PRML（Pattern Recognition And Machine Learning）读书会第八次讲课

第九章　Mixture Models and EM

主讲人 网络上的尼采

（我的微博 http://weibo.com/dmalgorithms）

QQ 群 177217565

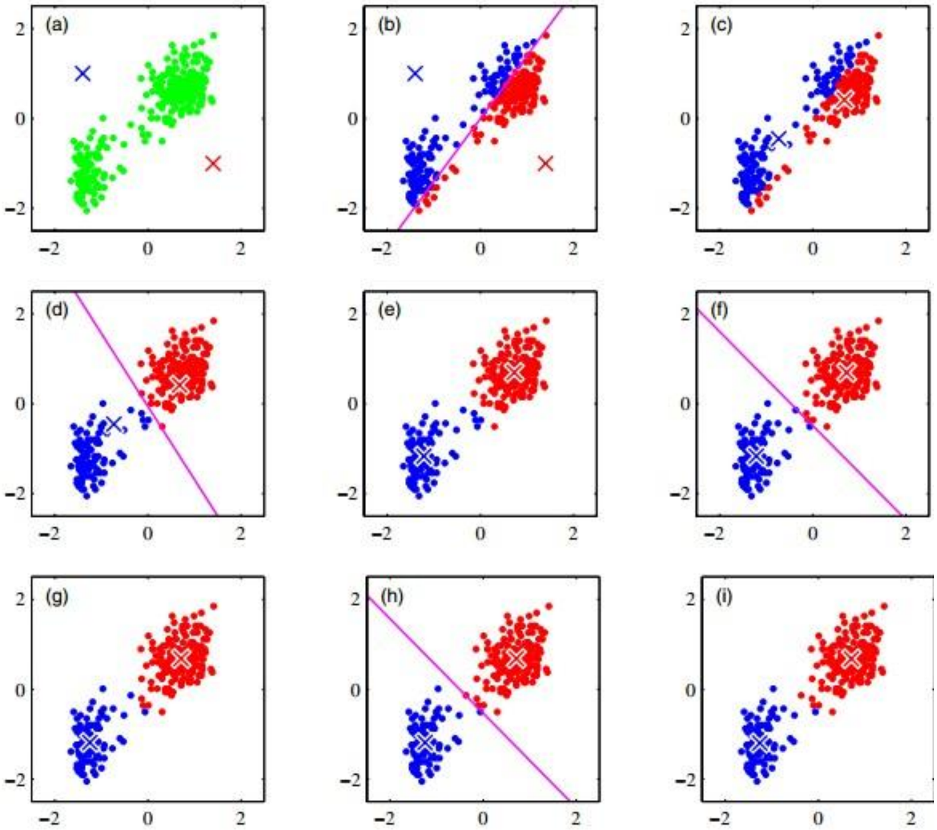
网络上的尼采(813394698) 9:10:56
今天的主要内容有 k-means 混合高斯 EM
karnon(447457116) 9:12:00
EM。。
karnon(447457116) 9:12:06
我永远的痛
网络上的尼采(813394698) 9:12:16
对于 k-means 大家都不会太陌生，非常经典的一个算法，50 多年了到现在这个算法还发着文章。
k-means 表达的思想非常经典，就是对于复杂问题分解成两步不停的迭代进行逼近，并且每一步相对于前一步都是递减的。
k-means 有个目标函数

n $r_{nk}=1$, and $r_{nj}=0$ for $j \neq k$. This is known as
e can then define an objective function, sometimes
ven by

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

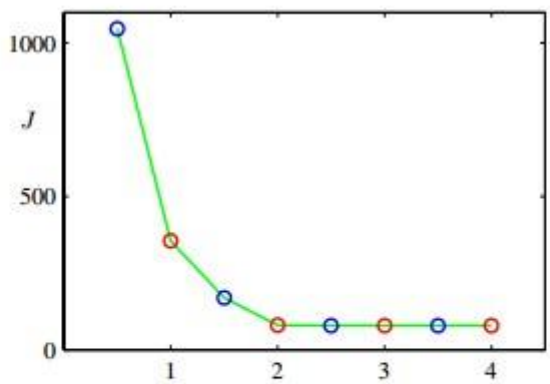
假设有 k 个簇，μ_k 是第 k 个簇的均值
滴水勤泉(8834388) 9:18:04
支持尼采！
网络上的尼采(813394698) 9:18:43
r_{nk} 是个向量的元素，每个数据点都有一个向量表示属于哪个簇，表示如果点 x_n 属于第 k 个簇，则 r_{nk} 是 1，向量的其他元素是 0.
whuSky(102030175) 9:20:03

网络上的尼采(813394698) 9:20:53
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 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，大家可以思考下，无论 e 步还是 m 步,目标函数都比上一步是减少的。



这是划分两个簇的过程

.2 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.



