}_k$ that maximize the expected complete log-likelihood:

$$\mathbb{E}_{z_{ic} \sim \widehat{p}(z_{ic}|\ldots)} [\log p(X, Z; \boldsymbol{\pi}, \boldsymbol{\mu}_1, \ldots)] = \sum_{i=1}^{n} \sum_{c=1}^{k} \gamma_{ic} (\log \pi_c + \log \mathcal{N}(\mathbf{x}_i;\, \boldsymbol{\mu}_c, \boldsymbol{\Sigma}_c))$$

- However, note that the posterior distributions (that we computed the expectation with respect to) actually depend on the parameters $\boldsymbol{\pi}$, $\boldsymbol{\mu}_1, \ldots, \boldsymbol{\Sigma}_k$
- We treated the posterior probabilities (the γ_{ic}) as constants, which simplified the optimization problem

Are we done?

- We found values for $\boldsymbol{\pi}$, $\boldsymbol{\mu}_1, \ldots, \boldsymbol{\Sigma}_k$ that maximize the expected complete log-likelihood:

$$\mathbb{E}_{z_{ic} \sim \widehat{p}(z_{ic}|\ldots)} \left[\log p(X, Z; \boldsymbol{\pi}, \boldsymbol{\mu}_1, \ldots)\right] = \sum_{i=1}^{n} \sum_{c=1}^{k} \gamma_{ic} (\log \pi_c + \log \mathcal{N}(\mathbf{x}_i; \boldsymbol{\mu}_c, \boldsymbol{\Sigma}_c))$$

- However, note that the posterior distributions (that we computed the expectation with respect to) actually depend on the parameters $\boldsymbol{\pi}$, $\boldsymbol{\mu}_1, \ldots, \boldsymbol{\Sigma}_k$
- We treated the posterior probabilities (the γ_{ic}) as constants, which simplified the optimization problem
- Since we have new estimates for $\boldsymbol{\pi}$, $\boldsymbol{\mu}_1, \ldots, \boldsymbol{\Sigma}_k$, the posteriors are now out of date!

Summary so far

which
component

- If we know the **parameters** and **indicators** (assignments) we are done.
- If we know the **indicators** but not the parameters, we can do ML estimation of the parameters – and we are done.
- If we know the **parameters** but not the indicators, we can compute the posteriors of indicators; Previous slides
  - With known posteriors, we can estimate parameters that maximize the *expected* likelihood – and then we are done.
- But in reality we know neither the parameters nor the indicators.

We cannot get the posteriors in this case

The expectation-maximization (EM) algorithm

- Start with a guess of $\boldsymbol{\pi}, \boldsymbol{\mu}_1, \ldots$
  - Typically, random Gaussians and $\pi_c = 1/k$
- Iterate between:

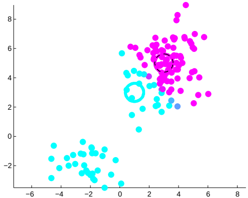
  E-step Compute values of expected assignments, i.e., calculate γ_{ic}, using current estimates of $\boldsymbol{\pi}, \boldsymbol{\mu}_1, \ldots$

  M-step Maximize the expected complete log-likelihood, under current γ_{ic}
- Repeat until convergence.

EM for Gaussian mixture: an example

- Colors represent γ_{ic} after the E-step.

1st iteration



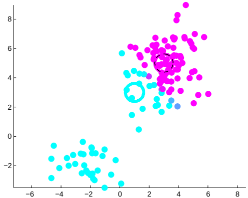
start with random parameter of two gaussians

reesitmate the parameter based on the posterior

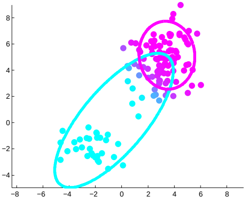
EM for Gaussian mixture: an example

- Colors represent γ_{ic} after the E-step.

1st iteration



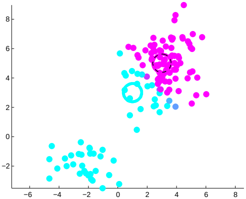
2nd iteration



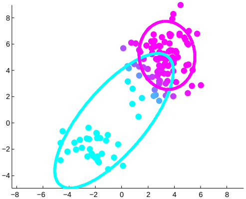
EM for Gaussian mixture: an example

- Colors represent γ_{ic} after the E-step.

