# Expectation Maximization

Dam Quang Tuan

# Building intuition for EM: a case study

Let events be "grades in a class"

| | |
|---|---|
| $w_1$ = Gets an A | $P(A) = \frac{1}{2}$ |
| $w_2$ = Gets a B | $P(B) = \mu$ |
| $w_3$ = Gets a C | $P(C) = 2\mu$ |
| $w_4$ = Gets a D | $P(D) = \frac{1}{2} - 3\mu$ |

(Note $0 \leq \mu \leq 1/6$)

Assume we want to estimate $\mu$ from data. In a given class there were

a   A's
b   B's
c   C's
d   D's

What's the maximum likelihood estimate of $\mu$ given a,b,c,d ?

# Max likelihood solution

P(A) = ½    P(B) = μ    P(C) = 2μ    P(D) = ½-3μ

P( $a,b,c,d$ | μ) = K(½)$^a$(μ)$^b$(2μ)$^c$(½-3μ)$^d$

log P( $a,b,c,d$ | μ) = log K + $a$log ½ + $b$log μ + $c$log 2μ + $d$log (½-3μ)

$$\text{FOR MAX LIKE } \mu, \text{ SET } \frac{\partial \text{LogP}}{\partial \mu} = 0$$

$$\frac{\partial \text{LogP}}{\partial \mu} = \frac{b}{\mu} + \frac{2c}{2\mu} - \frac{3d}{1/2 - 3\mu} = 0$$

$$\text{Gives max like } \mu = \frac{b+c}{6(b+c+d)}$$

So if class got

| A | B | C | D |
|---|---|---|---|
| 14 | 6 | 9 | 10 |

$$\text{Max like } \mu = \frac{1}{10}$$

# Same Problem with Hidden Information

Someone tells us that

Number of High grades (A's + B's) = $h$

Number of C's $\quad\quad\quad\quad\quad\quad\quad$ = $c$

Number of D's $\quad\quad\quad\quad\quad\quad\quad$ = $d$

What is the max. like estimate of $\mu$ now?

REMEMBER
$P(A) = \frac{1}{2}$
$P(B) = \mu$
$P(C) = 2\mu$
$P(D) = \frac{1}{2} - 3\mu$

# Same Problem with Hidden Information

Someone tells us that

Number of High grades (A's + B's) = $h$

Number of C's                     = $c$

Number of D's                     = $d$

What is the max. like estimate of μ now?

We can answer this question circularly:

**EXPECTATION**

If we know the value of μ we could compute the expected value of $a$ and $b$

Since the ratio a:b should be the same as the ratio ½ : μ

$$a = \frac{1/2}{1/2 + \mu} h \qquad b = \frac{\mu}{1/2 + \mu} h$$

**MAXIMIZATION**

If we know the expected values of $a$ and $b$ we could compute the maximum likelihood value of μ

$$\mu = \frac{b + c}{6(b + c + d)}$$

# EM for our problem

We begin with a guess for $\mu$

We iterate between EXPECTATION and MAXIMIZATION to improve our estimates of $\mu$ and $a$ and $b$.

Define   $\mu(t)$  the estimate of $\mu$ on the t'th iteration

   $b(t)$  the estimate of $b$ on t'th iteration

$$\mu(0) = \text{initial guess}$$

$$b(t) = \frac{\mu(t)h}{\frac{1}{2} + \mu(t)} = E\big[b \mid \mu(t)\big]$$

E -step

$$\mu(t+1) = \frac{b(t)+c}{6\big(b(t)+c+d\big)}$$

M -step

$$= \text{max like est of } \mu \text{ given } b(t)$$

# EM Convergence

- Convergence proof based on fact that Prob(data | μ) must increase or remain same between each iteration
  [NOT OBVIOUS]

