附录

1.1 PCA

PCA 算法的公式为：

$$z = \mathrm{w}x$$

其中为x原始数据，w为变换矩阵，z为降维后的矩阵。

1.1.1 基于最大投影方差

在降维过程中，我们想的是能够在维度降低的保留越多的信息越好，这可以由方差来量化，方差越大，一组数据所包含的信息越多。如下图中，将左边的 L 做为新坐标比用 L'更好。

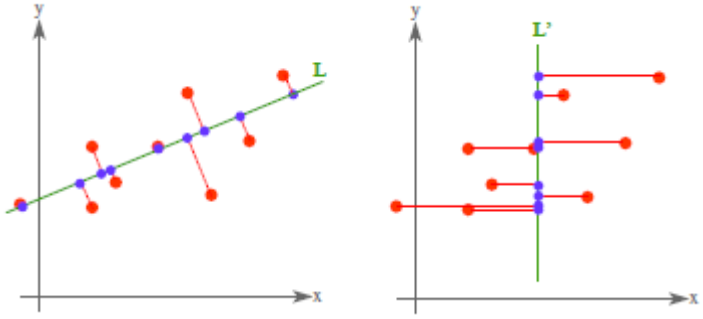


图 1 基于最大投影方差的 PCA

首先考虑降至一维：

$$Var(z_1) = \sum_{z_1} (z_1 - \overline{z_1})^2$$

$$s.t. \|w^1\|_2 = 1$$

其中z_1表示数据降至一维，w^1表示w第一行权重。

P.s.为什么这里的$\|\mathrm{w}^1\|_2 = 1$?

这里的w可以看作x投影到新空间里的一组基，基的模是 1；

$$Var(z_1) = \sum_{z_1} (z_1 - \overline{z_1})^2 = \sum_{z_1} (w^1 x - w^1 \overline{x_1})^2 = \sum_{z_1} [w^1 (x - \overline{x_1})]^2$$

若$\boldsymbol{a}$，$\boldsymbol{b}$为向量，则有$(\boldsymbol{a} \cdot \boldsymbol{b})^2 = (\boldsymbol{a}^T \boldsymbol{b})^2 = \boldsymbol{a}^T \boldsymbol{b} \boldsymbol{a}^T \boldsymbol{b} = \boldsymbol{a}^T \boldsymbol{b} (\boldsymbol{a}^T \cdot \boldsymbol{b})^T = \boldsymbol{a}^T \boldsymbol{b} \boldsymbol{b}^T \boldsymbol{a}$

所以

$$Var(z_1) = {w^1}^T \sum (x - \overline{x_1})(x - \overline{x_1})^T w^1 = {w^1}^T cov(x) w^1$$

$$s.t. \|w^1\|_2 = 1$$

由拉格朗日乘数法可以解到上式的极小值。

$$f(w^1) = {w^1}^T cov(x) w^1 + \lambda(\|w^1\|_2 - 1)$$

$$\begin{cases} \dfrac{\partial f(w^1)}{\partial \lambda} = \|w^1\|_2 - 1 = 0 \\ \dfrac{\partial f(w^1)}{\partial w^1} = cov(x)w^1 + \lambda w^1 = 0 \end{cases}$$

解得：

$$cov(x)w^1 = -\lambda w^1$$

令$\alpha = -\lambda$，则

$$cov(x)w^1 = \alpha w^1$$

明显可以看出，α为矩阵cov(x)特征值，w^1为其对应的特征向量，代入原式中得到：

$$Var(z_1) = {w^1}^T cov(x)w^1 = {w^1}^T \alpha w^1 = \alpha$$

所以，当α最大时，即cov(x)的特征值最大时，得到的z_1方差最大，且w^1就是最大特征值对应的特征向量。

若要降至n维，只需依次取w^2，w^3……注意$\|w^i\|_2 = 1$，且要与前面的$w^{1\sim i-1}$正交（标准正交基）。

1.1.2 基于最小投影距离（最小平方误差）：

从求解直线的思路出发，容易想到数学中的线性回归问题，其目标也是求解一个线性函数使得对应直线能够更好的拟合样本点集合。以这个思路为指导，在高维空间中，我们实际上是要找到一个超平面，使得数据点到这个超平面的距离平方和最小，如下图

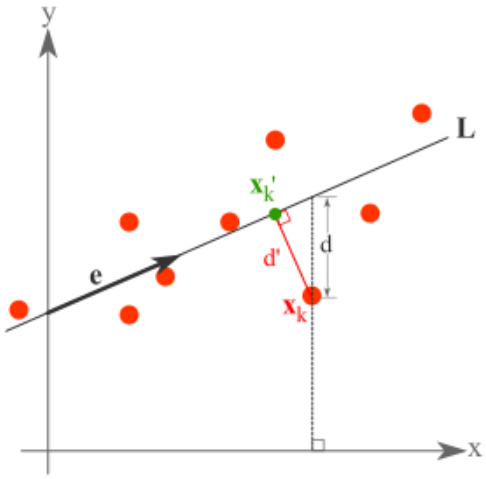


图 2 基于最小投影距离的 PCA

假设m个n维数据$(x^{(1)}, x^{(2)}, \ldots, x^{(m)})$都已经进行了中心化，即$\sum_{i=1}^{m} x^{(i)} = 0$，经过投影变换后得到的新坐标系为$\{w_1, w_2, \ldots, w_n\}$，其中$w$是标准正交基，

即$\| w \|_2 = 1, w_i^T w_j = 0.$

如果将数据从n维降到n'维（低维坐标系），即丢弃新坐标中的部分坐标，则样本点在n'维坐标系中的投影为：

$$z^{(i)} = (z_1^i, z_2^i, \dots, z_{n'}^i)^T$$

其中，$z_j^i = w_j^T x^{(i)}$是$x^{(i)}$在低维坐标系里第 j 维的坐标。

用$z^{(i)}$来恢复$x^{(i)}$，则得到的恢复数据$\bar{x}^{(i)} = \sum_{j=1}^{n'} z_j^{(i)} w_j = Wz^{(i)}$，其中 W 为标准正交基组成的矩阵。

现在考虑整个样本集，要使所有样本到这个超平面的距离最近，即最小化下式：

$$\sum_{i=1}^{m} \left\| \bar{x}^{(i)} - x^{(i)} \right\|_2^2$$

整理得

$$\sum_{i=1}^{m} \left\| \bar{x}^{(i)} - x^{(i)} \right\|_2^2 = \sum_{i=1}^{m} \| Wz^{(i)} - x^{(i)} \|_2^2$$

$$= \sum_{i=1}^{m} (Wz^{(i)})^T Wz^{(i)} - 2\sum_{i=1}^{m} (Wz^{(i)})^T x^{(i)} + \sum_{i=1}^{m} (x^{(i)})^T x^{(i)}$$

$$= \sum_{i=1}^{m} (z^{(i)})^T z^{(i)} - 2\sum_{i=1}^{m} (z^{(i)})^T W^T x^{(i)} + \sum_{i=1}^{m} (x^{(i)})^T x^{(i)}$$

$$= \sum_{i=1}^{m} (z^{(i)})^T z^{(i)} - 2\sum_{i=1}^{m} (z^{(i)})^T z^{(i)} + \sum_{i=1}^{m} (x^{(i)})^T x^{(i)}$$

$$= -\mathrm{tr}(W^T (\sum_{i=1}^{m} x^{(i)} (x^{(i)})^T) W) + \sum_{i=1}^{m} (x^{(i)})^T x^{(i)}$$

$$= -\sum_{i=1}^{m} (z^{(i)})^T z^{(i)} + \sum_{i=1}^{m} (x^{(i)})^T x^{(i)}$$

$$= -\mathrm{tr}(W^T X X^T W) + \sum_{i=1}^{m} (x^{(i)})^T x^{(i)}$$

则最小化上式等价于：

$$argmin_W -\mathrm{tr}\,(W^T X X^T W) \;\; s.t. W^T W = I$$

利用拉格朗日乘数法可以得到：

$$J(W) = -\mathrm{tr}(W^T X X^T W + \lambda(W^T W - I))$$

对 W 求导有$-XX^T W + \lambda W = 0$, 整理即为：

$$XX^T W = (-\lambda) W$$

则 W 为XX^T的特征向量组成的矩阵。和上一节基于最大投影距离的推导结果一致。

1.1.3 PCA 具体操作步骤：

（1） 去除平均值，让每一维特征减去各自特征的平均值
Why？可参考 http://sofasofa.io/forum_main_post.php?postid=1000375
以及步骤末尾博客。
（2） 计算样本协方差矩阵
（3） 计算协方差矩阵的特征值与特征向量
（4） 将特征值从大到小进行排序
（5） 选择最大的 K 个特征值，对应的特征向量
（6） 将数据转换到 K 个特征向量构建的新空间中
具体步骤详解及原因可见
https://blog.csdn.net/lanyuelvyun/article/details/82384179