这是目标函数单调递减，经过三轮迭代就收敛了，由于目标函数只减不增，所以 k-means 是可以保证收敛的。

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。

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 transı
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

压缩后图像的大小是 k 的函数

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

对孤立点的干扰鲁棒性差

月苔河璞(360961410) 9:42:43

能不能自适应选初值

网络上的尼采(813394698) 9:43:20

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，一个原离簇的孤立点对簇的均值的拉动作用是非常大的。

liyitan__ML<liyitan2144@163.com> 9:44:20

‘异质’是指？

网络上的尼采(813394698) 9:44:51
针对这些问题后来又有了 dbscan，不过那个算法好像已经没有了目标函数。
异质是指不是同一机制生成的。
月苔河璞(360961410) 9:46:13
但是收敛很快啊
网络上的尼采(813394698) 9:47:38
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
口水猫(465191936) 9:49:26
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
口水猫(465191936) 9:50:47
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
月苔河璞(360961410) 9:51:04
讨论一下各种距离算法用在什么情况下吧
口水猫(465191936) 9:51:34
是啊 这个在实际中很重要
网络上的尼采(813394698) 9:51:41
@η 关于那个 k 是最优的比较主观，有从结果稳定性来考虑的
口水猫(465191936) 9:51:48
现在刚好要完成这个调查的聚类分析
口水猫(465191936) 9:52:04
算法的原理比较好理解 但是实践中怎样用是个问题啊
月苔河璞(360961410) 9:53:31
还有 k 的选取是不是要更科学点啊
网络上的尼采(813394698) 9:54:32
k 的选择有很多方法，dp MDL 什么的
月苔河璞(360961410) 9:54:32
至少从理论上说我的这个 k 是最接近实际情况的
liyitan__ML<liyitan2144@163.com> 9:54:59
MDL 是啥？
网络上的尼采(813394698) 9:55:12
最短描述长度
liyitan__ML<liyitan2144@163.com> 9:55:24
可否科普一下？
网络上的尼采(813394698) 9:55:47
就是从压缩的角度来看 k-means
liyitan__ML<liyitan2144@163.com> 9:56:09
以及这里的 dp 又是指？

网络上的尼采(813394698) 9:57:05
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。
网络上的尼采(813394698) 9:58:11
我们继续，混合高斯模型
独孤圣者(303957511) 9:59:18
高斯过程，狄利克雷过程，可以简单介绍一下么？
牧云(1106207961) 9:59:29
jius
liyitan__ML<liyitan2144@163.com> 10:00:20
同样希望介绍一下，用于选参数的么？
网络上的尼采(813394698) 10:01:11
高斯过程讲第六章时再说，狄利克雷过程等讲完 PRML 还有时间读 MLAPP 时我来讲
牧云(1106207961) 10:01:41
这。。
独孤圣者(303957511) 10:01:46
MLAPP 全称？
独孤圣者(303957511) 10:01:55
尼采是博士还是老师啊？
网络上的尼采(813394698) 10:02:54
第二章我们说过，高斯分布有很多优点并且普遍存在，但对于复杂的分布表达能力差，于是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。
牧云(1106207961) 10:03:15
嗯
月苔河璞(360961410) 10:03:22
这是个思路
网络上的尼采(813394698) 10:03:39

ition of Gaussians in the form

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

GMM 的形式
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，zk=1,其他为 0
秦淮/sun 人家(76961223) 10:08:37
这是 em 经典的例子
网络上的尼采(813394698) 10:09:21

en that

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

zk=1 的概率的先验就是高斯分布前的那个系数

网络上的尼采(813394698) 10:11:10

yes theorem

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

这是 zk=1 概率的后验，由贝叶斯公式推导的，其实很好理解，如果如果没有 π_k 限制，数据点由哪个分布得出的概率大 zk=1 的期望就大，但前面还有一个系数限制，所以最后的期望形式是

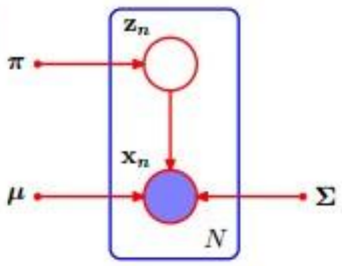
$$\frac{\pi_k \mathcal{N}(\mathbf{x}|\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)}{\sum_{j=1}^{K} \pi_j \mathcal{N}(\mathbf{x}|\boldsymbol{\mu}_j, \boldsymbol{\Sigma}_j)}.$$

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。

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 $n = 1, \ldots, N$.



