机器学习（西瓜书）

注　解

（第 9 章　聚类）

https://blog.csdn.net/jbb0523

前言

经常听人说南大周老师所著的《机器学习》（以下统称为西瓜书）是一本入门教材，是一本科普性质的教科书。在该书第十次印刷之际，周老师在“如何使用本书”中也提到“这是一本入门级教科书”。然而，本人读起来却感觉该书远不止“科普”“入门”那么简单，书中的很多公式需要思考良久方能推导，很多概念需要反复咀嚼才能消化。边读边想着是不是应该将自己学习时遇到的一些知识难点的解析分享出来，以帮助更多的人入门。自己的确也随手做过一些笔记，但由于怀疑这仅是自己的个别现象，毕竟读书期间，思考更多的是如何使用单片机、DSP、ARM、FGPA 等，而这些基本是不需要推导任何公式的，因此作罢。偶然间在周老师的新浪微博看到如下对话：

此时方知，可能“读不懂”并不是个别现象。因此决定写一本“西瓜书注解”或者称为“西瓜书读书笔记”，对自己研读西瓜书时遇到的“台阶”进行解释和推导，以帮助更多的人能够更快地进入到这个领域。另外，近期越来越强地意识到，扎扎实实地推导一些基础算法的公式，无论是对于理解算法本身机理还是进行学术研究，都是非常有必要的。

自己会根据个人学习进度和研究需要按章发布，不知道能不能坚持写完，加油！

毕竟自己也是一名初学者，所以可能一些概念解释并不完整、一些公式推导并不优美，甚至会存在错误，这是不可避免的，不接受漫骂，但欢迎将问题反馈给我，共同学习进步！

（网盘链接：https://pan.baidu.com/s/1QtEiNnk8jMzmbs0KPBN-_w）

第 9 章目录

第 9 章 聚类.....1
9.1 聚类任务.....1
9.2 性能度量.....1
1、式(9.7)的解释.....1
2、式(9.8)的解释.....1
3、式(9.12)的解释.....1
9.3 距离计算.....1
1、欧氏距离和曼哈顿距离.....2
2、式(9.21)的解释.....2
9.4 原型聚类.....2
1、式(9.28)的解释.....3
2、式(9.29)的解释.....3
3、式(9.30)的解释.....3
4、式(9.31)的解释.....4
5、式(9.32)的解释.....4
6、式(9.33)的推导.....4
7、式(9.34)的推导.....6
8、式(9.35)的推导.....6
9、式(9.36)的解释.....8
10、式(9.37)的推导.....8
11、式(9.38)的推导.....8
12、图 9.6 的解释.....9
9.5 密度聚类.....9
1、密度直达、密度可达、密度相连.....10
2、图 9.9 的解释.....10
9.6 层次聚类.....11
9.7 本章小节.....11

第 9 章 聚类

到目前为止，前面章节介绍的方法都是针对监督学习(supervised learning)的，本章介绍的聚类(clustering)和下一章介绍的降维属于无监督学习(unsupervised learning)。

9.1 聚类任务

本节内容通俗易懂， 几乎不需要什么注解。

单词“cluster”既是动词也是名词，作为名词时翻译为“簇”，即聚类得到的子集；一般谈到“聚类”这个概念时对应其动名词形式“clustering”。

9.2 性能度量

本节给出了聚类性能度量的三种外部指标和两种内部指标，其中式(9.5)~式(9.7)是基于式(9.1)~式(9.4)导出的三种外部指标，而式(9.12)和式(9.13)是基于式(9.8)~式(9.11)导出的两种内部指标。读本节内容需要心里清楚的一点：本节给出的指标仅是该领域的前辈们定义的指标，在个人研究过程中可以根据需要自己定义，说不定就会被圈内同行广泛使用，然后以你的名字命名该指标写入教科书了呢^_^

1、式(9.7)的解释

将该式变形为

$$\mathrm{RI}=\frac{a+d}{m(m-1)/2}=\frac{a+d}{a+b+c+d}$$

因此RI肯定不大于 1。之所以$a+b+c+d=m(m-1)/2$，这是因为式(9.1)~式(9.4)遍历了所有$(\boldsymbol{x}_i,\boldsymbol{x}_j)$组合对$(i\neq j)$：其中$i=1$时$j$可以取 2 到$m$共$m-1$个值，$i=2$时$j$可以取 3 到$m$共$m-2$个值，……，$i=m-1$时$j$仅可以取$m$共 1 个值，因此$(\boldsymbol{x}_i,\boldsymbol{x}_j)$组合对的个数为从 1 到$m-1$求和，根据等差数列求和公式即得$m(m-1)/2$ 。

2、式(9.8)的解释

式中，$|C|$表示簇C中样本的个数，因此该式求和号之前为$(\boldsymbol{x}_i,\boldsymbol{x}_j)$组合对个数的倒数，而求和号是求簇$C$中样本间的距离之和，再除以$(\boldsymbol{x}_i,\boldsymbol{x}_j)$组合对个数，即簇内平均距离。

3、式(9.12)的解释

式中，k表示聚类结果中簇的个数。该式的DBI值越小越好，因为我们希望“物以类聚”，即同一簇的样本尽可能彼此相似，$\mathrm{avg}(C_i)$和$\mathrm{avg}(C_j)$越小越好；我们希望不同簇的样本尽可能不同，即$d_{\mathrm{cen}}(C_i,C_j)$越大越好。

勘误：第 25 次印刷将分母$d_{\mathrm{cen}}(\boldsymbol{\mu}_i,\boldsymbol{\mu}_j)$改为$d_{\mathrm{cen}}(C_i,C_j)$。

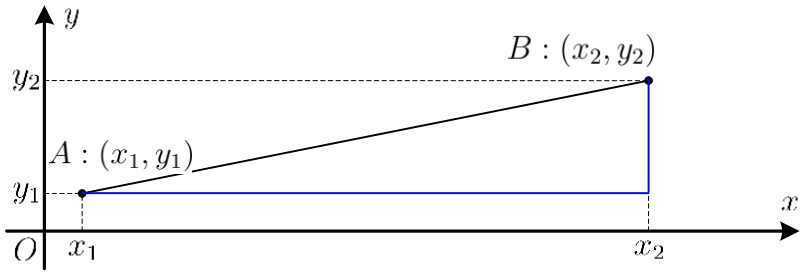
9.3 距离计算

距离计算在各种算法中都很常见，本节介绍的距离计算方式和 10.6 节介绍的马氏距离基本囊括了一般的距离计算方法。另外可能还会碰到 10.5 节的测地线距离。

本节有很多概念和名词很常见，比如本节开篇介绍的距离度量的四个基本性质、闵可夫斯基距离、欧氏距离、曼哈顿距离、切比雪夫距离（详见边注）、数值属性、离散属性、有序属性、无序属性、非度量距离等，注意对应的中文和英文。