以下为使用实验室数据进行的 PCA 处理：

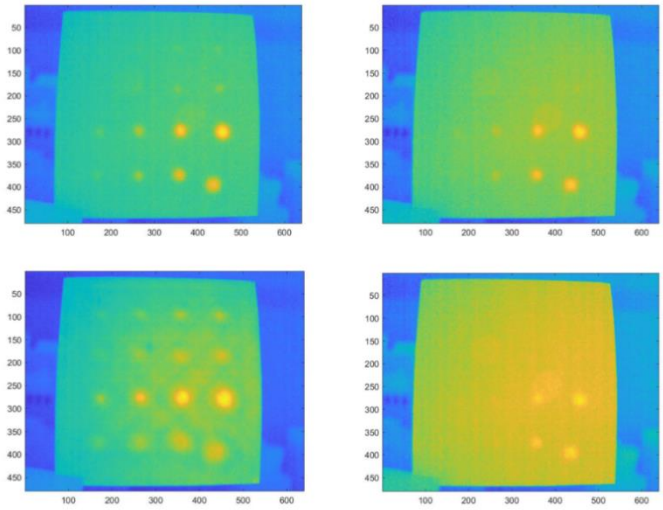


图 3(a) 实验室采集红外数据原图

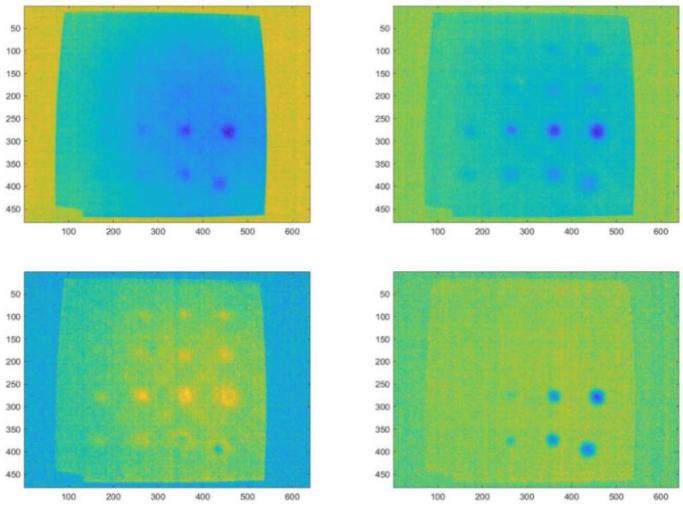


图 3(b) 使用 MATLAB 内置 PCA 函数处理数据

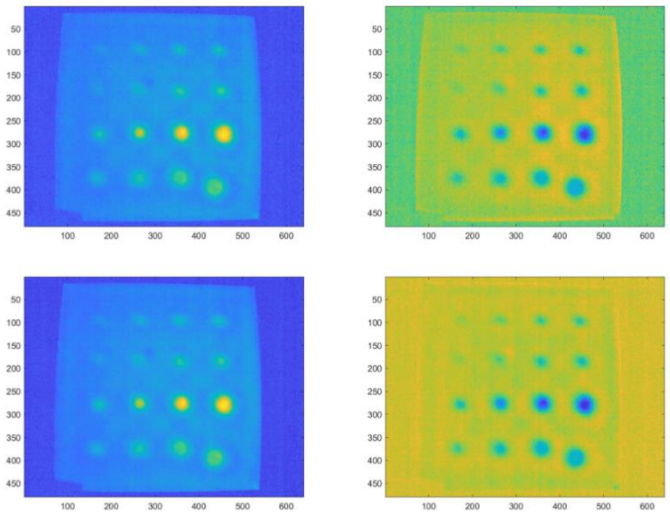


图 3(c) 根据步骤 PCA 函数处理数据

可以看到经过 PCA 处理之后，某些原来观测不到的缺陷可以被观测到了。

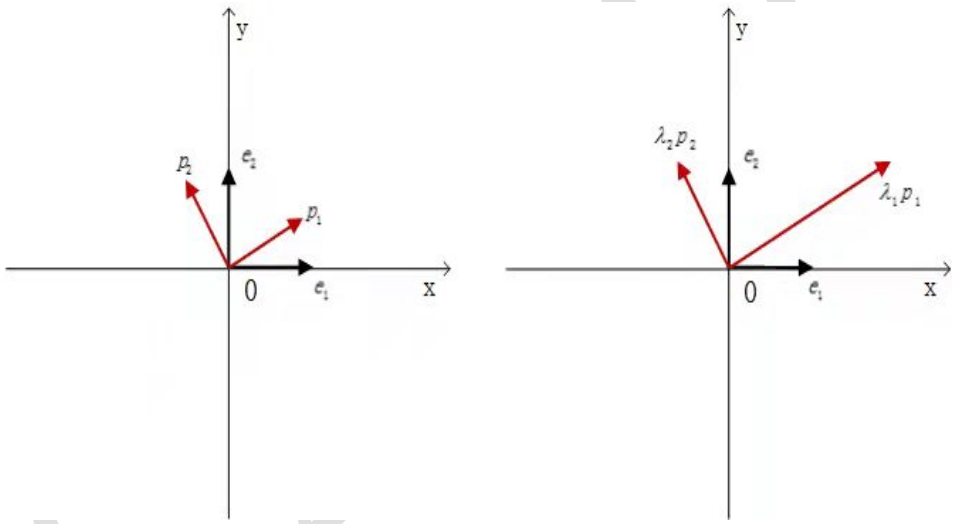
在 PCA 的 matlab 实现过程中，通过一些资料了解到 matlab 内置 PCA（）函数，使用 SVD 分解的方法，于是对 SVD 分解，作出总结。

1.1.4 特征值分解的几何意义

为了完成矩阵的特征值分解，最关键还是要回归到其基本性质上来：

$$\boldsymbol{Cq_i = \lambda_i q_i}。$$

结合主成分分析的推导过程我们知道，协方差矩阵 C 之所以能够分解，是因为在原始空间 R m 中，我们原本默认是用 e 1 , e 2 , . . , e m 这组默认基向量来表示我们空间中的任意一个向量 a，如果我们采用基变换，将 a 用 q 1 , q 2 , . . . , q m 这组标准正交基来表示后，乘法运算就变得很简单了，只需要在各个基向量的方向上对应伸长 λi 倍即可，如图所示：



因为协方差矩阵具备对称性、正定性，保证了他可以被对角化，并且特征值一定为正，从而使得特征值分解的过程一定能够顺利完成。因此利用特征值分解进行主成分分析，核心就是获取协方差矩阵，然后对其进行矩阵分解，获得一组特征值和其对应的方向。

1.1.5 PCA 性质：

（1）**缓解维度灾难**：PCA 算法通过舍去一部分信息之后能使得样本的采样密度增大（因为维数降低了），这是缓解维度灾难的重要手段；

（2）**降噪**：当数据受到噪声影响时，最小特征值对应的特征向量往往与噪

声有关，将它们舍弃能在一定程度上起到降噪的效果；

（3）**过拟合**：PCA 保留了主要信息，但这个主要信息只是针对训练集的，而且这个主要信息未必是重要信息。有可能舍弃了一些看似无用的信息，但是这些看似无用的信息恰好是重要信息，只是在训练集上没有很大的表现，所以 PCA 也可能加剧了过拟合；

（4）**特征独立**：PCA 不仅将数据压缩到低维，它也使得降维之后的数据各特征相互独立。

1.1.6 SVD（奇异值分解）

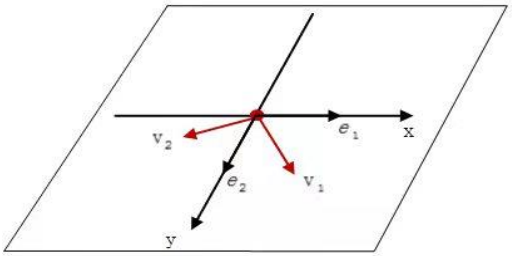
SVD 是提取出一个矩阵重要的特征的方法，它与特征值分解很相似，不同的是可以对非方阵进行分解，即不进行协方差矩阵 C 的求取，绕开它直接对原始的数据采样矩阵 A 进行矩阵分解，从而进行降维操作。

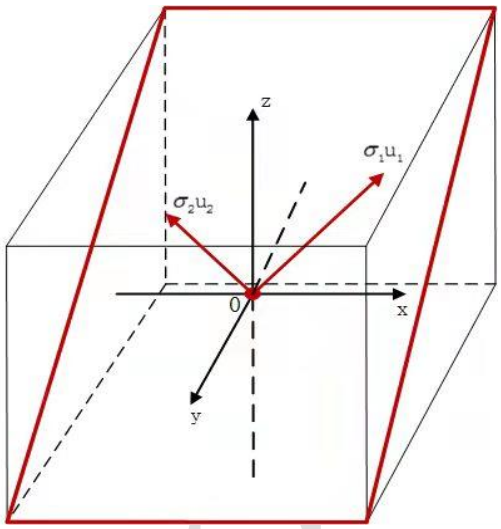
在特征值分解中，对矩阵的要求很严，首先得是一个方阵，其次在方阵的基础上，还得满足可对角化的要求。但是原始的 m × n 数据采样矩阵 A 连方阵这个最基本的条件都不满足，是根本无法进行特征值分解的。

在这里介绍一个对于任意 m × n 矩阵的更具普遍意义的一般性质：

对于一个 m × n，秩为 r 的矩阵 A ，暂且假设 m > n ，于是就有 r ≤ n < m 的不等关系。我们在 R n 空间中一定可以找到一组标准正交向量 v 1 , v 2 , . . . v n，在 R m 空间中一定可以找到另一组标准正交向量 u 1 , u 2 , . . . , u m ，使之满足 n 组相等关系：Avi = σiui，其中（i 取 1 ~n ）。

矩阵 A 是一个 m × n 的矩阵，他所表示的线性变换是将 n 维原空间中的向量映射到更高维的 m 维目标空间中，而 Avi = σiui 这个等式意味着，在原空间中找到一组新的标准正交向量[v 1 v 2 . . . v n]，在目标空间中存在着对应的一组标准正交向量[u 1 u 2 . . . u n]。此时 v i 与 u i 线性无关。当矩阵 A 作用在原空间上的某个基向量 v i 上时，其线性变换的结果就是：对应在目标空间中的 u i 向量沿着自身方向伸长 σi 倍，并且任意一对(v i , u i) 向量都满足这种关系（显然特征值分解是这里的一种特殊情况，即两组标准正交基向量相等）。如图所示：