GMM 的图表示
由于直接求解困难，这也是引入 EM 的原因。
我们可以试想下，如果隐变量 zn 是可以观测的，也就是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 zn 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。

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

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

网络上的尼采(813394698) 10:30:34
EM 对 GMM 的步骤
我们先对模型的参数初始化

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

E 步就是我们利用这些参数得 znk 的期望 这种形式我们前面已经提到了
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

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

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

高斯分布的好处就在这儿，可以直接得出来新的参数。
想飞的猪(425914818) 10:35:49
怎么得啊？
想飞的猪(425914818) 10:36:04
新的参数是什么意思？就是新的样本数据？
网络上的尼采(813394698) 10:36:33

然后不断迭代 E 步和 M 步直到满足收敛条件。
HEHE(61992090) 10:38:32
网络上的尼采是干什么的？
企业开发人员还是科研工作者
很牛的感觉
网络上的尼采(813394698) 10:38:59
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

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

单调递增的。

karnon(447457116) 10:39:42
用 EM 来解 GMM 其实是有问题的
karnon(447457116) 10:39:51
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。
karnon(447457116) 10:40:18
所以 EM 来解 GMM 其实就是在搞笑啊。。
月苔河璞(360961410) 10:40:44
参与跌代的是什么数据，是总体样本吗
monica(909117539) 10:41:09
EM 可以用在哪些地方，貌似很多地方都在用，究竟效果怎么样啊?
网络上的尼采(813394698) 10:41:36
我们再来看 EM 更一般的形式。

$$\ln p(\mathbf{X}|\boldsymbol{\theta}) =$$

这是我们的目标函数
网络上的尼采(813394698) 10:43:11
d function is given by

$$\ln p(\mathbf{X}|\boldsymbol{\theta}) = \ln \left\{ \sum_{\mathbf{Z}} p(\mathbf{X}, \mathbf{Z}|\boldsymbol{\theta}) \right\}.$$

加入隐藏变量可以写成这种形式

HEHE(61992090) 10:43:55
然后呢？
网络上的尼采(813394698) 10:45:43
先初始化模型的参数，由于隐藏变量无法观测到，我们用原来的参数来得到它的后验

. Our state of knowledge of the value
the posterior distribution $p(\mathbf{Z}|\mathbf{X}, \boldsymbol{\theta})$.

网络上的尼采(813394698) 10:46:22

nt parameter values θ
en by $p(\mathbf{Z}|\mathbf{X}, \boldsymbol{\theta}^{\text{old}})$. W
the complete-data log

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

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

M 步也就是求这个似然函数的 Q 函数的最优解

网络上的尼采(813394698) 10:50:37

$\ln p(\mathbf{X}|\boldsymbol{\pi}, \boldsymbol{\mu}, \boldsymbol{\Sigma})$

注意这个 Q 函数是完整数据包含隐变量的似然函数，不是

网络上的尼采(813394698) 10:51:57
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。
网络上的尼采(813394698) 10:53:27
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 $\boldsymbol{\theta}^{\text{old}}$.

9.3. An Alternative View of EM 441

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.

liyitan__ML<liyitan2144@163.com> 10:54:43

为何觉得用 EM 解 GMM “可笑”@karnon ？
网络上的尼采(813394698) 10:54:54
EM 算法只所以用途广泛就在于有潜在变量的场合都能用，并不局限于用在 GMM 上。
karnon(447457116) 10:56:05
最后会讲的，别急
HEHE(61992090) 10:56:23
Q（ ）
那个函数怎么来的？
网络上的尼采(813394698) 10:56:40

