

NPTEL

NPTEL ONLINE CERTIFICATION COURSE

Introduction to Machine Learning

Lecture-78

Expectation Maximization Continued

Prof: Balaraman Ravindran
Computer Science and Engineering
Indian Institute of Technology Madras

So we started with defining Gaussian mixture models.

(Refer Slide Time: 00:19)

Mixture Models

- Superpositions or linear combinations of simple distributions (density: $p(x_n) = \sum_{k=1}^K \pi_k p(x_n|\theta_k)$)
- Example, mixture of Gaussians; density:

$$p(x_n) = \sum_{k=1}^K \pi_k \mathcal{N}(x_n|\mu_k, \Sigma_k)$$
$$\mathcal{N}(x|\mu, \Sigma) = \frac{1}{(2\pi)^{p/2}} \frac{1}{|\Sigma|^{1/2}} \exp\left\{-\frac{1}{2}(x - \mu)^T \Sigma^{-1}(x - \mu)\right\}$$

- Each Gaussian \mathcal{N} is a *component* of the mixture with its own mean μ_k and covariance Σ_k ($\theta_k = \{\mu_k, \Sigma_k\}$)
- For $p(x_n)$ to be a valid density, we need:

$$\sum_{k=1}^K \pi_k = 1, \quad 0 \leq \pi_k \leq 1$$

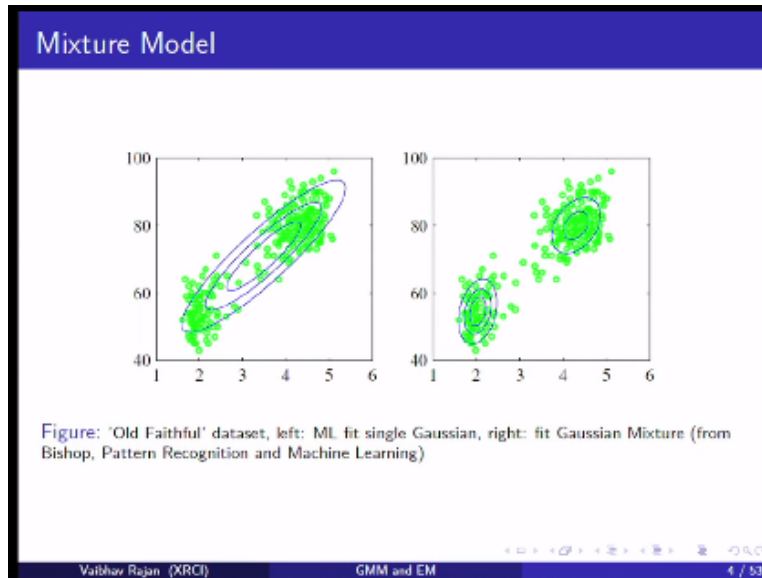
π_k : mixing coefficients

Navigation icons: back, forward, search, etc.

Footer: Vaibhav Rajan (XRCI) GMM and EM 3 / 53

Which are just one second super position of A different Gaussians and in a general mixture model instead of a Gaussian you can use any other probability distribution, so the three important setoff parameter are the mixture weights the mean and the covariance matrices of each of the Gaussians and there are K components I have still not come back to sum and ask how do we estimate K we will see that today.

(Refer Slide Time: 00:56)



We saw some examples of how to fit Gaussian mixture models we saw that Gaussian mixture models are good models when there is naturally good models when there is a cluster structure in a data so that each of those clusters can be nicely fitted with the Gaussian.

(Refer Slide Time: 01:15)

Generative Model

- Sample $z_n \sim \text{Mult}_K(1, \pi)$: k^{th} component with parameters θ_k
- Sample $x_n \sim p(x_n | \theta_k)$


Figure: Graphical Representation of Mixture Model. Circles: random variables (observed – shaded, latent – unshaded).

Valbhav Rajan (XRCI) GMM and EM 7 / 53

Then we saw that the Gaussian mixture model can be very intuitively explain through the generative procedure where you assume that there is a latent variable that basically tells you which Gaussian to pick and once you pick that you sample your or you generate your data from that particular Gaussian, okay. I think this is very important to remember as it makes lot of the math make sense of lot of math.

(Refer Slide Time: 01:50)

- $p(x_n) = \sum_{k=1}^K p(z_n = k)p(x_n|z_n = k) = \sum_{z_n} p(x_n, z_n)$
- $p(z_n = k)$: *Prior probability* of datapoint x_n from component k
- $p(z_n = k|x_n)$: *Posterior probability* of datapoint x_n from component k
- $\gamma(z_{nk}) = p(z_n = k|x_n)$: *Responsibility* of component k for x_n
- $\gamma(z_{nk}) = p(z_n = k|x_n) = \frac{p(z_n=k)p(x_n|z_n=k)}{\sum_{j=1}^K p(z_n=j)p(x_n|z_n=j)} = \frac{\pi_k p(x_n|\theta_k)}{\sum_{j=1}^K \pi_j p(x_n|\theta_j)}$
- $\gamma(z_{nk}) = \frac{\pi_k p(x_n|\theta_k)}{\sum_{j=1}^K \pi_j p(x_n|\theta_j)}$



Variklav Rajan (XRCI) GMM and EM 8 / 53

Then we saw our posterior probability which are also called responsibility the posterior probability of for the latent variable taking a value K given the data and we saw it coming up repeatedly in all our calculation.

(Refer Slide Time: 02:11)

Parameter Estimation

- For a GMM with k components, on p -dimensional data, parameters $\vartheta = \{\pi_k, \mu_k, \Sigma_k\}$ to estimate:
 - k mixing coefficients
 - k p -dimensional mean vectors
 - k $(p \times p)$ -dimensional covariance matrices

- Likelihood of N data points drawn independently

$$p(X|\vartheta) = \prod_{n=1}^N \left(\sum_{k=1}^K \pi_k \mathcal{N}(x_n | \mu_k, \Sigma_k) \right)$$

- Log Likelihood:

$$\log p(X|\vartheta) = \sum_{n=1}^N \log \left(\sum_{k=1}^K \pi_k \mathcal{N}(x_n | \mu_k, \Sigma_k) \right)$$

- $\theta_{ML} = \operatorname{argmax}_{\vartheta} \{\log p(X|\vartheta)\}$, $\theta_{MAP} = \operatorname{argmax}_{\vartheta} \{\log p(X|\vartheta) + \log p(\vartheta)\}$
- Summation ($\sum_{k=1}^K$) inside the logarithm: makes ML/MAP estimate difficult, no closed form solution


So the estimation assuming that we know k on P dimensional data the estimation problem is to estimate these π_k , μ_k , and σ_k for each of the Gaussians right.