即

$$A[v_1 \quad v_2 \quad \dots \quad v_n] = [u_1 \quad u_2 \quad \dots \quad u_n]\begin{bmatrix}\sigma_1 & & & \\ & \sigma_2 & & \\ & & \dots & \\ & & & \sigma_n\end{bmatrix}$$

加入 u n + 1 , u n + 2 , . . . , u m 到矩阵右侧，形成完整的 m 阶方阵 U = [u 1 u 2 . . . u n u n + 1 . . . u m] ，在对角矩阵下方加上 m − n 个全零行，形成 m × n 的矩阵

$$\Sigma = \begin{bmatrix}\sigma_1 & & & \\ & \sigma_2 & & \\ & & \dots & \\ & & & \sigma_n \\ & & & \\ & & & \end{bmatrix}$$

由于 Σ 矩阵最下面的 m − n 行全零，因此右侧的计算结果不变，等式依然成立。此时就有：A V = U Σ，由于 V 的各列是标准正交向量，因此$V^{-1} = V^T$，移到等式右侧，得到了一个矩阵分解的式子：$A = U\Sigma V^T$，其中 U 和 V 是由标准正交向量构成的 m 阶和 n 阶方阵，而 Σ 是一个 m × n 的对角矩阵。

此时还有一个最关键的地方没有明确：就是方阵 U 和方阵 V 该如何取得，以及 Σ 矩阵中的各个值应该为多少。

由于：

$$A^TA = V\Sigma^2V^T$$

$$AA^T = U\Sigma^2U^T$$

他们的秩相等，为 r，因此他们拥有完全相同的 r 个非零特征值，从大到小排列为：λ1 , λ 2 , . . . λ r ，两个对称矩阵的剩余 n − r 和 m − r 个特征值为 0 。

同时，由对称矩阵的性质可知：A^TA一定含有 n 个标准正交特征向量，对应特征值从大到小的顺序排列为：[v 1 v 2 . . . v n]，而AA^T也一定含有 m 个标准正交特征向量，对应特征值从大到小依次排列为[u 1 u 2 . . . u m]。

对应的 Σ 矩阵即求出AA^T或A^TA的非零特征值，从大到小排列为：$\lambda_1, \lambda_2, \dots, \lambda_r$，Σ 矩阵中对角线上的非零值 σi 则依次为：$\sqrt{\lambda_1}, \sqrt{\lambda_2}, \dots, \sqrt{\lambda_r}$ 。因此，Σ 矩阵对角线上 σr 以后的元素均为 0 了。

整个推导分析过程结束，我们隐去零特征值，最终得到了最完美的 SVD 分解结果：

$$A = [u_1 \quad u_2 \quad \dots \quad u_m]\begin{bmatrix}\sigma_1 & & & \\ & \sigma_2 & & \\ & & \dots & \\ & & & \sigma_r\end{bmatrix}\begin{bmatrix}v_1^T \\ v_2^T \\ \dots \\ v_n^T\end{bmatrix}$$，这里 r ≤ n < m

同时，SVD 可以通过行压缩、列压缩、和矩阵整体压缩三种思路进行数据降维。

这里对AA^T进行特征值分解，不就是 PCA 中求解 w的过程吗？所以 PCA 可以通过部分 SVD 的求解而得到，但显然 SVD 的计算量要大于 PCA，那 matlab 内置函数为什么会使用 SVD 来解 PCA 呢。

因为在 matlab PCA（）中的 SVD 是使用 QR 分解近似得来的，其计算复杂度比较低。

1.1.7 QR 分解

$$A = QR$$

使用施密特正交化将矩阵分解为单位正交矩阵 Q 与上三角矩阵 R。

施密特正交化：

假设有一组线性无关的向量 α_1、α_2，将其构造为正交向量组。

我们先控制住一个向量：$\beta_1 = \alpha_1$，那现在只要找到另外一个新的向量β_2，使得β_2与β_1内积为 0 即可。

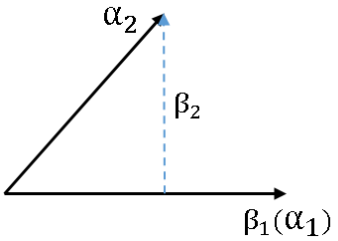


图 4 投影图示

因为β_2是由β_1、α_2构造而成，所以我们可以令：

$$\beta_2 = \mathrm{k}_1\beta_1 + k_2\alpha_2$$

β_2，β_1内积为零可得：

$$(\beta_1，\ \beta_2) = (\beta_1，\ \mathrm{k}_1\beta_1 + k_2\alpha_2) = 0$$

$$\mathrm{k}_1(\beta_1，\ \beta_1) + k_2(\alpha_2，\ \beta_1) = 0 \Rightarrow \mathrm{k}_1 = -\frac{k_2(\alpha_2，\ \beta_1)}{(\beta_1，\ \beta_1)}$$

$$\beta_2 = k_2\alpha_2 - \frac{k_2(\alpha_2，\ \beta_1)}{(\beta_1，\ \beta_1)}\beta_1 = k_2\alpha_2 - \frac{k_2(\alpha_2，\ \beta_1)}{(\beta_1，\ \beta_1)}\alpha_1$$

更多向量使用相同的思想，分别与前面正交化后的向量内积等于 0。

1.2 ICA

ICA(Independent component analysis)即独立成分分析，ICA 来自于经典的鸡尾酒问题：在一个鸡尾酒会现场，如果用安放在不同位置的多个麦克风现场录音，则所记录的信号实际上是不同声源的混合信号。人们希望从这些混合录音信号中把不同的声源分离出来。其基本模型为：

$$X = AS$$

其中S为源信号矩阵，X为观测信号矩阵，A为混合矩阵。ICA 所做的事情就是求S的一个估计Y，使得Y能与S十分近似。

即：$Y = WX$所以我们需要做的事情就是求出W (即目标函数)。

总的来说，我们可以用一个“方程式”表达 ICA 的方法：

ICA 方法 = 目标函数 + 优化方法

在目标函数明确的情况下，我们可以使用任何经典的优化算法，如梯度下降和牛顿法。ICA 的估计原理也可以分为几个类别：极大化观测数据的非高斯性、基于信息理论的估计方法。在第一种中可分为峭度法和负熵法，在第二种中可分为最大似然估计、信息极大化法和最小互信息法。这里从两个原理中分别选择最常用的做推导，其余方法在参考文献中均有说明。

ICA 中有一些假设：1）成分s_i之间是统计独立的 2）假设独立成分是非高斯分布的

注：

为什么是非高斯？

假设$s(s_1 s_2) \sim N(0, I)$，$cov(ss^T) = 1$

X = AS，则 X 也为高斯分布，均值为零，$cov(xx^T) = cov(Ass^T A^T) = AA^T$

令 R 为标准正交阵，$A' = AR \rightarrow X' = A'S$（X′均值为 0，协方差为$AA^T$，与 X 相同），因此无法通过 X 确定混合矩阵，也就无法得到源信号。

1.2.1 极大化数据的非高斯性——基于负熵的估计方法

负熵总是非负的，当且仅当 x（观测数据）是高斯分布时为其值为 0，因此需要极大化负熵得到独立分量。

这里采用负熵的近似表达式：

$$J(y) = [E\{G(y)\} - E\{G(V)\}]^2$$

其中 v 是与 y 具有同样方差的高斯变量，通常 y 是具有零均值,单位方差的向量,y 就是具有零均值 、单位方差的高斯变量。

公式中的 G 函数可以选取以下三种：

$$G_1(y) = \frac{1}{a} logcosh\, a\, y, a为常数，\ 1 < a < 2$$

$$G_2(y) = -exp(-y^2/2)$$

$$G_3(y) = \frac{1}{4} y^4$$

G_2适用于超高斯源信号的恢复，G_3适用于高斯分布的源信号恢复，如果超高斯源和亚高斯源都存在，选择 G_1比较合适。

对负熵的近似式求导可得：

$$\begin{Bmatrix} \Delta w \propto rE\{zg(w^Tz)\} \\ w \leftarrow w/||w|| \end{Bmatrix}$$

公式中$r = E\{G(y)\} - E\{G(v)\}$,函数 g 是 G 的导数

梯度算法的在线形式为：

$$\begin{Bmatrix} \Delta w \propto rzg(w^Tz) \\ w \leftarrow w/||w|| \end{Bmatrix}$$

（在线算法是用了瞬时值来代替期望值，也是随机梯度迭代的思路）

（||w||表示向量的二范数用于正交化）

基于负熵的 Fast ICA 算法推导：

白化：

$$z = Vx$$

其中$cov(zz^T) = I$，V为变换矩阵，x为观测数据。

$$V = D^{-\frac{1}{2}}P^T$$

其中，D，P 分别为$cov(xx^T)$的特征值分解所得到的特征值矩阵与特征向量矩阵。

FastICA 学习规则是找到一个方向以便$W^TX(Y = W^TX)$具有最大的非高斯性。这里的非高斯性可用下式衡量：

$$J(y) = [E\{G(y)\} - E\{G(V)\}]^2$$

首先,W^TX的负熵的最大近似值能通过对$E\{G(W^TX)\}$,进行优化来获得。根据 K-T 条件，在$E\{(W^TX^2)\} = ||W||^2 = 1$的约束下，$E\{G(W^TX)\}$的最优值能在满足下式的点上获得。

$$E\{Xg(W^TX)\} + \beta W = 0$$

其中 β 为拉格朗日常数，考虑用牛顿迭代的优化算法求解此式，令上式的左边为函数 F，即$F(w) = E\{zg(w^Tz)\} + \beta w$，可得 F(W)的导数，也就是拉格朗日目标函数的二阶导数为：

$$\frac{\delta F}{\delta w} = E\{zz^Tg'(w^Tz)\} + \beta I$$

其中 g`是 g 的导数，因为数据经过白化处理，上式可以进一步处理：

$$E\{zz^Tg'(w^Tz)\} \approx E\{\mathrm{zz}^T\}E\{g'(w^Tz)\} = E\{g'(w^Tz)\}I$$

从而得到近似牛顿迭代：

$$w \leftarrow w - \frac{E\{zg(w^Tz)\} + \beta w}{E\{g'(w^Tz\text{）}\} + \beta}$$

