PRML（Pattern Recognition And Machine Learning）读书会

第九章　Mixture Models and EM

主讲人 网络上的尼采

（新浪微博：@Nietzsche_复杂网络机器学习）

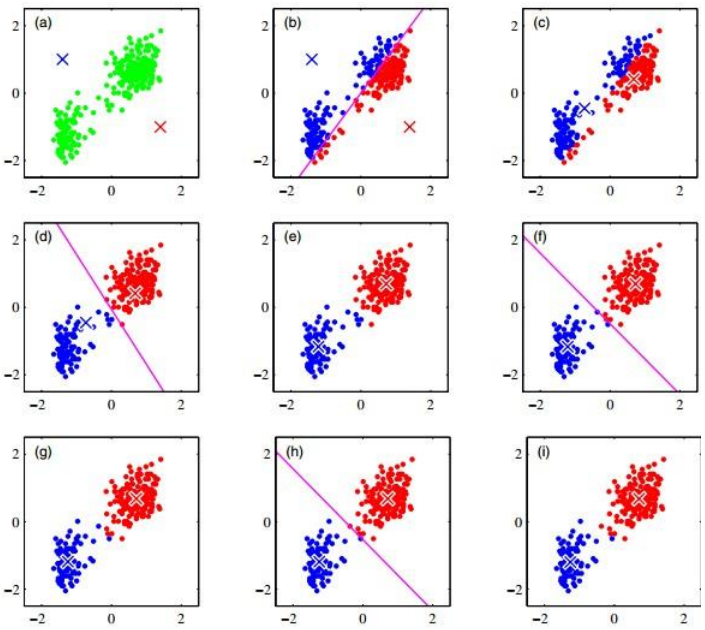
QQ 群 177217565

读书会微信公众平台请扫描下面的二维码

网络上的尼采(813394698) 9:10:56
今天的主要内容有 k-means、混合高斯模型、 EM 算法。
对于 k-means 大家都不会陌生，非常经典的一个聚类算法，已经 50 多年了，关于 clustering 推荐一篇不错的 survey: Data clustering: 50 years beyond K-means。k-means 表达的思想非常经典，就是对于复杂问题分解成两步不停的迭代进行逼近，并且每一步相对于前一步都是递减的。
k-means 有个目标函数 ：

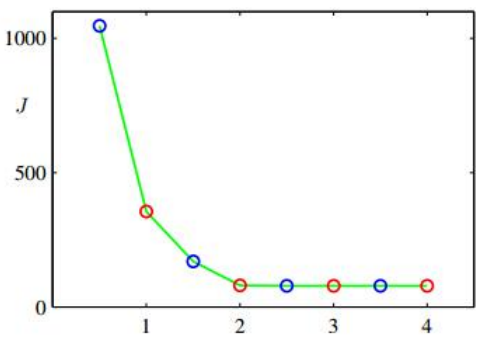
$$J = \sum_{n=1}^{N}\sum_{k=1}^{K} r_{nk}\|\mathbf{x}_n - \boldsymbol{\mu}_k\|^2$$

假设有 k 个簇，$\boldsymbol{\mu}_k$是第 k 个簇的均值；每个数据点都有一个向量表示属于哪个簇，r_{nk} 是向量的元素，如果点 x_n 属于第 k 个簇，则 r_{nk} 是 1，向量的其他元素是 0。
上面这个目标函数就是各个簇的点与簇均值的距离的总和，k-means 要做的就是使这个目标函数最小。这是个 NP-hard 问题，k-means 只能收敛到局部最优。
算法的步骤非常简单：
先随机选 k 个中心点
第一步也就是 E 步把离中心点近的数据点划分到这个簇里；
第二步 M 步根据各个簇里的数据点重新确定均值，也就是中心点。
然后就是迭代第一步和第二步，直到满足收敛条件为止。
自强<ccab4209211@qq.com> 9:29:00
收敛是怎么判断的呀？
网络上的尼采(813394698) 9:30:16
不再发生大的变化。大家思考下不难得出：无论 E 步还是 M 步,目标函数都比上一步是减少的。 下面是划分两个簇的过程 ：



下面这个图说明聚类过程中目标函数单调递减，经过三轮迭代就收敛了，由于目标函数只减不增，并且有界，所以 k-means 是可以保证收敛的：

Plot of the cost function J given by (9.1) after each E step (blue points) and M step (red points) of the K-means algorithm for the example shown in Figure 9.1. The algorithm has converged after the third M step, and the final EM cycle produces no changes in either the assignments or the prototype vectors.



