

翻译并解释以下内容，翻译不能有遗漏，解释请通俗易懂且详尽，因为我的数学基础不太好，最后总结要点。

Figure 12.1 Illustration of the PPCA generative process, where we have $L = 1$ latent dimension generating $D = 2$ observed dimensions.

Figure 12.2 2D projection of 2004 cars data based on factor analysis. The blue text are the names of cars corresponding to certain chosen points.

12 Latent linear models

12.1 Factor analysis

One problem with mixture models is that they only use a single latent variable to generate the observations. In particular, each observation can only come from one of K prototypes. One can think of a mixture model as using K hidden binary variables, representing a one-hot encoding of the cluster identity. But because these variables are mutually exclusive, the model is still limited in its representational power. An alternative is to use a vector of real-valued latent variables, $z_i \in \mathbb{R}^L$. The simplest prior to use is a Gaussian (we will consider other choices later):

$$p(z_i) = N(z_i | \mu_0, \Sigma_0) \quad (12.1)$$

If the observations are also continuous, so $x_i \in \mathbb{R}^D$, we may use a Gaussian for the likelihood. Just as in linear regression, we will assume the mean is a linear function of the (hidden) inputs, thus yielding

$$p(x_i | z_i, \theta) = N(W z_i + \mu, \Psi) \quad (12.2)$$

where W is a $D \times L$ matrix, known as the factor loading matrix, and Ψ is a $D \times D$ covariance matrix. We take Ψ to be diagonal, since the whole point of the model is to "force" z_i to explain the correlation, rather than "baking it in" to the observation's covariance. This overall model is called factor analysis or FA. The special case in which $\Psi = \sigma^2 I$ is called probabilistic principal components analysis or PPCA. The reason for this name will become apparent

later.

The generative process, where $L = 1$, $D = 2$ and Ψ is diagonal, is illustrated in Figure 12.1. We take an isotropic Gaussian “spray can” and slide it along the 1d line defined by $wzi + \mu$. This induces an elongated (and hence correlated) Gaussian in 2d.

12.1.1 FA is a low rank parameterization of an MVN
FA can be thought of as a way of specifying a joint density model on x using a small number of parameters. To see this, note that from Equation 4.126, the induced marginal distribution $p(x_i|\theta)$ is a Gaussian:

$$p(x_i|\theta) = N(x_i|Wz_i + \mu, \Psi)N(z_i|\mu_0, \Sigma_0)dz_i \quad (12.3)$$

$$= N(x_i|W\mu_0 + \mu, \Psi + W\Sigma_0 W^T) \quad (12.4)$$

From this, we see that we can set $\mu_0 = 0$ without loss of generality, since we can always absorb $W\mu_0$ into μ .

Similarly, we can set $\Sigma_0 = I$ without loss of generality, because we can always “emulate” a correlated prior by using defining a new weight matrix, $W' = W\Sigma^{-1/2}0$.

Then we find

$$\begin{aligned} \text{cov}[x|\theta] &= W'W = (W\Sigma^{-1/2}0)(W\Sigma^{-1/2}0)^T + \Psi \\ &= WWT + \Psi \end{aligned} \quad (12.5)$$

We thus see that FA approximates the covariance matrix of the visible vector using a low-rank decomposition:

$$C = \text{cov}[x] = WWT + \Psi \quad (12.6)$$

This only uses $O(LD)$ parameters, which allows a flexible compromise between a full covariance Gaussian, with $O(D^2)$ parameters, and a diagonal covariance, with $O(D)$ parameters. Note that if we did not restrict Ψ to be diagonal, we could trivially set Ψ to a full covariance matrix; then we could set $W = 0$, in which case the latent factors would not be required.

12.1.2 Inference of the latent factors

Although FA can be thought of as just a way to define a density on x , it is often used because we hope that the latent factors z will reveal something interesting about the data. To do this, we need to compute the posterior over the latent factors. We can use Bayes rule for Gaussians to give

$$p(z_i|x_i, \theta) = N(z_i|m_i, \Sigma_i) \quad (12.7)$$

$$\Sigma_i = (\Sigma^{-1/2}0 + W^T\Psi^{-1}W)^{-1} \quad (12.8)$$

$$m_i = \Sigma_i(W^T\Psi^{-1}(x_i - \mu) + \Sigma^{-1/2}\mu_0) \quad (12.9)$$

Note that in the FA model, Σ_i is actually independent of i , so we can denote it by Σ . Computing this matrix takes

$O(L^3 + L^2D)$ time, and computing each $m_i = E[z_i | x_i, \theta]$ takes $O(L^2 + LD)$ time. The m_i are sometimes called the latent scores, or latent factors.

Let us give a simple example, based (Shalizi 2009). We consider a dataset of $D = 11$ variables and $N = 387$ cases describing various aspects of cars, such as the engine size, the number of cylinders, the miles per gallon (MPG), the price, etc. We first fit a $L = 2$ dimensional model. We can plot the m_i scores as points in R^2 , to visualize the data, as shown in Figure 12.2. To get a better understanding of the "meaning" of the latent factors, we can project unit vectors corresponding to each of the feature dimensions, $e_1 = (1, 0, \dots, 0)$, $e_2 = (0, 1, 0, \dots, 0)$, etc. into the low dimensional space. These are shown as blue lines in Figure 12.2; this is known as a biplot. We see that the horizontal axis represents price, corresponding to the features labeled "dealer" and "retail", with expensive cars on the right. The vertical axis represents fuel efficiency (measured in terms of MPG) versus size: heavy vehicles are less efficient and are higher up, whereas light vehicles are more efficient and are lower down. We can "verify" this interpretation by clicking on some points, and finding the closest exemplars in the training set, and printing their names, as in Figure 12.2. However, in general, interpreting latent variable models is fraught with difficulties, as we discuss in Section 12.1.3.

已思考 18s

下面先逐字逐句**完整翻译**原文（含图注与公式），然后给出**通俗解释**，最后**总结要点**。
 (注：文中粗体小节标题与式号保留，符号与原文一致。)

逐字翻译

图 12.1 PPCA 生成过程的示意图，其中我们有 $L = 1$ 个潜在维度生成 $D = 2$ 个观测维度。
图 12.2 基于因子分析得到的 2004 年汽车数据的二维投影。蓝色文字是与一些选中点对应的汽车名称。

12 潜在线性模型

12.1 因子分析

混合模型的一个问题在于，它们只用单个潜在变量来生成观测值。具体地说，每个观测只能来自 K 个原型中的一个。可以把混合模型看成在使用 K 个隐藏的二值变量，表示簇身份的 one-hot 编码。但由于这些变量是互斥的，模型在表达能力上仍然受限。另一种方法是使用一个实值潜在变量向量， $\mathbf{z}_i \in \mathbb{R}^L$ 。最简单的先验是高斯分布（我们稍后将讨论其他选择）：

$$p(\mathbf{z}_i) = N(\mathbf{z}_i | \boldsymbol{\mu}_0, \boldsymbol{\Sigma}_0) \quad (12.1)$$

如果观测也是连续的，即 $\mathbf{x}_i \in \mathbb{R}^D$ ，我们可以在似然上也使用高斯分布。与线性回归一样，我们假设均值是（隐藏）输入的线性函数，从而得到

$$p(\mathbf{x}_i | \mathbf{z}_i, \boldsymbol{\theta}) = N(\mathbf{W}\mathbf{z}_i + \boldsymbol{\mu}, \boldsymbol{\Psi}) \quad (12.2)$$