将两边乘以$-(E\{g'(w^Tz)\} + \beta)$，并考虑到 w 的标准单位化处理，可得不动点算法为：

$$w \leftarrow E\{zg(w^Tz)\} - E\{g'(w^Tz\text{）})\}w$$

$$w \leftarrow w/||w||$$

1.2.2 基于信息理论的估计方法——似然最大

假定s_i的概率密度P_s，则源信号的联合概率密度为（假设独立）

$$p(S) = \prod_{i=1}^{n} p_s(s_i)$$

则：

$$p(X) = p_s(Wx)|W| = |W| \cdot \prod_{i=1}^{n} p_s(w_i^T x)$$

推导：$\mathrm{X} = \mathrm{AS} \rightarrow \mathrm{S} = \mathrm{WX}$，$s$的分布$P(s) = P\{S < s\}$

$$P_X(x) = P\{X < x\} = P\{AS < x\} = P\{S < Wx\} = P_S(Wx)$$

所以：

$$p(x) = P_X'(x) = P_S'(Wx) \cdot |W| = p_s(Wx)|W|$$

于是要解 W，我们还需要 S 的分布$P_s(\mathrm{s}_i)$，这里一般选择 sigmoid function。即：

$$P_S(s) = g(s) = \frac{1}{1 + e^{-s}}$$

$$p_S(s) = P_S^{\prime(s)} = g^{\prime(s)} = \frac{e^s}{(1 + e^s)^2}$$

有了概率密度，我们就可以构造似然函数，求解 W，使 X 等于我们观测值的概率最大。

$$L(w) = \log g\left(\prod_{i=1}^{m} p(x_i)\right) = \sum_{i=1}^{m}\left[\sum_{j=1}^{n} \log g'\left(w_j^T x^{(i)}\right) + \log|W|\right]$$

注：这里$[\log g'(s)]' = 1 - 2g(s)$

将上式对 w 求导，即可得出 w 的梯度：

$$\frac{\partial L(w)}{\partial w} = \begin{pmatrix} 1 - 2g\left(w_1^T x^{(i)}\right) \\ 1 - 2g\left(w_2^T x^{(i)}\right) \\ \vdots \\ 1 - 2g\left(w_n^T x^{(i)}\right) \end{pmatrix} x^{(i)^T} + (w^T)^{-1}$$

注：加号右边：$\nabla_w|W| = |W|(W^{-1})^T$

使用梯度下降法可以得到最终的 W。但在 ICA 的具体应用过程中，由于以上过程涉及到矩阵的求逆运算，计算量很大，我们选择使用自然梯度法在我们之前得到的梯度基础上右乘矩阵 W^TW 进行计算，这样即可避免求矩阵的逆。

下图为根据基于负熵最大 FastICA 步骤书写的 matlab 程序处理的数据结果。

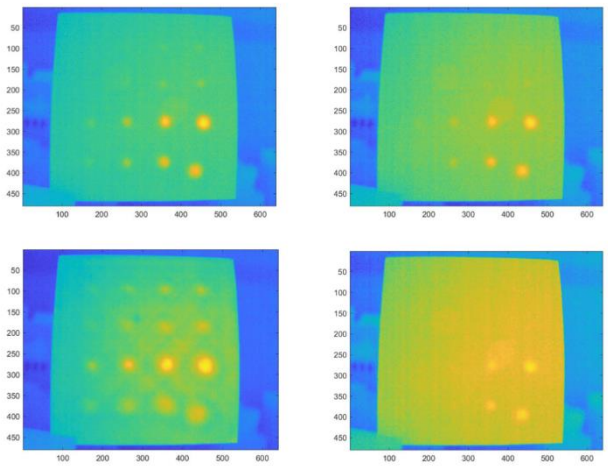


图 5(a) 实验室采集红外数据原图

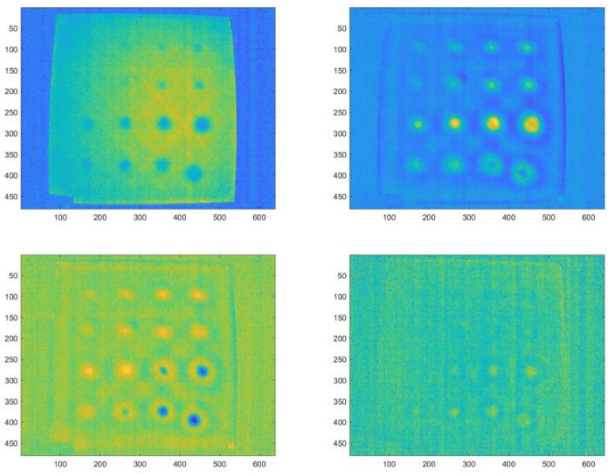


图 5(b) ICA 得到的检测结果

1.3 EM 算法

1.3.1 基本知识预备

1.3.1.1 最大似然估计（MLE）

最大似然估计的基本想法是从一堆样本数据中，有未知参数时，通过概率模型列出似然函数取得最大值时的未知参数值。

最大似然函数 L:

$$L(\theta|X) = \prod_{i=1}^{N} p(x_i|\theta)$$

θ—未知参数

x—样本数据

$p(x_i|\theta)$则表示在样本数据x_i（已知）的条件下，得到该样本数据x_i的概率，因为θ是未知参数，所以$p(x_i|\theta)$是一个关于 θ 的函数。

我们所要求的参数 θ 的最大似然估计值

$$\theta = argmax(L(\theta|X))$$

我们要求出似然函数 L 的极大值，最直接的方法便是求导。然而对多项式乘积求导复杂的多，于是我们对 L 取对数。

$$\log L(\theta|x) = \log \prod_{i=1}^{N} p(\theta|x_i) = \sum_{i=1}^{N} \log p(x_i|\theta)$$

1.3.1.2 贝叶斯最大后验概率估计（MAP）

在 MLE 中，是使似然函数$L(\theta|x)$最大的时候参数 θ 的值，并没有考虑先验概率$p(\theta)$。而在 MAP 中，则是求$p(x|\theta)p(\theta)$最大时的参数 θ，这就要求 θ 不仅仅是让似然函数最大，同时要求 θ 本身的 先验概率也较大

可以认为 MAP 是贝叶斯算法的一种应用

$$P(\theta|X) = \frac{P(\theta)P(X|\theta)}{P(X)} \propto P(\theta)P(X|\theta)$$

$$\theta = argmaxP(\theta)P(X|\theta)$$

1.3.1.3 混合高斯模型(GMM)

混合高斯模型基本思想是将事件的概率模型分解为若干个高斯模型加权相加。其概率密度函数如下：

$$P(x) = \sum_{k=1}^{K} P(k)P(x|k) = \sum_{k=1}^{K} \pi_k N(x|\mu_k, \sum_k)$$

$\mu_k, \sum_k$为第 k 个高斯模型的期望和方差。其中$P(k)$为样本 x 属于第 k 个高斯分布的概率，其值等于权重π_k。$P(x|k) = N(x|\mu_k, \sum_k)$。$\sum_{k=1}^{K} \pi_k = 1$。

GMM 的参数 θ 有：

$$\theta = \{\mu_1, \mu_2, \dots, \mu_k, \sum_1, \sum_2, \dots, \sum_k, \pi_1, \pi_2, \dots, \pi_k\}$$

1.3.2 EM 算法流程

有样本数据 X= $\{x_1, x_2, \dots, x_k\}$,p(x,z|θ)是样本 x 和隐变量 z 的联合概率分布。隐变量 z 表示样本 x 在哪一类。EM 算法首先会初始化模型参数 θ，然后进行 E-step 和 M-step 的迭代。

E-step

利用样本数据和的上一次迭代的模型参数估计值θ，计算引入隐变量 z 后的似然函数的期望最大时的$Q(z_i)$。

$$Q(z_i) = p(z_i|x_i, \theta)。$$

M-step

求在 E-step 所确定的隐变量条件下的似然函数最大化的参数。

$$\theta = argmax_\theta \sum_{i=1}^{N} \sum_{l \in z_i}^{K} Q_i(l) \log \frac{p(x_i, l|\theta)}{Q_i(l)}$$

其中 N 为样本的个数，K 为隐变量的个数。

1.3.3 EM 推导

1.3.3.1 琴生不等式

Jensen 不等式描述如下：

如果 f 是凸函数，XX 是随机变量，则 E[f(X)]>=f(E[X])，当 f 是严格凸函数时，则 E[f(X)]>f(E[X])E[f(X)]>f(E[X])；

如果 f 是凹函数，XX 是随机变量，则 f(E[X])<=E[f(X)]，当 f 是（严格）凹函数当且仅当−f 是（严格）凸函数。（这里定义的凹凸性与之前高数中学的相反）