(Refer Slide Time: 02:23)

Parameter Estimation

- Log Likelihood: $l = \log p(X|\theta) = \sum_{n=1}^N \log \left(\sum_{k=1}^K \pi_k \mathcal{N}(x_n | \mu_k, \Sigma_k) \right)$
- $\frac{\partial l}{\partial \mu_k} = \sum_{n=1}^N \frac{\pi_k \mathcal{N}(x_n | \mu_k, \Sigma_k)}{\sum_{j=1}^K \pi_j \mathcal{N}(x_n | \mu_j, \Sigma_j)} \Sigma^{-1} (x_n - \mu_k) = \sum_{n=1}^N \gamma(z_{nk}) \Sigma^{-1} (x_n - \mu_k)$
($\frac{d \log x}{dx} = \frac{1}{x}$ for $x > 0$, $\frac{\partial}{\partial s} (x-s)^T W (x-s) = -2W(x-s)$ for symmetric W)
- Setting $\frac{\partial l}{\partial \mu_k} = 0$, multiplying by Σ_k .

$$\mu_k = \frac{\sum_{n=1}^N \gamma(z_{nk}) x_n}{\sum_{n=1}^N \gamma(z_{nk})}$$
- Weighted mean of all data points, weight: responsibility (posterior probability of latent variable)



Vaibhav Rajan (XRC) GMM and EM 11 / 53

And we saw that initially we first saw that if we assume that we know the responsibility then the math works out very nicely and we get very intuitive forms for the different parameters.

(Refer Slide Time: 02:49)

Iterative Algorithm

- Initialize $\vartheta = \{\pi_k, \mu_k, \Sigma_k\}$
- Compute log-likelihood

$$l = \log p(X|\vartheta) = \sum_{n=1}^N \log \left(\sum_{k=1}^K \pi_k \mathcal{N}(x_n | \mu_k, \Sigma_k) \right)$$
- Repeat until convergence:
 - Set responsibility: $\gamma(z_{nk}) = \frac{\pi_k p(x_n | \theta_k)}{\sum_{j=1}^K \pi_j p(x_n | \theta_j)}$
 - Update parameters:
 - $\mu_k = \frac{\sum_{n=1}^N \gamma(z_{nk}) x_n}{\sum_{n=1}^N \gamma(z_{nk})}$
 - $\Sigma_k = \frac{\sum_{n=1}^N \gamma(z_{nk}) (x_n - \mu_k)(x_n - \mu_k)^T}{\sum_{n=1}^N \gamma(z_{nk})}$
 - $\pi_k = \frac{\sum_{n=1}^N \gamma(z_{nk})}{N}$
 - Recompute log-likelihood l

⏪ ⏩ ⏴ ⏵ ⏶ ⏷ ⏸ ⏹ ⏺

Vartharajan (XRCI)
GMM and EM
15 / 53

And then we design an iterative algorithm that essentially guesses the parameter first and compute the responsibilities and then refines the guess in each iteration and later we saw that actually is the EM algorithm for Gaussian mixture models, right.

(Refer Slide Time: 03:09)

Expectation Maximization (EM)

- ML Estimation with Missing Data
 - X : Observed/Incomplete Data
 - Z : Hidden data (assume discrete)
 - $\{X, Z\}$: Complete data
 - Assume parameterized family: $p(X, Z|\theta)$, unknown parameters θ
 - Aim: Estimate $\operatorname{argmax}_{\theta} \log p(X|\theta)$
- $\log p(X|\theta) = \log \sum_Z p(X, Z|\theta)$ (\sum_Z inside log)
- E.g. Exponential $p(X, Z|\theta) \Rightarrow$ Exponential marginal $p(X|\theta)$
- Assume maximizing joint likelihood $\log p(X, Z|\theta)$ is easy

⏪ ⏩ 🔍 🔄
 Vaibhav Rajan (XRCI) GMM and EM 17 / 33

So let us see this more carefully in general EM had been proposed for data which had some hidden data points not known when you get the data set and so z is we denote that hidden data by z and for the purpose of this discussion we assumed this discrete and data we saw that in case of the Gaussian mixture models we can take latent variables to be hidden that is the common trick used in many other models.

And then we saw that EM is a good approach to take when the joint likelihood the complete data likelihood can be easily parameterized and the if that is a if you make this assumption then we see that we can get the imagine likelihood also.

(Refer Slide Time: 04:02)

Expectation Maximization (EM)

- Initialize $\vartheta^{(0)}$, Evaluate $f^{(0)} = \log p(X|\vartheta^{(0)})$
- For $m = 1, \dots, T$
 - Posterior distribution of Z : $p(Z|X, \vartheta^{(m-1)})$
 - Expected Complete Likelihood under this distribution of Z :

$$\begin{aligned} Q(\vartheta, \vartheta^{(m-1)}) &= \sum_Z \underbrace{p(Z|X, \vartheta^{(m-1)})}_{\substack{\text{distribution of } Z \\ \text{assuming } \vartheta^{(m-1)}}} \underbrace{\log p(X, Z|\vartheta)}_{\substack{\text{complete data likelihood} \\ \text{unknown } \vartheta}} \\ &= \mathbb{E}_{Z|X, \vartheta^{(m-1)}} \log p(X, Z|\vartheta) \end{aligned}$$

- $\vartheta^{(m)} = \operatorname{argmax}_{\vartheta} Q(\vartheta, \vartheta^{(m-1)})$
- Check for convergence: stop if $f^{(m)} - f^{(m-1)} < \epsilon$

So what is EM, the key idea is that so this is the key idea we take the expectation of the log likelihood of the complete data under the distribution of latent variables assuming the guesses of the parameter that we had made, right.

(Refer Slide Time: 04:27)

Expectation Maximization (EM)

- $\operatorname{argmax}_{\theta} \log p(X|\theta)$
- $\operatorname{argmax}_{\theta} \log \sum_Z p(X, Z|\theta)$ summation inside log
- $\operatorname{argmax}_{\theta} \mathbb{E}_{Z|X, \theta} \log p(X, Z|\theta)$ we don't know $p(Z|X, \theta)$
- $\operatorname{argmax}_{\theta} \mathbb{E}_{Z|X, \theta^{(m-1)}} \log p(X, Z|\theta)$ guess and iterate: works!

$$\theta^{(m)} = \operatorname{argmax}_{\theta} \mathbb{E}_{Z|X, \theta^{(m-1)}} \log p(X, Z|\theta)$$

$$\text{E Step } Q(\theta, \theta^{(m-1)}) = \mathbb{E}_{Z|X, \theta^{(m-1)}} \log p(X, Z|\theta)$$

$$\text{M Step } \theta^{(m)} = \operatorname{argmax}_{\theta} Q(\theta, \theta^{(m-1)})$$

And instead of computing the maximum likelihood we compute the parameters that maximizes this expectation and this is the key idea of EM, right. + Actually if you remember this I mean this should be the main take away of the class this from now.

(Refer Slide Time: 04:49)

EM for GMM

- Gaussian Mixture Model:
$$p(x_n) = \sum_{k=1}^K \pi_k \mathcal{N}(x_n | \mu_k, \Sigma_k) = \sum_{k=1}^K p(z_n = k) p(x_n | z_n = k)$$
- Parameters $\vartheta = \{\pi_k, \mu_k, \Sigma_k\}$:
 - k mixing coefficients
 - k p -dimensional mean vectors
 - k $(p \times p)$ -dimensional covariance matrices
- $\vartheta_{ML} = \operatorname{argmax}_{\vartheta} \{\log p(X | \vartheta)\}$
- Hidden Variables = Latent Variables

◀ ▶ ↻ 🔍

So then we saw that if we use this formulation then for Gaussian mixture models we essentially get back the iterative algorithm that we are guessed.