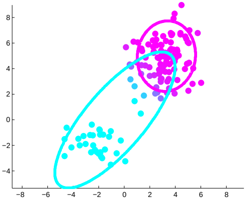
1st iteration



2nd iteration

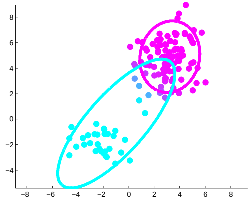


3rd iteration



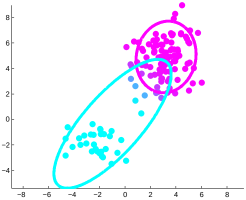
EM for Gaussian mixture: an example

4th iteration

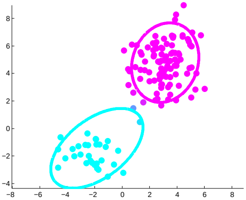


EM for Gaussian mixture: an example

4th iteration

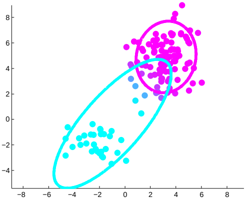


7th iteration

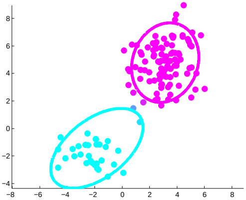


EM for Gaussian mixture: an example

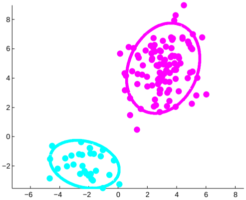
4th iteration

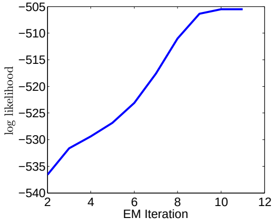


7th iteration



10th iteration





Marginal log-likelihood progress with iterations

Generic EM for mixture models

- General mixture models: $p(\mathbf{x}) = \sum_{c=1}^{k} \pi_c \, p(\mathbf{x};\, \boldsymbol{\theta}_c)$
- Initialize $\boldsymbol{\pi}$, $\boldsymbol{\theta}_c^{old}$ for all c, and iterate until convergence:

  E-step: compute responsibilities

$$\gamma_{ic} \;=\; \frac{\pi_c^{old}\, p(\mathbf{x}_i; \boldsymbol{\theta}_c^{old})}{\sum_{l=1}^{k} \pi_l^{old}\, p(\mathbf{x}_i; \boldsymbol{\theta}_l^{old})}$$

  M-step: re-estimate mixture parameters:

$$\boldsymbol{\pi}^{new},\, \boldsymbol{\theta}^{new} \;=\; \underset{\boldsymbol{\theta},\boldsymbol{\pi}}{\operatorname{argmax}} \sum_{i=1}^{n} \sum_{c=1}^{k} \gamma_{ic} \left(\log \pi_c \,+\, \log p(\mathbf{x}_i;\, \boldsymbol{\theta}_c)\right)$$

The EM algorithm in general

- Observed data X, hidden variables Z.
  - E.g., *missing data*.
- Initialize θ^{old} and iterate until convergence:

  E-step: Compute the expected complete log-likelihood as a function of θ.

$$Q\Big(\theta;\, \theta^{old}\Big) \;=\; \mathbb{E}_{p(Z \,|\, X,\, \theta^{old})}\Big[\log p(X, Z;\, \theta) \,|\, X,\, \theta^{old}\Big]$$

  M-step: Compute

$$\theta^{new} \;=\; \operatorname*{argmax}_{\theta}\, Q\Big(\theta;\, \theta^{old}\Big)$$

Why does EM work?

- Recall: our initial goal was to maximize marginal log-likelihood of the observed data:

$$\theta^* \;=\; \underset{\theta}{\operatorname{argmax}} \log p(X;\theta)$$

- Let $\log p^{(t)}$ be $\log p(X;\theta^{new})$ after t iterations of EM
- Can show:

$$\log p^{(0)} \;\leq\; \log p^{(1)} \;\leq \ldots \leq \log p^{(t)} \ldots$$

Understanding EM (1/2)