- But it can never exceed 1    [OBVIOUS]
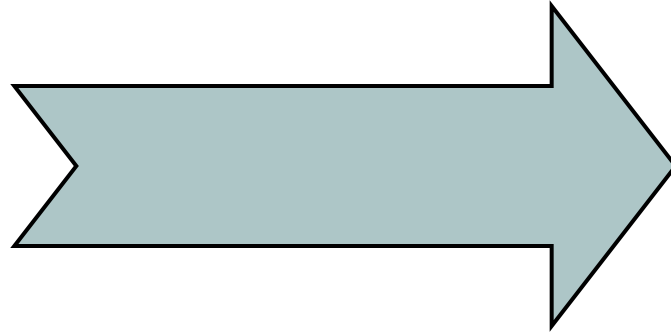
So it must therefore converge    [OBVIOUS]

---

In our example, suppose we had

      h = 20
      c = 10
      d = 10
      μ(0) = 0

Convergence is generally <u>linear</u>: error decreases by a constant factor each time step.

| t | μ(t) | b(t) |
|---|------|------|
| 0 | 0 | 0 |
| 1 | 0.0833 | 2.857 |
| 2 | 0.0937 | 3.158 |
| 3 | 0.0947 | 3.185 |
| 4 | 0.0948 | 3.187 |
| 5 | 0.0948 | 3.187 |
| 6 | 0.0948 | 3.187 |

# ALT-OPT/EM for Gaussian Mixture Model

# MLE for Gaussian Discriminant Analysis

- Assume a $K$ class generative classification model with Gaussian class-conditionals

- Assume class $k = 1, 2, \ldots, K$ is modeled by a Gaussian with mean $\mu_k$ and cov matrix $\Sigma_k$

- Can assume label $y_n$ to be one-hot and then $y_{nk} = 1$ if $y_n = k$, and $y_{nk} = 0$, o/w

- Assuming class prior as $p(y_n = k) = \pi_k$, the model has params $\Theta = \{\pi_k, \mu_k, \Sigma_k\}_{k=1}^{K}$

- Given training data $\{\boldsymbol{x}_n, y_n\}_{n=1}^{N}$, the MLE solution will be

$$\hat{\pi}_k = \frac{1}{N} \sum_{n=1}^{N} y_{nk} \qquad \text{Same as } \frac{N_k}{N} \text{ where } N_k \text{ is \# of training ex. for which } y_n = k$$

$$\hat{\mu}_k = \frac{1}{N_k} \sum_{n=1}^{N} y_{nk} \boldsymbol{x}_n \qquad \text{Same as } \frac{1}{N_k} \sum_{n:y_n=k}^{N} \boldsymbol{x}_n$$

$$\hat{\Sigma}_k = \frac{1}{N_k} \sum_{n=1}^{N} y_{nk} (\boldsymbol{x}_n - \hat{\mu}_k)(\boldsymbol{x}_n - \hat{\mu}_k)^{\top} \qquad \text{Same as } \frac{1}{N_k} \sum_{n:y_n=k}^{N} (\boldsymbol{x}_n - \hat{\mu}_k)(\boldsymbol{x}_n - \hat{\mu}_k)^{\top}$$

See here for a derivation of the MLE for GDA

# Observations on the GDA objective function

- Here is a formal derivation of the MLE solution for $\Theta = \{\pi_k, \mu_k, \Sigma_k\}_{k=1}^{K}$

$$\widehat{\Theta} = \text{argmax}_\Theta \, p(\boldsymbol{X}, \boldsymbol{y}|\Theta) = \text{argmax}_\Theta \prod_{n=1}^{N} p(\boldsymbol{x}_n, y_n|\Theta)$$

multinoulli    Gaussian

$$= \text{argmax}_\Theta \prod_{n=1}^{N} p(y_n|\Theta) \, p(x_n|y_n, \Theta)$$

In general, in models with probability distributions from the **exponential family**, the MLE problem will usually have a simple analytic form

$$= \text{argmax}_\Theta \prod_{n=1}^{N} \prod_{k=1}^{K} \pi_k^{y_{nk}} \prod_{k=1}^{K} p(x_n|y_n = k, \Theta)^{y_{nk}}$$