(Refer Slide Time: 05:04)

E Step

$$\begin{aligned}
 Q(\vartheta, \vartheta^{(m-1)}) &= \mathbb{E}_{Z|X, \vartheta^{(m-1)}} \log p(X, Z | \vartheta) \\
 &= \mathbb{E}_{Z|X, \vartheta^{(m-1)}} \left[\sum_{n=1}^N \log p(x_n, z_n | \vartheta) \right] \\
 &= \sum_{n=1}^N \mathbb{E}_{Z|X, \vartheta^{(m-1)}} \left[\log \prod_{k=1}^K (\pi_k p(x_n | \theta_k))^{I(z_n=k)} \right] \\
 &= \sum_{n=1}^N \sum_{k=1}^K \mathbb{E}_{Z|X, \vartheta^{(m-1)}} [I(z_n = k)] \log (\pi_k p(x_n | \theta_k)) \\
 &= \sum_{n=1}^N \sum_{k=1}^K p(z_n = k | X, \vartheta^{(m-1)}) \log (\pi_k p(x_n | \theta_k)) \\
 &= \sum_{n=1}^N \sum_{k=1}^K \gamma(z_{nk})_{|\vartheta^{(m-1)}} \log (\pi_k p(x_n | \theta_k)) \\
 &= \sum_{n=1}^N \sum_{k=1}^K \gamma(z_{nk})_{|\vartheta^{(m-1)}} \log \pi_k + \gamma(z_{nk})_{|\vartheta^{(m-1)}} \log p(x_n | \theta_k)
 \end{aligned}$$

We used this so this expectation is also called Q function in the literature we get a very nice form for the Q function the reason so reason we get a nice form is one because we using an expectation operator which pushes the summation to the outside and the second reason is that we get so we get the logarithm of the Gaussian without any summation inside this is the expectation position outside, right this was the reason why the math's worked out.

(Refer Slide Time: 05: 37)

M Step

$$\begin{aligned} \frac{\partial Q}{\partial \Sigma_k} &= \frac{\partial}{\partial \Sigma_k} \left\{ \sum_{n=1}^N \sum_{k=1}^K \gamma(z_{nk})_{\theta^{(n-1)}} \log \pi_k + \gamma(z_{nk})_{\theta^{(n-1)}} \log p(x_n | \theta_k) \right\} \\ &= \frac{\partial}{\partial \Sigma_k} \left\{ \sum_{n=1}^N \gamma(z_{nk})_{\theta^{(n-1)}} \log \left[\frac{1}{(2\pi_k)^{p/2}} \frac{1}{|\Sigma_k|^{1/2}} \exp\left\{-\frac{1}{2}(x_n - \mu_k)^T \Sigma_k^{-1} (x_n - \mu_k)\right\} \right] \right\} \\ &= \frac{\partial}{\partial \Sigma_k} \left\{ \sum_{n=1}^N \gamma(z_{nk})_{\theta^{(n-1)}} \left[\log \frac{1}{(2\pi_k)^{p/2}} - \frac{1}{2} \log |\Sigma_k| - \frac{1}{2} (x_n - \mu_k)^T \Sigma_k^{-1} (x_n - \mu_k) \right] \right\} \end{aligned}$$

(Use $\frac{\partial |X|}{\partial X} = |X|(X^T)^{-1}$, $\frac{\partial}{\partial X}(a^T X^{-1} b) = -(X^T)^{-1} a b^T (X^T)^{-1}$ to simplify and set $\frac{\partial Q}{\partial \Sigma_k} = 0$ to get

$$\Sigma_k = \frac{\sum_{n=1}^N \gamma(z_{nk})_{\theta^{(n-1)}} (x_n - \mu_k)(x_n - \mu_k)^T}{\sum_{n=1}^N \gamma(z_{nk})_{\theta^{(n-1)}}} \quad k = 1, \dots, K$$

And the derivatives become easy to calculate for the case of Gaussian, right.

(Refer Slide Time: 05:46)

M Step: Summary

- $\mu_k^{(m)} = \frac{\sum_{n=1}^N \gamma(z_{nk})_{|\phi^{(m-1)}} x_n}{\sum_{n=1}^N \gamma(z_{nk})_{|\phi^{(m-1)}}$
- $\Sigma_k^{(m)} = \frac{\sum_{n=1}^N \gamma(z_{nk})_{|\phi^{(m-1)}} (x_n - \mu_k)(x_n - \mu_k)^T}{\sum_{n=1}^N \gamma(z_{nk})_{|\phi^{(m-1)}}$
- $\pi_k^{(m)} = \frac{\sum_{n=1}^N \gamma(z_{nk})_{|\phi^{(m-1)}}}{N}, \quad k = 1, \dots, K$

We essentially got back the same formulas for μ_k , σ_k and Π_k that we had guessed earlier summing we know the responsibilities, right.

(Refer Slide Time: 06:00)

Expectation Maximization (EM)

- Initialize $\vartheta^{(0)}$, Evaluate $l^{(0)} = \log p(X|\vartheta^{(0)})$
- For $m = 1, \dots, T$
 - Posterior distribution of Z : $p(Z|X, \vartheta^{(m-1)})$
 - Expected Complete Likelihood under this distribution of Z :
 $Q(\vartheta, \vartheta^{(m-1)}) = \mathbb{E}_{Z|X, \vartheta^{(m-1)}} \log p(X, Z|\vartheta)$
 - $\vartheta^{(m)} = \operatorname{argmax}_{\vartheta} Q(\vartheta, \vartheta^{(m-1)})$
 - Check for convergence: stop if $l^{(m)} - l^{(m-1)} < \epsilon$

So general EEM algorithm is this guess the posterior distribution of the hidden data or the latent variables and then refine your guess by taking by maximizing the Q function which is the expectation of the complete data likelihood under the distribution of z with your current guess and today we are going to see that this procedure is nice because it guarantees that the likelihood will increase in every iteration.

So whatever likelihood you start with at every iteration the likelihood is going to increase so that is what we are going to show today. This is the complete EM algorithm for estimating the parameter of the Gaussian mixture.

(Refer Slide Time: 06:50)

Special Case

- Assume a GMM, where covariance of each component $\epsilon \mathbf{I}$, fixed constant ϵ (spherical) and $\pi_k = 1/K$
- Parameter to estimate: μ_k

$$p(x_n | \theta_k) = \frac{1}{(2\pi)^{p/2}} \frac{1}{|\Sigma_k|^{1/2}} \exp\left\{-\frac{1}{2}(x - \mu)^T \Sigma_k^{-1}(x - \mu)\right\}$$

$$= \frac{1}{(2\pi\epsilon)^{p/2}} \exp\left\{-\frac{1}{2\epsilon} \|x - \mu_k\|^2\right\}$$

- $\gamma(z_{nk}) = \frac{\pi_k p(x_n | \theta_k)}{\sum_{j=1}^K \pi_j p(x_n | \theta_j)} = \frac{\pi_k \exp\{-\|x_n - \mu_k\|^2 / 2\epsilon\}}{\sum_{j=1}^K \pi_j \exp\{-\|x_n - \mu_j\|^2 / 2\epsilon\}}$
- $\epsilon \rightarrow 0$, term for which $\|x_n - \mu_j\|^2$ is smallest will go to 0 most slowly
 $\implies \gamma(z_{nj}) \rightarrow 1$ and $\gamma(z_{nk}) \rightarrow 0, k \neq j$
- $\gamma(z_{ni}) = \begin{cases} 1 & \text{if } k = \operatorname{argmin}_j \|x_n - \mu_j\|^2 \\ 0 & \text{otherwise} \end{cases}$

And we also saw that if we take if we assume that the only parameter to be determine is μ_k which means all we assume that all the Gaussians are spherical with known covariance matrices and $\pi_k = 1/k$ then essentially what we get back is the K means algorithm, okay.