书里还举例一个 k-means 对图像分割和压缩的例子：

Figure 9.3 Two examples of the application of the K-means clustering algorithm to image segmentation showing the initial images together with their K-means segmentations obtained using various values of K. This also illustrates of the use of vector quantization for data compression, in which smaller values of K give higher compression at the expense of poorer image quality.

图像分割后，每个簇由均值来表示，每个像素只存储它属于哪个簇就行了。压缩后图像的大小是 k 的函数 ：

pixel intensity vectors we transmit the identity of the nearest vector $\boldsymbol{\mu}_k$. Becau
there are K such vectors, this requires $\log_2 K$ bits per pixel. We must also transm
the K code book vectors $\boldsymbol{\mu}_k$, which requires $24K$ bits, and so the total number
bits required to transmit the image is $24K + N \log_2 K$ (rounding up to the near

现在讨论下 k-means 的性质和不足 ：

首先对初值敏感 ，由于只能收敛到局部最优，初值很大程度上决定它收敛到哪里；

从算法的过程可以看出，k-means 对椭球形状的簇效果最好 ，不能发现任意形状的簇；

对孤立点干扰的鲁棒性差，孤立点是异质的，可以说是均值杀手，k-means 又是围绕着均值展开的，试想下，原离簇的孤立点对簇的均值的拉动作用是非常大的。

针对这些问题后来又有了基于密度的 DBSCAN 算法，最近 Science 上发了另一篇基于密度的方法：Clustering by fast search and find of density peaks。基于密度的方法无论对 clustering 还是 outliers detection 效果都不错，和 k-means 不同，这些算法已经没有了目标函数，但鲁棒性强，可以发现任意形状的簇。

另外如何自动确定 k-means 的 k 是目前研究较多的一个问题。k-means 就到这里，现在一块讨论下。

口水猫(465191936) 9:49:20

如果对于这批数据想做 k-mean 聚类，那么如果去换算距离？

网络上的尼采(813394698) 9:49:53

k-means 一般基于欧式距离，关于距离度量是个专门的方向，点集有了度量才能有拓扑，有专门的度量学习这个方向。

口水猫(465191936) 9:50:08

嗯嗯 有没有一些参考意见了，k 的选择可以参考 coursera 上的视频 选择 sse 下降最慢的那个拐点的 k

网络上的尼采(813394698) 9:51:41

关于选哪个 k 是最优的比较主观，有从结果稳定性来考虑的，毕竟一个算法首先要保证的是多次运行后结果相差不大，关于这方面有一篇 survey: Clustering stability an overview。另外还有其他自动选 k 的方法，DP、MDL 什么的。MDL 是最短描述长度，从压缩的角度来看 k-means；DP 是狄利克雷过程，一种贝叶斯无参方法，感兴趣可以看 JORDAN 小组的文章。

我们下面讲混合高斯模型 GMM，第二章我们说过，高斯分布有很多优点并且普遍存在，但是单峰函数，所以对于复杂的分布表达能力差，我们可以用多个高斯分布的线性组合来逼近这些复杂的分布。我们看下 GMM 的线性组合形式：

$$p(\mathbf{x}) = \sum_{k=1}^{K} \pi_k \mathcal{N}(\mathbf{x}|\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k).$$

对于每个数据点是哪个分布生成的，我们假设有个 Z 隐变量 ，和 k-means 类似，对于每个数据点都有一个向量 z，如果是由第 k 个分布生成，元素 z_k=1,其他为 0。
z_k=1 的概率的先验就是高斯分布前的那个系数：

$$p(z_k = 1) = \pi_k$$

下面是 z_k=1 概率的后验，由贝叶斯公式推导的：

$$\begin{aligned}\gamma(z_k) \equiv p(z_k = 1|\mathbf{x}) &= \frac{p(z_k = 1)p(\mathbf{x}|z_k = 1)}{\sum_{j=1}^{K} p(z_j = 1)p(\mathbf{x}|z_j = 1)} \\ &= \frac{\pi_k \mathcal{N}(\mathbf{x}|\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)}{\sum_{j=1}^{K} \pi_j \mathcal{N}(\mathbf{x}|\boldsymbol{\mu}_j, \boldsymbol{\Sigma}_j)}.\end{aligned}$$