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

$$\boldsymbol{\mu}_k^{\text{new}} = \frac{1}{N_k}\sum_{n=1}^{N}\gamma(z_{nk})\mathbf{x}_n \quad (9.$$

$$\boldsymbol{\Sigma}_k^{\text{new}} = \frac{1}{N_k}\sum_{n=1}^{N}\gamma(z_{nk})\left(\mathbf{x}_n-\boldsymbol{\mu}_k^{\text{new}}\right)\left(\mathbf{x}_n-\boldsymbol{\mu}_k^{\text{new}}\right)^{\mathrm{T}} \quad (9.$$

$$\pi_k^{\text{new}} = \frac{N_k}{N} \quad (9.$$

其实我们回过头来看 GMM,
这些新参数就是 Q 函数的最优解

monica(909117539) 10:58:03
@Learner ，你别打岔，让尼采讲完先
HEHE(61992090) 10:58:19
哦
karnon(447457116) 10:58:27
EM，不自己花几小时看，是不可能弄懂的。
HEHE(61992090) 10:58:35
先听听吧
网络上的尼采(813394698) 11:00:20
有什么问题过会讨论，现在再思考下 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。

.40), becomes

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

hus we see that in this limit, maximizing the expected complete-data log li他们是等价的

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。

网络上的尼采(813394698) 11:05:36

Now let us consider a finite mixture of these distributions giv

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

混合形式

月苔河璞(360961410) 11:05:47

伯努利，先复习一下

月苔河璞(360961410) 11:06:40

01 分布

网络上的尼采(813394698) 11:06:44

model is given by

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

目标函数，log 里面同样有加和

网络上的尼采(813394698) 11:08:53

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

znk=1 的期望

网络上的尼采(813394698) 11:09:29

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

Q 函数

网络上的尼采(813394698) 11:10:02

上面是 E 步

网络上的尼采(813394698) 11:10:34

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

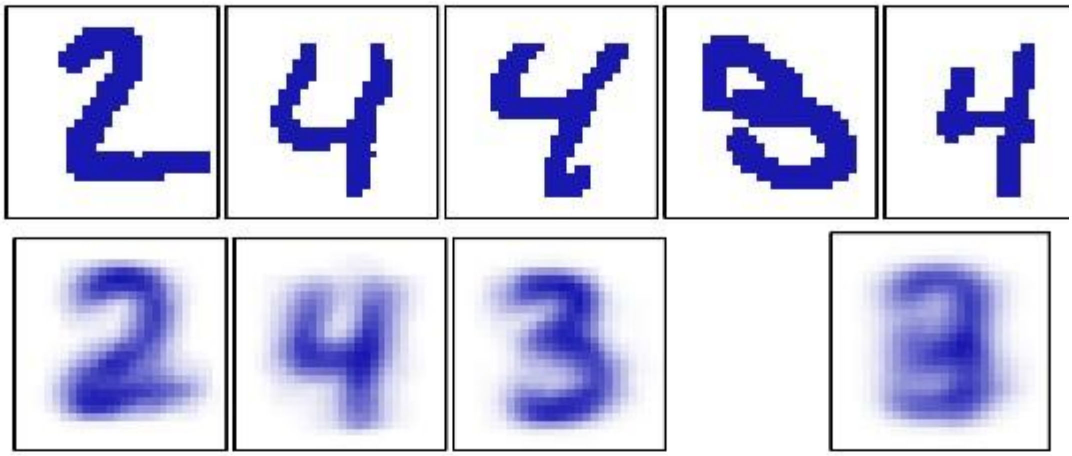
$$\overline{\mathbf{x}}_k = \frac{1}{N_k} \sum_{n=1}^{N} \gamma(z_{nk}) \mathbf{x}_n$$

M 步的解

网络上的尼采(813394698) 11:10:47

$\mu_k = \overline{\mathbf{x}}_k.$　　　$\pi_k = \frac{N_k}{N}$　这两个

网络上的尼采(813394698) 11:11:09
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

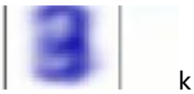


ure 9.10 Illustration of the Bernoulli mixture model in which the top row shows examples from the digits c

网络上的尼采(813394698) 11:12:25

 这是 3 个簇的代表

网络上的尼采(813394698) 11:12:53

 k=1 时簇的代表

liyitan__ML<liyitan2144@163.com> 11:13:47
不得已要离开一下，请一定要回答为何 EM 做 GMM 不合适的问题啊！
@karnon
网络上的尼采(813394698) 11:14:01
刚才有个图忘记发了
karnon(447457116) 11:14:36
EM 不能用来做 GMM 的问题在 vapnic 的书上有说
网络上的尼采(813394698) 11:15:36