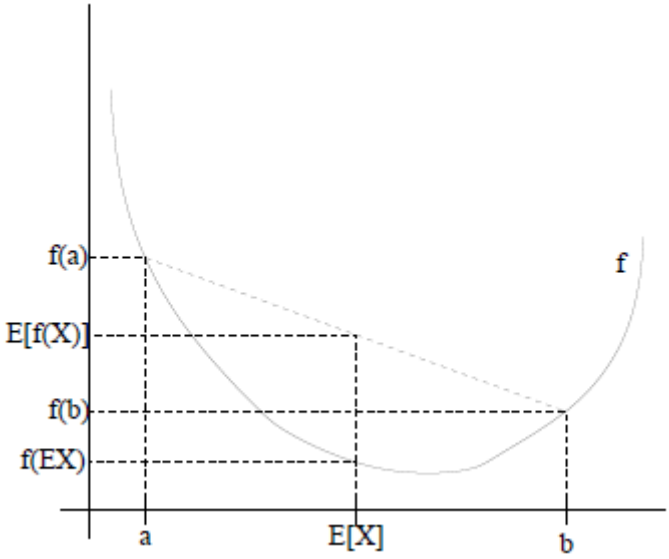


图 6 琴生不等式示意图

1.3.3.2 推导

- E-step

  引入隐变量 z 后的对数似然函数为

$$L(\theta|X) = \sum_{i=1}^{N} \log p(x_i|\theta^t)$$

$$= \sum_{i=1}^{N} \log \sum_{z_i} p(x_i, z_i|\theta^t)$$

$$= \sum_{i=1}^{N} \log \sum_{z_i} Q(z_i) \frac{p(x_i, z_i|\theta^t)}{Q(z_i)}$$

$$\geq \sum_{i=1}^{N} \sum_{z_i} Q(z_i) \log \frac{p(x_i, z_i|\theta^t)}{Q(z_i)}$$

最后一个不等式由琴生不等式推导得出。$Q(z_i)$是关于隐变量z_i的概率，而且$\sum_{z_i} Q(z_i) = 1$。

我们就得到了似然函数的下界$\sum_{i=1}^{N} \sum_{z_i} Q(z_i) \log \frac{p(x_i, z_i|\theta^t)}{Q(z_i)}$。要使≥取到=，即下界最大化，那么就要使如图 1 中的 a，b 相等。a，b 对应式中的$\frac{p(x_i, z_i|\theta^t)}{Q(z_i)}$，令$\frac{p(x_i, z_i|\theta^t)}{Q(z_i)} = c$。进行以下变换。

$$p(x_i, z_i|\theta^t) = cQ(z_i)$$

$$\sum_{z_i} p(x_i, z_i|\theta^t) = c\sum_{z_i} Q(z_i)$$

因为$\sum_{z_i} Q(z_i) = 1$

所以

$$\sum_{z_i} p(x_i, z_i|\theta^t) = c$$

也就是说，似然函数取到最大时

$$Q(z_i) = \frac{p(x_i, z_i|\theta^t)}{c} = \frac{p(x_i, z_i|\theta^t)}{\sum_{z_i} p(x_i, z_i|\theta^t)} = p(z_i|x_i, \theta^t)$$

以上就是E-step的 推导过程，其中所求的期望E就是$\sum_{z_i} Q(z_i)\log\frac{p(x_i, z_i|\theta^t)}{Q(z_i)}$，求出期望最大时的$Q(z_i)$。

- M-step

  执行完 E-step 后，≥取到=，原来的似然函数变为

$$L(\theta^t|X) = \sum_{i=1}^{N}\sum_{z_i} Q^t(z_i)\log\frac{p(x_i, z_i|\theta^t)}{Q^t(z_i)}$$

  在对似然函数对参数求偏导等于 0，得到最优参数。

$$\theta^{t+1} = argmax_\theta L(\theta^t|X)$$

- 收敛性证明

  我们还要证明通过以上 EM 迭代后的似然函数会收敛。因为似然函数有上界 1，故只要证明递增：

$$L(\theta^{t+1}|X) \geq L(\theta^t|X)$$

  证明：

$$L(\theta^{t+1}|X) = \sum_{i=1}^{N}\sum_{z_i}[\log p(x_i, z_i|\theta^{t+1}) - \log p(z_i|x_i, \theta^{t+1})]p(z_i|x_i, \theta)$$

$$\geq \sum_{i=1}^{N}\sum_{z_i}[\log p(x_i, z_i|\theta^{t+1}) - \log p(z_i|x_i, \theta^t)]p(z_i|x_i, \theta)$$

$$\geq \sum_{i=1}^{N}\sum_{z_i}[\log p(x_i, z_i|\theta^t) - \log p(z_i|x_i, \theta^t)]p(z_i|x_i, \theta)$$

$$= L(\theta^t|X)$$

  第二式≥第三式的解释：

  因为

$$\theta^{t+1} = argmax_\theta L(\theta^t|X)$$

  所以

$$\sum_{i=1}^{N}\sum_{z_i}\log p(x_i,z_i|\theta)\,p(z_i|x_i,\theta^{t+1}) \geq \sum_{i=1}^{N}\sum_{z_i}\log p(x_i,z_i|\theta)\,p(z_i|x_i,\theta^{t})$$

$$\sum_{z_i}[\log p(z_i|x_i,\theta^t)\,p(z_i|x_i,\theta^t) - \log p(z_i|x_i,\theta)\,p(z_i|x_i,\theta^t)]$$

$$= \sum_{z_i} -\log\left(\frac{p(z_i|x_i,\theta)}{p(z_i|x_i,\theta^t)}\right) p(z_i|x_i,\theta^t)$$

$$\geq -\log\sum_{z_i}\frac{p(z_i|x_i,\theta)}{p(z_i|x_i,\theta^t)}p(z_i|x_i,\theta^t) = -\log 1 = 0$$

所以

$$\sum_{z_i}\log p(z_i|x_i,\theta^t) \geq \sum_{z_i}\log p(z_i|x_i,\theta) \geq \sum_{z_i}\log p(z_i|x_i,\theta^{t+1})$$

所以

$$\sum_{i=1}^{N}\sum_{z_i}[\log p(x_i,z_i|\theta^{t+1}) - \log p(z_i|x_i,\theta^{t+1})]p(z_i|x_i,\theta)$$

$$\geq \sum_{i=1}^{N}\sum_{z_i}[\log p(x_i,z_i|\theta^{t+1}) - \log p(z_i|x_i,\theta^{t})]p(z_i|x_i,\theta)$$

故收敛。

1.3.4 EM 算法在 GMM 中的应用

EM 算法应用在 GMM 中，引入的隐变量 z 就是当前样本数据 x 属于 GMM 中哪个高斯分布的概率。

1.3.4.1 E-step

$$Q^{t+1}(z_i) = p(z_i|x_i,\theta)$$

$$= \frac{p(x_i,z_i|\theta^t)}{p(x_i|\theta^t)}$$

$$= \frac{p(x_i,z_i|\theta^t)}{\sum_{l\in z_i}p(x_i,l|\theta^t)}$$

$$= \frac{p(x_i|z_i,\theta^t)p(z_i|\theta^t)}{\sum_{l\in z_i}p(x_i|l,\theta^t)\,p(l|\theta^t)}$$

$$= \frac{N(\mu_{z_i}, \sum_{z_i})\pi_{z_i}}{\sum_{l \in z_i} N(\mu_l, \sum_l)\pi_l}$$

z_i表示第 i 个样本数据x_i属于 GMM 哪个高斯分布，π_{z_i}表示属于该高斯分布的概率，即之后的权重。

1.3.4.2 M-step

- 似然函数 L

在得到隐变量 z 的期望Q^{t+1}后，M-step 就是再此基础上求 GMM 的参数 θ 。

$$\theta = \{\mu_1, \mu_2, \dots, \mu_k, \sum_1, \sum_2, \dots, \sum_k, \pi_1, \pi_2, \dots, \pi_k\}$$

此时的似然函数 L:

$$L(\theta|X) = \sum_{i=1}^{N}\sum_{l \in z_i}^{K} Q_i(l) \log \frac{p(x_i, l|\theta)}{Q_i(l)}$$

$$= \sum_{i=1}^{N}\sum_{l \in z_i}^{K} Q_i(l) \log p(x_i, l|\theta) - \sum_{i=1}^{N}\sum_{l \in z_i}^{K} Q_i(l) \log Q_i(l)$$

$$= \sum_{i=1}^{N}\sum_{l \in z_i}^{K} Q_i(l) \log p(x_i, l|\theta) - constant$$

$$= \sum_{i=1}^{N}\sum_{l \in z_i}^{K} Q_i(l) \log N(\mu_l, \sum_l)\pi_l - constant$$

$$= \sum_{i=1}^{N}\sum_{l \in z_i}^{K} Q_i(l) \log \pi_l + \sum_{i=1}^{N}\sum_{l \in z_i}^{K} Q_i(l) \log N(\mu_l, \sum_l) - constant$$

- 计算参数 π

$$\begin{cases} \dfrac{\partial L(\theta|X)}{\partial \pi_l} = 0 \\ \sum_{l}^{K} \pi_l = 1 \end{cases}$$

拉格朗日数乘法得到拉格朗日行列式：

$$\begin{cases} \dfrac{\partial L(\theta|X)}{\partial \pi_l} + \lambda(\sum_{l}^{K} \pi_l - 1) = 0 \\ \sum_{l}^{K} \pi_l - 1 = 0 \end{cases}$$

求偏导后的形式为;

$$\begin{cases} \frac{1}{\pi_1}\sum_i^N Q_i(1) - \lambda = 0 \\ \quad\cdots \\ \quad\cdots \\ \frac{1}{\pi_l}\sum_i^N Q_i(l) - \lambda = 0 \end{cases}$$

将以上等式组相加得

$$\sum_l^K \sum_i^N Q_i(l) = \lambda \sum_l^K \pi_l = \lambda$$

由于$Q_i(l) = p(l|x_i, \theta)$，所以

$$\sum_l^K \sum_i^N Q_i(l) = \sum_i^N \sum_l^K p(l|x_i, \theta) = N = \lambda$$

$$\sum_i^N Q_i(l) = \lambda \pi_l$$

$$\pi_l = \frac{1}{\lambda}\sum_i^N Q_i(l) = \frac{1}{N}\sum_i^N Q_i(l) = \frac{1}{N}\sum_i^N p(l|x_i, \theta)$$