其实很好理解，如果没有 π_k限制，数据点由哪个分布得出的概率大 z_k=1 的期望就大，但前面还有一个系数限制，所以期望形式是：

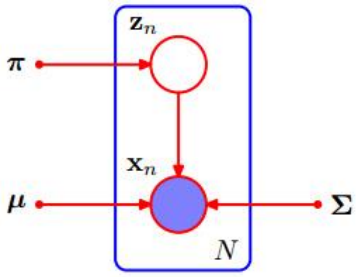
$$\frac{\pi_k \mathcal{N}(\mathbf{x}|\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)}{\sum_{j=1}^{K} \pi_j \mathcal{N}(\mathbf{x}|\boldsymbol{\mu}_j, \boldsymbol{\Sigma}_j)}.$$

刚才有人问如何确定模型的参数，我们首先想到的就是 log 最大似然：

$$\ln p(\mathbf{X}|\boldsymbol{\pi}, \boldsymbol{\mu}, \boldsymbol{\Sigma}) = \sum_{n=1}^{N} \ln \left\{ \sum_{k=1}^{K} \pi_k \mathcal{N}(\mathbf{x}_n|\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k) \right\}$$

但是我们可以观察下这个目标函数，log 里面有加和，求最优解是非常困难的，混合高斯和单个高斯的参数求法差别很大，如果里面有一个高斯分布坍缩成一个点，log 似然函数会趋于无穷大。由于直接求解困难，这也是引入 EM 的原因。
下面是 GMM 的图表示 ：

Graphical representation of a Gaussian mixture model for a set of N i.i.d. data points $\{\mathbf{x}_n\}$, with corresponding latent points $\{\mathbf{z}_n\}$, where $n = 1, \ldots, N$.



我们可以试想下，如果隐变量 z_n 是可以观测的，也就是知道哪个数据点是由哪个分布生成的，那么我们求解就会很方便，可以利用高斯分布直接得到解析解。 但关键的是 z_n 是隐变量，我们没法观测到。但是我们可以用它的期望来表示。 现在我们来看一下 EM 算法在 GMM 中的应用：

EM for Gaussian Mixtures

Given a Gaussian mixture model, the goal is to maximize the likelihood function with respect to the parameters (comprising the means and covariances of the components and the mixing coefficients).

1. Initialize the means $\boldsymbol{\mu}_k$, covariances $\boldsymbol{\Sigma}_k$ and mixing coefficients π_k, and evaluate the initial value of the log likelihood.
2. **E step**. Evaluate the responsibilities using the current parameter values

$$\gamma(z_{nk}) = \frac{\pi_k \mathcal{N}(\mathbf{x}_n|\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)}{\sum_{j=1}^{K} \pi_j \mathcal{N}(\mathbf{x}_n|\boldsymbol{\mu}_j, \boldsymbol{\Sigma}_j)}. \tag{9.23}$$

上面是 EM 对 GMM 的 E 步

我们首先对模型的参数初始化

E 步就是我们利用这些参数得 z_{nk} 的期望，这种形式我们前面已经提到了：

$$\gamma(z_{nk}) = \frac{\pi_k \mathcal{N}(\mathbf{x}_n|\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)}{\sum_{j=1}^{K} \pi_j \mathcal{N}(\mathbf{x}_n|\boldsymbol{\mu}_j, \boldsymbol{\Sigma}_j)}$$

现在我们有隐藏变量的期望了，由期望得新的模型参数也就是 M 步，高斯分布的好处就在这儿，可以推导出新参数的闭式解：

3. **M step**. Re-estimate the parameters using the current responsibilities

$$\boldsymbol{\mu}_k^{\text{new}} = \frac{1}{N_k} \sum_{n=1}^{N} \gamma(z_{nk}) \mathbf{x}_n \tag{9.24}$$

$$\boldsymbol{\Sigma}_k^{\text{new}} = \frac{1}{N_k} \sum_{n=1}^{N} \gamma(z_{nk}) (\mathbf{x}_n - \boldsymbol{\mu}_k^{\text{new}}) (\mathbf{x}_n - \boldsymbol{\mu}_k^{\text{new}})^{\mathrm{T}} \tag{9.25}$$