1、欧氏距离和曼哈顿距离

当提及“距离”的概念，通常意味着从 A 到 B 的“最短”路径。下图显示出在二维平面上欧氏距离（黑色）和曼哈顿距离（蓝色）：



即 A 到 B 的欧氏距离是两点之间的直线最短路径（空间任意位置均可作为路径），而 A 到 B 的曼哈顿距离则是当只能沿平行各坐标轴方向行走时的最短路径。

更多有关曼哈顿距离的解释参见本章末尾第 224 页的小故事。

2、式(9.21)的解释

该式符号较为抽象，下面计算第 76 页表 4.1 西瓜数据集 2.0 属性根蒂上“蜷缩”和“稍蜷”两个离散值之间的距离。

此时，u为“根蒂”，a为属性根蒂上取值为“蜷缩”，b为属性根蒂上取值为“稍蜷”，根据边注，此时样本类别已知（好瓜/坏瓜），因此$k=2$。

从表 4.1 中可知，根蒂为蜷缩的样本共有 8 个（编号 1~5、编号 12、编号 16~17），即$m_{u,a}=8$，根蒂为稍蜷的样本共有 7 个（编号 6~9 和编号 13~15），即$m_{u,b}=7$；设$i=1$对应好瓜，$i=2$对应坏瓜，好瓜中根蒂为蜷缩的样本共有 5 个（编号 1~5），即$m_{u,a,1}=5$，好瓜中根蒂为稍蜷的样本共有 3 个（编号 6~8），即$m_{u,b,1}=3$，坏瓜中根蒂为蜷缩的样本共有 3 个（编号 12 和编号 16~17），即$m_{u,a,2}=3$，坏瓜中根蒂为稍蜷的样本共有 4 个（编号 9 和编号 13~15），即$m_{u,b,2}=4$，因此 VDM 距离为

$$\begin{aligned}\mathrm{VDM}_p(a,b) &= \left|\frac{m_{u,a,1}}{m_{u,a}}-\frac{m_{u,b,1}}{m_{u,b}}\right|^p+\left|\frac{m_{u,a,2}}{m_{u,a}}-\frac{m_{u,b,2}}{m_{u,b}}\right|^p \\ &= \left|\frac{5}{8}-\frac{3}{7}\right|^p+\left|\frac{3}{8}-\frac{4}{7}\right|^p\end{aligned}$$

9.4 原型聚类

本节介绍了三个原型聚类算法，其中k均值算法最为经典，几乎成为聚类的代名词，在 Matlab 中有 kmeans 函数供调用。学习向量量化看似监督聚类，但根据第 206 页的举例来看，似乎也主要是对同类别样本内部再进行聚类，虽然图 9.5 的聚类结果并不完全是每个簇中仅包含同一类样本。有关学习向量量化的应用，书中所有提及，具体还没在实际中遇到。

前两个聚类算法比较易懂，下面主要推导第三个聚类算法：高斯混合聚类。

1、式(9.28)的解释

该式就是多元高斯分布概率密度函数的定义式：

$$p(\boldsymbol{x}) = \frac{1}{(2\pi)^{\frac{n}{2}} |\boldsymbol{\Sigma}|^{\frac{1}{2}}} e^{-\frac{1}{2}(\boldsymbol{x}-\boldsymbol{\mu})^{\top}\boldsymbol{\Sigma}^{-1}(\boldsymbol{x}-\boldsymbol{\mu})}$$

对应到我们常见的一元高斯分布概率密度函数的定义式：

$$p(x) = \frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{(x-\mu)^2}{2\sigma^2}}$$

其中$\sqrt{2\pi} = (2\pi)^{\frac{1}{2}}$对应$(2\pi)^{\frac{n}{2}}$，$\sigma$对应$|\boldsymbol{\Sigma}|^{\frac{1}{2}}$，指数项中分母中的方差$\sigma^2$对应协方差矩阵$\boldsymbol{\Sigma}$，$\frac{(x-\mu)^2}{\sigma^2}$对应$(\boldsymbol{x}-\boldsymbol{\mu})^{\top}\boldsymbol{\Sigma}^{-1}(\boldsymbol{x}-\boldsymbol{\mu})$。

概率密度函数$p(\boldsymbol{x})$是$\boldsymbol{x}$的函数。其中对于某个特定的$\boldsymbol{x}$来说，函数值$p(\boldsymbol{x})$就是一个数，若$\boldsymbol{x}$的维度为 2，则可以将函数$p(\boldsymbol{x})$的图像可视化，是三维空间的一个曲面。类似于一元高斯分布$p(x)$与横轴$p(x)=0$之间的面积等于 1（即$\int p(x)\mathrm{d}x = 1$），$p(\boldsymbol{x})$曲面与平面$p(\boldsymbol{x}) = 0$之间的体积等于 1（即$\int p(\boldsymbol{x})\mathrm{d}\boldsymbol{x} = 1$）。

注意，西瓜书中后面将$p(\boldsymbol{x})$记为$p(\boldsymbol{x} \mid \boldsymbol{\mu}, \boldsymbol{\Sigma})$。

2、式(9.29)的解释

对于该式表达的高斯混合分布概率密度函数$p_{\mathcal{M}}(\boldsymbol{x})$，与式(9.28)中的$p(\boldsymbol{x})$不同的是，它由$k$个不同的多元高斯分布加权而来。具体来说，$p(\boldsymbol{x})$仅由参数$\boldsymbol{\mu}, \boldsymbol{\Sigma}$确定，而$p_{\mathcal{M}}(\boldsymbol{x})$由$k$个“混合系数”$\alpha_i$以及$k$组参数$\boldsymbol{\mu}_i, \boldsymbol{\Sigma}_i$确定。

在西瓜书中该式下方(P207 最后一段)中介绍了样本的生成过程，实际也反应了“混合系数”α_i的含义，即α_i为选择第i个混合成分的概率，或者反过来说，α_i为样本属于第i个混合成分的概率。重新描述一下样本生成过程，根据先验分布$\alpha_1, \alpha_2, \ldots, \alpha_k$选择其中一个高斯混合成分（即第$i$个高斯混合成分被选到的概率为$\alpha_i$），假设选到了第$i$个高斯混合成分，其参数为$\boldsymbol{\mu}_i, \boldsymbol{\Sigma}_i$; 然后根据概率密度函数$p(\boldsymbol{x} \mid \boldsymbol{\mu}_i, \boldsymbol{\Sigma}_i)$（即将式(9.28)中的$\boldsymbol{\mu}, \boldsymbol{\Sigma}$替换为$\boldsymbol{\mu}_i, \boldsymbol{\Sigma}_i$）进行采样生成样本$\boldsymbol{x}$。两个步骤的区别在于第 1 步选择高斯混合成分时是从k个之中选其一（相当于概率密度函数是离散的），而第 2 步生成样本时是从$\boldsymbol{x}$定义域中根据$p(\boldsymbol{x} \mid \boldsymbol{\mu}_i, \boldsymbol{\Sigma}_i)$选择其中一个样本，样本$\boldsymbol{x}$被选中的概率即为$p(\boldsymbol{x} \mid \boldsymbol{\mu}_i, \boldsymbol{\Sigma}_i)$。即第 1 步对应于离散型随机变量，第 2 步对应于连续型随机变量。

3、式(9.30)的解释

若由上述样本生成方式得到训练集$D = \{\boldsymbol{x}_1, \boldsymbol{x}_2, \ldots, \boldsymbol{x}_m\}$，现在的问题是对于给定样本$\boldsymbol{x}_j$，它是由哪个高斯混合成分生成的呢？该问题即求后验概率$p_{\mathcal{M}}(z_j \mid \boldsymbol{x}_j)$，其中$z_j \in \{1, 2, \ldots, k\}$。下面对式(9.30)进行推导。

对于任意样本，在不考虑样本本身之前（即先验），若瞎猜一下它由第i个高斯混合成分生成的概率$P(z_j = i)$，那么肯定按先验概率$\alpha_1, \alpha_2, \ldots, \alpha_k$进行猜测，即$P(z_j = i) = \alpha_i$。若考虑样本本身带来的信息（即后验），此时再猜一下它由第i个高斯混合成分生成的概率$p_{\mathcal{M}}(z_j = i \mid \boldsymbol{x}_j)$，根据贝叶斯公式，后验概率$p_{\mathcal{M}}(z_j = i \mid \boldsymbol{x}_j)$可写为

$$p_{\mathcal{M}}(z_j = i \mid \boldsymbol{x}_j) = \frac{P(z_j = i) \cdot p_{\mathcal{M}}(\boldsymbol{x}_j \mid z_j = i)}{p_{\mathcal{M}}(\boldsymbol{x}_j)}$$