M-step maximizes expected complete log-likelihood, which is a lower bound on marginal log-likelihood:

$$\sum_{i=1}^{n} \log \sum_{z} p(\mathbf{x}_i, z;\, \theta)$$

$$= \sum_{i=1}^{n} \log \sum_{z} q_i(z) \frac{p(\mathbf{x}_i, z;\, \theta)}{q_i(z)} \quad \text{(for any distribution } q_i(z) \text{ s.t. } q_i(z) \neq 0\, \forall z\text{)}$$

$$\geq \sum_{i=1}^{n} \sum_{z} q_i(z) \log \frac{p(\mathbf{x}_i, z;\, \theta)}{q_i(z)} \quad \text{(by Jensen's inequality (}\log\text{ is concave))}$$

like a expectation

$$= \sum_{i=1}^{n} \sum_{z} q_i(z) \log p(\mathbf{x}_i, z;\, \theta) - \sum_{i=1}^{n} \sum_{z} q_i(z) \log q_i(z)$$

Understanding EM (2/2)

M-step maximizes expected complete log-likelihood, which is a lower bound on marginal log-likelihood:

$$\begin{aligned}
\sum_{i=1}^{n} \log \sum_{z} p(\mathbf{x}_i, z;\, \theta) &\geq \sum_{i=1}^{n} \sum_{z} q_i(z) \log \frac{p(\mathbf{x}_i, z;\, \theta)}{q_i(z)} \\
&= \sum_{i=1}^{n} \sum_{z} q_i(z) \log p(\mathbf{x}_i, z;\, \theta) - \sum_{i=1}^{n} \sum_{z} q_i(z) \log q_i(z)
\end{aligned}$$

Understanding EM (2/2)

M-step maximizes expected complete log-likelihood, which is a lower bound on marginal log-likelihood:

$$\begin{aligned}\sum_{i=1}^{n} \log \sum_{z} p(\mathbf{x}_i, z;\, \theta) &\geq \sum_{i=1}^{n} \sum_{z} q_i(z) \log \frac{p(\mathbf{x}_i, z;\, \theta)}{q_i(z)} \\ &= \sum_{i=1}^{n} \sum_{z} q_i(z) \log p(\mathbf{x}_i, z;\, \theta) - \sum_{i=1}^{n} \sum_{z} q_i(z) \log q_i(z)\end{aligned}$$

If we define $q_i(z) = p(z \mid \mathbf{x}_i)$, this becomes:

$$\underbrace{\sum_{i=1}^{n} \sum_{z} p(z \mid \mathbf{x}_i) \log p(\mathbf{x}_i, z;\, \theta)}_{\text{expected complete log-likelihood}} + \sum_{i=1}^{n} \underbrace{\left(-\sum_{z} p(z \mid \mathbf{x}_i) \log p(z \mid \mathbf{x}_i) \right)}_{\text{entropy of } p(z \mid \mathbf{x}_i)}$$

Understanding EM (2/2)

M-step maximizes expected complete log-likelihood, which is a lower bound on marginal log-likelihood:

$$\sum_{i=1}^{n} \log \sum_{z} p(\mathbf{x}_i, z;\, \theta) \geq \sum_{i=1}^{n} \sum_{z} q_i(z) \log \frac{p(\mathbf{x}_i, z;\, \theta)}{q_i(z)}$$

$$= \sum_{i=1}^{n} \sum_{z} q_i(z) \log p(\mathbf{x}_i, z;\, \theta) - \sum_{i=1}^{n} \sum_{z} q_i(z) \log q_i(z)$$

If we define $q_i(z) = p(z \mid \mathbf{x}_i)$, this becomes: entropy: how flat is the distribution

$$\underbrace{\sum_{i=1}^{n} \sum_{z} p(z \mid \mathbf{x}_i) \log p(\mathbf{x}_i, z;\, \theta)}_{\text{expected complete log-likelihood}} + \sum_{i=1}^{n} \underbrace{\left(-\sum_{z} p(z \mid \mathbf{x}_i) \log p(z \mid \mathbf{x}_i) \right)}_{\text{entropy of } p(z \mid \mathbf{x}_i)}$$

Since entropy is non-negative, expected complete log-likelihood $\leq$ marginal log-likelihood

Direct optimization vs. EM

- EM is an algorithm for maximizing the marginal log-likelihood:

$$\log p(X;\, \theta) \;=\; \sum_{i=1}^{n} \log \sum_{z} p(\mathbf{x}_i, z;\, \theta)$$

- What about directly maximizing this quantity using gradient ascent?
- Computing gradients requires iterating over all hidden variable values for each example, but EM has to do that too!
- So then what advantages does EM provide?