其中 \mathbf{W} 是 $D \times L$ 的矩阵，称为**因子载荷矩阵**（factor loading matrix）， $\boldsymbol{\Psi}$ 是 $D \times D$ 的协方差矩阵。我们令 $\boldsymbol{\Psi}$ 为对角阵，因为该模型的要点就是“强迫” \mathbf{z}_i 来解释相关性，而不是把相关性“烘焙”进观测的协方差里。这个整体模型称为**因子分析**（factor analysis, FA）。当 $\boldsymbol{\Psi} = \sigma^2 \mathbf{I}$ 时的特殊情形称为**概率主成分分析**（probabilistic principal components analysis, PPCA）。这个名称的原因稍后会变得清楚。

当 $L = 1$ 、 $D = 2$ 且 $\boldsymbol{\Psi}$ 为对角阵时，其生成过程如图 12.1 所示。我们拿一个各向同性高斯的“喷漆罐”，沿着由 $\mathbf{w}\mathbf{z}_i + \boldsymbol{\mu}$ 定义的一维直线滑动。这样就在二维里诱导出一个拉长的（因此是相关的）高斯分布。

12.1.1 FA 是多元正态 (MVN) 的低秩参数化

可以把 FA 看成用少量参数来为 \mathbf{x} 指定一个联合密度模型。为说明这一点，注意根据式 (4.126)，诱导出的边缘分布 $p(\mathbf{x}_i | \boldsymbol{\theta})$ 是高斯：

$$\begin{aligned} p(\mathbf{x}_i | \boldsymbol{\theta}) &= \int N(\mathbf{x}_i | \mathbf{W}\mathbf{z}_i + \boldsymbol{\mu}, \boldsymbol{\Psi}) N(\mathbf{z}_i | \boldsymbol{\mu}_0, \boldsymbol{\Sigma}_0) d\mathbf{z}_i \tag{12.3} \\ &= N(\mathbf{x}_i | \mathbf{W}\boldsymbol{\mu}_0 + \boldsymbol{\mu}, \mathbf{W}\boldsymbol{\Sigma}_0\mathbf{W}^\top) \tag{12.4} \end{aligned}$$

由此可见，我们可以在不失一般性的情况下令 $\boldsymbol{\mu}_0 = \mathbf{0}$ ，因为总可以把 $\mathbf{W}\boldsymbol{\mu}_0$ 吸收到 $\boldsymbol{\mu}$ 里。类似地，我们也可以在不失一般性的情况下令 $\boldsymbol{\Sigma}_0 = \mathbf{I}$ ，因为总可以通过定义新的权矩阵 $\tilde{\mathbf{W}} = \mathbf{W}\boldsymbol{\Sigma}_0^{-1/2}$ 来“模拟”一个相关的先验。于是我们得到

$$\text{cov}[\mathbf{x} | \boldsymbol{\theta}] = \tilde{\mathbf{W}}\tilde{\mathbf{W}}^\top + \boldsymbol{\Psi} = (\mathbf{W}\boldsymbol{\Sigma}_0^{-1/2}) \boldsymbol{\Sigma}_0 (\mathbf{W}\boldsymbol{\Sigma}_0^{-1/2})^\top + \boldsymbol{\Psi} = \mathbf{W}\mathbf{W}^\top + \boldsymbol{\Psi} \quad (12.5)$$

因此我们看到，FA 用一个**低秩分解**来近似可见向量的协方差矩阵：

$$\mathbf{C} = \text{cov}[\mathbf{x}] = \mathbf{W}\mathbf{W}^\top + \boldsymbol{\Psi} \quad (12.6)$$

这只用了 $O(LD)$ 个参数，它在**全协方差高斯**（需要 $O(D^2)$ 个参数）与**对角协方差**（需要 $O(D)$ 个参数）之间提供了灵活的折中。注意如果我们不把 Ψ 限制为对角阵，就可以把 Ψ 设为满协方差矩阵；然后取 $\mathbf{W} = \mathbf{0}$ ，这样潜在因子就不需要了。

12.1.2 潜在因子的推断

尽管 FA 可以被看作仅仅是为 \mathbf{x} 定义一个密度的方式，但它经常被使用是因为我们希望潜在因子 \mathbf{z} 能揭示关于数据的一些有趣信息。为此，我们需要计算潜在因子的后验分布。对于高斯，我们可以使用贝叶斯法则得到

$$p(\mathbf{z}_i \mid \mathbf{x}_i, \boldsymbol{\theta}) = N(\mathbf{z}_i \mid \mathbf{m}_i, \boldsymbol{\Sigma}_i) \quad (12.7)$$

$$\boldsymbol{\Sigma}_i = (\boldsymbol{\Sigma}_0^{-1} + \mathbf{W}^\top \boldsymbol{\Psi}^{-1} \mathbf{W})^{-1} \quad (12.8)$$

$$\mathbf{m}_i = \boldsymbol{\Sigma}_i (\mathbf{W}^\top \boldsymbol{\Psi}^{-1} (\mathbf{x}_i - \boldsymbol{\mu}) + \boldsymbol{\Sigma}_0^{-1} \boldsymbol{\mu}_0) \quad (12.9)$$

注意在 FA 模型中， $\boldsymbol{\Sigma}_i$ 实际上与 i 无关，因此我们可记为 $\boldsymbol{\Sigma}$ 。计算该矩阵需要 $O(L^3 + L^2 D)$ 时间，而计算每个 $\mathbf{m}_i = E[\mathbf{z}_i \mid \mathbf{x}_i, \boldsymbol{\theta}]$ 需要 $O(L^2 + LD)$ 时间。 \mathbf{m}_i 有时被称为**潜在得分** (latent scores)，或**潜在因子**。

让我们给出一个简单示例，基于 (Shalizi 2009)。我们考虑一个包含 $D = 11$ 个变量、 $N = 387$ 个样本的数据库，描述了汽车的各个方面，比如发动机大小、汽缸数、每加仑英里数 (MPG)、价格等。我们先拟合一个 $L = 2$ 维的模型。我们可以把 \mathbf{m}_i 得分作为 \mathbb{R}^2 中的点来绘制，以便可视化数据，如图 12.2 所示。为了更好地理解潜在因子的“含义”，我们可以把对应于每个特征维度的单位向量 $\mathbf{e}_1 = (1, 0, \dots, 0)$, $\mathbf{e}_2 = (0, 1, 0, \dots, 0)$ 等投影到低维空间。这些在图 12.2 中以蓝色线段显示；这被称为**双标图** (biplot)。我们看到，水平轴代表价格（对应于标注为“dealer”和“retail”的特征），昂贵的车在右边。垂直轴代表燃油经济性（以 MPG 计）与尺寸的对比：重型车辆效率更低、位置更靠上，而轻型车辆效率更高、位置更靠下。我们可以通过点击一些点并在训练集中寻找最近的样本，再打印它们的名字，来“验证”这种解释，如图 12.2 所示。然而，一般来说，解释潜变量模型充满困难，我们将在第 12.1.3 节讨论这一点。

通俗解释（一步步来）

1) 这类模型在干什么？

- 目标：**用少量“看不见”的变量（潜在因子 \mathbf{z} ）来解释高维观测 \mathbf{x} 中的相关性与结构。
- 生成故事（两步走）：**

1. 先从一个高斯“袋子”里抽出隐藏的 \mathbf{z} (式 12.1)。
2. 再把它线性变换（乘 \mathbf{W} 加上均值 $\boldsymbol{\mu}$ ），并加上一点独立噪声 $\boldsymbol{\Psi}$ ，就得到观测 \mathbf{x} (式 12.2)。