分子第 1 项$P(z_j=i)=\alpha_i$；第 2 项即第i个高斯混合成分生成样本$\boldsymbol{x}_j$的概率$p(\boldsymbol{x}_j \mid \boldsymbol{\mu}_i, \boldsymbol{\Sigma}_i)$，根据式(9.28)将$\boldsymbol{x}, \boldsymbol{\mu}, \boldsymbol{\Sigma}$替换为$\boldsymbol{x}_j, \boldsymbol{\mu}_i, \boldsymbol{\Sigma}_i$即得；分母$p_{\mathcal{M}}(\boldsymbol{x}_j)$即为将$\boldsymbol{x}_j$代入式(9.29)即得。

注意，西瓜书中后面将$p_{\mathcal{M}}(z_j=i \mid \boldsymbol{x}_j)$记为$\gamma_{ji}$，其中$1 \leq j \leq m$，$1 \leq i \leq k$。

4、式(9.31)的解释

若将所有γ_{ji}组成一个矩阵Γ，其中γ_{ji}为第j行第i列的元素，矩阵Γ大小为$m \times k$，即

$$\Gamma=\begin{bmatrix} \gamma_{11} & \gamma_{12} & \cdots & \gamma_{1k} \\ \gamma_{21} & \gamma_{22} & \cdots & \gamma_{2k} \\ \vdots & \vdots & \ddots & \vdots \\ \gamma_{m1} & \gamma_{m2} & \cdots & \gamma_{mk} \end{bmatrix}_{m \times k}$$

其中m为训练集样本个数，k为高斯混合模型包含的混合模型个数。

可以看出，式(9.31)就是找出矩阵Γ第j行的所有k个元素中最大的那个元素的位置。维基百科中有符号“arg max”的解释：https://en.wikipedia.org/wiki/Arg_max，可以学习一下。

进一步说，式(9.31)就是最大后验概率。

5、式(9.32)的解释

对于训练集$D=\{\boldsymbol{x}_1, \boldsymbol{x}_2, \ldots, \boldsymbol{x}_m\}$，现在要把$m$个样本划分为$k$个簇，即认为训练集$D$的样本是根据$k$个不同的多元高斯分布加权而得的高斯混合模型生成的。

现在的问题是，k个不同的多元高斯分布的参数$\{(\boldsymbol{\mu}_i, \boldsymbol{\Sigma}_i) \mid 1 \leqslant i \leqslant k\}$及它们各自的权重$\alpha_1, \alpha_2, \ldots, \alpha_k$不知道，$m$个样本归到底属于哪个簇也不知道，该怎么办呢？

其实这跟k均值算法类似，开始时既不知道k个簇的均值向量，也不知道m个样本归到底属于哪个簇，最后我们采用了贪心策略，通过迭代优化来近似求解式(9.24)。

本节的高斯混合聚类求解方法与k均值算法，只是具体问题具体解法不同，从整体上来说，它们都应用了 7.6 节的期望最大化算法(EM 算法)。

具体来说，现假设已知式(9.30)的后验概率，此时即可通过式(9.31)知道m个样本归到底属于哪个簇，再来求解参数$\{(\alpha_i, \boldsymbol{\mu}_i, \boldsymbol{\Sigma}_i) \mid 1 \leqslant i \leqslant k\}$，怎么求解呢？对于每个样本$\boldsymbol{x}_j$来说，它出现的概率是$p_{\mathcal{M}}(\boldsymbol{x}_j)$，既然现在训练集$D$中确实出现了$\boldsymbol{x}_j$，我们当然希望待求解的参数$\{(\alpha_i, \boldsymbol{\mu}_i, \boldsymbol{\Sigma}_i) \mid 1 \leqslant i \leqslant k\}$能够使这种可能性$p_{\mathcal{M}}(\boldsymbol{x}_j)$最大；又因为我们假设$m$个样本是独立的，因此它们恰好一起出现的概率就是$\prod_{j=1}^{m} p_{\mathcal{M}}(\boldsymbol{x}_j)$，即所谓的似然函数；一般来说，连乘容易造成下溢（$m$个大于 0 小于 1 的数相乘，当$m$较大时，乘积会非常非常小，以致于计算机无法表达这么小的数，产生下溢），所以常用对数似然替代，即式(9.32)。

6、式(9.33)的推导

该式等号左侧即偏导$\frac{\partial LL(D)}{\partial \boldsymbol{\mu}_i}$，下面先推导$\frac{\partial LL(D)}{\partial \boldsymbol{\mu}_i}$的表达式。重写式(9.32)如下：

$$LL(D)=\sum_{j=1}^{m} \ln \left(\sum_{l=1}^{k} \alpha_l \cdot p(\boldsymbol{x}_j \mid \boldsymbol{\mu}_l, \boldsymbol{\Sigma}_l) \right)$$

这里将第 2 个求和号的求和变量由式(9.32)的i改为了l，这是为了避免与$p(\boldsymbol{x}_j \mid \boldsymbol{\mu}_i, \boldsymbol{\Sigma}_i)$中的变量$i$相混淆；再结合式(9.28)，重写$p(\boldsymbol{x}_j \mid \boldsymbol{\mu}_i, \boldsymbol{\Sigma}_i)$的表达式如下：

$$p(\boldsymbol{x}_j \mid \boldsymbol{\mu}_i, \boldsymbol{\Sigma}_i)=\frac{1}{(2\pi)^{\frac{n}{2}} |\boldsymbol{\Sigma}_i|^{\frac{1}{2}}} e^{-\frac{1}{2}(\boldsymbol{x}_j-\boldsymbol{\mu}_i)^{\top} \boldsymbol{\Sigma}_i^{-1} (\boldsymbol{x}_j-\boldsymbol{\mu}_i)}$$

接下来开始推导$\frac{\partial LL(D)}{\partial \boldsymbol{\mu}_i}$的表达式。根据链接求导规则

$$\frac{\partial LL(D)}{\partial \boldsymbol{\mu}_i} = \frac{\partial LL(D)}{\partial p(\boldsymbol{x}_j \mid \boldsymbol{\mu}_i, \boldsymbol{\Sigma}_i)} \cdot \frac{\partial p(\boldsymbol{x}_j \mid \boldsymbol{\mu}_i, \boldsymbol{\Sigma}_i)}{\partial \boldsymbol{\mu}_i}$$

第 1 部分很容易进行求导：

$$\begin{aligned}
\frac{\partial LL(D)}{\partial p(\boldsymbol{x}_j \mid \boldsymbol{\mu}_i, \boldsymbol{\Sigma}_i)} &= \frac{\partial \sum_{j=1}^{m} \ln\left(\sum_{l=1}^{k} \alpha_l \cdot p(\boldsymbol{x}_j \mid \boldsymbol{\mu}_l, \boldsymbol{\Sigma}_l)\right)}{\partial p(\boldsymbol{x}_j \mid \boldsymbol{\mu}_i, \boldsymbol{\Sigma}_i)} \\
&= \sum_{j=1}^{m} \frac{\partial \ln\left(\sum_{l=1}^{k} \alpha_l \cdot p(\boldsymbol{x}_j \mid \boldsymbol{\mu}_l, \boldsymbol{\Sigma}_l)\right)}{\partial p(\boldsymbol{x}_j \mid \boldsymbol{\mu}_i, \boldsymbol{\Sigma}_i)} \\
&= \sum_{j=1}^{m} \frac{\alpha_i}{\sum_{l=1}^{k} \alpha_l \cdot p(\boldsymbol{x}_j \mid \boldsymbol{\mu}_l, \boldsymbol{\Sigma}_l)}
\end{aligned}$$