EM might gave us closed form solution in
M step.

no need to tune the learning rate.

Direct optimization vs. EM

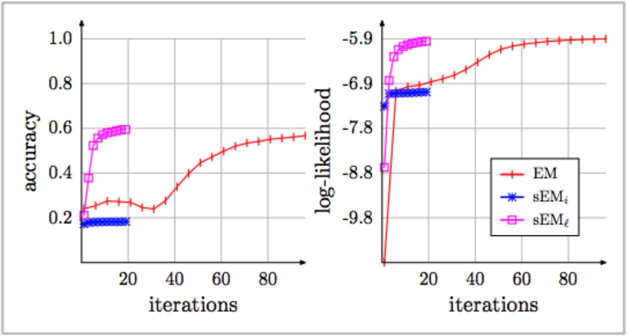
- EM is an algorithm for maximizing the marginal log-likelihood:

$$\log p(X;\, \theta) \;=\; \sum_{i=1}^{n} \log \sum_{z} p(\mathbf{x}_i, z;\, \theta)$$

- What about directly maximizing this quantity using gradient ascent?
- Computing gradients requires iterating over all hidden variable values for each example, but EM has to do that too!
- So then what advantages does EM provide?
  - For certain models (e.g., Gaussian mixtures), the M-step has a closed form solution
  - This closed form solution handles constraints on parameters (e.g., that entries of $\boldsymbol{\pi}$ are non-negative and sum to 1)
  - No need to tune learning rates or other optimizer-related decisions (when closed-form solutions are available)

Batch vs. mini-batch versions of EM

- EM is a "batch" algorithm
- It requires going through entire dataset for each update
- But there are mini-batch versions (online EM, incremental EM, stepwise EM, etc.) that generally work much better than standard EM

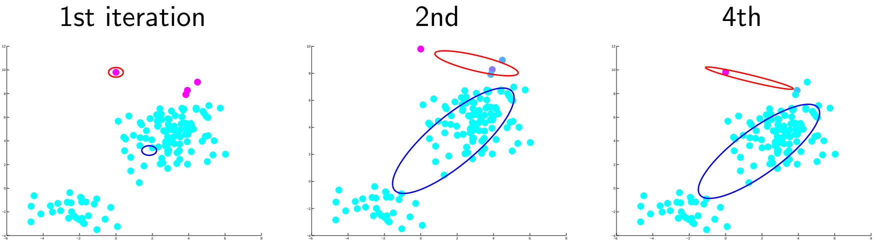


(a) POS tagging

Compare red ticks (EM) to pink squares (stepwise EM with hyperparameters tuned to maximize log-likelihood) (Liang and Klein, 2009)

EM and overfitting

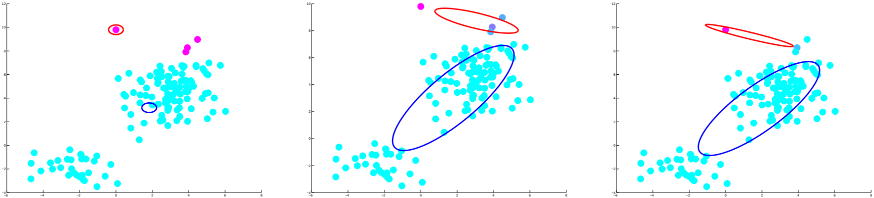
- We can be very unlucky with the initial guess.



EM and overfitting

- We can be very unlucky with the initial guess.

1st iteration 2nd 4th



- The problem:

$$\lim_{\sigma^2 \to 0} \mathcal{N}\left(\mathbf{x};\, \mu = \mathbf{x}, \mathbf{\Sigma} = \sigma^2 \mathbf{I}\right) \;=\; \infty$$

- i.e., we can increase likelihood by shrinking the variance of a Gaussian centered on a single point

Regularized EM

- Impose a prior on θ.
- Instead of maximizing the likelihood in the M-step, maximize the posterior:

$$\theta^{new} \; = \; \underset{\theta}{\operatorname{argmax}} \left\{ \mathbb{E}_{p(Z|X;\theta)} \left[\log p(X, Z; \theta) \mid X; \theta^{old} \right] \; + \; \log p(\theta) \right\}$$