$$\pi_k^{\text{new}} = \frac{N_k}{N} \tag{9.26}$$

where

$$N_k = \sum_{n=1}^{N} \gamma(z_{nk}). \tag{9.27}$$

然后不断迭代 E 步和 M 步直到满足收敛条件。

为什么要这么做，其实 EM 算法对我们前面提到的 log 最大似然目标函数：

$$\ln p(\mathbf{X}|\boldsymbol{\pi},\boldsymbol{\mu},\boldsymbol{\Sigma}) = \sum_{n=1}^{N} \ln \left\{ \sum_{k=1}^{K} \pi_k \mathcal{N}(\mathbf{x}_n|\boldsymbol{\mu}_k,\boldsymbol{\Sigma}_k) \right\}$$ 是单调递增的。

karnon(447457116) 10:39:42
用 EM 来解 GMM 其实是有问题的，解出来的解并不是最优的。。
网络上的尼采(813394698) 10:40:14
嗯，这个问题最后讲。
我们再来看 EM 更一般的形式：

$\ln p(\mathbf{X}|\boldsymbol{\theta})$ 是我们的目标函数，加入隐藏变量可以写成这种形式：$$\ln p(\mathbf{X}|\boldsymbol{\theta}) = \ln \left\{ \sum_{\mathbf{Z}} p(\mathbf{X},\mathbf{Z}|\boldsymbol{\theta}) \right\}$$

先初始化模型的参数，由于隐变量无法观测到，我们用参数来得到它的后验 $p(\mathbf{Z}|\mathbf{X},\boldsymbol{\theta}^{\text{old}})$

然后呢，我们通过隐藏变量的期望得到新的完整数据的最大似然函数：

$$\mathcal{Q}(\boldsymbol{\theta},\boldsymbol{\theta}^{\text{old}}) = \sum_{\mathbf{Z}} p(\mathbf{Z}|\mathbf{X},\boldsymbol{\theta}^{\text{old}}) \ln p(\mathbf{X},\mathbf{Z}|\boldsymbol{\theta})$$

以上是 E 步，M 步是求这个似然函数的 Q 函数的最优解，也就是新的参数：$$\boldsymbol{\theta}^{\text{new}} = \arg\max_{\boldsymbol{\theta}} \mathcal{Q}(\boldsymbol{\theta},\boldsymbol{\theta}^{\text{old}})$$

注意这个 Q 函数是包含隐变量的完整数据的似然函数，不是我们一开始的目标函数 $\ln p(\mathbf{X}|\boldsymbol{\pi},\boldsymbol{\mu},\boldsymbol{\Sigma})$

其实求完整数据的最大似然是在逼近我们目标函数的局部最优解，这个在后面讲。
下面这个是一般化 EM 算法的步骤：

The General EM Algorithm

Given a joint distribution $p(\mathbf{X},\mathbf{Z}|\boldsymbol{\theta})$ over observed variables $\mathbf{X}$ and latent variables $\mathbf{Z}$, governed by parameters $\boldsymbol{\theta}$, the goal is to maximize the likelihood function $p(\mathbf{X}|\boldsymbol{\theta})$ with respect to $\boldsymbol{\theta}$.

1. Choose an initial setting for the parameters $\boldsymbol{\theta}^{\text{old}}$.

---

2. **E step** Evaluate $p(\mathbf{Z}|\mathbf{X},\boldsymbol{\theta}^{\text{old}})$.

3. **M step** Evaluate $\boldsymbol{\theta}^{\text{new}}$ given by

$$\boldsymbol{\theta}^{\text{new}} = \arg\max_{\boldsymbol{\theta}} \mathcal{Q}(\boldsymbol{\theta},\boldsymbol{\theta}^{\text{old}}) \tag{9.32}$$

where

$$\mathcal{Q}(\boldsymbol{\theta},\boldsymbol{\theta}^{\text{old}}) = \sum_{\mathbf{Z}} p(\mathbf{Z}|\mathbf{X},\boldsymbol{\theta}^{\text{old}}) \ln p(\mathbf{X},\mathbf{Z}|\boldsymbol{\theta}). \tag{9.33}$$

4. Check for convergence of either the log likelihood or the parameter values. If the convergence criterion is not satisfied, then let

$$\boldsymbol{\theta}^{\text{old}} \leftarrow \boldsymbol{\theta}^{\text{new}} \tag{9.34}$$

and return to step 2.