第 2 部分求导略显复杂：

$$\begin{aligned}
\frac{\partial p(\boldsymbol{x}_j \mid \boldsymbol{\mu}_i, \boldsymbol{\Sigma}_i)}{\partial \boldsymbol{\mu}_i} &= \frac{\partial \frac{1}{(2\pi)^{\frac{n}{2}} |\boldsymbol{\Sigma}_i|^{\frac{1}{2}}} e^{-\frac{1}{2}(\boldsymbol{x}_j - \boldsymbol{\mu}_i)^\top \boldsymbol{\Sigma}_i^{-1} (\boldsymbol{x}_j - \boldsymbol{\mu}_i)}}{\partial \boldsymbol{\mu}_i} \\
&= \frac{1}{(2\pi)^{\frac{n}{2}} |\boldsymbol{\Sigma}_i|^{\frac{1}{2}}} \cdot \frac{\partial e^{-\frac{1}{2}(\boldsymbol{x}_j - \boldsymbol{\mu}_i)^\top \boldsymbol{\Sigma}_i^{-1} (\boldsymbol{x}_j - \boldsymbol{\mu}_i)}}{\partial \boldsymbol{\mu}_i}
\end{aligned}$$

上面仅把常数项拿出来，使求导形式看起来更直观一些；剩下的求导部分又是复合函数求导：

$$\frac{\partial e^{-\frac{1}{2}(\boldsymbol{x}_j - \boldsymbol{\mu}_i)^\top \boldsymbol{\Sigma}_i^{-1} (\boldsymbol{x}_j - \boldsymbol{\mu}_i)}}{\partial \boldsymbol{\mu}_i} = e^{-\frac{1}{2}(\boldsymbol{x}_j - \boldsymbol{\mu}_i)^\top \boldsymbol{\Sigma}_i^{-1} (\boldsymbol{x}_j - \boldsymbol{\mu}_i)} \cdot -\frac{1}{2} \frac{\partial (\boldsymbol{x}_j - \boldsymbol{\mu}_i)^\top \boldsymbol{\Sigma}_i^{-1} (\boldsymbol{x}_j - \boldsymbol{\mu}_i)}{\partial \boldsymbol{\mu}_i}$$

又因为协方差矩阵的逆矩阵$\boldsymbol{\Sigma}_i^{-1}$是对称阵，因此：

$$\begin{aligned}
-\frac{1}{2} \frac{\partial (\boldsymbol{x}_j - \boldsymbol{\mu}_i)^\top \boldsymbol{\Sigma}_i^{-1} (\boldsymbol{x}_j - \boldsymbol{\mu}_i)}{\partial \boldsymbol{\mu}_i} &= -\frac{1}{2} \cdot 2\boldsymbol{\Sigma}_i^{-1} (\boldsymbol{\mu}_i - \boldsymbol{x}_j) \\
&= \boldsymbol{\Sigma}_i^{-1} (\boldsymbol{x}_i - \boldsymbol{\mu}_j)
\end{aligned}$$

上式有关矩阵求导可以搜索《The Matrix Cookbook(Version: November 15, 2012)》（式 86）：
http://www2.imm.dtu.dk/pubdb/views/edoc_download.php/3274/pdf/imm3274.pdf
因此第 2 部分求导结果为

$$\begin{aligned}
\frac{\partial p(\boldsymbol{x}_j \mid \boldsymbol{\mu}_i, \boldsymbol{\Sigma}_i)}{\partial \boldsymbol{\mu}_i} &= \frac{1}{(2\pi)^{\frac{n}{2}} |\boldsymbol{\Sigma}_i|^{\frac{1}{2}}} e^{-\frac{1}{2}(\boldsymbol{x}_j - \boldsymbol{\mu}_i)^\top \boldsymbol{\Sigma}_i^{-1} (\boldsymbol{x}_j - \boldsymbol{\mu}_i)} \cdot \boldsymbol{\Sigma}_i^{-1} (\boldsymbol{x}_j - \boldsymbol{\mu}_i) \\
&= p(\boldsymbol{x}_j \mid \boldsymbol{\mu}_i, \boldsymbol{\Sigma}_i) \cdot \boldsymbol{\Sigma}_i^{-1} (\boldsymbol{x}_j - \boldsymbol{\mu}_i)
\end{aligned}$$

综上所述

$$\frac{\partial LL(D)}{\partial \boldsymbol{\mu}_i} = \sum_{j=1}^{m} \frac{\alpha_i}{\sum_{l=1}^{k} \alpha_l \cdot p(\boldsymbol{x}_j \mid \boldsymbol{\mu}_l, \boldsymbol{\Sigma}_l)} \cdot p(\boldsymbol{x}_j \mid \boldsymbol{\mu}_i, \boldsymbol{\Sigma}_i) \cdot \boldsymbol{\Sigma}_i^{-1} (\boldsymbol{x}_j - \boldsymbol{\mu}_i)$$

注意$\boldsymbol{\Sigma}_i^{-1}$对于求和变量$j$来说是常量，因此可以提到求和号外面，当令$\frac{\partial LL(D)}{\partial \boldsymbol{\mu}_i} = 0$时可以将

该常量略掉（即等号两边同时左乘以$\boldsymbol{\Sigma}_i$），即得式(9.33)：

$$\frac{\partial LL(D)}{\partial \boldsymbol{\mu}_i} = \sum_{j=1}^{m} \frac{\alpha_i \cdot p(\boldsymbol{x}_j \mid \boldsymbol{\mu}_i, \boldsymbol{\Sigma}_i)}{\sum_{l=1}^{k} \alpha_l \cdot p(\boldsymbol{x}_j \mid \boldsymbol{\mu}_l, \boldsymbol{\Sigma}_l)} (\boldsymbol{x}_j - \boldsymbol{\mu}_i) = 0$$

7、式(9.34)的推导

根据式(9.30)可知：

$$\gamma_{ji} = p_{\mathcal{M}}(z_j = i \mid \boldsymbol{x}_j) = \frac{\alpha_i \cdot p(\boldsymbol{x}_j \mid \boldsymbol{\mu}_i, \boldsymbol{\Sigma}_i)}{\sum_{l=1}^{k} \alpha_l \cdot p(\boldsymbol{x}_j \mid \boldsymbol{\mu}_l, \boldsymbol{\Sigma}_l)}$$

则式(9.33)可重写为

$$\sum_{j=1}^{m} \gamma_{ji} (\boldsymbol{x}_j - \boldsymbol{\mu}_i) = 0$$

移项，得

$$\sum_{j=1}^{m} \gamma_{ji} \boldsymbol{x}_j = \sum_{j=1}^{m} \gamma_{ji} \boldsymbol{\mu}_i = \boldsymbol{\mu}_i \cdot \sum_{j=1}^{m} \gamma_{ji}$$

第二个等号是因为$\boldsymbol{\mu}_i$对于求和变量j来说是常量，因此可以提到求和号外面；因此

$$\boldsymbol{\mu}_i = \frac{\sum\limits_{j=1}^{m} \gamma_{ji} \boldsymbol{x}_j}{\sum\limits_{j=1}^{m} \gamma_{ji}}$$

8、式(9.35)的推导

该式推导过程与(9.33)(9.34)推导过程基本相同，根据链接求导规则

$$\frac{\partial LL(D)}{\partial \boldsymbol{\Sigma}_i} = \frac{\partial LL(D)}{\partial p(\boldsymbol{x}_j \mid \boldsymbol{\mu}_i, \boldsymbol{\Sigma}_i)} \cdot \frac{\partial p(\boldsymbol{x}_j \mid \boldsymbol{\mu}_i, \boldsymbol{\Sigma}_i)}{\partial \boldsymbol{\Sigma}_i}$$