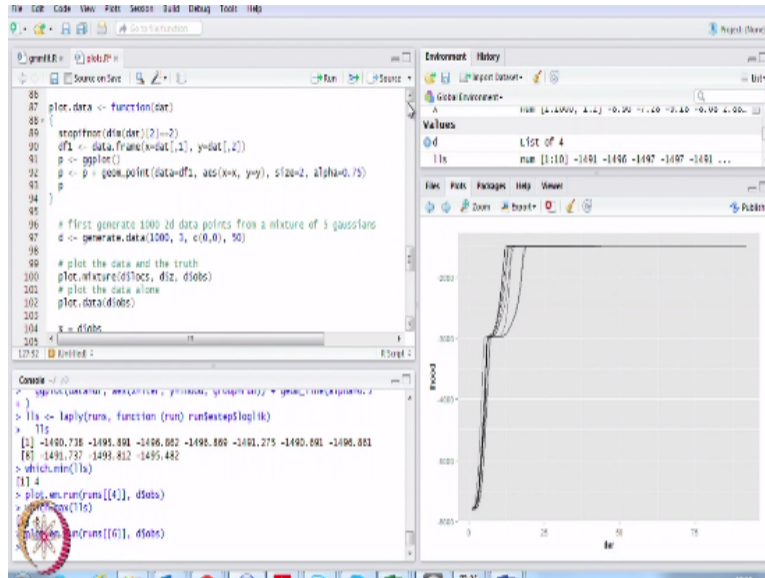
(Refer Slide Time: 07:22)

Theoretical Guarantee

- EM monotonically increases observed data likelihood
- Until local maximum (or saddle point)

We can see this so this is the theoretical guarantee I was talking about EM monotonically increases the observed data likelihood and until it reaches some local maximum it can also get stuck in some saddle points but it yeah.

(Refer Slide Time: 07:46)

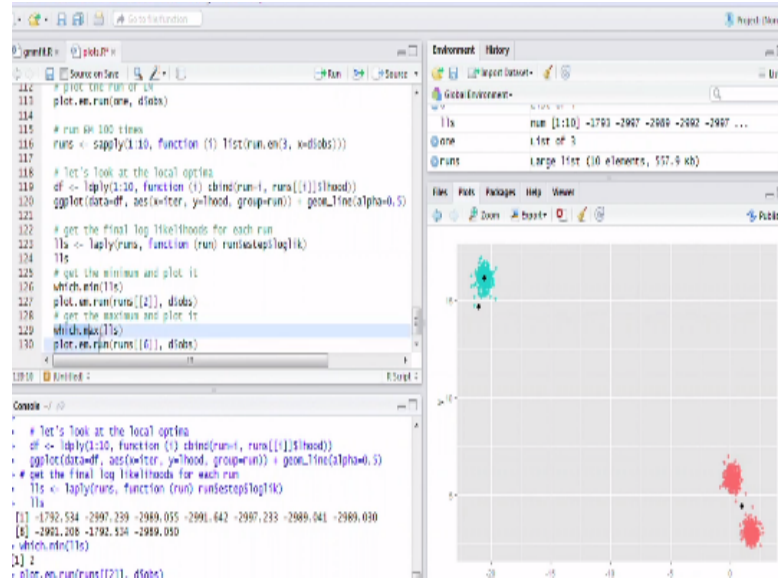


So it does not give you the global maximum it only gives you a it only reaches takes you to the local maximum, so let me show you that simulation that I have shown you last time, so I generate some data 3 Gaussians this is what the data looks like it was generated like this by taking these 3 means and `covariance matrices this is what the fitted density looks like, if I run EM once, right.

So this time EM did not do well you can see what happen the means that is inferred where two of them are here and the third one is here because these two clusters are very close together it assumed that it is coming it been generated from the same Gaussian, right. Let us now done this so I ran this 10 times and I what I see is that the likelihood for each of this run the likelihood keeps increasing.

Every time for each iteration the likelihood increases sometimes it get stuck and at a saddle point or fixed point and then it does not increase, so this is a typical behavior of EM right this is a very good debugging tool if you are writing EM algorithms for your models and if you see that the likelihood is not increasing there is some debug in your program.

(Refer Slide Time: 09:36)



Now let us see in these 10 runs these are the likelihood values it got stuck at, at the end I stopped at the end not necessarily stuck at so if we see the minimum of these this is the second one and we see the fitted density when we use that run yeah see the fit is not very good. If you take the maximum, maximum likelihood among those 10 runs. So 9th one the fit is much better now, you see this time when the likelihood was in the 9th run the likelihood was the highest among these 10 runs and the fit was also much better.

(Refer Slide Time: 10:36)

Background

- Jensen's Inequality. f : convex function on interval I , for $x_1, \dots, x_n \in I$, $\lambda_1, \dots, \lambda_n \geq 0$ with $\sum_{i=1}^n \lambda_i = 1$,

$$f\left(\sum_{i=1}^n \lambda_i x_i\right) \leq \sum_{i=1}^n \lambda_i f(x_i)$$

- f convex $\implies -f$ concave ($-\log x$: convex)
- $\log\left(\sum_{i=1}^n \lambda_i x_i\right) \geq \sum_{i=1}^n \lambda_i \log x_i$

So the main results we will lead to prove this is Jensen's inequality are you familiar with this? No okay this is very simple, so if you have a convex function and you have linear combination of these points, then the convex function applied to the linear combination is \sum is applied to each of the x . and now what we are interested in you might have guessed, is the algorithm function because log is what appears.

And if you use the fact $-\log x$ is convex and put it here then we get this inequality. The function is just logarithm, so what we see is that the log of \sum is \geq those $\gamma_8 \log x_i$.

(Refer Slide Time: 11:59)

Monotonicity of EM

- $q(z_n)$: arbitrary distribution over the latent variables
- $q(z_n) \geq 0$ with $\sum_{z_n} q(z_n) = 1$

$$\begin{aligned}
 \log p(X|\theta) &= \sum_{n=1}^N \log \left[\sum_{z_n} p(x_n, z_n|\theta) \right] \\
 &= \sum_{n=1}^N \log \left[\sum_{z_n} q(z_n) \frac{p(x_n, z_n|\theta)}{q(z_n)} \right] \\
 &\geq \sum_{n=1}^N \sum_{z_n} q(z_n) \log \left[\frac{p(x_n, z_n|\theta)}{q(z_n)} \right] \\
 &= \sum_{n=1}^N \underbrace{\sum_{z_n} q(z_n) \log p(x_n, z_n|\theta)}_{E_q[\log p(x_n, z_n|\theta)]} - \underbrace{\sum_{z_n} q(z_n) \log q(z_n)}_{\text{entropy}[q]} = Q(\theta, q)
 \end{aligned}$$

- $\log p(X|\theta) \geq Q(\theta, q)$
- Which distribution q should we choose?

So let us start with, so we have these latent variables or the hidden variables in some cases, let assume that cube is some arbitrate distribution over the latent variables, we will not define what q is right now. So because these are probability values each of these Q_{zn} for each latent is > 0 and this sum over all Z_n is $= 1$. So now let take the lightly hood of the data, and we express it again as usual in terms of joint lightly hood.