Also, due to the form of the likelihood (Gaussian) and prior (multinoulli), the MLE problem had a nice separable structure after taking the log

$$= \text{argmax}_\Theta \prod_{n=1}^{N} \prod_{k=1}^{K} [\pi_k p(x_n|y_n = k, \Theta)]^{y_{nk}}$$

$$= \text{argmax}_\Theta \log \prod_{n=1}^{N} \prod_{k=1}^{K} [\pi_k p(x_n|y_n = k, \Theta)]^{y_{nk}}$$

Can see that, when estimating the parameters of the $k^{th}$ Gaussian ($\pi_k, \mu_k, \Sigma_k$), we only will only need training examples from the $k^{th}$ class, i.e., examples for which $y_{nk} = 1$

$$\text{argmax}_\Theta \sum_{n=1}^{N} \sum_{k=1}^{K} y_{nk} [\log \pi_k + \log \mathcal{N}(\boldsymbol{x}_n|\mu_k, \Sigma_k)]$$

**The form of this expression is important; will encounter this in GMM too**

# Need for EM/ALT-OPT: Two Equivalent Perspectives

1. Consider an LVM with latent variables and parameters. Trying to estimate parameters without also estimating the latent variables (by marginalizing them) is difficult

A Gaussian Mixture Model (GMM)

$$p(\boldsymbol{x}_n|\Theta) = \sum_{k=1}^{K} p(\boldsymbol{x}_n, \boldsymbol{z}_n = k|\Theta) = \sum_{k=1}^{K} p(\boldsymbol{z}_n = k|\phi)p(\boldsymbol{x}_n|\boldsymbol{z}_n = k, \theta) = \sum_{k=1}^{K} \pi_k \mathcal{N}(\boldsymbol{x}_n|\mu_k, \Sigma_k)$$

MLE for GMM with cluster ids marginalized/summed/integrated out

$$\Theta_{MLE} = \underset{\Theta}{\text{argmax}} \sum_{n=1}^{N} \log \sum_{k=1}^{K} \pi_k \mathcal{N}(\boldsymbol{x}_n|\mu_k, \Sigma_k)$$

Can't get closed form expressions for the $\pi_k, \mu_k, \Sigma_k$ due to "log of sum". Have to use gradient based methods

This issue not just for MLE for GMM but MLE for other LVMs too

EM/ALT-OPT will help us "simulate" this condition by making guesses about the values of $\boldsymbol{z}_n$'s

If we knew the $\boldsymbol{z}_n$'s, the problem will be much simpler; just like MLE for generative classification with Gaussian class-conditional

**Since no marginalization of the $\boldsymbol{z}_n$'s required**

2. Consider a complex prob. density (without any latent vars) for which MLE is hard

Directly defining a probability density as a mixture of Gaussians ($x_n$ is generated by the $k^{th}$ Gaussian with probability $\pi_k$) without any reference to any latent variable whatsoever (we didn't define it as an LVM)

$$p(\boldsymbol{x}_n|\Theta) = \sum_{k=1}^{K} \pi_k \mathcal{N}(\boldsymbol{x}_n|\mu_k, \Sigma_k)$$

MLE for the params Θ of this distribution will again be hard (as we already saw above). However, we can artificially introduce a latent variable $z_n$ with each data point $x_n$, denoting which Gaussian generated $x_n$

Can now apply ALT-OPT/EM to estimate parameters Θ + we get the latent variables $z_n$ as a "by-product" (though we may not be interested in learning $z_n$'s if our goal is just density estimation, not clustering)

Even though we didn't need the artificially introduced $z_n$'s, their presence and doing ALT-OPT/EM made our job of estimating Θ easier!