第 1 部分与式(9.33)推导过程一样，第 2 部分与式(9.33)的区别较大且较为复杂：

$$\begin{aligned}
\frac{\partial p(\boldsymbol{x}_j \mid \boldsymbol{\mu}_i, \boldsymbol{\Sigma}_i)}{\partial \boldsymbol{\Sigma}_i} &= \frac{\partial \frac{1}{(2\pi)^{\frac{n}{2}} |\boldsymbol{\Sigma}_i|^{\frac{1}{2}}} e^{-\frac{1}{2}(\boldsymbol{x}_j - \boldsymbol{\mu}_i)^\top \boldsymbol{\Sigma}_i^{-1} (\boldsymbol{x}_j - \boldsymbol{\mu}_i)}}{\partial \boldsymbol{\Sigma}_i} \\
&= \frac{\frac{\partial e^{-\frac{1}{2}(\boldsymbol{x}_j - \boldsymbol{\mu}_i)^\top \boldsymbol{\Sigma}_i^{-1} (\boldsymbol{x}_j - \boldsymbol{\mu}_i)}}{\partial \boldsymbol{\Sigma}_i} \cdot (2\pi)^{\frac{n}{2}} |\boldsymbol{\Sigma}_i|^{\frac{1}{2}} - e^{-\frac{1}{2}(\boldsymbol{x}_j - \boldsymbol{\mu}_i)^\top \boldsymbol{\Sigma}_i^{-1} (\boldsymbol{x}_j - \boldsymbol{\mu}_i)} \cdot \frac{\partial (2\pi)^{\frac{n}{2}} |\boldsymbol{\Sigma}_i|^{\frac{1}{2}}}{\partial \boldsymbol{\Sigma}_i}}{\left((2\pi)^{\frac{n}{2}} |\boldsymbol{\Sigma}_i|^{\frac{1}{2}}\right)^2}
\end{aligned}$$

上式看起来复杂，实际就是函数求导规则$\left(\frac{u}{v}\right)' = \frac{u'v - uv'}{v^2}$；下面先求分子中的两项求导：

$$\begin{aligned}
\frac{\partial e^{-\frac{1}{2}(\boldsymbol{x}_j - \boldsymbol{\mu}_i)^\top \boldsymbol{\Sigma}_i^{-1} (\boldsymbol{x}_j - \boldsymbol{\mu}_i)}}{\partial \boldsymbol{\Sigma}_i} &= e^{-\frac{1}{2}(\boldsymbol{x}_j - \boldsymbol{\mu}_i)^\top \boldsymbol{\Sigma}_i^{-1} (\boldsymbol{x}_j - \boldsymbol{\mu}_i)} \cdot -\frac{1}{2} \frac{\partial (\boldsymbol{x}_j - \boldsymbol{\mu}_i)^\top \boldsymbol{\Sigma}_i^{-1} (\boldsymbol{x}_j - \boldsymbol{\mu}_i)}{\partial \boldsymbol{\Sigma}_i} \\
&= e^{-\frac{1}{2}(\boldsymbol{x}_j - \boldsymbol{\mu}_i)^\top \boldsymbol{\Sigma}_i^{-1} (\boldsymbol{x}_j - \boldsymbol{\mu}_i)} \cdot \frac{1}{2} \boldsymbol{\Sigma}_i^{-\top} (\boldsymbol{x}_j - \boldsymbol{\mu}_i)(\boldsymbol{x}_j - \boldsymbol{\mu}_i)^\top \boldsymbol{\Sigma}_i^{-\top} \\
&= e^{-\frac{1}{2}(\boldsymbol{x}_j - \boldsymbol{\mu}_i)^\top \boldsymbol{\Sigma}_i^{-1} (\boldsymbol{x}_j - \boldsymbol{\mu}_i)} \cdot \frac{1}{2} \boldsymbol{\Sigma}_i^{-1} (\boldsymbol{x}_j - \boldsymbol{\mu}_i)(\boldsymbol{x}_j - \boldsymbol{\mu}_i)^\top \boldsymbol{\Sigma}_i^{-1}
\end{aligned}$$

其中第一个等号就是复合函数求导；第三个等号是因为$\boldsymbol{\Sigma}_i^{-1}$为对称阵，其中$\boldsymbol{\Sigma}_i^{-\top}$表示$\boldsymbol{\Sigma}_i^{-1}$的转置；第二个等号参见《The Matrix Cookbook(Version: November 15, 2012)》（式 61）。

$$\begin{aligned}\frac{\partial(2\pi)^{\frac{n}{2}}|\boldsymbol{\Sigma}_i|^{\frac{1}{2}}}{\partial\boldsymbol{\Sigma}_i}&=\frac{(2\pi)^{\frac{n}{2}}}{2}|\boldsymbol{\Sigma}_i|^{-\frac{1}{2}}\cdot\frac{\partial|\boldsymbol{\Sigma}_i|}{\partial\boldsymbol{\Sigma}_i}\\&=\frac{(2\pi)^{\frac{n}{2}}}{2}|\boldsymbol{\Sigma}_i|^{-\frac{1}{2}}\cdot|\boldsymbol{\Sigma}_i|\cdot\boldsymbol{\Sigma}_i^{-\top}\\&=\frac{(2\pi)^{\frac{n}{2}}}{2}|\boldsymbol{\Sigma}_i|^{\frac{1}{2}}\cdot\boldsymbol{\Sigma}_i^{-1}\end{aligned}$$

上式推导中，第一个等号就是运用了复合函数求导规则，其中$\frac{\partial(2\pi)^{\frac{n}{2}}|\boldsymbol{\Sigma}_i|^{\frac{1}{2}}}{\partial|\boldsymbol{\Sigma}_i|}=\frac{(2\pi)^{\frac{n}{2}}}{2}|\boldsymbol{\Sigma}_i|^{-\frac{1}{2}}$；第二个等号中的$\frac{\partial|\boldsymbol{\Sigma}_i|}{\partial\boldsymbol{\Sigma}_i}=|\boldsymbol{\Sigma}_i|\cdot\boldsymbol{\Sigma}_i^{-\top}$为行列式求导，参见《The Matrix Cookbook(Version: November 15, 2012)》（式 49），第三个等号是由于$\boldsymbol{\Sigma}_i^{-1}$为对称阵。