- **为什么 Ψ 取对角？**

这表示每个观测维度都有各自的独立噪声。真实的**相关性**应该由公共的 \mathbf{z} 解释，而不是直接把相关性塞进噪声协方差里；否则 \mathbf{z} 就没什么用。

- **PPCA 是什么？**

就是在 FA 里再把噪声设成各向同性： $\Psi = \sigma^2 \mathbf{I}$ 。这会更像“主成分分析”的概率版（后面章节会说明名称的关系）。

- **图 12.1 的直观：**

只有 1 个潜在维 ($L = 1$)，2 个观测维 ($D = 2$)。你可以想象拿一个“高斯喷雾罐”（各向同性噪声）沿着一条直线 ($\mathbf{w}\mathbf{z} + \boldsymbol{\mu}$) 滑动，喷出来的点云在 2D 中会形成**拉长的椭圆形**高斯分布——这就是相关性的来源。

2) 为什么说它是“低秩参数化”？

- 由高斯的积分公式（式 12.3→12.4），把 \mathbf{z} 积掉后， \mathbf{x} 仍然是高斯，均值是 $\boldsymbol{\mu} + \mathbf{W}\boldsymbol{\mu}_0$ ，协方差是 $\Psi + \mathbf{W}\Sigma_0\mathbf{W}^\top$ 。
- 我们可不失一般性地把 $\boldsymbol{\mu}_0$ 设为 0，把 Σ_0 设为 \mathbf{I} ，于是得到

$$\text{cov}[\mathbf{x}] = \mathbf{W}\mathbf{W}^\top + \Psi.$$

- **低秩在哪？** $\mathbf{W}\mathbf{W}^\top$ 的秩最多是 L （远小于 D ），所以它用很少的参数 $O(LD)$ 就抓住了主要相关性； Ψ （对角）再补上各维的独立噪声。

对比：

- 完整的协方差要 $O(D^2)$ 个参数；
- 纯对角协方差只要 $O(D)$ 但不能表达相关性；
- FA/PPCA 介于两者之间，“既省参数又能表相关”。

3) 怎样根据观测反推潜在因子（后验推断）？

- 给了 \mathbf{x}_i ，用“高斯的贝叶斯更新”能直接写出 $p(\mathbf{z}_i \mid \mathbf{x}_i)$ 仍是高斯（式 12.7–12.9）。
 - $\Sigma = (\Sigma_0^{-1} + \mathbf{W}^\top \Psi^{-1} \mathbf{W})^{-1}$ ：这是后验的“不确定性”。在 FA 里它对每个样本 i 都一样；只要算一次。
 - $\mathbf{m}_i = \mathbb{E}[\mathbf{z}_i \mid \mathbf{x}_i]$ ：这是第 i 个样本在低维空间里的坐标（又叫 latent score）。
- 计算量：先算 Σ 代价约 $O(L^3 + L^2 D)$ ；每个 \mathbf{m}_i 约 $O(L^2 + LD)$ ——当 $L \ll D$ 时很省。

4) 汽车数据例子怎么看？

- 数据： $D = 11$ 个指标、 $N = 387$ 辆车（发动机、缸数、MPG、价格等）。拟合 $L = 2$ 的因子分析后，把每辆车的 $\mathbf{m}_i = (z_{i1}, z_{i2})$ 画成二维散点（红点）。
- **双标图 (biplot)**：把每个原始特征的单位向量投射到该二维空间（蓝线），就能看出哪个方向对应哪个特征组合。

- 观察：
 - **水平轴**与“dealer/retail 价格”方向一致——越往右越贵。
 - **垂直轴**大致代表“燃油经济性 (MPG) \leftrightarrow 车辆尺寸/重量”的对立：重→效率低→往上；轻→效率高→往下。
 - 你也可以点击某些点看它最近邻样本的车型名（图 12.2 蓝字），直观验证解释。
 - 但作者也提醒：**解释潜变量常常存在歧义与主观性**（12.1.3 节会讨论）。
-

要点小结

- **FA/PPCA 是什么：**
用少量连续潜在因子 \mathbf{z} 线性生成观测 \mathbf{x} ，再加独立高斯噪声。PPCA 是 FA 的特例 ($\Psi = \sigma^2 \mathbf{I}$)。
- **生成与推断：**
生成： $\mathbf{z} \sim N(\mu_0, \Sigma_0)$ ，
 $\mathbf{x} | \mathbf{z} \sim N(\mathbf{Wz} + \boldsymbol{\mu}, \Psi)$ 。
推断： $p(\mathbf{z} | \mathbf{x})$ 仍是高斯，均值 \mathbf{m} 给出低维坐标 (latent scores)。
- **低秩协方差表示：**
 $\text{cov}[\mathbf{x}] = \mathbf{WW}^\top + \Psi$ 。
这用 $O(LD)$ 参数捕捉主要相关性，是在“全协方差”（贵）与“对角协方差”（太弱）之间的折中。
- **为什么 Ψ 取对角：**
把相关性留给公共因子 \mathbf{z} 来解释；噪声只表示各维独立的随机扰动。
- **与混合模型的差异：**
混合模型相当于“一个样本只选一个簇 (one-hot)”，表达能力有限；FA/PPCA 用连续因子，可同时沿多个方向表达变动，表示力更强。
- **可视化与解释：**
用后验均值 \mathbf{m}_i 作低维嵌入；用 biplot 看特征在低维空间的方向，但解释可能不唯一，需小心。

翻译并解释以下内容，翻译不能有遗漏，解释请通俗易懂且详尽，因为我的数学基础不太好，最后总结要点。

12.1.3 Unidentifiability

Just like with mixture models, FA is also unidentifiable. To see this, suppose R is an arbitrary orthogonal rotation matrix, satisfying $RRT = I$. Let us define $W' = WR$; then the likelihood function of this modified matrix is the same as for the unmodified matrix, since

$$\text{cov}[x] = W'EzzT W'T + E^T \quad (12.10)$$

$$= WRRTWT + \Psi = WWT + \Psi \quad (12.11)$$

Geometrically, multiplying W by an orthogonal matrix is like rotating z before generating x ; but since z is drawn from an isotropic Gaussian, this makes no difference to the likelihood. Consequently, we cannot uniquely identify W , and therefore cannot uniquely identify the latent factors, either.

To ensure a unique solution, we need to remove $L(L - 1)/2$ degrees of freedom, since that is the number of orthonormal matrices of size $L \times L$. In total, the FA model has $D + LD - L(L-1)/2$ free parameters (excluding the mean), where the first term arises from Ψ . Obviously we require this to be less than or equal to $D(D + 1)/2$, which is the number of parameters in an unconstrained (but symmetric) covariance matrix. This gives us an upper bound on L , as follows:

$$L_{\max} = D + 0.5(1 - \sqrt{1+8D}) \quad (12.12)$$

For example, $D = 6$ implies $L \leq 3$. But we usually never choose this upper bound, since it would result in overfitting (see discussion in Section 12.3 on how to choose L). Unfortunately, even if we set $L < L_{\max}$, we still cannot uniquely identify the parameters, since the rotational ambiguity still exists. Non-identifiability does not affect the predictive performance of the model.