EM 算法只所以用途广泛在于有潜在变量的场合都能用，并不局限于用在 GMM 上。现在我们回过头来看混合高斯模型的 M 步，这些得到的新参数就是 Q 函数的最优解：

3. **M step**. Re-estimate the parameters using the current responsibilities

$$\boldsymbol{\mu}_k^{\text{new}} = \frac{1}{N_k}\sum_{n=1}^{N}\gamma(z_{nk})\mathbf{x}_n \tag{9.24}$$

$$\boldsymbol{\Sigma}_k^{\text{new}} = \frac{1}{N_k}\sum_{n=1}^{N}\gamma(z_{nk})\left(\mathbf{x}_n-\boldsymbol{\mu}_k^{\text{new}}\right)\left(\mathbf{x}_n-\boldsymbol{\mu}_k^{\text{new}}\right)^{\mathrm{T}} \tag{9.25}$$

$$\pi_k^{\text{new}} = \frac{N_k}{N} \tag{9.26}$$

where

$$N_k = \sum_{n=1}^{N}\gamma(z_{nk}). \tag{9.27}$$

再思考下 k-means，其实它是 EM 的特例，只不过是 k-means 对数据点的分配是硬性的，在 E 步每个数据点必须分配到一个簇，z 里面只有一个 1 其他是 0,而 EM 用的是 z 的期望：

$$\mathbb{E}_{\mathbf{Z}}[\ln p(\mathbf{X},\mathbf{Z}|\boldsymbol{\mu},\boldsymbol{\Sigma},\boldsymbol{\pi})] \rightarrow -\frac{1}{2}\sum_{n=1}^{N}\sum_{k=1}^{K} r_{nk}\|\mathbf{x}_n-\boldsymbol{\mu}_k\|^2 + \text{const.}$$

下面这个图说明 k-means 是 EM 算法特例，这与前面 k-means 聚类的过程的图对应：

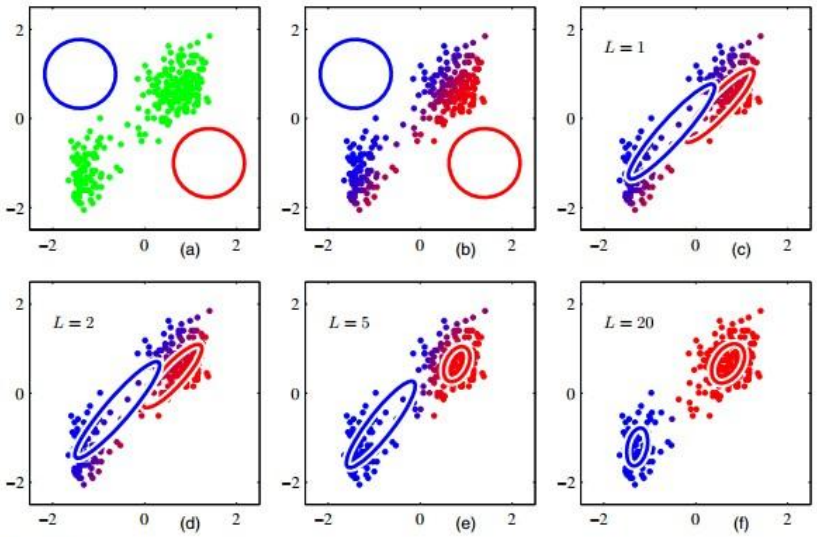


Figure 9.8 Illustration of the EM algorithm using the Old Faithful set as used for the illustration of the K-means algorithm in Figure 9.1. See the text for details.

对于 EM 算法性质的证明最后讲，下面讲混合伯努利模型，高斯分布是针对连续的属性，伯努利是针对离散属性。混合形式：

$$p(\mathbf{x}|\boldsymbol{\mu},\boldsymbol{\pi}) = \sum_{k=1}^{K}\pi_k p(\mathbf{x}|\boldsymbol{\mu}_k)$$