Also in any LVM, given Θ, you can always estimate $z_n$'s. Likewise, given $z_n$, you can always estimate Θ

# MLE for GMM

- Already saw that MLE is hard for GMM

$$\Theta_{MLE} = \underset{\Theta}{\operatorname{argmax}} \log p(\boldsymbol{X}|\Theta) = \underset{\Theta}{\operatorname{argmax}} \sum_{n=1}^{N} \log \sum_{k=1}^{K} \pi_k \mathcal{N}(\boldsymbol{x}_n|\mu_k, \Sigma_k)$$

- Two possible ways to solve this MLE problem

  *Will soon see how to get these guesses*

  1. If someone gave us optimal "point" guesses $\hat{z}_n$'s of cluster ids $z_n$'s, we could do MLE for the parameters just like we did for generative classification with Gaussian class-conditionals

  $$\Theta_{MLE} = \underset{\Theta}{\operatorname{argmax}} \log p(\boldsymbol{X}, \hat{\boldsymbol{Z}}|\Theta) = \operatorname{argmax}_{\Theta} \sum_{n=1}^{N} \sum_{k=1}^{K} \hat{z}_{nk}[\log \pi_k + \log \mathcal{N}(\boldsymbol{x}_n|\mu_k, \Sigma_k)]$$

  *In form of a probability distribution instead of a singe "optimal" guess*

  2. Alternatively, if someone gave a "probabilistic" guess of $z_n$'s, we can do MLE for $\Theta$ as follows

  $$\Theta_{MLE} = \underset{\Theta}{\operatorname{argmax}} \mathbb{E}[\log p(\boldsymbol{X}, \boldsymbol{Z}|\Theta)] = \operatorname{argmax}_{\Theta} \sum_{n=1}^{N} \sum_{k=1}^{K} \mathbb{E}[z_{nk}][\log \pi_k + \log \mathcal{N}(\boldsymbol{x}_n|\mu_k, \Sigma_k)]$$

  *Similar to Approach 1 but maximizes an expectation*

  *The expectation is w.r.t a distribution of Z which we will see shortly*

- Approach 1 is **ALT-OPT** and Approach 2 is **Expectation Maximization** ("soft" ALT-OPT). Both require alternating between estimating $\boldsymbol{Z}$ and $\Theta$ until convergence

# ALT-OPT for GMM

- We will assume we have a "hard" (most probable) guess of $z_n$, say $\hat{z}_n$

- Then ALT-OPT would look like this

  - Initialize $\Theta = \{\pi_k, \mu_k, \Sigma_k\}_{k=1}^{K}$ as $\widehat{\Theta}$

  - Repeat the following until convergence

    - For each $n$, compute most probable value (our best guess) of $z_n$ as

$$\hat{z}_n = \text{argmax}_{k=1,2,\ldots,K} \, p(z_n = k | \widehat{\Theta}, \mathbf{x}_n)$$

> Proportional to prior prob times likelihood, i.e., $p(z_n = k|\Theta) \, p(x_n|z_n = k, \Theta) = \pi_k \, \mathcal{N}(\mathbf{x}_n|\mu_k, \Sigma_k)$

> Posterior probability of point $x_n$ belonging to cluster $k$

    - Solve MLE problem for $\Theta$ using most probable $z_n$'s

$$\widehat{\Theta} = \text{argmax}_{\Theta} \, \sum_{n=1}^{N} \sum_{k=1}^{K} \hat{z}_{nk} [\log \pi_k + \log \mathcal{N}(\mathbf{x}_n|\mu_k, \Sigma_k)]$$

> Same objective function as generative $K$-class classification with Gaussian class-conditionals

> Note: The objective function is $\sum_{n=1}^{N} \log p(\mathbf{x}_n, \hat{\mathbf{z}}_n|\Theta) = \sum_{n=1}^{N} \log p(\hat{\mathbf{z}}_n|\Theta) + \log p(\mathbf{x}_n|\hat{\mathbf{z}}_n, \Theta)$