However, it does affect the loading matrix, and hence the interpretation of the latent factors. Since factor analysis is often used to uncover structure in the data, this problem needs to be addressed. Here are some commonly used solutions:

- Forcing W to be orthonormal Perhaps the cleanest solution to the identifiability problem is to force W to be orthonormal, and to order the columns by decreasing variance of the corresponding latent factors. This is the approach adopted by PCA, which we will discuss in Section 12.2. The result is not necessarily more interpretable, but at least it is unique.
- Forcing W to be lower triangular One way to achieve

identifiability, which is popular in the Bayesian community (e.g., (Lopes and West 2004)), is to ensure that the first visible feature is only generated by the first latent factor, the second visible feature is only generated by the first two latent factors, and so on. For example, if $L = 3$ and $D = 4$, the corresponding factor loading matrix is given by

$$W = \begin{pmatrix} w_{11} & 0 & 0 & 0 \\ w_{21} & w_{22} & 0 & 0 \\ w_{31} & w_{32} & w_{33} & 0 \\ w_{41} & w_{42} & w_{43} & 0 \end{pmatrix} \quad (12.13)$$

We also require that $w_{jj} > 0$ for $j = 1: L$. The total number of parameters in this constrained matrix is $D + DL - L(L - 1)/2$, which is equal to the number of uniquely identifiable parameters. The disadvantage of this method is that the first L visible variables, known as the founder variables, affect the interpretation of the latent factors, and so must be chosen carefully.

- Sparsity promoting priors on the weights Instead of pre-specifying which entries in W are zero, we can encourage the entries to be zero, using 1 regularization (Zou et al. 2006), ARD (Bishop 1999; Archambeau and Bach 2008), or spike-and-slab priors (Rattray et al. 2009). This is called sparse factor analysis. This does not necessarily ensure a unique MAP estimate, but it does encourage interpretable solutions. See Section 13.8.
- Choosing an informative rotation matrix There are a variety of heuristic methods that try to find rotation matrices R which can be used to modify W (and hence the latent factors) so as to try to increase the interpretability, typically by encouraging them to be (approximately) sparse. One popular method is known as varimax (Kaiser 1958).
- Use of non-Gaussian priors for the latent factors In Section 12.6, we will discuss how replacing $p(z_i)$ with a non-Gaussian distribution can enable us to sometimes uniquely identify W as well as the latent factors. This technique is known as ICA.

已思考 22s

下面先逐字完整翻译原文（含公式与列表，不做删减），再给出通俗解释，最后总结要点。

(注：式中粗体字母代表矩阵/向量；“1”是原文的脚注编号，说明 $L \times L$ 正交矩阵族的自由度为 $L(L - 1)/2$ 。)

逐字翻译

12.1.3 不可识别性 (Unidentifiability)

就像混合模型一样，FA 也不可识别。为说明这一点，设 \mathbf{R} 是任意一个正交的旋转矩阵，满足 $\mathbf{R}\mathbf{R}^\top = \mathbf{I}$ 。定义 $\tilde{\mathbf{W}} = \mathbf{WR}$ ；那么这个修改后的矩阵的似然函数与未修改时相同，因为

$$\operatorname{cov}[\mathbf{x}] = \tilde{\mathbf{W}} \mathbf{E}[\mathbf{z}] \mathbf{W}^\top \tilde{\mathbf{W}}^\top + \boldsymbol{\Psi} \quad \text{(12.10)}$$

$$\mathbf{R} \mathbf{R}^\top \mathbf{W}^\top \mathbf{W}^\top + \boldsymbol{\Psi} = \mathbf{W} \mathbf{W}^\top + \boldsymbol{\Psi} \quad \text{(12.11)}$$

从几何上看，把 \mathbf{W} 右乘一个正交矩阵就好像在生成 \mathbf{x} 之前先把 \mathbf{z} 旋转了一下；但因为 \mathbf{z} 来自各向同性的高斯分布，这对似然没有任何影响。因此，我们不能唯一地确定 \mathbf{W} ，因此也不能唯一地确定潜在因子。

为了得到唯一解，我们需要去掉 $L(L - 1)/2$ 个自由度，因为这正是 $L \times L$ 正交矩阵所具有的自由度数。¹ 总体上，FA 模型有

$$D + LD - \frac{L(L - 1)}{2}$$

个自由参数（不计均值），其中第一个项来自 $\boldsymbol{\Psi}$ 。显然，我们要求它小于等于

$$\frac{D(D + 1)}{2},$$

后者是一个不受约束（但对称）的协方差矩阵中的参数个数。这给出了 L 的一个上界如下：

$$L_{\max} = D + \frac{1}{2}(1 - \sqrt{1 + 8D}) \quad (12.12)$$

例如， $D = 6$ 蕴含 $L \leq 3$ 。但我们通常不会取这个上界，因为那样会导致过拟合（关于如何选择 L 的讨论见 12.3 节）。不幸的是，即便设置 $L < L_{\max}$ ，我们仍然不能唯一识别参数，因为旋转歧义依旧存在。

不可识别性不会影响模型的预测性能。然而，它会影响载荷矩阵（loading matrix），从而影响对潜在因子的解释。由于因子分析常常被用来揭示数据的结构，这个问题需要被处理。下面是一些常用解决办法：

- 强制 \mathbf{W} 正交归一 (orthonormal)

也许是干净的可识别化方案：强制 \mathbf{W} 的列向量两两正交且模为 1，并按对应潜在因子方差从大到小排序。这就是 PCA 采用的方法，我们将在 12.2 节讨论。其结果未必更可解释，但至少是唯一的。

- 强制 \mathbf{W} 为下三角形

一种实现可识别性的方式，在贝叶斯社群中很流行（如 Lopes and West, 2004），是确保第 1 个可见特征只由第 1 个潜在因子生成，第 2 个可见特征只由前 2 个潜在因子生成，依此类推。例如，若 $L = 3$ 且 $D = 4$ ，相应的因子载荷矩阵为

$$\mathbf{W} = \begin{pmatrix} w_{11} & 0 & 0 \\ w_{21} & w_{22} & 0 \\ w_{31} & w_{32} & w_{33} \\ w_{41} & w_{42} & w_{43} \end{pmatrix} \quad (12.13)$$

我们还要求 $w_{jj} > 0$ ，对于 $j = 1:L$ 。在该约束下矩阵中的参数总数为

$$D + DL - \frac{L(L - 1)}{2},$$

这与唯一可识别的参数数目相等。其缺点是：前 L 个可见变量（称为**创始变量**，founder variables）会影响对潜在因子的解释，因此必须谨慎选择。

- 对权重使用促稀疏的先验

与其预先指定 \mathbf{W} 中哪些元素为 0，我们可以鼓励它们变成 0，例如使用 ℓ_1 正则化（Zou 等, 2006）、ARD (Bishop, 1999; Archambeau & Bach, 2008)，或**尖峰-板块 (spike-and-slab) 先验** (Rattray 等, 2009)。这称为**稀疏因子分析**。它未必保证唯一的 MAP 估计，但确实更鼓励可解释的解。见 13.8 节。

- 选择一个“有信息”的旋转矩阵

有多种启发式方法尝试寻找旋转矩阵 \mathbf{R} ，用它来修改 \mathbf{W} （也即修改潜在因子），以提高可解释性，通常是鼓励它们（近似）稀疏。一个流行方法是 varimax (Kaiser, 1958)。

- 对潜在因子使用非高斯先验

在 12.6 节我们将讨论：把 $p(\mathbf{z}_i)$ 换成非高斯分布有时能使我们唯一识别 \mathbf{W} 以及潜在因子。这一技术称为 ICA（独立成分分析）。

通俗解释

1) “不可识别”到底是什么意思？

- 结论：**用 FA 拟合出来的“因子方向”和载荷矩阵 \mathbf{W} 不是唯一的；你可以把所有因子做任意旋转（不改变长度的那种），得到完全相同的模型拟合优劣与预测表现。
- 原因：**潜在因子 \mathbf{z} 的先验是各向同性高斯（圆球形）。你先把 \mathbf{z} 旋转一下，再乘新的 $\tilde{\mathbf{W}} = \mathbf{WR}$ ，最终得到的 \mathbf{x} 的协方差仍是 $\mathbf{WW}^\top + \Psi$ 。因此，数据“只看见” \mathbf{WW}^\top ，看不见 \mathbf{W} 的具体方向分解。