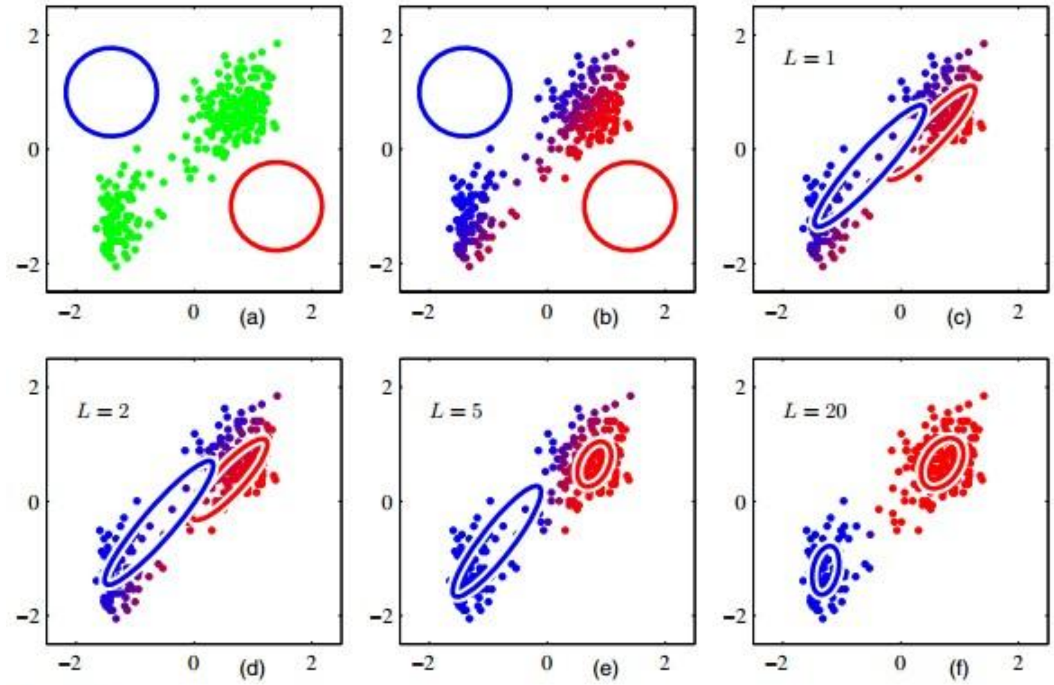


Figure 9.8 Illustration of the EM algorithm using the Old Faithful set as used for the illustration of the K-means algorithm in Figure 9.1. See the text for details.

说

明 k-means 是 em 算法特例，这与与前面 k-means 聚类的过程的图对应。

网络上的尼采(813394698) 11:16:41

最后说下 EM 算法为什么能收敛到似然函数的局部最优解

网络上的尼采(813394698) 11:18:04

$$p(\mathbf{X}|\boldsymbol{\theta}) =$$

我们的目标函数

网络上的尼采(813394698) 11:18:26

on that is given by

$$p(\mathbf{X}|\boldsymbol{\theta}) = \sum_{\mathbf{Z}} p(\mathbf{X}, \mathbf{Z}|\boldsymbol{\theta}).$$

引入潜在变量

定义隐藏变量的分布为 q(z)，目标函数可以表达为这种形式

erve that, for any choice of $q(\mathbf{Z})$, the following decomposition ho

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

where we have defined

$$\mathcal{L}(q, \boldsymbol{\theta}) = \sum_{\mathbf{Z}} q(\mathbf{Z}) \ln \left\{ \frac{p(\mathbf{X}, \mathbf{Z}|\boldsymbol{\theta})}{q(\mathbf{Z})} \right\}$$

$$\mathrm{KL}(q\|p) = -\sum_{\mathbf{Z}} q(\mathbf{Z}) \ln \left\{ \frac{p(\mathbf{Z}|\mathbf{X}, \boldsymbol{\theta})}{q(\mathbf{Z})} \right\}.$$

网络上的尼采(813394698) 11:22:03

其中 $\mathrm{KL}(q\|p)$ 是 $p(\mathbf{Z}|\mathbf{X},\boldsymbol{\theta})$ 与 $q(\mathbf{Z})$ 的 KL 距离

网络上的尼采(813394698) 11:22:37

$\mathcal{L}(q,\boldsymbol{\theta})$ 是 q(z)的泛函形式

网络上的尼采(813394698) 11:23:51

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。

网络上的尼采(813394698) 11:26:38

$\mathcal{L}(q,\boldsymbol{\theta})$ 与目标函数什么时候相等呢？其实就是 $\mathrm{KL}(q\|p)$ 等于 0

网络上的尼采(813394698) 11:27:35

也就是 q(z)与 z 的后验分布 $p(\mathbf{Z}|\mathbf{X},\boldsymbol{\theta})$ 相同时，这个时候就是 E 步，z 取它的期望时候。