> $N_k$ : Effective number of points in cluster k

$$\hat{\pi}_k = \frac{1}{N} \sum_{n=1}^{N} \hat{z}_{nk} \qquad \hat{\mu}_k = \frac{1}{N_k} \sum_{n=1}^{N} \hat{z}_{nk} \mathbf{x}_n$$

$$\hat{\Sigma}_k = \frac{1}{N_k} \sum_{n=1}^{N} \hat{z}_{nk} (\mathbf{x}_n - \hat{\mu}_k)(\mathbf{x}_n - \hat{\mu}_k)^{\mathsf{T}}$$

> Does that matter? Should we worry that we aren't solving the actual problem anymore?

> Not really; will see the justification soon ☺

> But wait! This is not the same as $\sum_{n=1}^{N} \log p(\mathbf{x}_n|\Theta)$ which was the original MLE objective for this LVM ☹

# Expectation-Maximization (EM) for GMM

- EM finds $\Theta_{MLE}$ by maximizing $\mathbb{E}[\log p(\boldsymbol{X}, \boldsymbol{Z}|\Theta)]$ rather than $\log p(\boldsymbol{X}, \widehat{\boldsymbol{Z}}|\Theta)$

  Expectation of CLL

  .. which we maximized in ALT-OPT

- Note: Expectation will be w.r.t. the <u>conditional</u> posterior distribution of $\boldsymbol{Z}$, i.e., $p(\boldsymbol{Z}|\boldsymbol{X}, \Theta)$

  It is "conditional" posterior because it is also conditioned on Θ, not just data $X$

  Why w.r.t. this distribution? Will see justification in a bit

- The EM algorithm for GMM operates as follows

  Requires knowing Θ

  - Initialize $\Theta = \{\pi_k, \mu_k, \Sigma_k\}_{k=1}^K$

    Needed to get the expected CLL

  - Repeat until convergence

    - Compute conditional posterior $p(\boldsymbol{Z}|\boldsymbol{X}, \widehat{\Theta})$. Since obs are i.i.d, compute separately for each $n$ (and for $k =$

      Same as $p(z_{nk} = 1| \boldsymbol{x}_n, \widehat{\Theta})$, just a different notation

      $$p(\boldsymbol{z}_n = k|\boldsymbol{x}_n, \widehat{\Theta}) \propto p(\boldsymbol{z}_n = k|\widehat{\Theta}) \, p(\boldsymbol{x}_n|\boldsymbol{z}_n = k, \widehat{\Theta}) = \hat{\pi}_k \mathcal{N}(x_n|\hat{\mu}_k, \hat{\Sigma}_k)$$

      $$\widehat{\Theta} = \operatorname{argmax}_\Theta \mathbb{E}_{p(\boldsymbol{Z}|\boldsymbol{X}, \widehat{\Theta})}[\log p(\boldsymbol{X}, \boldsymbol{Z}|\Theta)] = \sum_{n=1}^N \mathbb{E}_{p(\boldsymbol{z}_n|\boldsymbol{x}_n, \widehat{\Theta})}[\log p(\boldsymbol{x}_n, \boldsymbol{z}_n|\Theta)]$$

      Solution has a similar form as ALT-OPT (or gen. class.), except we now have the **expectation** of $z_{nk}$ being used

    by maximizing the expected complete data log-likelihood

$$\hat{\pi}_k = \frac{1}{N} \sum_{n=1}^N \mathbb{E}[z_{nk}] \quad \hat{\mu}_k = \frac{1}{N_k} \sum_{n=1}^N \mathbb{E}[z_{nk}] \boldsymbol{x}_n$$