将分子中的两项求导结果代入

$$\begin{aligned}\frac{\partial p(\boldsymbol{x}_j\mid\boldsymbol{\mu}_i,\boldsymbol{\Sigma}_i)}{\partial\boldsymbol{\Sigma}_i}&=\frac{\frac{\partial e^{-\frac{1}{2}(\boldsymbol{x}_j-\boldsymbol{\mu}_i)^\top\boldsymbol{\Sigma}_i^{-1}(\boldsymbol{x}_j-\boldsymbol{\mu}_i)}}{\partial\boldsymbol{\Sigma}_i}\cdot(2\pi)^{\frac{n}{2}}|\boldsymbol{\Sigma}_i|^{\frac{1}{2}}-e^{-\frac{1}{2}(\boldsymbol{x}_j-\boldsymbol{\mu}_i)^\top\boldsymbol{\Sigma}_i^{-1}(\boldsymbol{x}_j-\boldsymbol{\mu}_i)}\cdot\frac{\partial(2\pi)^{\frac{n}{2}}|\boldsymbol{\Sigma}_i|^{\frac{1}{2}}}{\partial\boldsymbol{\Sigma}_i}}{\left((2\pi)^{\frac{n}{2}}|\boldsymbol{\Sigma}_i|^{\frac{1}{2}}\right)^2}\\&=e^{-\frac{1}{2}(\boldsymbol{x}_j-\boldsymbol{\mu}_i)^\top\boldsymbol{\Sigma}_i^{-1}(\boldsymbol{x}_j-\boldsymbol{\mu}_i)}\cdot\frac{\frac{1}{2}\boldsymbol{\Sigma}_i^{-1}(\boldsymbol{x}_j-\boldsymbol{\mu}_i)(\boldsymbol{x}_j-\boldsymbol{\mu}_i)^\top\boldsymbol{\Sigma}_i^{-1}\cdot(2\pi)^{\frac{n}{2}}|\boldsymbol{\Sigma}_i|^{\frac{1}{2}}-\frac{(2\pi)^{\frac{n}{2}}}{2}|\boldsymbol{\Sigma}_i|^{\frac{1}{2}}\cdot\boldsymbol{\Sigma}_i^{-1}}{\left((2\pi)^{\frac{n}{2}}|\boldsymbol{\Sigma}_i|^{\frac{1}{2}}\right)^2}\\&=e^{-\frac{1}{2}(\boldsymbol{x}_j-\boldsymbol{\mu}_i)^\top\boldsymbol{\Sigma}_i^{-1}(\boldsymbol{x}_j-\boldsymbol{\mu}_i)}\cdot\frac{\boldsymbol{\Sigma}_i^{-1}(\boldsymbol{x}_j-\boldsymbol{\mu}_i)(\boldsymbol{x}_j-\boldsymbol{\mu}_i)^\top-\boldsymbol{I}}{\left((2\pi)^{\frac{n}{2}}|\boldsymbol{\Sigma}_i|^{\frac{1}{2}}\right)^2}\cdot\frac{(2\pi)^{\frac{n}{2}}}{2}|\boldsymbol{\Sigma}_i|^{\frac{1}{2}}\cdot\boldsymbol{\Sigma}_i^{-1}\\&=\frac{1}{(2\pi)^{\frac{n}{2}}|\boldsymbol{\Sigma}_i|^{\frac{1}{2}}}e^{-\frac{1}{2}(\boldsymbol{x}_j-\boldsymbol{\mu}_i)^\top\boldsymbol{\Sigma}_i^{-1}(\boldsymbol{x}_j-\boldsymbol{\mu}_i)}\cdot\left(\boldsymbol{\Sigma}_i^{-1}(\boldsymbol{x}_j-\boldsymbol{\mu}_i)(\boldsymbol{x}_j-\boldsymbol{\mu}_i)^\top-\boldsymbol{I}\right)\cdot\frac{1}{2}\boldsymbol{\Sigma}_i^{-1}\\&=p(\boldsymbol{x}_j\mid\boldsymbol{\mu}_i,\boldsymbol{\Sigma}_i)\cdot\left(\boldsymbol{\Sigma}_i^{-1}(\boldsymbol{x}_j-\boldsymbol{\mu}_i)(\boldsymbol{x}_j-\boldsymbol{\mu}_i)^\top-\boldsymbol{I}\right)\cdot\frac{1}{2}\boldsymbol{\Sigma}_i^{-1}\end{aligned}$$

注意$\frac{(2\pi)^{\frac{n}{2}}}{2}|\boldsymbol{\Sigma}_i|^{\frac{1}{2}}$为一个数，矩阵$\boldsymbol{I}$为大小与协方差矩阵$\boldsymbol{\Sigma}_i$相同的单位阵。

综上所述

$$\begin{aligned}\frac{\partial LL(D)}{\partial\boldsymbol{\Sigma}_i}&=\sum_{j=1}^{m}\frac{\alpha_i\cdot p(\boldsymbol{x}_j\mid\boldsymbol{\mu}_i,\boldsymbol{\Sigma}_i)}{\sum_{l=1}^{k}\alpha_l\cdot p(\boldsymbol{x}_j\mid\boldsymbol{\mu}_l,\boldsymbol{\Sigma}_l)}\cdot\left(\boldsymbol{\Sigma}_i^{-1}(\boldsymbol{x}_j-\boldsymbol{\mu}_i)(\boldsymbol{x}_j-\boldsymbol{\mu}_i)^\top-\boldsymbol{I}\right)\cdot\frac{1}{2}\boldsymbol{\Sigma}_i^{-1}\\&=\sum_{j=1}^{m}\gamma_{ji}\cdot\left(\boldsymbol{\Sigma}_i^{-1}(\boldsymbol{x}_j-\boldsymbol{\mu}_i)(\boldsymbol{x}_j-\boldsymbol{\mu}_i)^\top-\boldsymbol{I}\right)\cdot\frac{1}{2}\boldsymbol{\Sigma}_i^{-1}\end{aligned}$$

当令$\frac{\partial LL(D)}{\partial\boldsymbol{\Sigma}_i}=0$时可以将该常量略掉（即等号两边同时右乘以$2\boldsymbol{\Sigma}_i$）：

$$\frac{\partial LL(D)}{\partial\boldsymbol{\Sigma}_i}=\sum_{j=1}^{m}\gamma_{ji}\cdot\left(\boldsymbol{\Sigma}_i^{-1}(\boldsymbol{x}_j-\boldsymbol{\mu}_i)(\boldsymbol{x}_j-\boldsymbol{\mu}_i)^\top-\boldsymbol{I}\right)=0$$

移项，得

$$\sum_{j=1}^{m}\gamma_{ji}\cdot\boldsymbol{\Sigma}_i^{-1}(\boldsymbol{x}_j-\boldsymbol{\mu}_i)(\boldsymbol{x}_j-\boldsymbol{\mu}_i)^\top=\sum_{j=1}^{m}\gamma_{ji}\boldsymbol{I}$$

两边同时左乘以$\boldsymbol{\Sigma}_i$，得

$$\sum_{j=1}^{m}\gamma_{ji}\cdot(\boldsymbol{x}_j-\boldsymbol{\mu}_i)(\boldsymbol{x}_j-\boldsymbol{\mu}_i)^\top=\sum_{j=1}^{m}\gamma_{ji}\boldsymbol{\Sigma}_i=\boldsymbol{\Sigma}_i\sum_{j=1}^{m}\gamma_{ji}$$

第二个等号是因为$\mathbf{\Sigma}_i$对于求和变量j来说是常量，因此可以提到求和号外面；因此

$$\mathbf{\Sigma}_i = \frac{\sum_{j=1}^{m} \gamma_{ji} \cdot (\boldsymbol{x}_j - \boldsymbol{\mu}_i)(\boldsymbol{x}_j - \boldsymbol{\mu}_i)^\top}{\sum_{j=1}^{m} \gamma_{ji}}$$

9、式(9.36)的解释

该式就是$LL(D)$添加了等式约束$\sum_{i=1}^{k} \alpha_i = 1$的拉格朗日形式。

有个疑问是除了等式约束，还有不等式约束$\alpha_i \geqslant 0$，该式如何保证呢？

10、式(9.37)的推导

重写式(9.32)如下：

$$LL(D) = \sum_{j=1}^{m} \ln \left(\sum_{l=1}^{k} \alpha_l \cdot p(\boldsymbol{x}_j \mid \boldsymbol{\mu}_l, \mathbf{\Sigma}_l) \right)$$

这里将第 2 个求和号的求和变量由式(9.32)的i改为了l，这是为了避免对α_i求导时与变量i相混淆。将式(9.36)中的两项分别对α_i求导，得

$$\begin{aligned}
\frac{\partial LL(D)}{\partial \alpha_i} &= \frac{\partial \sum_{j=1}^{m} \ln \left(\sum_{l=1}^{k} \alpha_l \cdot p(\boldsymbol{x}_j \mid \boldsymbol{\mu}_l, \mathbf{\Sigma}_l) \right)}{\partial \alpha_i} \\
&= \sum_{j=1}^{m} \frac{1}{\sum_{l=1}^{k} \alpha_l \cdot p(\boldsymbol{x}_j \mid \boldsymbol{\mu}_l, \mathbf{\Sigma}_l)} \cdot \frac{\partial \sum_{l=1}^{k} \alpha_l \cdot p(\boldsymbol{x}_j \mid \boldsymbol{\mu}_l, \mathbf{\Sigma}_l)}{\partial \alpha_i} \\
&= \sum_{j=1}^{m} \frac{1}{\sum_{l=1}^{k} \alpha_l \cdot p(\boldsymbol{x}_j \mid \boldsymbol{\mu}_l, \mathbf{\Sigma}_l)} \cdot p(\boldsymbol{x}_j \mid \boldsymbol{\mu}_i, \mathbf{\Sigma}_i)
\end{aligned}$$

$$\frac{\partial \left(\sum_{l=1}^{k} \alpha_l - 1 \right)}{\partial \alpha_i} = \frac{\partial \left(\alpha_1 + \alpha_2 + \ldots + \alpha_i + \ldots + \alpha_k - 1 \right)}{\partial \alpha_i} = 1$$