With respect to latent variables and then we just multiply and divide by q of z_n . Now because of this condition $Q(z_n)$ is same as γ it follows the assumptions of Jensen in equality. So we can Jensen in equality here and get a lower bound on this expression right basically take the \sum outside and get the log inside and this lower bound as follows from Jensen inequality right all we have done is applied this in equality.

(Refer Slide Time: 13:22)

Monotonicity of EM

- $q(z_n)$: arbitrary distribution over the latent variables
- $q(z_n) \geq 0$ with $\sum_{z_n} q(z_n) = 1$

$$\begin{aligned} \log p(X|\vartheta) &= \sum_{n=1}^N \log \left[\sum_{z_n} p(x_n, z_n|\vartheta) \right] \\ &= \sum_{n=1}^N \log \left[\sum_{z_n} q(z_n) \frac{p(x_n, z_n|\vartheta)}{q(z_n)} \right] \\ &\geq \sum_{n=1}^N \sum_{z_n} q(z_n) \log \left[\frac{p(x_n, z_n|\vartheta)}{q(z_n)} \right] \\ &= \sum_{n=1}^N \sum_{z_n} \underbrace{q(z_n) \log p(x_n, z_n|\vartheta)}_{\mathbb{E}_q[\log p(x_n, z_n|\vartheta)]} - \underbrace{q(z_n) \log q(z_n)}_{\text{entropy}_q} = Q(\vartheta, q) \end{aligned}$$

- $\log p(X|\vartheta) \geq Q(\vartheta, q)$
- Which distribution q should we choose?

NPTEL Vaibhav Rajan (XRCI) GMM and EM 39 / 53

And λ_i are the qz because they are probability is the assumptions at true okay so now this logarithm can be written as a difference of the log of the numerator- log of the denominator and what we get here of this expression should start looking familiar to you this is just an expectation is expectation of the complete data likelihood under the distribution q right.

So this is something that we, we have been working with in EM and on this side we have an entropy right this so this entropy term is not going to be not going to play a big role here but we are going to be interested in this so let us call this q this is although it will be the same q eventually I have used different here because right now we do not know the risk same thing okay.

So what have we got we have got a lower bound on the log likelihood right and we have proved this for any arbitrary distribution right we have not said that it is the distribution of the latent variables under the guesses of the parameters that we had we did not say anything about that so now the question is which distribution q should be chose any guesses so what we have is a lower bound what kind of distribution would you like to choose no guesses think iteratively alright.

So we since it is the lower bound we want the bound to be as tight as possible okay so we will choose the q such that we want to maximize such that we maximize the lower bound to reach the actual likelihood right so that is the natural choice when you are dealing with bounds right.


(Refer Slide Time: 15:36)

Monotonicity of EM

- Maximize the lower bound to reach the actual likelihood

$$\begin{aligned}
 L(\theta, q) &= \sum_{z_n} q(z_n) \log \left[\frac{p(x_n, z_n | \theta)}{q(z_n)} \right] \\
 &= \sum_{z_n} q(z_n) \log \left[\frac{p(z_n | x_n, \theta) p(x_n | \theta)}{q(z_n)} \right] \\
 &= \underbrace{\sum_{z_n} q(z_n) \log \left[\frac{p(z_n | x_n, \theta)}{q(z_n)} \right]}_{=-K(q(z_n) || p(z_n | x_n, \theta))} + \underbrace{\sum_{z_n} q(z_n) \log [p(x_n | \theta)]}_{=-\log p(x_n | \theta) \text{ independent of } q}
 \end{aligned}$$

- $q(z_n) = p(z_n | x_n, \theta) \implies q(z_n) \log \left[\frac{p(z_n | x_n, \theta)}{q(z_n)} \right] = 0$
- But real θ is unknown, lets use $q^m(z_n) = p(z_n | x_n, \theta^{(m)})$
- $Q(\theta^m, q^m) = \sum_{m=1}^M \mathbb{E}_{q^m} [\log p(x_n, z_n | \theta^m)] + \underbrace{H(q^m)}_{\text{independent of } \theta}$
- $\theta^{(m+1)} = \operatorname{argmax}_{\theta} \mathbb{E}_{q^m} [\log p(x, z | \theta^m)]: M \text{ Step}$
- So what?



Vaibhav Rajan (XRCI)
GMM and EM
40 / 53

So let us see how we can choose such a q okay to do that let us look at this expression again we will ignore the \sum_n because we will bring it back later but I have just not written it so this is the original expression for the lower bound right q function is here I have just written that again here and now we I am just expressing this joint likelihood I am factorizing it in this way so you have the probability of X_n and Z_n and the joint probability is just the probability of Z_n .

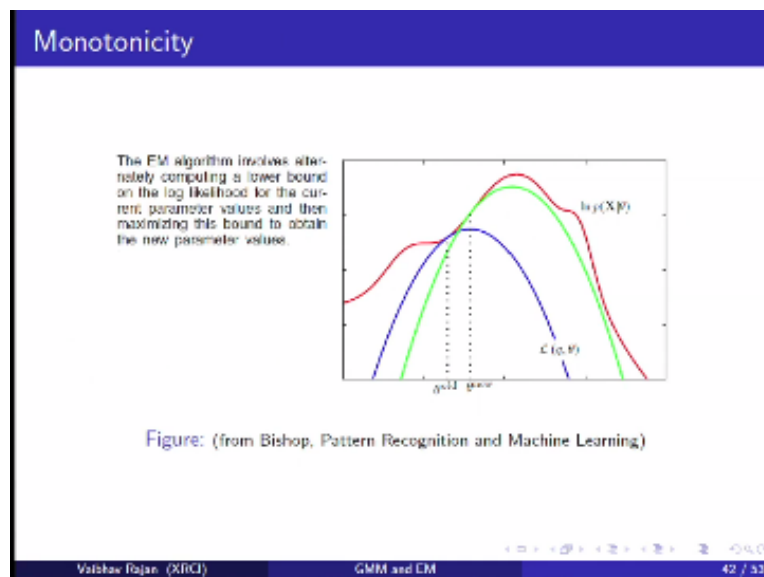
And in the probability of Z_n is given here so they should be X_n, Z_n here yeah it is fine so this is just factorization of this probability and then I just separate it out in different way this time and what we get here is a term which is just the back distance between qz and this probability this distribution right it is negative of the curl divergence between qz and probability of Z_n given as X_n .

And this term is essentially summing over all Z_n for this so this is independent of q and we just get logarithm of you just kept the likelihood back here right and here we have the negative curl divergence between these two distributions so if we want the lower bound to reach the actual likelihood which we are getting here we want this term to become 0 right.

And that we can do by just putting qz and equal to this probability, probability of Z_n given x and θ but again we come back to the same problem that we do not the actual θ because θ but in an iteration of EM we have guess the value of θ EM so we can use that value of θ EM to and use that probability distribution as Q okay.

So at the M step we took q^m the distributions z_n to be exactly this probability distribution the posterior distribution Z_n give the data points and the current cases and we saw that this likelihood is exactly equal to the kl divergence + the log likelihood and because this kl divergence becomes 0 at this point this q function is exactly = to the log likelihood which means the lower bound is tight after E step which is what we wanted and so maximizing Q after this is going to maximize the data log likelihood also.