Regularized EM

- Impose a prior on θ.
- Instead of maximizing the likelihood in the M-step, maximize the posterior:

$$\theta^{new} \;=\; \underset{\theta}{\operatorname{argmax}} \left\{ \mathbb{E}_{p(Z|X;\theta)} \left[\log p(X, Z; \theta) \mid X; \theta^{old} \right] \;+\; \log p(\theta) \right\}$$

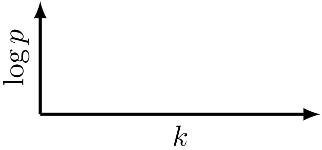
- A common prior on a covariance matrix: the **Wishart** distribution

$$p(\mathbf{\Sigma}; \mathbf{S}, n) \;\propto\; \frac{1}{|\mathbf{\Sigma}|^{n/2}} \exp\left(-\frac{1}{2} \operatorname{Tr}\left(\mathbf{\Sigma}^{-1} \mathbf{S} \right) \right)$$

- Intuition: $\mathbf{S}$ is the covariance of n “hallucinated” observations.

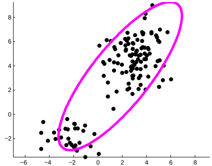
Model selection: setting k

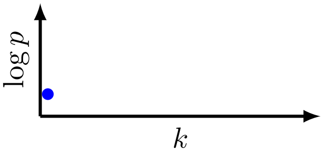
- So far we have assumed known k.
- Idea: select k that maximizes the likelihood.



Model selection: setting k

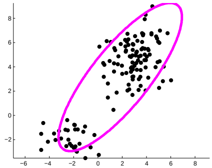
- So far we have assumed known k.
- Idea: select k that maximizes the likelihood.

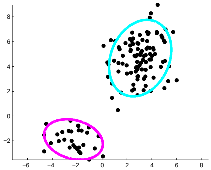


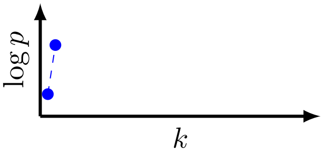


Model selection: setting k

- So far we have assumed known k.
- Idea: select k that maximizes the likelihood.

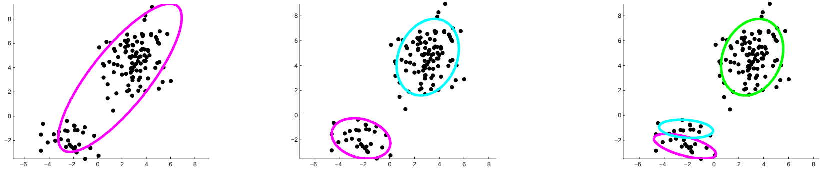


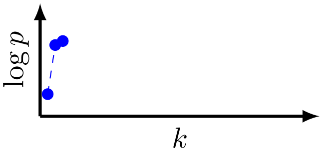




Model selection: setting k

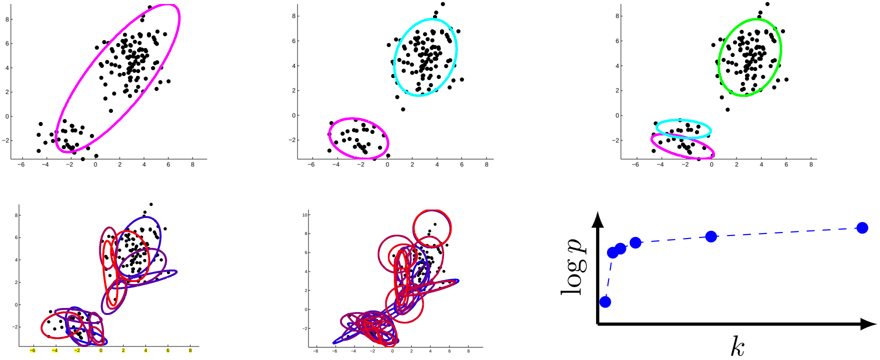
- So far we have assumed known k.
- Idea: select k that maximizes the likelihood.