综合两项求导结果，并令导数等于零即得式(9.37)。

11、式(9.38)的推导

注意，在西瓜书第 14 次印刷中式(9.38)上方的一行话进行了勘误：“两边同乘以α_i，对所有**混合成分**求和可知$\lambda = -m$”，将原来的“样本”修改为“混合成分”。

对式(9.37)两边同乘以α_i，得

$$\sum_{j=1}^{m} \frac{\alpha_i p(\boldsymbol{x}_j \mid \boldsymbol{\mu}_i, \mathbf{\Sigma}_i)}{\sum_{l=1}^{k} \alpha_l \cdot p(\boldsymbol{x}_j \mid \boldsymbol{\mu}_l, \mathbf{\Sigma}_l)} + \lambda \alpha_i = 0$$

结合式(9.30)发现，求和号内即为后验概率γ_{ji}，即

$$\sum_{j=1}^{m} \gamma_{ji} + \lambda \alpha_i = 0$$

对所有**混合成分**求和，得

$$\sum_{i=1}^{k}\sum_{j=1}^{m}\gamma_{ji}+\sum_{i=1}^{k}\lambda\alpha_i=0$$

注意$\sum_{i=1}^{k}\alpha_i=1$，因此$\sum_{i=1}^{k}\lambda\alpha_i=\lambda\sum_{i=1}^{k}\alpha_i=\lambda$，根据式(9.30)中$\gamma_{ji}$表达式可知

$$\begin{aligned}\sum_{i=1}^{k}\gamma_{ji}&=\sum_{i=1}^{k}\frac{\alpha_i\cdot p(\boldsymbol{x}_j\mid\boldsymbol{\mu}_i,\boldsymbol{\Sigma}_i)}{\sum_{l=1}^{k}\alpha_l\cdot p(\boldsymbol{x}_j\mid\boldsymbol{\mu}_l,\boldsymbol{\Sigma}_l)}\\&=\frac{\sum_{i=1}^{k}\alpha_i\cdot p(\boldsymbol{x}_j\mid\boldsymbol{\mu}_i,\boldsymbol{\Sigma}_i)}{\sum_{l=1}^{k}\alpha_l\cdot p(\boldsymbol{x}_j\mid\boldsymbol{\mu}_l,\boldsymbol{\Sigma}_l)}\\&=1\end{aligned}$$

再结合加法满足交换律，所以

$$\sum_{i=1}^{k}\sum_{j=1}^{m}\gamma_{ji}=\sum_{j=1}^{m}\sum_{i=1}^{k}\gamma_{ji}=\sum_{j=1}^{m}1=m$$

将以上分析结果代入$\sum_{i=1}^{k}\sum_{j=1}^{m}\gamma_{ji}+\sum_{i=1}^{k}\lambda\alpha_i=0$，移项即得$\lambda=-m$。将此结果代入$\sum_{j=1}^{m}\gamma_{ji}+\lambda\alpha_i=0$，即$\sum_{j=1}^{m}\gamma_{ji}-m\alpha_i=0$，整理即得式(9.38)。

12、图 9.6 的解释

第 1 行初始化参数，本页接下来的例子是按如下策略初始化的：混合系数$\alpha_i=\frac{1}{k}$；任选训练集中的k个样本分别初始化k个均值向量$\boldsymbol{\mu}_i(1\leqslant i\leqslant k)$；使用对角元素为0.1的对角阵初始化$k$个协方差矩阵$\boldsymbol{\Sigma}_i(1\leqslant i\leqslant k)$。

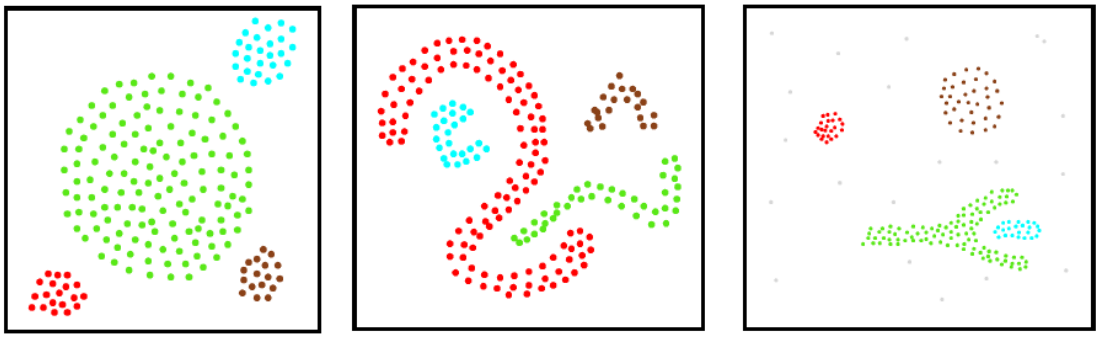
第 3~5 行根据式(9.30)计算共$m\times k$个γ_{ji}。

第 6~10 行分别根据式(9.34)、式(9.35)、式(9.38)使用刚刚计算得到的γ_{ji}更新均值向量、协方差矩阵、混合系数；注意第 8 行计算协方差矩阵时使用的是第 7 行计算得到的均值向量，这并没错，因为协方差矩阵$\boldsymbol{\Sigma}_i'$与均值向量$\boldsymbol{\mu}_i'$是对应的，而非$\boldsymbol{\mu}_i$；第 7 行的$\boldsymbol{\mu}_i'$在第 8 行使用之后会在下一轮迭代中第 4 行计算γ_{ji}再次使用。

整体来说，第 2~12 行就是一个 EM 算法的具体使用例子，学习完 7.6 节 EM 算法可能根本无法理解其思想。此例中有两组变量，分别是γ_{ji}和$(\alpha_i,\boldsymbol{\mu}_i,\boldsymbol{\Sigma}_i)$，它们之间相互影响，但都是未知的，因此 EM 算法就有了用武之地：初始化其中一组变量$(\alpha_i,\boldsymbol{\mu}_i,\boldsymbol{\Sigma}_i)$，然后计算$\gamma_{ji}$；再根据$\gamma_{ji}$根据最大似然推导出的公式更新$(\alpha_i,\boldsymbol{\mu}_i,\boldsymbol{\Sigma}_i)$，反复迭代，直到满足停止条件。

9.5 密度聚类

本节介绍的 DBSCAN 算法并不难懂，只是本节在最后举例时并没有说清楚密度聚类算法与前面原型聚类算法的区别，当然这也可能是作者有意为之，因为在第 220 页本章习题 9.7 题就提到了“凸聚类”的概念。具体来说，前面介绍的聚类算法只能产生“凸聚类”，而本节介绍的 DBSCAN 则能产生“非凸聚类”，其本质原因，个人感觉在于聚类时使用的距离度量，k均值算法使用欧氏距离，而 DBSCAN 使用类似于测地线距离（只是类似，并不相同，测地线距离参见 10.5 节），因此可以产生如下聚类结果（中间为典型的非凸聚类）：



注意：虽然左图为 “凸聚类”（四个簇都有一个凸包），但k均值算法却无法产生此结果，因为最大的簇太大了，其外围样本与另三个小簇的中心之间的距离更近，因此中间最大的簇肯定会被k均值算法划分到不同的簇之中，这显然不是我们希望的结果。