(Refer Slide Time: 21:04)



To see that see this picture so this is your current value the guest value of θ now the E step and this red curve here is the actual data log likelihood with the original parameters that you do not know now what the E step as ensured is that you get a lower bound using the q function that we had so that lower bound is L so this is the is the lower bound right.

(Refer Slide Time: 21:36)

Monotonicity of EM

- $q(z_n)$: arbitrary distribution over the latent variables
- $q(z_n) \geq 0$ with $\sum_{z_n} q(z_n) = 1$

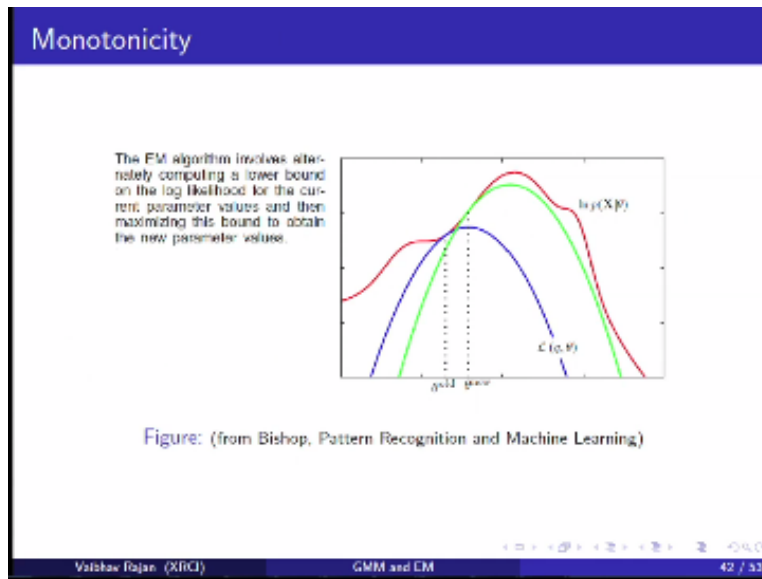
$$\begin{aligned} \log p(X|\theta) &= \sum_{n=1}^N \log \left[\sum_{z_n} p(x_n, z_n|\theta) \right] \\ &= \sum_{n=1}^N \log \left[\sum_{z_n} q(z_n) \frac{p(x_n, z_n|\theta)}{q(z_n)} \right] \\ &\geq \sum_{n=1}^N \sum_{z_n} q(z_n) \log \left[\frac{p(x_n, z_n|\theta)}{q(z_n)} \right] \\ &= \sum_{n=1}^N \underbrace{\sum_{z_n} q(z_n) \log p(x_n, z_n|\theta)}_{\mathbb{E}_q[\log p(x_n, z_n|\theta)]} - \underbrace{\sum_{z_n} q(z_n) \log q(z_n)}_{\text{entropy } H(q)} = Q(\theta, q) \end{aligned}$$

- $\log p(X|\theta) \geq Q(\theta, q)$
- Which distribution q should we choose?

Valtslav Fajen (XIPIC) GMM and EM 30 / 53

And which is exactly the expectation that we are trying to maximize.

(Refer Slide Time: 21:42)



So you use this L function and you get a no that this is lower bound which means it is always lesser than the that curve right the important point is that at the E step this bound is tight which means this is touching the red curve right and if you maximize this you will get a new set of parameters which will increase the L value right but because this is touching it and because this is the lower bound it will also increase the likelihood value for the with respect to the original θ right.

So it is a trick because you cannot we cannot compute this likelihood but we know the lower bound we have computed the lower bound and we have maximizing this but it is guaranteed to be the new values are guaranteed to increase the likelihood in the original likelihood also because at this point the approximation is tight and we are maximizing it okay so now again the at the next step the E step will ensure that the lower bound that you calculate the green curve will be tight.

And once again you maximize it you will get a value somewhere here or any way here and the next value of θ is again going to increase the likelihood because every time you are at each step the E step will ensure that you get a proper lower bound and you always get to the you always make sure that it is tight because of the choice of the distribution of q that we take at each step.

Sorry yeah because what if we get the saddle point the likelihood curve need not always be like this right so for example the likelihood value can be something like this, where you can suppose

it goes like this then how at this point it is not guarantee to go up that way it will just be here in this region so the usual problem with optimization, so we can say now we can do this formally.

(Refer Slide Time: 24: 15)

Monotonicity of EM

$$\begin{aligned}
 l(\theta^{m+1}) &= \log p(X|\theta^{m+1}) \\
 &\geq Q(\theta^{m+1}, q^{m+1}) && \text{lower bound} \\
 &= \max_{\theta} Q(\theta, q^m) && \text{M step} \\
 &\geq Q(\theta^{(m)}, q^m) \\
 &= \log p(X|\theta^{(m)}) && \text{E step bound tight} \\
 &= l(\theta^m)
 \end{aligned}$$

⏪ ⏩ ⏴ ⏵ 🔍
43 / 52

Vaibhav Rajan (XIIIT) GMM and EM

We at the M + 1 M+ first round we have some parameters that is the log likelihood of those parameters and we know that q function is lower bound we provide it for any q any choice of the distribution a small q and then this q value was chosen by the pervious iteration M step so this equality follows this is the maximum value of q which maximizes the maximum at all parameters curly theta and then this by definition is greater than any q here and because this E step bond is tight we get that this is equal to the logarithm equal to the likelihood in the previous step which is just the likelihood of the previous step.

(Refer Slide Time: 25:14)

Singularity in ML solution

Set $\mu_1 = x_1, \Sigma_1 = \sigma_1^2 \mathbb{I}_p, 0 < \pi_1 < 1$

$$\begin{aligned}l(\theta) &= \sum_{n=1}^N \log \left(\sum_{k=1}^K \pi_k \mathcal{N}(x_n | \mu_k, \Sigma_k) \right) \\&= \log \left(\sum_{k=1}^K \pi_k \mathcal{N}(x_1 | \mu_k, \Sigma_k) \right) + \sum_{n=2}^N \log \left(\sum_{k=1}^K \pi_k \mathcal{N}(x_n | \mu_k, \Sigma_k) \right) \\&\geq \log \left(\pi_1 \mathcal{N}(x_1 | \mu_1, \Sigma_1) \right) + \sum_{n=2}^N \log \left(\sum_{k=1}^K \pi_k \mathcal{N}(x_n | \mu_k, \Sigma_k) \right) \\&= \log \left(\pi_1 \mathcal{N}(x_1 | \mu_1, \sigma_1^2 \mathbb{I}_p) \right) + \sum_{n=2}^N \log \left(\sum_{k=1}^K \pi_k \mathcal{N}(x_n | \mu_k, \Sigma_k) \right) \\&= \log \pi_1 - \frac{p}{2} \log 2\pi - \frac{p}{2} \log \sigma_1^2 + \sum_{n=2}^N \log \left(\sum_{k=1}^K \pi_k \mathcal{N}(x_n | \mu_k, \Sigma_k) \right)\end{aligned}$$

- $\sigma_1^2 \rightarrow 0 \implies l(\theta) \rightarrow \infty$
- Singularity when Gaussian collapses onto a data point during fitting

Okay.