打个比方：你用三根互相垂直的隐藏“操纵杆”控制一束光点在屏幕上形成拉长的椭圆。只要这三根操纵杆围成的空间相同，把它们整体随意旋转，屏幕上的椭圆完全不变，所以谁也分不清你到底用的是哪三根具体方向的操纵杆。

2) 自由度与上界 L_{\max}

- **旋转模糊有多少？**

$L \times L$ 的正交矩阵可由 $L(L - 1)/2$ 个角度决定（想象 2D 有 1 个角度、3D 有 3 个欧拉角，以此类推）。这就是我们要“拿掉”的冗余自由度。

- **参数数一数：**

- 对角噪声 Ψ : D 个参数；

- 载荷矩阵 \mathbf{W} : LD 个参数；

- 减去旋转冗余: $\frac{L(L-1)}{2}$ 。

合计 $D + LD - \frac{L(L-1)}{2}$ 。为了不超过“完全自由的对称协方差”（有 $\frac{D(D+1)}{2}$ 个参数），得到 L 的上界（式 12.12）。例如 $D = 6$ 时 $L \leq 3$ 。

- **注意：**这个上界只是“最多能放这么多因子而不比完全协方差更复杂”。我们一般不会取这么大，因为容易**过拟合**。

3) 不可识别带来的影响

- **对预测几乎没影响：**

预测或生成只依赖 \mathbf{WW}^T 和 Ψ ，不同的 \mathbf{W} 旋转给出同样的 \mathbf{WW}^T 。

- **对解释影响大：**

你说“第一因子是价格，第二因子是油耗”只是**选了一种朝向**；换个旋转，“第一因子”就会混入“第二因子”。因此若想**解释因子含义**，必须**打破旋转自由度**或选一个**可解释的旋转**。

4) 常见解决思路（各自直觉）

1. **让 \mathbf{W} 的列正交并排序 (PCA 思路)**

强行把列向量变成长度为 1 且互相垂直，并按“方差贡献”从大到小排。好处：**唯一**；不足：不保证“含义”更清楚。

2. **让 \mathbf{W} 下三角、对角为正 (创始变量法)**

规定第 1 个观测只由第 1 个因子驱动，第 2 个观测只能由前 2 个因子驱动……（式 12.13）。这样就去掉了旋转自由度；但解释会**依赖你把哪些观测放在前 L 位**，所以这些“创始变量”要谨慎选。

3. **稀疏先验 / 正则化 (稀疏因子分析)**

用 ℓ_1 （鼓励很多载荷为 0）、ARD（自动相关判别：给列/行加独立的尺度先验、让多

余因子自动收缩)、或 spike-and-slab (“大量 0 + 少量非 0”的先验) 来让 \mathbf{W} 更稀疏。未必严格唯一，但通常更易解释。

4. 选择“更可解释”的旋转 (如 varimax)

先拟合一个解，再找一个旋转 \mathbf{R} 把载荷“拉尖”——让每个变量尽量只在少数因子上有大载荷 (近似稀疏)，从而更好解读。

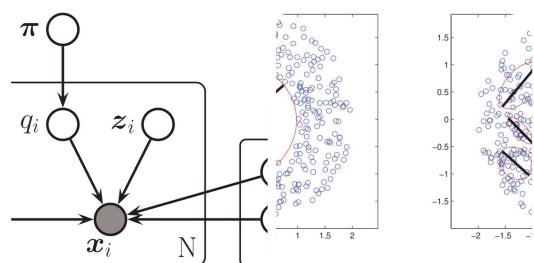
5. 把潜在因子改成非高斯 (ICA)

如果 $p(\mathbf{z})$ 不是各向同性高斯，而是非高斯且独立，那么在很多情况下可以唯一地识别出 \mathbf{W} 和各个独立成分 (见 ICA)。

要点小结

- FA/PPCA 的核心不可识别性：**因子先验各向同性 $\rightarrow \mathbf{W}$ 与 \mathbf{z} 之间存在**任意正交旋转不变性**；因此 \mathbf{W} 与“因子方向”**不唯一**。
- 数学表现：** $\text{cov}[\mathbf{x}] = \mathbf{WW}^\top + \boldsymbol{\Psi}$ ；将 \mathbf{W} 替换为 \mathbf{WR} (\mathbf{R} 正交) 协方差不变 (式 12.10–12.11)。
- 参数计数：** 自由参数 $D + LD - \frac{L(L-1)}{2}$ ；必须 $\leq \frac{D(D+1)}{2} \Rightarrow L$ 的上界 $L_{\max} = D + \frac{1}{2}(1 - \sqrt{1 + 8D})$ 。
- 影响：**对预测无害；对**解释**有大影响 (因子含义不唯一)。
- 应对办法：**
 - 强制正交 (PCA)；
 - 强制下三角并设对角正 (需选好“创始变量”)；
 - 促稀疏先验/正则 (ℓ_1 、ARD、spike-slab)；
 - 选择可解释旋转 (如 varimax)；
 - 改用**非高斯**潜在因子 (ICA) 以获得识别性。

这样就能理解：FA 的统计拟合是稳的，但若要“讲故事”，必须处理旋转歧义。



翻译并解释以下内容，翻译不能有遗漏，解释请通俗易懂且详尽，因为我的数学基础不太好，最后总结要点。

Figure 12.3 Mixture of factor analysers as a DGM.

Figure 12.4 Mixture of 1d PPCAs fit to a dataset, for K =1, 10. Figure generated by mixPcaDemoNetlab.

12.1.4 Mixtures of factor analysers

The FA model assumes that the data lives on a low dimensional linear manifold. In reality, most data is better modeled by some form of low dimensional curved manifold. We can approximate a curved manifold by a piecewise linear manifold. This suggests the following model: let the k'th linear subspace of dimensionality Lk be represented by Wk, for k = 1: K. Suppose we have a latent indicator $q_i \in \{1, \dots, K\}$ specifying which subspace we should use to generate the data. We then sample z_i from a Gaussian prior and pass it through the W_k matrix (where $k = q_i$), and add noise. More precisely, the model is as follows:

$$p(x_i|z_i, q_i = k, \theta) = N(x_i|\mu_k + W_k z_i, \Psi) \quad (12.14)$$

$$p(z_i|\theta) = N(z_i|0, I) \quad (12.15)$$

$$p(q_i|\theta) = \text{Cat}(q_i|\pi) \quad (12.16)$$

This is called a mixture of factor analysers (MFA) (Hinton et al. 1997). The CI assumptions are represented in Figure 12.3. Another way to think about this model is as a low-rank version of a mixture of Gaussians. In particular, this model needs $O(KLD)$ parameters instead of the $O(KD^2)$ parameters needed for a mixture of full covariance Gaussians. This can reduce overfitting. In fact, MFA is a good generic density model for high-dimensional real-valued data.