网络上的尼采(813394698) 11:28:32

$\mathcal{L}(q,\boldsymbol{\theta})$ 与目标函数相等是为了取一个比较紧的 bound

网络上的尼采(813394698) 11:30:23

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

网络上的尼采(813394698) 11:30:33

$\mathrm{KL}(q\|p)$ 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。

网络上的尼采(813394698) 11:31:31

这个过程是使目标函数一直单调递增的。

网络上的尼采(813394698) 11:32:29

shown in Figure 9.13. If we substitute $q(\mathbf{Z}) = p(\mathbf{Z}|\mathbf{X},\boldsymbol{\theta}^{\text{old}})$ into (9.71), we see that, after the E step, the lower bound takes the form

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

网络上的尼采(813394698) 11:33:07

$\mathcal{L}(q,\boldsymbol{\theta})$ 与 Q 函数的关系

网络上的尼采(813394698) 11:33:56

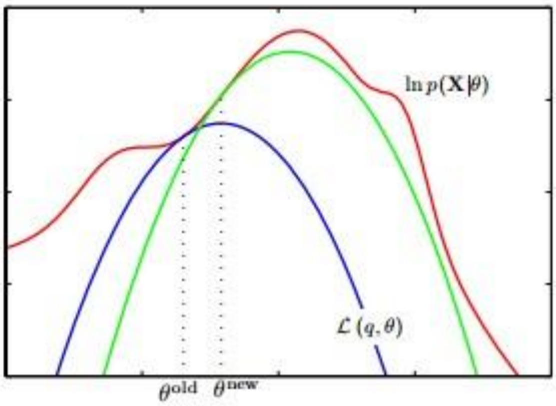
这也是我们为什么取完整数据的最大似然解的原因。

网络上的尼采(813394698) 11:34:49

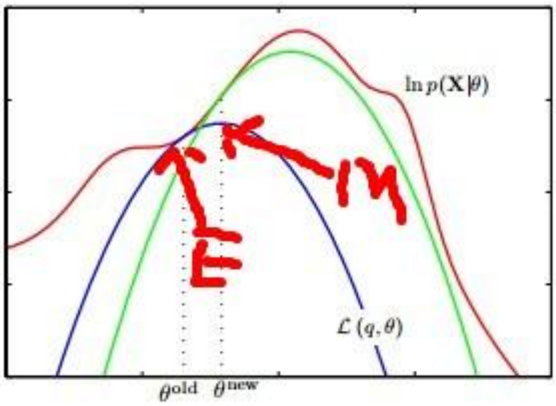
最后上一张非常形象的图，解释为什么 EM 能收敛到目标函数的局部最优。

网络上的尼采(813394698) 11:35:10

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.



网络上的尼采(813394698) 11:35:31
红的曲线是目标函数
网络上的尼采(813394698) 11:35:41
蓝的绿的是两步迭代
网络上的尼采(813394698) 11:36:05
咱们先看蓝的 e 步和 m 步
lth(46359905) 11:36:13
额
牧云(1106207961) 11:36:22
吃饭了
网络上的尼采(813394698) 11:38:14



monica(909117539) 11:39:03
嗯
网络上的尼采(813394698) 11:39:36

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
网络上的尼采(813394698) 11:40:07

M 步就是最大化 $\mathcal{L}(q,\boldsymbol{\theta})$
网络上的尼采(813394698) 11:42:38

绿线就是下一轮的迭代，可以看出目标函数一直是单调上升的，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，目标函数的曲线稍微变动下，EM 就收敛到局部最优了。
网络上的尼采(813394698) 11:43:36
好，今天就到这里，吃完饭再回来讨论交流吧。
monica(909117539) 11:43:51
辛苦辛苦
网络上的尼采(813394698) 11:44:02

HEHE(61992090) 11:44:21
辛苦了
最好
后面整理一下
麦穗(27633854) 11:44:39
辛苦辛苦
网络上的尼采(813394698) 11:44:52
下午再整理，你们继续交流吧。
牧云(1106207961) 11:45:38

月苔河璞(360961410) 11:46:54
辛苦了
karnon(447457116) 11:47:47

流不止(751679856) 11:49:04

巴川(53403506) 11:56:19

甲乙丙丁(328214796) 12:09:25

sunsusu(727128979) 12:18:08