(Refer Slide Time: 25:15)

Monotonicity

The EM algorithm involves alternately computing a lower bound on the log likelihood for the current parameter values and then maximizing this bound to obtain the new parameter values.

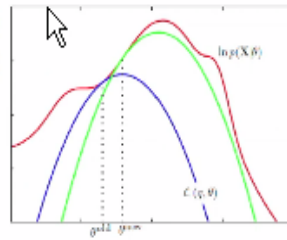


Figure: (from Bishop, Pattern Recognition and Machine Learning)

So any question now is this clear, why it is increasing the likelihood at each point, at each iteration. All right, so now, so that covers the basics of EM, now let us look at some strange cases.

(Refer Slide Time: 25:52)

Singularity in ML solution

Set $\mu_1 = x_1, \Sigma_1 = \sigma_1^2 \mathbb{I}_p, 0 < \sigma_1 < 1$

$$\begin{aligned}
 l(\theta) &= \sum_{n=1}^N \log \left(\sum_{k=1}^K \pi_k \mathcal{N}(x_n | \mu_k, \Sigma_k) \right) \\
 &= \log \left(\sum_{k=1}^K \pi_k \mathcal{N}(x_1 | \mu_k, \Sigma_k) \right) + \sum_{n=2}^N \log \left(\sum_{k=1}^K \pi_k \mathcal{N}(x_n | \mu_k, \Sigma_k) \right) \\
 &\geq \log \left(\pi_1 \mathcal{N}(x_1 | \mu_1, \Sigma_1) \right) + \sum_{n=2}^N \log \left(\sum_{k=1}^K \pi_k \mathcal{N}(x_n | \mu_k, \Sigma_k) \right) \\
 &= \log \left(\pi_1 \mathcal{N}(x_1 | \mu_1, \sigma_1^2 \mathbb{I}) \right) + \sum_{n=2}^N \log \left(\sum_{k=1}^K \pi_k \mathcal{N}(x_n | \mu_k, \Sigma_k) \right) \\
 &= \log \pi_1 - \frac{p}{2} \log 2\pi - \frac{p}{2} \log \sigma_1^2 + \sum_{n=2}^N \log \left(\sum_{k=1}^K \pi_k \mathcal{N}(x_n | \mu_k, \Sigma_k) \right)
 \end{aligned}$$

- $\sigma_1^2 \rightarrow 0 \implies l(\theta) \rightarrow \infty$
- Singularity when Gaussian collapses onto a data point during fitting

◀ ▶ ↺ ↻

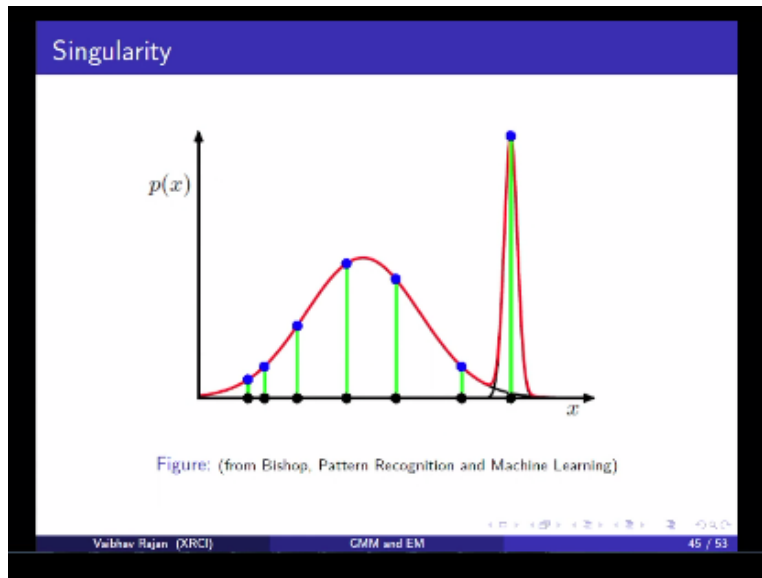
Vaibhav Rajan (XRCI) GMM and EM 44 / 53

Sometimes what happens is when you are running here, you tend to get very strange solutions and this could be one of the reasons. So I am going to motivate this mathematically, so suppose you take your likelihood for that you want to maximize and you set, now suppose you have two components okay, it does not matter. So take one of the components and set μ_1 the mean to be equal to x_1 , one of the data points.

And set Σ_1 to be equal to some diagonal matrix of dimensionality, and take some pivot. So this can be just split into two parts we are looking at just one Gaussian here and when you plug in these values you essentially get this expression okay. Now what happens if Σ_1^2 the variance tends to 0, this likelihood essentially tends to ∞ , I mean this total likelihood, because this value goes to ∞ right.

So this is a problem in general with maximum likelihood solutions, your, the likelihood will tend to ∞ although the fit is really bad.

(Refer Slide Time: 27:10)



This looks like, so the pictorial representation is something like this. What you are doing is you are taking two Gaussians and you are fitting just one data point with one Gaussian, and the other Gaussian is fitting the rest of the data points. So this is in most real life cases this is not a good thing to do, because yeah, it is very unlikely that the data has been generated by two Gaussians like this, one data point from one Gaussian and the rest from the other Gaussian right.

So any, so when you try to do this with just a single Gaussian, you think you will get this problem? Why? Yeah, but suppose you have, suppose I give you, I take uni-dimensional case, and I fit this one Gaussian here, this, there will be a nonzero probability of a point coming from somewhere here right, you know this is the mean. So intuitively we will think that the blue Gaussian is what might have generated this data with so much variance right.

But there is a nonzero probability that the data has been generated from such a Gaussian, so why we will not have this problem there. Yeah, so the maximum likelihood solution will never give you this, maximum likelihood solution is most likely to give you something like this right. When you work out the likelihood the likelihood for the pink Gaussian is definitely going to be lesser than the likelihood for this right.

And again this is just due to the mathematical form of the Gaussian mixture, so because of the summation this is really happening, because it is possible that you can fit the data like that in a way that the likelihood goes to ∞ .

(Refer Slide Time: 29:40)

Solutions

- Reinitialize parameters on detecting collapsing component
- MAP solution $\theta_{MAP} = \operatorname{argmax}_{\theta} \{ \log p(X|\theta) + \log p(\theta) \}$
 - E Step $Q(\theta, \theta^{(m-1)}) = E_{Z|X, \theta^{(m-1)}} \log p(X, Z|\theta)$
 - M Step $\theta^{(m)} = \operatorname{argmax}_{\theta} \{ Q(\theta, \theta^{(m-1)}) + \log p(\theta) \}$

Vaibhav Rajan (XRCI) CMM and EM 46 / 53

So how do you deal with this, the simplest way in a frequent framework is to just keep when you are running EM you check whether it is happening or not and if you, if it is happening then you just reinitialize the parameters, you keep trying to detect such collapsing components and try to do it. And in general actually it is better to restart EM several times, because EM is, as you know it can get stuck at a fixed point it is better to restart EM several times because EM is so as you know it can get stuck at a fix point or a saddle point so with different initialization parameters you can get much better solutions as we saw in this stimulation as well.

The basin solution is to take priors okay, you take priors on each of the parameters and it turns out you can work out the math and see that the expects the E step remains the same and the only difference necessary is the additional term in the M step that we need to maximize and this usually solves the problem by choosing right priors.