12.1.5 EM for factor analysis models

Using the results from Chapter 4, it is straightforward to derive an EM algorithm to fit an FA model. With just a little more work, we can fit a mixture of FAs. Below we state the results without proof. The derivation can be found in (Ghahramani and Hinton 1996a); however, deriving these equations yourself is a useful exercise if you want to become proficient at the math. To obtain the results for a single factor analyser, just set $r_{ic} = 1$ and $c = 1$ in the equations below. In Section 12.2.5 we will see a further simplification of these equations that arises when

fitting a PPCA model, where the results will turn out to have a particularly simple and elegant interpretation. In the E step, we compute the posterior responsibility of cluster c for data point i using

$$ric = p(q_i = c | x_i, \theta) \propto \pi^c N(x_i | \mu_c, W_c W_c^T c + \Psi) \quad (12.17)$$

The conditional posterior for z_i is given by

$$p(z_i | x_i, q_i = c, \theta) = N(z_i | \mu_{ic}, \Sigma_{ic}) \quad (12.18)$$

$$\Sigma_{ic} = (I_L + W_c W_c^T c \Psi^{-1} c W_c)^{-1} \quad (12.19)$$

$$\mu_{ic} = \Sigma_{ic}(W_c \Psi^{-1} c (x_i - \mu_c)) \quad (12.20)$$

In the M step, it is easiest to estimate μ_c and W_c at the same time, by defining $\tilde{W}^c = (W_c, \mu_c)$, $\tilde{z} = (z, 1)$. Also, define

$$bic = E [\tilde{z} | x_i, q_i = c] = [\mu_{ic}; 1] \quad (12.21)$$

$$Cic = E \tilde{z} \tilde{z}^T | x_i, q_i = c = E z z^T | x_i, q_i = c E [z | x_i, q_i = c] E [z | x_i, q_i = c]^T 1 \quad (12.22)$$

Then the M step is as follows:

$$\tilde{W}^c = i ric x_i b_i^T i ric Cic - 1 \quad (12.23)$$

$$\Psi^c = 1 N \text{ diag } ic ric xi - \tilde{W}^c bic x_i^T i . \quad (12.24)$$

$$\pi^c = 1 N \sum_{i=1}^N ric \quad (12.25)$$

Note that these updates are for “vanilla” EM. A much faster version of this algorithm, based on ECM, is described in (Zhao and Yu 2008).

12.1.6 Fitting FA models with missing data In many applications, such as collaborative filtering, we have missing data. One virtue of the EM approach to fitting an FA/PPCA model is that it is easy to extend to this case. However, overfitting can be a problem if there is a lot of missing data. Consequently it is important to perform MAP estimation or to use Bayesian inference. See e.g., (Iljin and Raiko 2010) for details.

已思考 18s

下面先逐字完整翻译原文（含图注与公式、不做删减），再给出通俗解释，最后总结要点。

(注：DGM=有向图模型；Cat=类别分布；式中粗体为向量/矩阵。)

逐字翻译

图 12.3 因子分析器混合 (Mixture of factor analysers) 作为一个 DGM。

图 12.4 将 1 维 PPCA 的混合拟合到一个数据集上，分别为 $K = 1, 10$ 。该图

由 `mixPpcaDemoNetlab` 生成。

12.1.4 因子分析器的混合

FA 模型假设数据位于一个**低维线性流形上**。实际上，大多数数据更适合由某种**低维的弯曲流形**来建模。我们可以用**分段线性流形**来近似一个弯曲流形。这启发了如下模型：令第 k 个维度为 L_k 的线性子空间由 \mathbf{W}_k 表示， $k = 1:K$ 。假设我们有一个潜在指示变量 $q_i \in \{1, \dots, K\}$ ，它指定我们应该用哪个子空间来生成数据。然后我们从一个高斯先验中抽样 \mathbf{z}_i ，把它通过矩阵 \mathbf{W}_k （其中 $k = q_i$ ），并加上噪声。更精确地，模型如下：

$$p(\mathbf{x}_i \mid \mathbf{z}_i, q_i = k, \theta) = N(\mathbf{x}_i \mid \boldsymbol{\mu}_k + \mathbf{W}_k \mathbf{z}_i, \boldsymbol{\Psi}) \quad (12.14)$$

$$p(\mathbf{z}_i \mid \theta) = N(\mathbf{z}_i \mid \mathbf{0}, \mathbf{I}) \quad (12.15)$$

$$p(q_i \mid \theta) = \text{Cat}(q_i \mid \boldsymbol{\pi}) \quad (12.16)$$

这称为**因子分析器的混合** (MFA) (Hinton 等, 1997)。条件独立 (CI) 的假设在图 12.3 中给出。理解该模型的另一种方式是：它是**高斯混合 (MoG) 的低秩版本**。具体地说，与使用满协方差高斯混合所需的 $O(KD^2)$ 个参数相比，该模型只需要 $O(KLD)$ 个参数。这可以减少过拟合。实际上，MFA 是针对高维实值数据的一个很好的通用密度模型。

12.1.5 因子分析模型的 EM

利用第 4 章的结论，推出一个用于拟合 FA 模型的 EM 算法是直接的。只需再做一点工作，我们就可以拟合 FA 的混合。下面我们将不给出证明而只给出结果。推导可见

(Ghahramani and Hinton 1996a)；然而，如果你想熟练掌握这些数学，亲自推导这些方程是很有用的练习。要得到单个因子分析器的结果，只需在下面的公式中令 $r_{ic} = 1$ 且 $c = 1$ 。在 12.2.5 节，我们将看到当拟合 PPCA 模型时这些方程的进一步简化，结果会有特别简单而优雅的解释。

在 E 步，我们计算对于数据点 i 的簇 c 的后验责任：

$$r_{ic} = p(q_i = c \mid \mathbf{x}_i, \theta) \propto \pi_c N(\mathbf{x}_i \mid \boldsymbol{\mu}_c, \mathbf{W}_c \mathbf{W}_c^\top + \boldsymbol{\Psi}) \quad (12.17)$$

对 \mathbf{z}_i 的条件后验为

$$p(\mathbf{z}_i \mid \mathbf{x}_i, q_i = c, \theta) = N(\mathbf{z}_i \mid \mathbf{m}_{ic}, \boldsymbol{\Sigma}_{ic}) \quad (12.18)$$

$$\boldsymbol{\Sigma}_{ic} = (\mathbf{I}_L + \mathbf{W}_c^\top \boldsymbol{\Psi}_c^{-1} \mathbf{W}_c)^{-1} \quad (12.19)$$

$$\mathbf{m}_{ic} = \boldsymbol{\Sigma}_{ic} (\mathbf{W}_c^\top \boldsymbol{\Psi}_c^{-1} (\mathbf{x}_i - \boldsymbol{\mu}_c)) \quad (12.20)$$

在 M 步，把 $\boldsymbol{\mu}_c$ 与 \mathbf{W}_c 同时估计最容易。定义

$$\tilde{\mathbf{W}}_c = (\mathbf{W}_c, \boldsymbol{\mu}_c), \quad \tilde{\mathbf{z}} = (\mathbf{z}, 1),$$

并且定义

$$\mathbf{b}_{ic} = \mathbb{E}[\tilde{\mathbf{z}} \mid \mathbf{x}_i, q_i = c] = \begin{bmatrix} \mathbf{m}_{ic} \\ 1 \end{bmatrix} \quad (12.21)$$

$$\mathbf{C}_{ic} = \mathbb{E}[\tilde{\mathbf{z}}\tilde{\mathbf{z}}^\top \mid \mathbf{x}_i, q_i = c] = \begin{bmatrix} \mathbb{E}[\mathbf{z}\mathbf{z}^\top \mid \mathbf{x}_i, q_i = c] & \mathbb{E}[\mathbf{z} \mid \mathbf{x}_i, q_i = c]^\top \\ \mathbb{E}[\mathbf{z} \mid \mathbf{x}_i, q_i = c] & 1 \end{bmatrix} \quad (12.22)$$