密度聚类算法可以产生任意形状的簇，不需要事先指定聚类个数k，并且对噪声鲁棒。

1、密度直达、密度可达、密度相连

$\boldsymbol{x}_j$由$\boldsymbol{x}_i$密度直达，该概念最易理解，但要特别注意：密度直达除了要求$\boldsymbol{x}_j$位于$\boldsymbol{x}_i$的ϵ-领域的条件之外，还额外要求$\boldsymbol{x}_i$是核心对象；ϵ-领域满足对称性，但$\boldsymbol{x}_j$不一定为核心对象，因此密度直达关系通常不满足对称性。

$\boldsymbol{x}_j$由$\boldsymbol{x}_i$密度可达，该概念基于密度直达，因此样本序列$\boldsymbol{p}_1, \boldsymbol{p}_2, \ldots, \boldsymbol{p}_n$中除了$\boldsymbol{p}_n = \boldsymbol{x}_j$之外，其余样本均为核心对象（当然包括$\boldsymbol{p}_1 = \boldsymbol{x}_i$），所以同理，一般不满足对称性。

以上两个概念中，若$\boldsymbol{x}_j$为核心对象，已知$\boldsymbol{x}_j$由$\boldsymbol{x}_i$密度直达/可达，则$\boldsymbol{x}_i$由$\boldsymbol{x}_j$密度直达/可达，即满足对称性（也就是说，核心对象之间的密度直达/可达满足对称性）。

$\boldsymbol{x}_i$与$\boldsymbol{x}_j$密度相连，不要求$\boldsymbol{x}_i$与$\boldsymbol{x}_j$为核心对象，所以满足对称性。

2、图 9.9 的解释

在第 1~7 行中，算法先根据给定的邻域参数$(\epsilon, MinPts)$找出所有核心对象，并存于集合Ω之中；第 4 行的 **if** 判断语句即在判别$\boldsymbol{x}_j$是否为核心对象。

在第 10~24 行中，以任一核心对象为出发点（由第 12 行实现），找出其密度可达的样本生成聚类簇（由第 14~21 行实现），直到所有核心对象被访问过为止（由第 10 行和第 23 行配合实现）。具体来说：

其中第 14~21 行 **while** 循环中的 **if** 判断语句（第 16 行）在第一次循环时一定为真（因为Q在第 12 行初始化为某核心对象），此时会往队列Q中加入$\boldsymbol{q}$密度直达的样本（已知$\boldsymbol{q}$为核心对象，$\boldsymbol{q}$的ϵ-领域中的样本即为$\boldsymbol{q}$密度直达的），队列遵循先进先出规则，接下来的循环将依次判别$\boldsymbol{q}$的ϵ-领域中的样本是否为核心对象（第 16 行），若为核心对象，则将密度直达的样本（ϵ-领域中的样本）加入Q。根据密度可达的概念，**while** 循环中的 **if** 判断语句（第 16 行）找出的核心对象之间一定是相互密度可达的，非核心对象一定是密度相连的。

第 14~21 行 **while** 循环每跳出一次，即生成一个聚类簇。每次生成聚类簇之前，会记录当前未访问过样本集合（第 11 行$\Gamma_{\text{old}} = \Gamma$），然后当前要生成的聚类簇每决定录取一个样本后会将该样本从Γ去除（第 13 行和第 19 行），因此第 14~21 行 **while** 循环每跳出一次后，Γ_{old}与Γ差别即为聚类簇的样本成员（第 22 行），并将该聚类簇中的核心对象从第 1~7 行生成的核心对象集合Ω中去除。

符号“$\setminus$”为集合求差，例如集合$A = \{a, b, c, d, e, f\}, B = \{a, d, f, g, h\}$，则$A \setminus B$为$A \setminus B = \{b, c, e\}$，即将$A, B$所有相同元素从$A$中去除。

9.6 层次聚类

本节主要介绍了层次聚类的代表算法 AGNES。

式(9.41)~(9.43)介绍了三种距离计算方式，这与 9.3 节中介绍的距离不同之处在于，此三种距离计算面向集合之间，而 9.3 节的距离则面向两点之间。正如第 215 页左上边注所示，集合间的距离计算常采用豪斯多夫距离(Hausdorff distance)，有关该距离的更多介绍可以参见 http://cgm.cs.mcgill.ca/~godfried/teaching/cg-projects/98/normand/main.html，网上搜索到的中文资料大多翻译自该英文网页。

算法 AGNES 很简单，就是不断重复执行合并距离最近的两个聚类簇。图 9.11 为具体实现方法，核心就是在合并两个聚类簇后更新距离矩阵（第 11-23 行），之所以看起来复杂，是因为该实现只更新原先距离矩阵中发生变化的行和列，因此需要为此做一些调整。

在第 1-9 行，算法先对仅含一个样本的初始聚类簇和相应的距离矩阵进行初始化。注意，距离矩阵中，第i行为聚类簇C_i到各聚类簇的距离，第i列为各聚类簇到聚类簇C_i的距离，由第 7 行可知，距离矩阵为对称矩阵，即使用的集合间的距离计算方法满足对称性。

第 17 行删除距离矩阵M的第j^*行与第j^*列，因为聚类簇C_{j^*}已经合并至C_{i^*}了；

第 18-21 行更新距离矩阵M的第i^*行与第i^*列，因为此时的聚类簇C_{i^*}已经合并了C_{j^*}，因此与其余聚类簇之间的距离都发生了变化，需要更新。

9.7 本章小节

本章介绍无监督学习中最典型的任务：聚类。正如书中第 217 页阅读材料说到：聚类也许是机器学习中“新算法”出现最多、最快的领域。因此本章勉强采用了“列举式”的叙述方式，相较于其它各章给出了更多的算法描述。

除了本章介绍的k均值算法、学习向量量化、高斯混合聚类，以及 DBSCAN 和 AGNES 之外，个人感觉阅读材料中提到的k-modes 与谱聚类也应该了解一下。

k-modes 针对离散属性的样本聚类，实际上就是替换了k均值算法的距离计算方式。k均值算法一般使用欧氏距离，但该距离仅针对连续属性；而当属性为离散值时，如第 76 页表 4.1 西瓜数据集 2.0（去除最后一列“好瓜”），假设要计算编号 1 和编号 2 之间的距离，式(9.21)给出的 VDM 当然是一种备选，但k-modes 使用了更简单的办法：若属性值相同，则距离为 0，若属性不同，则距离为 1，因此编号 1 和编号 2 之间的距离为 2，因为它们只有在色泽和敲声两个属性上面取值不同。

谱聚类是一种基于图论的聚类方法。简单来说将每个样本看作无向图的顶点，各顶点之间的某种关联性度量作为无向图的边，常用的度量方式是第 301 页的式(13.11)，聚类任务要将无向图分割成若干个簇，即无向图的部分边要被去除，谱聚类试图使去除的边的权重之和最小，而这实际上类似于图分割(Graph Partition)。有关谱聚类的详细介绍可参见参考文献[von Luxburg, 2007]或参见博客园两篇博客：https://www.cnblogs.com/pinard/p/6221564.html 和 https://www.cnblogs.com/Leo_wl/p/3156049.html。

George Karypis (http://glaros.dtc.umn.edu/)是图分割的知名学者，其开发的软件包 METIS 被广泛使用（在西瓜书作者所著的集成学习英文专著《Ensemble Methods: Foundations and Algorithms》的第 7.4 节，讲述基于图方法的聚类集成算法，其中就提到了 METIS 工具包中的 hMETIS）；但 METIS 并不容易琢磨明白，个人还发现一个非常简单易懂的 MATLAB 图分割程序 grPartition (https://www.ece.ucsb.edu/~hespanha/software/grPartition.html)，从其分割结果可以看出，图分割本身也是一种聚类。