目标函数，log 里面同样有加和：

$$\ln p(\mathbf{X}|\boldsymbol{\mu},\boldsymbol{\pi}) = \sum_{n=1}^{N}\ln\left\{\sum_{k=1}^{K}\pi_k p(\mathbf{x}_n|\boldsymbol{\mu}_k)\right\}$$，。

z_{nk}=1 的期望：

$$\gamma(z_{nk}) = \mathbb{E}[z_{nk}] = \frac{\sum_{z_{nk}} z_{nk}\left[\pi_k p(\mathbf{x}_n|\boldsymbol{\mu}_k)\right]^{z_{nk}}}{\sum_{z_{nj}}\left[\pi_j p(\mathbf{x}_n|\boldsymbol{\mu}_j)\right]^{z_{nj}}} = \frac{\pi_k p(\mathbf{x}_n|\boldsymbol{\mu}_k)}{\sum_{j=1}^{K}\pi_j p(\mathbf{x}_n|\boldsymbol{\mu}_j)}.$$

Q 函数：

$$\mathbb{E}_{\mathbf{Z}}[\ln p(\mathbf{X},\mathbf{Z}|\boldsymbol{\mu},\boldsymbol{\pi})] = \sum_{n=1}^{N}\sum_{k=1}^{K}\gamma(z_{nk})\left\{\ln \pi_k + \sum_{i=1}^{D}[x_{ni}\ln \mu_{ki} + (1-x_{ni})\ln(1-\mu_{ki})]\right\}$$

以上是 E 步，下面是 M 步的解，这两个：

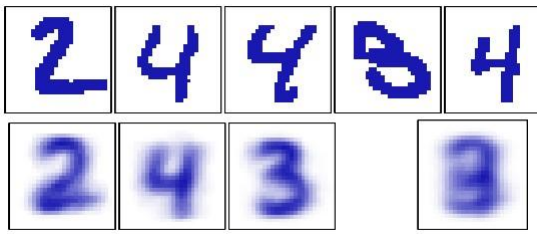
$$\boldsymbol{\mu}_k = \overline{\mathbf{x}}_k. \qquad \pi_k = \frac{N_k}{N}$$

其中：

$$N_k = \sum_{n=1}^{N}\gamma(z_{nk})$$

$$\overline{\mathbf{x}}_k = \frac{1}{N_k}\sum_{n=1}^{N}\gamma(z_{nk})\mathbf{x}_n$$

书里举了一个手写字聚类的例子 ，先对像素二值化 ，然后聚成 3 个簇：



ure 9.10 Illustration of the Bernoulli mixture model in which the top row shows examples from the digits d

这是 3 个簇的代表：

k=1 时簇的代表：

最后说下 EM 算法为什么能收敛到似然函数的局部最优解:

我们的目标函数是分布 $p(\mathbf{X}|\boldsymbol{\theta})$ 的最大 log 似然

引入潜在变量，分布表示为：$$p(\mathbf{X}|\boldsymbol{\theta}) = \sum_{\mathbf{Z}} p(\mathbf{X}, \mathbf{Z}|\boldsymbol{\theta})$$

定义隐藏变量的分布为 q(z)，目标函数可以表达为下面形式，这个地方是神来一笔：

$$\ln p(\mathbf{X}|\boldsymbol{\theta}) = \mathcal{L}(q, \boldsymbol{\theta}) + \mathrm{KL}(q\|p)$$

where we have defined

$$\begin{aligned}\mathcal{L}(q, \boldsymbol{\theta}) &= \sum_{\mathbf{Z}} q(\mathbf{Z}) \ln \left\{ \frac{p(\mathbf{X}, \mathbf{Z}|\boldsymbol{\theta})}{q(\mathbf{Z})} \right\} \\ \mathrm{KL}(q\|p) &= -\sum_{\mathbf{Z}} q(\mathbf{Z}) \ln \left\{ \frac{p(\mathbf{Z}|\mathbf{X}, \boldsymbol{\theta})}{q(\mathbf{Z})} \right\}\end{aligned}$$

其中$\mathrm{KL}(q\|p)$是$p(\mathbf{X}, \mathbf{Z}|\boldsymbol{\theta})$与 q(z)的 KL 散度；$\mathcal{L}(q, \boldsymbol{\theta})$是 q(z)的泛函形式。