于是 M 步为

$$\hat{\mathbf{W}}_c = \left(\sum_i r_{ic} \mathbf{x}_i \mathbf{b}_{ic}^\top \right) \left(\sum_i r_{ic} \mathbf{C}_{ic} \right)^{-1} \quad (12.23)$$

$$\Psi = \frac{1}{N} \operatorname{diag} \left(\sum_{i,c} r_{ic} (\mathbf{x}_i - \hat{\mathbf{W}}_c \mathbf{b}_{ic}) (\mathbf{x}_i - \hat{\mathbf{W}}_c \mathbf{b}_{ic})^\top \right) \quad (12.24)$$

$$\hat{\pi}_c = \frac{1}{N} \sum_{i=1}^N r_{ic} \quad (12.25)$$

注意：这些更新是“**原味 (vanilla) EM**”的。一个更快的版本（基于 ECM）在 (Zhao and Yu, 2008) 中描述。

12.1.6 带缺失数据的 FA 拟合

在许多应用（如协同过滤）中，我们会有缺失数据。用 EM 来拟合 FA/PPCA 的一个优点是它很容易扩展到这种情况。然而，如果缺失很多，**过拟合**可能成为问题。因此很重要的是进行 **MAP 估计**或使用**贝叶斯推断**。细节可参见例如 (Iljin and Raiko, 2010) 。

通俗解释（尽量白话）

1) 为什么要“因子分析器的混合” (MFA) ?

- FA 的局限：** FA/PPCA 用一条（或几条）**直线/平面**来近似数据。如果数据躺在**弯曲的环、曲面上**，一条直线就不够了（图 12.4(a)：一条斜线去拟合一个圆环，很差）。
- MFA 的想法：** 用很多小的“**线性片段**”拼成一条弯曲的路（图 12.4(b)：很多小椭圆/短线条沿着圆环排一圈）。
 - 每个片段由一个小的 FA/PPCA（参数 \mathbf{W}_k, μ_k ）描述；
 - 一个**隐藏的类别变量** q_i 选择“该样本属于哪一段”；
 - 然后在该段内从 $\mathbf{z}_i \sim N(0, \mathbf{I})$ 生成 $\mathbf{x}_i = \mu_k + \mathbf{W}_k \mathbf{z}_i + \text{噪声}$ 。
- 参数更省：** 相比“每段都是**满协方差高斯**”要 $O(KD^2)$ 参数，MFA 只要 $O(KLD)$ （因为协方差主要由低秩 $\mathbf{W}_k \mathbf{W}_k^\top$ 表达，噪声取对角）。当 $L \ll D$ 时更不易过拟合。

2) EM 如何训练 (先“分配”，再“更新”)

设共有 K 个因子分析器 (簇) :

E 步:

- 先算每个样本属于每个簇的概率 r_{ic} (“责任”)。它就是把该簇看成一个高斯，均值 μ_c ，协方差 $\mathbf{W}_c \mathbf{W}_c^\top + \Psi$ (相当于“该簇在观测空间的总体形状”)，乘以簇先验 π_c ；归一化即得式 (12.17)。
- 同时，在给定簇 c 的条件下，求出该样本的潜在坐标 \mathbf{z}_i 的后验：均值 \mathbf{m}_{ic} 、方差 Σ_{ic} (式 12.18–12.20)。

这些公式和单个 FA 的推断几乎一样，只是现在对每个簇各算一遍。

M 步:

- 用上一步的软分配 r_{ic} 和 \mathbf{z} 的后验矩 (\mathbf{b}_{ic} 、 \mathbf{C}_{ic}) 去更新参数。
- 采用“增广”写法：把 \mathbf{W}_c 与 μ_c 合在一起变成 $\tilde{\mathbf{W}}_c$ ，把 \mathbf{z} 末尾加 1 变成 $\tilde{\mathbf{z}}$ ，写成一个整洁的带截距的线性回归式 (12.23)。
- 噪声协方差 Ψ 取对角，用残差的加权平均来更新 (12.24)。
- 簇权重 π_c 就是平均责任 (12.25)。

小提醒：公式里有时写 Ψ_c ，有时写公共的 Ψ 。两种设定都有人用：

- 共享噪声 Ψ** (更少参数，较稳健)；
- 每簇各自 Ψ_c** (更灵活，但更易过拟合)。上面的更新式 (12.24) 对应共享对角噪声的情形。

加速版： ECM (期望-条件极大化) 能更快收敛，但思路与 EM 相同。

3) 为什么图 12.4 有趣？

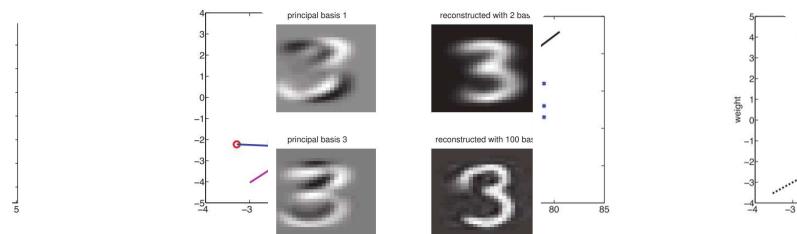
- $K = 1$: 只有一个 1D PPCA 子空间，一条线 + 各向同性噪声，只能粗糙穿过圆。
- $K = 10$: 十个 1D PPCA，像“多节木棒”顺着圆摆一圈，每节都近似局部切线，整体逼近“弯曲流形”。

4) 缺失数据怎么做？

- 在 FA/PPCA/MFA 中，**缺失的观测维**也当作“隐藏变量”。
- E 步**: 只用**观测到的维度**来计算责任与 \mathbf{z} 的后验；对缺失维自动进行**条件期望**的“填充”。
- M 步**: 用这些期望来更新参数。
- 但如果缺失很多，模型可能把噪声当成结构而**过拟合**。解决：
 - 用 MAP (给 \mathbf{W}, Ψ 加先验/正则，比如高斯先验、ARD 等)；
 - 或直接做**贝叶斯推断** (对参数做后验平均)。

要点小结

- **MFA 概念**: 用多个低维 FA/PPCA 片段 $(\mathbf{W}_k, \boldsymbol{\mu}_k)$ + 隐藏类别 q 拼出弯曲流形; 生成式:
 $\mathbf{z} \sim N(0, \mathbf{I}), \mathbf{x} = \boldsymbol{\mu}_k + \mathbf{W}_k \mathbf{z} + \varepsilon, \varepsilon \sim N(0, \boldsymbol{\Psi}), q \sim \text{Cat}(\boldsymbol{\pi})$ 。
- **参数量优势**: 满协方差 MoG 需 $O(KD^2)$; MFA 仅 $O(KLD)$ ($L \ll D$ 时显著更省), 更不易过拟合。
- **EM 训练**:
 - **E 步**: 责任 r_{ic} (把每个簇视作高斯 $N(\boldsymbol{\mu}_c, \mathbf{W}_c \mathbf{W}_c^\top + \boldsymbol{\Psi})$) ; \mathbf{z} 的条件后验 $(\mathbf{m}_{ic}, \boldsymbol{\Sigma}_{ic})$ 。
 - **M 步**: 增广回归更新 $\tilde{\mathbf{W}}_c = (\mathbf{W}_c, \boldsymbol{\mu}_c)$; 对角噪声 $\boldsymbol{\Psi}$ 由加权残差给出; π_c 由平均责任给出。
 - ECM 可加速。
- **图示直觉**: 一个 PPCA 拟合弯曲形状很差; 多个小 PPCA 拼接能很好逼近。
- **缺失数据**: EM 可自然处理 (把缺失视作隐藏变量) ; 但缺失多会过拟合, 宜用 MAP/贝叶斯做正则。
- **实用建议**: 高维连续数据建模时, 若 MoG 过参数、PCA 太线性, 尝试 MFA/MixPPCA; 选择较小 L 、共享对角 $\boldsymbol{\Psi}$ 、必要时加稀疏/ARD 正则, 既能表达非线性结构又较稳健。



翻译并解释以下内容，翻译不能有遗漏，解释请通俗易懂且详尽，因为我的数学基础不太好，最后总结要点。

Figure 12.5 An illustration of PCA and PPCA where $D = 2$ and $L = 1$. Circles are the original data points, crosses are the reconstructions. The red star is the data mean. (a) PCA. The points are orthogonally projected onto the line. Figure generated by pcaDemo2d. (b) PPCA. The projection is no longer orthogonal: the reconstructions are shrunk

towards the data mean (red star). Based on Figure 7.6 of (Nabney 2001). Figure generated by ppcaDemo2d.