Model selection: setting k

- So far we have assumed known k.
- Idea: select k that maximizes the likelihood.



could use BIC or other criteria

Overfitting mixture models

- Imagine placing a separate, very narrow Gaussian component on every training example.
- This solution yields infinite log-likelihood!
- Solution 2: validate on a held-out data set
- Solution 1: penalize model complexity directly, e.g., the Bayesian Information Criterion (BIC)
- For a model class $\mathcal{M}$ with parameters $\theta_{\mathcal{M}}$, we find ML (or MAP) estimates of the parameters on $X = [\mathbf{x}_1, \ldots, \mathbf{x}_n]$:

$$L^*(\mathcal{M}) \;\triangleq\; \max_{\theta_{\mathcal{M}}} \log p(X \mid \mathcal{M}; \theta_{\mathcal{M}})$$

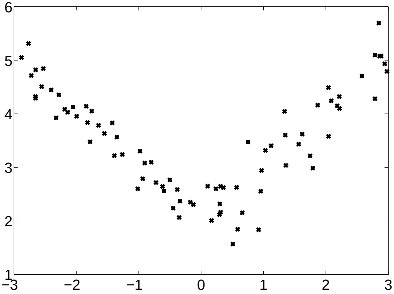
  e.g., $\mathcal{M} = \{\text{mixtures of 5 Gaussians}\}$
- The BIC score for the model $\mathcal{M}$ on data X:

$$BIC(\mathcal{M}) \;=\; L^*(\mathcal{M}) - \frac{1}{2}|\theta_{\mathcal{M}}| \log n$$

  where $|\theta_{\mathcal{M}}|$ is the number of parameters in a model from class $\mathcal{M}$

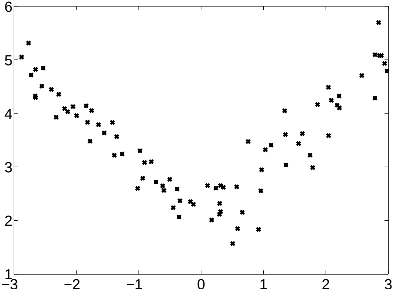
Mixture model for regression

- Example:



Mixture model for regression

- Example:



- We can represent this as a mixture of two regression models
  - Two *experts*;
  - Need to switch between them according to $\mathbf{x}$.

Mixture of experts model

- Expert j holds a parameteric model $p\left(y \,|\, \mathbf{x}; \theta_j\right)$, e.g.,

$$\theta_j \;=\; \{\mathbf{w}_j, \sigma_j^2\},$$

$$p\left(y \,|\, \mathbf{x}; \theta_j\right) \;=\; \mathcal{N}\left(y;\, \mathbf{w}_j^T \mathbf{x},\, \sigma_j^2\right)$$

- The distribution of y is a *conditional mixture* model:

$$p\left(y \,|\, \mathbf{x};\, \theta\right) \;=\; \sum_{j=1}^{c} p\left(j \,|\, \mathbf{x}\right) p\left(y \,|\, \mathbf{x}; \theta_j\right).$$

Gating network

$$p\left(y \,|\, \mathbf{x};\, \theta\right) \;=\; \sum_{j=1}^{c} p\left(j \,|\, \mathbf{x}\right) p\left(y \,|\, \mathbf{x}; \theta_j\right)$$

- A *gating network* specifies the conditional distribution $p\left(j \,|\, \mathbf{x};\, \eta\right)$
- Common choice for gaiting: softmax, $\eta = \{\mathbf{v}_1, \ldots, \mathbf{v}_k\}$

$$p\left(j \,|\, \mathbf{x}; \eta\right) \;=\; \frac{e^{\mathbf{v}_j^T \mathbf{x}}}{\sum_{t=1}^{k} e^{\mathbf{v}_t^T \mathbf{x}}}.$$

- Can think of it as classification into which expert should be responsible for an example
- Responsibilities:

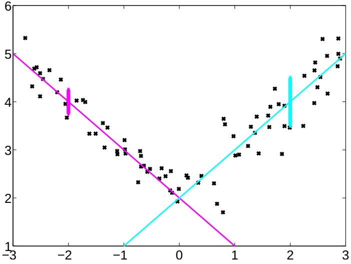
$$\gamma_{ij} \;=\; p\left(j \,|\, \mathbf{x}_i, y_i; \theta, \eta\right) \;=\; \frac{p\left(j \,|\, \mathbf{x}_i; \eta\right) p\left(y_i \,|\, \mathbf{x}_i; \theta_j\right)}{\sum_{c=1}^{k} p\left(c \,|\, \mathbf{x}_i; \eta\right) p\left(y_i \,|\, \mathbf{x}_i; \theta_c\right)}$$