- 计算参数 μ

$$\sum_{i=1}^N \sum_{l \in z_i}^K Q_i(l) \log N\ (\mu_l, \sum_l)$$

$$= \sum_{i=1}^N \sum_{l \in z_i}^K Q_i(l) \log \frac{1}{\sqrt{2\pi \sum_l}} e^{-\frac{(x_i - \mu_l)^2}{2\sum_l}}$$

$$= \sum_{i=1}^N \sum_{l \in z_i}^K Q_i(l) \left[-\log 2\pi - \frac{1}{2}\log \sum_l - \frac{(x_i - \mu_l)^2}{2\sum_l}\right]$$

对 μ 求导

$$\frac{\partial L(\theta|X)}{\partial \mu_l} = \sum_i^N Q_i(l) \frac{x_i - \mu_l}{\sum_l} = 0$$

得

$$\mu_l = \frac{\sum_i^N Q_i(l) x_i}{\sum_i^N Q_i(l)}$$

- 计算参数Σ

$\sum_{i=1}^N \sum_{l \in z_i}^K Q_i(l) \left[-\log 2\pi - \frac{1}{2}\log \sum_l - \frac{(x_i - \mu_l)^2}{2\sum_l}\right]$对$\sum_l$求偏导得：

$$\frac{\partial L(\theta|X)}{\partial \sum_l} = \sum_i^N Q_i(l)[-\frac{1}{2\sum_l} + \frac{(x_i - \mu_l)^2}{2\sum_l{}^2}] = 0$$

得

$$\sum_l = \frac{\sum_i^N Q_i(l)(x_i - \mu_l)^2}{\sum_i^N Q_i(l)}$$

1.3.5 Matlab 实验

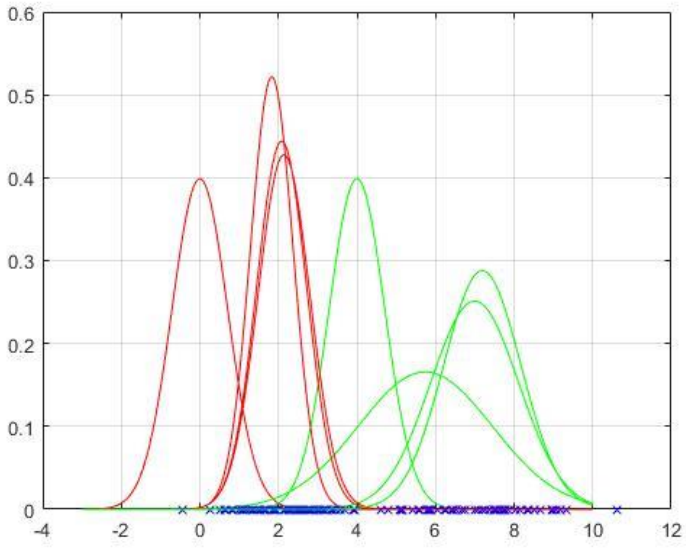


图 8 EM 算法拟合数据的高斯混合分布

参考博客：

1. http://www.csuldw.com/2015/12/02/2015-12-02-EM-algorithms/
2. https://www.zybuluo.com/zsh-o/note/1077331

视频讲解：

https://www.bilibili.com/video/BV1Wp411R7am

1.4 变分贝叶斯

1.4.1 变分贝叶斯原理

变分贝叶斯可看作 EM 算法的一种推广。它们的共同思想是：

1.将复杂的积分计算转化为期望的计算。

2.迭代过程中，一步更新潜在变量，一步更新模型参数。（通过一组相互依然的等式进行不断的迭代来获得最优解。）

变分贝叶斯的两个目的是：

1.近似不可观测变量的后验概率，以便通过这些变量做出统计判断。

2.对一个特定的模型，给出观测变量的边缘似然函数的下界。主要用于模型的选择，模型的边缘似然值越高，则模型对数据拟合程度越好，该模型产生 Data 的概率也越高。

模型中潜在变量的后验概率通常是很复杂的，因此在误差允许范围内，使用更简单、容易理解的数学形式$Q(Z)$来近似$P(Z|D)$。（注：D 表示观测样本，Z 表示潜在变量的集合）

描述两个随机分布之间距离的度量：KL 散度。

Q 分布与 P 分布的 KL 散度为：

$$D_{KL}(Q||P) = \sum_Z Q(Z) log \frac{Q(Z)}{P(Z|D)} = \sum_Z Q(Z) log \frac{Q(Z)}{P(Z,D)} + logP(D)$$

移项可得：

$$logP(D) = D_{KL}(Q||P) - \sum_Z Q(Z) log \frac{Q(Z)}{P(Z,D)} = D_{KL}(Q||P) + L(Q)$$

由于$logP(D)$已知，只要极大化$L(Q)$，就能使 KL 散度最小。

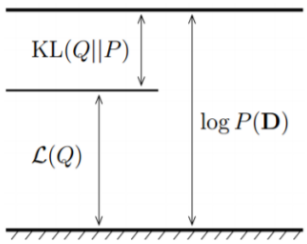


图 9 $logP(D)$、$L(Q)$与$D_{KL}(Q||P)$三者的关系，其中$logP(D)$固定

通过选择合适的$Q(Z)$，使$L(Q)$便于计算和求极值，这样就可以得到后验$P(Z|D)$的近似解析表达式。

$$L(Q) = \sum_Z Q(Z) lnP(Z,D) - \sum_Z Q(Z) lnQ(Z) = E_Q[lnP(Z,D)] + H(Q)$$

根据平均场理论，将可以对 Q(Z)通过参数和潜在变量的划分因式分解，将 Z 划分为 Z_1,.., Z_M。（各子集之间包含的随机变量所彼此统计独立）

$$Q(Z) = \prod_{i=1}^{M} Q_i(Z_i)$$

这里并非将每个不可观测的变量都作为一个划分，应该根据实际情况做决定。有时候将几个潜在变量放在一起更容易处理。（具体做法见后文例子）

于是，对于某个划分,我们可以先保持其他划分不变，然后用以上关系式更新它。相同步骤应用于其他划分的更新，使得每个划分之间充分相互作用，最终达到稳定值。

由上式可得：

$$L(Q) = E_Q[lnP(Z,D)] + H(Q)$$

$$= \sum_Z \left(\prod_i Q_i(Z_i)\right) lnP(Z,D) - \sum_Z \left[\left(\prod_k Q_k(Z_k)\right)\sum_i lnQ_i(Z_i)\right]$$

$$= \int \left(\prod_i Q_i(Z_i)\right) lnP(Z,D)dZ - \int \left[\left(\prod_k Q_k(Z_k)\right)\sum_i lnQ_i(Z_i)\right]dZ$$

固定除了 Zi 以外的所有划分 Z-i，更新划分 Zi。将上式两项分步计算：

$$E_{Q(Z)}[lnP(Z,D)] = \int \left(\prod_i Q_i(Z_i)\right) lnP(Z,D)dZ$$

$$= \int Q_i(Z_i)dZ_i \int Q_{-i}(Z_{-i})\, lnP(Z,D)dZ_{-i}$$

$$= \int Q_i(Z_i)\langle ln(Z,D)\rangle_{Q_{-i}(Z_{-i})}dZ_i$$

$$= \int Q_i(Z_i) ln\left(exp\langle ln(Z,D)\rangle_{Q_{-i}(Z_{-i})}\right)dZ_i$$

$$= \int Q_i(Z_i) lnQ_i^*(Z_i)dZ_i + lnC$$

其中定义$Q_i^*(Z_i) = \frac{1}{C} exp\langle ln(Z,D)\rangle_{Q_{-i}(Z_{-i})}$，C 为$Q_i^*(Z_i)$的归一化常数，$\langle F\rangle_{p(x)}$表示函数 F 对概率密度函数$p(x)$的期望。

再考虑第二项：

$$H(Q) = -\int \left[\left(\prod_k Q_k(Z_k)\right)\sum_i lnQ_i(Z_i)\right]dZ$$

$$= -\sum_i \int \left(\prod_k Q_k(Z_k)\right) lnQ_i(Z_i)dZ$$

$$= -\sum_i \int\int Q_i(Z_i)Q_{-i}(Z_{-i}) lnQ_i(Z_i)dZ_i dZ_{-i}$$

$$= -\sum_i \langle \int Q_i(Z_i) lnQ_i(Z_i)dZ_i\rangle_{Q_{-i}(Z_{-i})}$$

$$= -\sum_i \int Q_i(Z_i) lnQ_i(Z_i)dZ_i$$

此时可得到：

$$L\big(Q(Z)\big)=\int Q_i(Z_i)lnQ_i^*(Z_i)dZ_i-\sum_i\int Q_i(Z_i)lnQ_i(Z_i)dZ_i+lnC$$

$$=\left(\int Q_i(Z_i)lnQ_i^*(Z_i)dZ_i-\int Q_i(Z_i)lnQ_i(Z_i)dZ_i\right)$$

$$-\sum_{k\neq i}\int Q_k(Z_k)lnQ_k(Z_k)dZ_k+lnC$$

$$=\int Q_i(Z_i)ln\frac{Q_i^*(Z_i)}{Q_i(Z_i)}dZ_i-\sum_{k\neq i}\int Q_k(Z_k)lnQ_k(Z_k)dZ_k+lnC$$

$$=-D_{KL}[Q_i(Z_i)||Q_i^*(Z_i)]+H[Q_{-i}(Z_{-i})]+lnC$$

由 KL 散度的定义可知，$-D_{KL}[Q_i(Z_i)||Q_i^*(Z_i)]$小于等于零，当且仅当

$$Q_i(Z_i)=Q_i^*(Z_i)=\frac{1}{C}exp\langle ln(Z,D)\rangle_{Q_{-i}(Z_{-i})}$$

时，该 KL 散度为零，使得$L\big(Q(Z)\big)$取最大值。由此得到了变分贝叶斯的参数迭代更新思想：

$$lnQ_i(Z_i)=\langle ln(Z,D)\rangle_{Q_{-i}(Z_{-i})}-lnC=\langle ln(Z,D)\rangle_{Q_{-i}(Z_{-i})}+const$$