$N_k$ : Effective number of points in cluster k

$$\hat{\Sigma}_k = \frac{1}{N_k} \sum_{n=1}^N \mathbb{E}[z_{nk}](\boldsymbol{x}_n - \hat{\mu}_k)(\boldsymbol{x}_n - \hat{\mu}_k)^\top$$

$$= \operatorname{argmax}_\Theta \mathbb{E}\left[\sum_{n=1}^N \sum_{k=1}^K z_{nk}[\log \pi_k + \log \mathcal{N}(\boldsymbol{x}_n|\mu_k, \Sigma_k)]\right]$$

$$= \operatorname{argmax}_\Theta \sum_{n=1}^N \sum_{k=1}^K \mathbb{E}[z_{nk}][\log \pi_k + \log \mathcal{N}(\boldsymbol{x}_n|\mu_k, \Sigma_k)]$$

# EM for GMM (Contd)

- The EM algo for GMM required $\mathbb{E}[z_{nk}]$. Note $z_{nk} \in \{0,1\}$

Reason: $\sum_{k=1}^{K} \gamma_{nk} = 1$

Need to normalize: $\mathbb{E}[z_{nk}] = \frac{\hat{\pi}_k \mathcal{N}(x_n | \hat{\mu}_k, \hat{\Sigma}_k)}{\sum_{\ell=1}^{K} \hat{\pi}_\ell \mathcal{N}(x_n | \hat{\mu}_\ell, \hat{\Sigma}_\ell)}$

$$\mathbb{E}[z_{nk}] = \gamma_{nk} = 0 \times p(z_{nk} = 0 | x_n, \widehat{\Theta}) + 1 \times p(z_{nk} = 1 | x_n, \widehat{\Theta}) = p(z_{nk} = 1 | x_n, \widehat{\Theta}) \propto \hat{\pi}_k \mathcal{N}(x_n | \hat{\mu}_k, \hat{\Sigma}_k)$$

## EM for Gaussian Mixture Model

**①** Initialize $\Theta = \{\pi_k, \mu_k, \Sigma_k\}_{k=1}^{K}$ as $\Theta^{(0)}$, set $t = 1$

**②** E step: compute the expectation of each $\boldsymbol{z}_n$ (we need it in M step)

Accounts for fraction of points in each cluster

Accounts for cluster shapes (since each cluster is a Gaussian

Soft K-means, which are more of a heuristic to get soft-clustering, also gave us probabilities but didn't account for cluster shapes or fraction of points in each cluster

$$\mathbb{E}[z_{nk}^{(t)}] = \gamma_{nk}^{(t)} = \frac{\pi_k^{(t-1)} \mathcal{N}(\boldsymbol{x}_n | \boldsymbol{\mu}_k^{(t-1)}, \boldsymbol{\Sigma}_k^{(t-1)})}{\sum_{\ell=1}^{K} \pi_\ell^{(t-1)} \mathcal{N}(\boldsymbol{x}_n | \boldsymbol{\mu}_\ell^{(t-1)}, \boldsymbol{\Sigma}_\ell^{(t-1)})} \quad \forall n, k$$

**③** Given "responsibilities" $\gamma_{nk} = \mathbb{E}[z_{nk}]$, and $N_k = \sum_{n=1}^{N} \gamma_{nk}$, re-estimate $\Theta$ via MLE

Effective number of points in the $k^{th}$ cluster

M-step:

$$\boldsymbol{\mu}_k^{(t)} = \frac{1}{N_k} \sum_{n=1}^{N} \gamma_{nk}^{(t)} \boldsymbol{x}_n$$

$$\boldsymbol{\Sigma}_k^{(t)} = \frac{1}{N_k} \sum_{n=1}^{N} \gamma_{nk}^{(t)} (\boldsymbol{x}_n - \boldsymbol{\mu}_k^{(t)})(\boldsymbol{x}_n - \boldsymbol{\mu}_k^{(t)})^\top$$