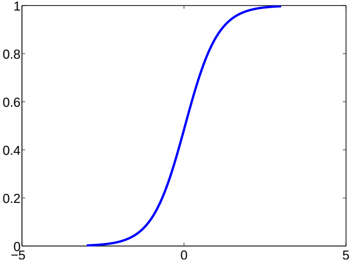
Conditional mixtures

- Parametrization
  - Regression models $p\,(y\,|\,\mathbf{x};\theta_j)$

    e.g., linear regressors, $\theta_j \;=\; \{\mathbf{w}_j, \sigma_j^2\}$.
  - Gating network $p\,(j\,|\,\mathbf{x};\eta)$

    e.g., logistic regression, $\eta \;=\; \{\mathbf{v}\}$





EM for mixtures of experts

Initialize random θ_j, σ_j^2, η.

E-step Compute responsibilities $\gamma_{ij} = p\left(j \mid \mathbf{x}_i, y_i; \theta^{old}, \eta^{old}\right)$

M-step Separately:

- For each expert j estimate

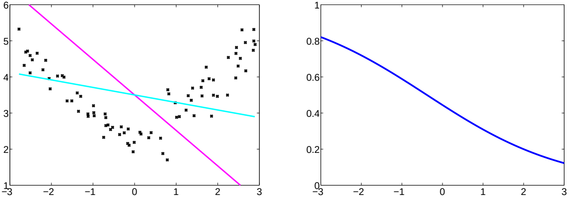
$$\theta_j^{new} \;=\; \operatorname*{argmax}_{\theta} \sum_{i=1}^{N} \gamma_{ij} \log p\left(y_i \mid \mathbf{x}_i; \theta\right)$$

- Estimate the gating network:

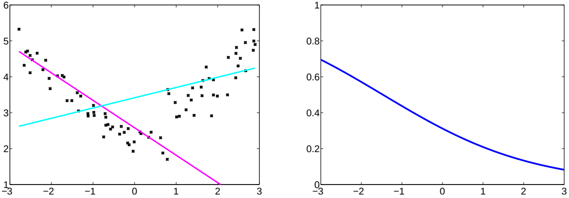
$$\eta^{new} \;=\; \operatorname*{argmax}_{\eta} \sum_{i=1}^{N} \sum_{j=1}^{k} \gamma_{ij} \log p\left(j \mid \mathbf{x}_i; \eta\right)$$

EM for mixtures of experts: example

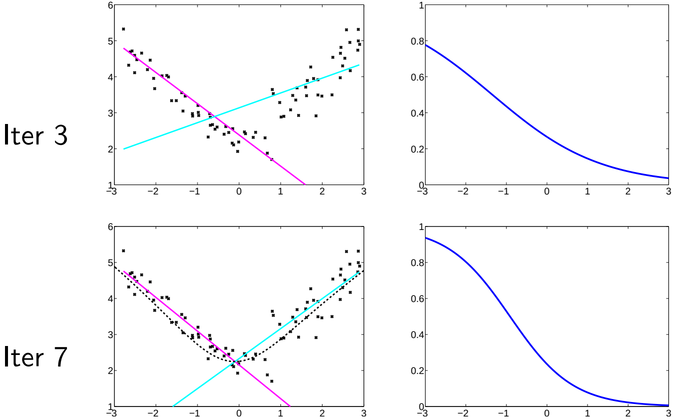
Iter 1



Iter 2



EM for mixtures of experts: example



Review: generative models, C classes

- Construct for each class c

$$\delta_c(\mathbf{x}) \triangleq \log p(\mathbf{x} \mid y = c) + \log p(y = c)$$

  based on our per-class (class-conditional) model $p(\mathbf{x} \mid y = c)$
- Generative classifier:

$$h^*(\mathbf{x}) = \operatorname*{argmax}_c \delta_c(\mathbf{x}).$$

- If assume equal priors $p(y = c) = 1/C$, then $h^*(\mathbf{x}) = \operatorname{argmax}_c \log p(\mathbf{x} \mid y = c)$.
- Learning = fitting a model of $\mathbf{x}$ for each class y
- Some models we have considered for $p(\mathbf{x} \mid y)$: Gaussian (with shared or individual covariances) $\Rightarrow$ML estimate (closed form) mixture of Gaussians (or of other densities) $\Rightarrow$ML via the EM algorithm