1.4.2 变分贝叶斯在 GMM 中的应用

假设现在有独立同分布的训练样本 X 符合下述混合高斯分布：

$$p(x)=\sum_{k=1}^{K}\pi_kN\big(x|\mu_k,\sum_k\big)$$

如何求解高斯混合分布的三组参数π_k，μ_k，$\sum_k$？

步骤一：选择无信息先验分布

先验分布不能随便取，要选择共轭分布才合适，即先验分布和后验分布属于同一分布类型。本例中不展开讨论，直接给出：

$$\pi_{i=1,\cdots,k}\sim SymDir(K,\alpha_0)$$
$$\Lambda_{i=1,\cdots,k}\sim W(w_0,v_0)$$
$$\mu_{i=1,\cdots,k}=N(m_0,(\beta_0\Lambda_i)^{-1})$$
$$z_{i=1,\cdots,N}\sim Mult(1,\pi)$$
$$X_{i=1,\cdots,N}=N(\mu_z)$$

K 表示单高斯分布的个数，N 表示样本个数，每个分布的解释如下：

$SymDir(\,)$表示 K 维对称 Dirichlet 分布；它是卡方分布或多项式分布的共轭先验分布。

$W(\,)$表示 Wishart 分布；对一个多元高斯分布，它是逆协方差矩阵的共轭先验。

$Mult(\,)$表示多项分布（此处也称卡方分布）；多项式分布是二项式分布的推

广，表示在一个 K 维向量中只有一项为 1，其它都为 0。

$N(\ )$为高斯分布，在这里特别指多元高斯分布。

对变量的解释：

$X=\{x_1,\cdots,x_N\}$是 N 个训练样本，每一项都是服从多元高斯分布的 K 维向量。

$Z=\{z_1,\cdots,z_N\}$是一组潜在变量，每一项$z_k=\{z_{1k},\cdots,z_{nk}\}$用于表示对应的样本$x_k$属于哪个混合部分。

$\pi=\{\pi_1,\cdots,\pi_K\}$表示每个单高斯分布混合比例。

$\mu_{i=1,\cdots,k}$和$\Lambda_{i=1,\cdots,k}$分别表示每个单高斯分布参数的均值和精度。

另外，为了区分联合分布的参数，以上分布的参数如 K, $\alpha_0,\beta_0,w_0,v_0,m_0$称为超参数，并且都是已知量。

步骤二：写出联合概率密度函数

用“盘子表示法”表示贝叶斯多元高斯混合模型，如下图所示。

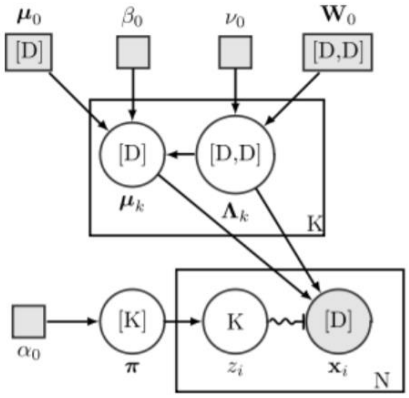


图 10 贝叶斯多元高斯混合模型的“盘子表示法”

小正方形表示不变的超参数，如$\alpha_0,\beta_0,w_0,v_0,\mu_0$；圆圈表示随机变量，如$\pi,z_i,x_i,\mu_k,\Lambda_k$；圆圈内的值为已知量。其中[K],[D]表示 K、D 维的向量，[D,D]表示 D×D 的矩阵，单个 K 表示一个有 K 个值的多项式分布变量。波浪线和一个开关表示变量x_i通过一个 K 维向量z_i来选择其他传入的变量(μ_k,Λ_k)

假设各参数与潜在变量条件独立，则联合概率密度函数可以表示为：

$$p(X,Z,\pi,\mu,\Lambda)=p(X|Z,\mu,\Lambda)p(Z|\pi)p(\pi)p(\mu|\Lambda)p(\Lambda)$$

每个因子为：

$$p(X|Z,\mu,\Lambda)=\prod_{n=1}^{N}\prod_{k=1}^{K}N\left(x_n|\mu_k,{\Lambda_k}^{-1}\right)^{z_{nk}}$$

$$p(Z|\pi)=\prod_{n=1}^{N}\prod_{k=1}^{K}{\pi_k}^{z_{nk}}$$

$$p(\pi)=\frac{\Gamma(K\alpha_0)}{\Gamma(\alpha_0)}\prod_{k=1}^{K}{\pi_k}^{\alpha_0-1}$$

$$p(\mu|\Lambda)=N(\mu_k|m_0,(\beta_0\Lambda_k)^{-1})$$

$$p(\Lambda)=W(\Lambda_k|w_0,v_0)$$

其中，

$$N(x|\mu,\sum)=\frac{1}{(2\pi)^{D/2}}\frac{1}{|\sum|^{1/2}}exp\left\{-\frac{1}{2}(x-\mu)^T\sum^{-1}(x-\mu)\right\}$$

$$W(\Lambda|w,v)=B(w,v)|\Lambda|^{(v-D-1)/2}exp\left(-\frac{1}{2}Tr(w^{-1}\Lambda)\right)$$

$$B(w,v)=|w|^{-v/2}\left(2^{vD/2}\pi^{D(D-1)/4}\prod_{i=1}^{D}\Gamma\left(\frac{v+1-i}{2}\right)\right)^{-1}$$

D 为各观测点的维度。

步骤三：计算边缘密度

（1） 计算 Z 的边缘密度,根据平均场假设，$q(Z,\pi,\mu,\Lambda)=q(Z)q(\pi,\mu,\Lambda)$

$$lnq^*(Z)=E_{\pi,\mu,\Lambda}[lnp(X,Z,\pi,\mu,\Lambda)]+const$$
$$=E_{\pi,\mu,\Lambda}[lnp(X|Z,\mu,\Lambda)p(Z|\pi)p(\pi)p(\mu|\Lambda)p(\Lambda)]+const$$
$$=E_{\pi}[lnp(Z|\pi)]+E_{\mu,\Lambda}[p(X|Z,\mu,\Lambda)]+const$$
$$=\sum_{n=1}^{N}\sum_{k=1}^{K}z_{nk}ln\rho_{nk}+const$$

其中，$ln\rho_{nk}=E[ln\pi_k]+\frac{1}{2}E[ln|\Lambda_k|]-\frac{D}{2}ln(2\pi)$

$$-\frac{1}{2}E_{\mu_k,\Lambda_k}[(x_n-\mu_k)^T\Lambda_k(x_n-\mu_k)]$$

两边分别取对数可得：

$$q^*(Z)\propto\prod_{n=1}^{N}\prod_{k=1}^{K}{\rho_{nk}}^{z_{nk}}$$

归一化，即对于观测变量的属于某个单高斯分布的概率相加应等于 1，则有

$$q^*(Z)\propto\prod_{n=1}^{N}\prod_{k=1}^{K}{r_{nk}}^{z_{nk}}$$

其中$r_{nk}=\frac{\rho_{nk}}{\sum_{j=1}^{K}\rho_{nj}}$

可见$q^*(Z)$是多个单观测多项式分布的乘积，可以因式分解成一个个以$r_{nk},(k=1,\cdots,K)$为参数的单观测多项式分布z_n。更进一步，根据 categorical 分布，有$E[z_{nk}]=r_{nk}$。

（2） 计算 π 的概率密度，$q(\pi,\mu,\Lambda)=q(\pi)\prod_{k=1}^{K}q(\mu_k,\Lambda_k)$

$$lnq^*(\pi)=E_{Z,\mu,\Lambda}[p(X|Z,\pi,\mu,\Lambda)]+const$$
$$lnp(\pi)+E_Z[lnp(Z|\pi)]+const$$
$$=(\alpha_0-1)\sum_{k=1}^{K}ln\pi_k+\sum_{n=1}^{N}\sum_{k=1}^{K}r_{nk}ln\pi_k+const$$

两边取对数$q^*(\pi)\sim\prod_{n=1}^{K}{\pi_k}^{\sum_{n=1}^{N}r_{nk}+\alpha_0-1}$，可见$q^*(\pi)$是 Dirichlet 分布，$q^*(\pi)\sim Dir(\alpha)$，其中$\alpha=\alpha_0+N_k$，$N_k=\sum_{n=1}^{N}r_{nk}$。

（3） 最后同时考虑，对于每一个单高斯分布有，

$$lnq^*(\mu_k,\Lambda_k)=E_{Z,\pi,\mu_{i\neq k},\Lambda_k}\left[lnp(X|Z,\mu_k,\Lambda_k)p(\mu_k,\Lambda_k)\right]$$

$$=lnp(\mu_k,\Lambda_k)+\sum_{n=1}^{N}E[z_{nk}]lnN\left(x_n|\mu_k,{\Lambda_k}^{-1}\right)+const$$

经过一系列重组化解将得到 Gaussian-Wishart 分布，

$$q^*(\mu_k,\Lambda_k)=N(\mu_k|m_k,(\beta_k\Lambda_k)^{-1})W(\Lambda_k|w_k,\mathrm{v}_k)$$