$$\pi_k^{(t)} = \frac{N_k}{N}$$

**④** Set $t = t + 1$ and go to step 2 if not yet converged

# EM for GMM in action
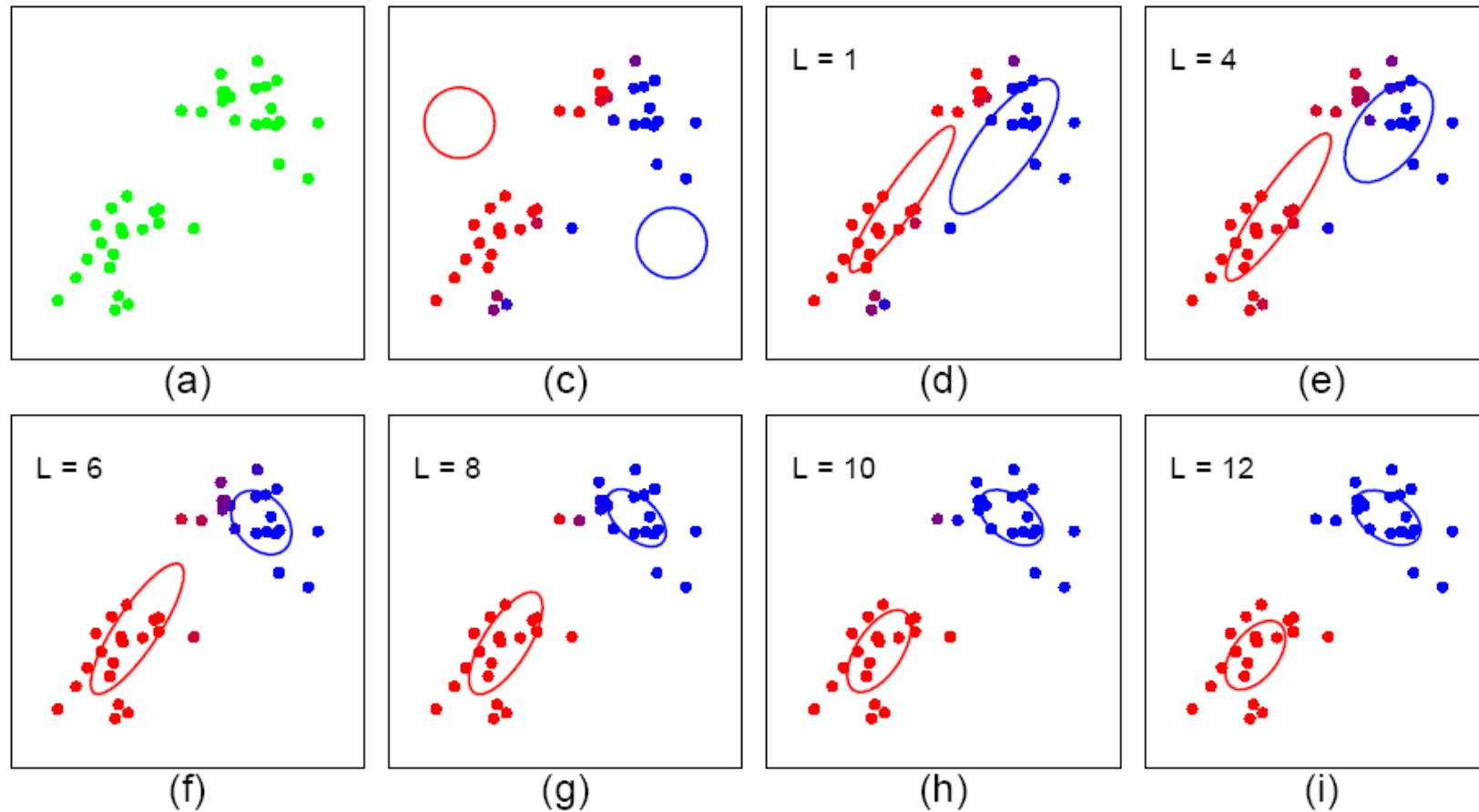


Note: Just like with k-means, cluster initialization matters. EM only finds local optima.

# References

- Multi Modal Machine Learning, 11-777 • Fall 2023 • Carnegie Mellon University