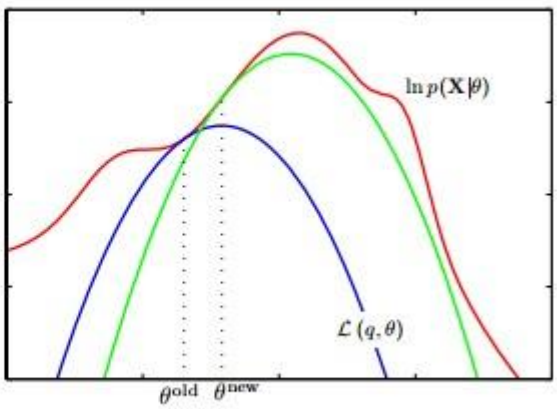
我们原来讲过根据 Jensen 不等式来证明 KL 散度是非负的，这个性质在这里发挥了作用。由于$\mathrm{KL}(q\|p)$是大于等于零的，所以$\mathcal{L}(q, \boldsymbol{\theta})$是目标函数$\ln p(\mathbf{X}|\boldsymbol{\theta})$的下界。$\mathcal{L}(q, \boldsymbol{\theta})$与目标函数什么时候相等呢？其实就是$\mathrm{KL}(q\|p)$等于 0 的时候，也就是 q(z)与 z 的后验分布$p(\mathbf{X}, \mathbf{Z}|\boldsymbol{\theta})$相同时，这个时候就是 E 步 z 取它的期望时候。$\mathcal{L}(q, \boldsymbol{\theta})$与目标函数相等是为了取一个比较紧的 bound。

M 步就是最大化$\mathcal{L}(q, \boldsymbol{\theta})$，随着$\mathcal{L}(q, \boldsymbol{\theta})$的增大，$\mathrm{KL}(q\|p)$开始大于 0，也就是目标函数比$\mathcal{L}(q, \boldsymbol{\theta})$增大的幅度更大。这个过程是使目标函数一直单调递增的。下面是$\mathcal{L}(q, \boldsymbol{\theta})$与 Q 函数的关系，这也是为什么 M 步新参数取完整数据的最大似然解的原因：

$$\begin{aligned}\mathcal{L}(q, \boldsymbol{\theta}) &= \sum_{\mathbf{Z}} p(\mathbf{Z}|\mathbf{X}, \boldsymbol{\theta}^{\mathrm{old}}) \ln p(\mathbf{X}, \mathbf{Z}|\boldsymbol{\theta}) - \sum_{\mathbf{Z}} p(\mathbf{Z}|\mathbf{X}, \boldsymbol{\theta}^{\mathrm{old}}) \ln p(\mathbf{Z}|\mathbf{X}, \boldsymbol{\theta}^{\mathrm{old}}) \\ &= \mathcal{Q}(\boldsymbol{\theta}, \boldsymbol{\theta}^{\mathrm{old}}) + \mathrm{const} \qquad (9.74)\end{aligned}$$

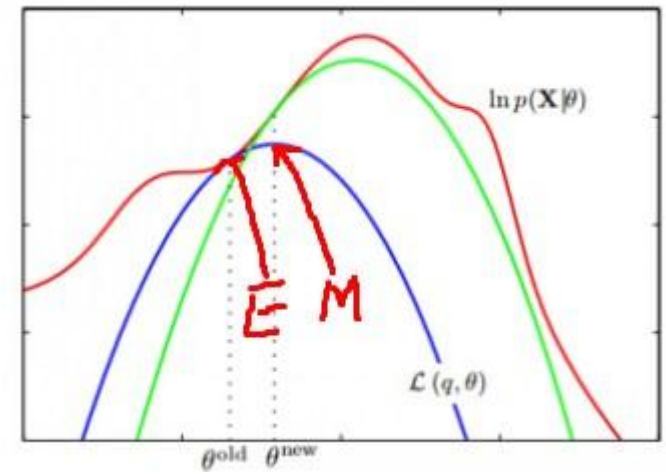
最后上一张非常形象的图，解释为什么 EM 能收敛到目标函数的局部最优：

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. See the text for a full discussion.



红的曲线是目标函数；蓝的绿的曲线是两步迭代。

咱们先看蓝的 E 步和 M 步：



E 步时就是取 z 的期望的时候，这时目标函数与$\mathcal{L}(q,\boldsymbol{\theta})$相同；

M 步就是最大化$\mathcal{L}(q,\boldsymbol{\theta})$

绿线是下一轮的迭代，EM 过程中，目标函数一直是单调上升的，并且有界，所以 EM 能够保证收敛。但不一定能收敛到最优解，这与初始值有很大关系，试想一下，目标函数的曲线变动下，EM 就有可能收敛到局部最优了。