其中定义，

$$\begin{cases}\beta_k=\beta_0+N_k\\ m_k=\dfrac{1}{\beta_k}(\beta_0 m_0+N_k\bar{x}_k)\\ {w_k}^{-1}={w_0}^{-1}+N_kS_k+\dfrac{\beta_0N_k}{\beta_0+N_k}(\bar{x}_k-m_0)(\bar{x}_k-m_0)^T\\ v_k=v_0+N_k\\ \bar{x}_k=\dfrac{1}{N_k}\sum_{n=1}^{N}r_{nk}x_n\\ S_k=\dfrac{1}{N_k}\sum_{n=1}^{N}r_{nk}\,(\bar{x}_k-x_k)(\bar{x}_k-x_k)^T\end{cases}$$

步骤四：迭代收敛

最后，注意到对 π，μ，Λ的边缘概率都只需要r_{nk}；另一方面，r_{nk}的计算需要ρ_{nk}，而这又是基于$E[ln\pi_k]$，$E[ln\Lambda_k]$，$E[(\bar{x}_k-x_k)^T\Lambda_k(\bar{x}_k-x_k)]$，即需要知道π，$\mu$，Λ的值。不难确定这三个期望的一般表达式为：

$$\begin{cases}ln\widetilde{\pi_k}\equiv E[ln|\pi_k|]\equiv\psi(\alpha_k)-\psi\left(\sum_{i=1}^{K}\alpha_i\right)\\ ln\widetilde{\Lambda}_k\equiv E[ln|\Lambda_k|]\equiv\sum_{i=1}^{D}\psi\left(\dfrac{v_k+1-i}{2}\right)+Dln2+ln|\Lambda_k|\\ E_{\mu_k,\Lambda_k}[(x_k-\mu_k)^T\Lambda(x_k-\mu_k)]=D{\beta_k}^{-1}+v_k(x_n-m_k)^T\mathrm{W}_k(x_n-m_k)\end{cases}$$

这些结果能够导出：

$$r_{nk}\propto\widetilde{\pi_k}\widetilde{\Lambda_k}^{1/2}exp\left\{-\frac{D}{2\beta_k}-\frac{v_k}{2}(x_n-m_k)^T\mathrm{W}_k(x_n-m_k)\right\}$$

由于需要归一化使得$\sum_{k=1}^{K}r_{nk}=1$，这样可以从线性相对值转化为绝对值。

再次分析各参数，

1.参数变量μ_k，Λ_k更新方程中的超参数β_k，m_k，w_k，v_k都依赖于统计量N_k，$\bar{x}_k$，S_k，而这些统计量又依赖于r_{nk}。

2.参数变量π更新方程中的超参数$\alpha_{1\cdots K}$都依赖于统计量N_k，即r_{nk}。

3.潜在变量r_{nk}的更新方程对超变量β_k，m_k，w_k，v_k有直接的依赖关系，同时对w_k，v_k，$\alpha_{1\cdots K}$通过$\widetilde{\pi_k}$，$\widetilde{\Lambda_k}$有简洁的依赖关系。

这样迭代过程便很清楚了，可以总结为如下两个迭代步骤：

1.在 VBE-Step，用参数和超参数的旧值计算潜在变量r_{nk}。

2.在 VBM-Step，用潜在变量计算参数和超参数的新值。

1.4.3 变分贝叶斯与 EM 算法的不同

两者在迭代中，逼近最优值的过程是不一样的，如下图所示。在有限制的情况下，EM 算法极大似然值是动态变化的。刚开始与当前最优值相差一个 KL。在 E 步骤，下界逼近最大似然值（或者由于条件限制相差一点）；然后在 E 步骤中，根据新参数重新确定新的似然值。如此往复，直到达到稳定。而在 VBEM 算法中，极大似然值是不变的。VBE 与 VBM 步骤，都是逼近极大似然值的过程。

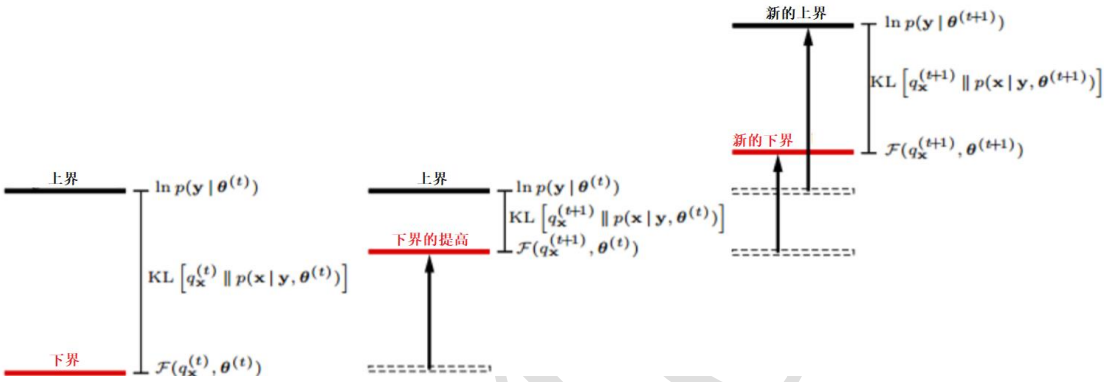


图 11(a) EM 算法中的极大似然值的变化情况

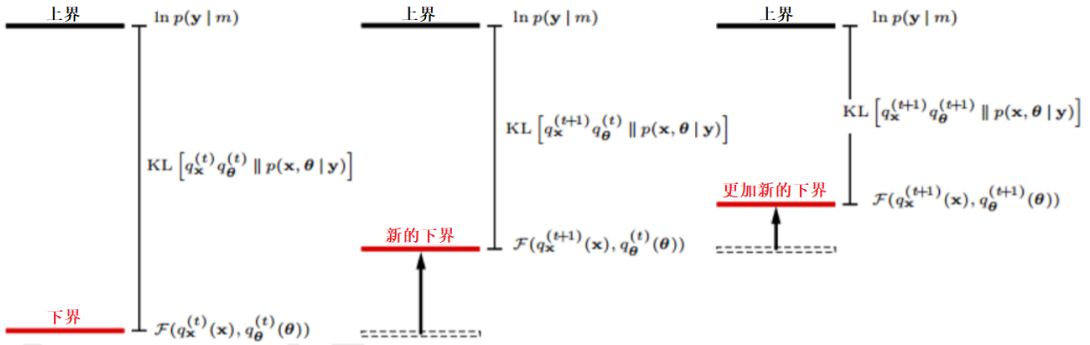


图 11(b) VBEM 算法中的极大似然值的变化情况

注：实验室师兄们应用变分贝叶斯方法处理张量形式热图数据的实际操作：

（1）首先对数据进行预处理，将视频每一帧的像素矩阵列向量化，再将这些列向量按时间顺序排列为矩阵，其过程如下图所示。

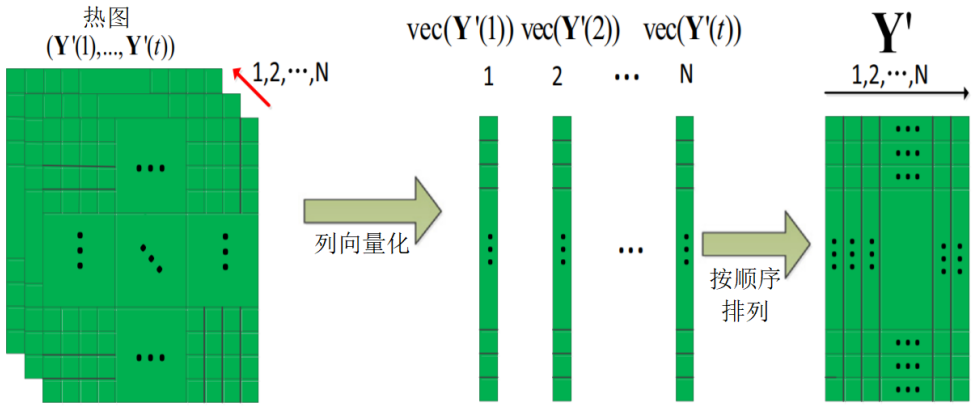


图 12 将张量数据转换为矩阵数据

（2）根据 RPCA 的思想，将预处理所得到的矩阵视为一个低秩矩阵、一个稀疏矩阵、一个噪声矩阵的叠加结果，即：

$$Y = X + S + N$$

（3）根据变分贝叶斯的思想设置超参数以及各参数之间的先验关系或独立关系。以下图为例。其中超参数$b_\lambda{}^1$，$b_\lambda{}^1$，$b_\gamma{}^1$，$b_\gamma{}^1$，$b_\tau{}^1$，$b_\tau{}^1$的值均需根据实际情况，人为预先设置。

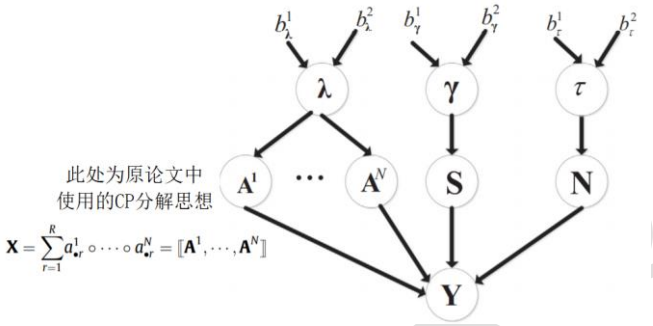


图 13 模型所包含参数的先验关系或独立关系

（4）利用变分贝叶斯的参数更新思想，固定其余所有参数以计算当前被更新参数的新值。

注：变分贝叶斯方法处理实验室采集的数据

ECPT 数据	大灯数据	便携式数据