(Refer Slide Time: 30:51)

Finding K

- Generate candidate models for $K = K_{\min}, \dots, K_{\max}$
- Select $K^* = \operatorname{argmin}_k \{C(\hat{\theta}_k, k), k = k_{\min}, \dots, k_{\max}\}$
- $C(\hat{\theta}_k, k) = -\log p(X|\hat{\theta}_k) + f(k)$
 - f increasing function penalizing high values of k
 - AIC: $C(\hat{\theta}_k, k) = -2 \log p(X|\hat{\theta}_k) + 2k$
 - BIC: $C(\hat{\theta}_k, k) = -2 \log p(X|\hat{\theta}_k) + k \log n$

Vishal Papan (XRD)
GMM and EM 47 / 68

So now it is come to finding K , till now we have assumed that we know the number of components and how do we find K this is there is no really good solution to finding K and what states this in usually prefer and what works well in practice is to generate many candidate models you look at the data and you assume that okay, they cannot be lesser than three components here they can be more than 12 components here.

So let us run EM for all these different values of K and you choose that K which minimizes some criterion okay, and this there are different criteria that people have discussed for example it is something like the regularization that you do in another models you basically penalize high values of K .

So the AIC a K information criterion is this, this is just the log likelihood $+K$, so minimizing this will give you the least number of components which can explain the data well okay. There is a BIC information criterion which uses $K \log n$ similar general idea and then are other approaches of finding K which are basin nonparametric approaches where you assume some derive process priors and then the method itself automatically estimates K , right.

So the algorithm that we discussed in that form was given in 1977, so you can imagine that a lot of work has been done on EM since 1977.

(Refer Slide Time: 32:43)

The slide is titled "EM Variants" and contains a bulleted list of seven different EM algorithm variants. At the bottom of the slide, there is a footer with the text "Vishnu Rajan (XRO)", "GMM and EM", and "48 / 68".

- Online EM: large or streaming datasets
- Annealed EM: to increase chances of finding global maximum
- Variational EM: computationally intractable E step
- Monte Carlo EM: computationally intractable E step
- Generalized EM: computationally intractable M step (increase expected likelihood)
- Expected Conditional Maximization (ECM): dependent parameters, sequentially optimized
- Over-relaxed EM: slow, lots of missing data

There are a lot of different kinds of EM algorithms, there are online versions that work on large streaming data sets like I said EM is designed to find local maximum, so there are annealed versions that increase the chances of finding global maximum, the simplest solution is random restarts but annealed does something more variation, so sometimes so in the case of Gaussian we saw that the E step and the M steps they were computationally tractable we could derive analytical formula for these.

But in a lot of cases if there is time I can show one, we will see that they are not computationally intractable and sometimes you need to do additional things. So there are variational versions of EM, there are stochastic versions of EM, Monte Carlo version where you have intractable E steps there is something called generalized EM which was one of the earliest algorithms where you have computationally intractable m steps.

Then when we have sequential parameters dependent parameters then there are other versions of EM and in general EM is quite slow right, so your each step within EM with the iteration is computationally not very expensive, but convergence usually very slow and is especially slow when you have lots of missing data or lots of latent variables to infer okay. So there are many approaches to deal with that these I can acceleration techniques over relaxed EM and so on. So to summarize like what I said.

(Refer Slide Time: 34:25)

Summary

Advantages of EM

- Each iteration monotonically increases the likelihood
 - except at fixed points
 - can monitor convergence and debug programs by watching likelihood
- Numerically stable, easily implemented
- Many problems can be modeled as incomplete data problems
- Cost per iteration is low (but may require large number of iterations)

Disadvantages of EM

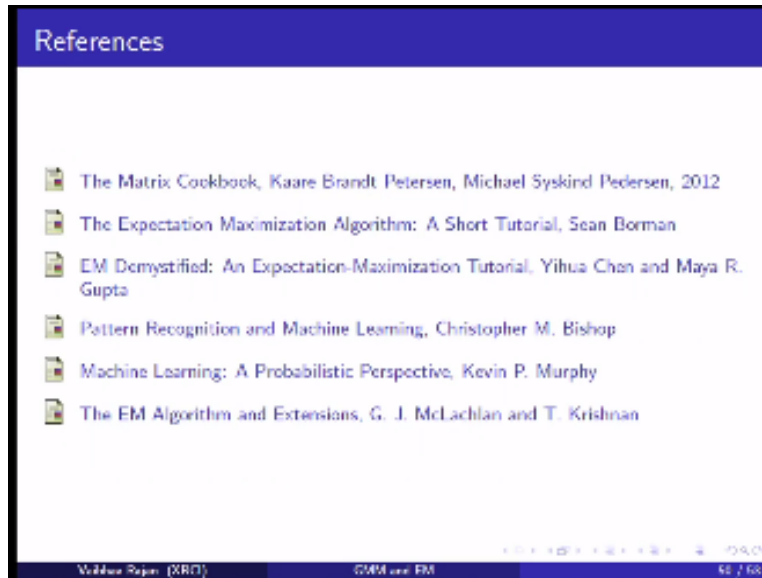
- Slow convergence
- No guarantees of finding global maximum
- E or M Step may be analytically intractable

Vishal Rajan (XRD) GMM and EM 48 / 68

The major advantage of EM is that it monotonically increases the likelihood it guarantees that, if you take any distribution any mixture model or anything any likelihood computation where there are hidden variables or latent variables and you apply EM and if you follow the formulas carefully you will guarantee that the likelihood is increased except at fixed points.

And it is usually numerically very stable compared to other techniques like gradient descent, it is easily implemented and the interesting thing is that many problems can be modeled as incomplete data problems we saw that in the case of Gaussian mixture there is no missing data and the beginning but we assume the latent variables to be missing. The disadvantages as I mentioned is slow convergence and there is no guarantees of finding global maximum, and the steps maybe analytical intractable okay.

(Refer Slide Time: 35:24)



Yeah, so the two standard references have very nice explanation for you and there are very nice tutorials also available Matrix Cookbook you should be familiar with to get all your matrix derivatives and this is these standard reference if you want to go really deep into EM McLachlan and Krishnan's book and EM the whole book is on EM algorithm. The EM can always solve it but it not maybe able to solve well, well lots of missing data then usually the it does not give good.

Right, so there is some there is lot of work on so these estimates that you are getting you may need to, you may sometimes want to know how good those estimates are right, so you want to get these standard errors on those estimates, so there is so in fact that is one of the flaws of EM it does not automatically give you that but there are methods to deal with that for example, there are some boots trap methods that can give you estimates of the error that you error estimates for the estimated parameters.

Yeah, there also guessed so that is something you have guess based on the data that you have, so if you take these standard r packages like M clusters or something like that they usually have some default parameters 2 and 12 or something like that but then you can set them so when M cluster trans and gives you like what I showed in this stimulation when it runs and tries to find the parameters it runs it for all those different values of K and takes the best one, best one with respect to the likelihood, okay.

So it will be a good exercise I think like if you take some different distribution so take something like Bernoulli's and very simple distribution and work out the math it will be quite nice to see how it works out and yeah, even the other thing that I did not work out here, this part is also quite simple to do yeah, assume that there is a prior and see how it works out. But the general idea is clear right okay.

IIT Madras Production

Funded by
Department of Higher Education
Ministry of Human Resource Development
Government of India

www.nptel.ac.in

Copyrights Reserved