Figure 12.6 (a) The mean and the first three PC basis vectors (eigendigits) based on 25 images of the digit 3 (from the MNIST dataset). (b) Reconstruction of an image based on 2, 10, 100 and all the basis vectors. Figure generated by pcalmageDemo.

Figure 12.7 Effect of standardization on PCA applied to the height/ weight dataset. Left: PCA of raw data. Right: PCA of standardized data. Figure generated by pcaDemoHeightWeight.

12.2 Principal components analysis (PCA)

Consider the FA model where we constrain $\Psi = \sigma^2 I$, and W to be orthonormal. It can be shown (Tipping and Bishop 1999) that, as $\sigma^2 \rightarrow 0$, this model reduces to classical (nonprobabilistic) principal components analysis (PCA), also known as the Karhunen Loeve transform. The version where $\sigma^2 > 0$ is known as probabilistic PCA (PPCA) (Tipping and Bishop 1999), or sensible PCA (Roweis 1997). (An equivalent result was derived independently, from a different perspective, in (Moghaddam and Pentland 1995).) To make sense of this result, we first have to learn about classical PCA. We then connect PCA to the SVD. And finally we return to discuss PPCA.

12.2.1 Classical PCA: statement of the theorem

The synthesis view of classical PCA is summarized in the following theorem.

Theorem 12.2.1. Suppose we want to find an orthogonal set of L linear basis vectors $w_j \in R^D$, and the corresponding scores $z_i \in R^L$, such that we minimize the average reconstruction error

$$J(W, Z) = \frac{1}{N} \sum_{i=1}^N \|x_i - \hat{x}_i\|^2 \quad (12.26)$$

where $\hat{x}_i = Wz_i$, subject to the constraint that W is orthonormal. Equivalently, we can write this objective as follows:

$$J(W, Z) = \|X - WZ\|^2_F \quad (12.27)$$

where Z is an $N \times L$ matrix with the z_i in its rows, and $\|A\|_F$ is the Frobenius norm of matrix A , defined by

$$\|A\|_F^2 = \sqrt{\sum_{i=1}^m \sum_{j=1}^n a_{ij}^2} = \sqrt{\text{tr}(A^T A)} = \sqrt{\sum_{i=1}^m \sum_{j=1}^n a_{ij} a_{ij}} \quad (12.28)$$

The optimal solution is obtained by setting $W^* = V_L$, where V_L contains the L eigenvectors with largest

eigenvalues of the empirical covariance matrix, $\hat{\Sigma} = \frac{1}{N} \sum_{i=1}^N \mathbf{x}_i \mathbf{x}_i^T$. (We assume the \mathbf{x}_i have zero mean, for notational simplicity.) Furthermore, the optimal low-dimensional encoding of the data is given by $\hat{\mathbf{z}}_i = \mathbf{W}^T \mathbf{x}_i$, which is an orthogonal projection of the data onto the column space spanned by the eigenvectors.

An example of this is shown in Figure 12.5(a) for $D = 2$ and $L = 1$. The diagonal line is the vector w_1 ; this is called the first principal component or principal direction. The data points $\mathbf{x}_i \in \mathbb{R}^2$ are orthogonally projected onto this line to get $\mathbf{z}_i \in \mathbb{R}$. This is the best 1-dimensional approximation to the data. (We will discuss Figure 12.5(b) later.) In general, it is hard to visualize higher dimensional data, but if the data happens to be a set of images, it is easy to do so. Figure 12.6 shows the first three principal vectors, reshaped as images, as well as the reconstruction of a specific image using a varying number of basis vectors. (We discuss how to choose L in Section 11.5.) Below we will show that the principal directions are the ones along which the data shows maximal variance. This means that PCA can be “misled” by directions in which the variance is high merely because of the measurement scale. Figure 12.7(a) shows an example, where the vertical axis (weight) uses a large range than the horizontal axis (height), resulting in a line that looks somewhat “unnatural”. It is therefore standard practice to standardize the data first, or equivalently, to work with correlation matrices instead of covariance matrices. The benefits of this are apparent from Figure 12.7(b).

下面先逐字完整翻译原文（含图注与公式，不做删减），再给出通俗解释，最后总结要点。

（记号：粗体为向量/矩阵； $\|\cdot\|_F$ 为弗罗贝尼乌斯范数。）

逐字翻译

图 12.5 当 $D = 2$ 、 $L = 1$ 时 PCA 与 PPCA 的示意。圆圈是原始数据点，叉号是重构点。红色五角星是数据均值。 (a) PCA：把点正交投影到直线上。图由 `pcaDemo2d` 生成。 (b) PPCA：投影不再是正交的：重构被朝数据均值（红星）收缩。基于 (Nabney 2001) 图 7.6。图由 `ppcaDemo2d` 生成。

图 12.6 (a) 基于 25 张数字“3”(MNIST 数据集) 的**均值与前三个主基 (主方向)**

(eigendigits)。(b) 用 2、10、100 以及全部基向量对一幅图像的重构。图由 `pcaImageDemo` 生成。

图 12.7 标准化对应用于身高/体重数据集的 PCA 的影响。左：对原始数据做 PCA。右：对**标准化的数据**做 PCA。图由 `pcaDemoHeightWeight` 生成。

12.2 主成分分析 (PCA)

考虑 FA 模型，并对其施加约束 $\Psi = \sigma^2 \mathbf{I}$ ，且 \mathbf{W} 为正交归一 (orthonormal)。可以证明 (Tipping and Bishop 1999)，当 $\sigma^2 \rightarrow 0$ 时，该模型**退化为经典 (非概率) 的主成分分析 (PCA)**，又称 Karhunen–Loëve 变换。当 $\sigma^2 > 0$ 的版本称为**概率 PCA (PPCA)** (Tipping and Bishop 1999)，或 sensible PCA (Roweis 1997)。(等价的结果也在 (Moghaddam and Pentland 1995) 中从不同视角独立得到。) 为了理解这一结果，我们首先学习经典的 PCA；随后把 PCA 与 SVD 联系起来；最后再回来讨论 PPCA。

12.2.1 经典 PCA：定理的表述

从“合成 (synthesis)”视角看，经典 PCA 可以由下述定理概括。

定理 12.2.1. 设我们希望找到一组互相正交的 L 个线性基向量 $\mathbf{w}_j \in \mathbb{R}^D$ ，以及相应的得分 $\mathbf{z}_i \in \mathbb{R}^L$ ，使得**平均重构误差最小**：

$$J(\mathbf{W}, \mathbf{Z}) = \frac{1}{N} \sum_{i=1}^N \|\mathbf{x}_i - \hat{\mathbf{x}}_i\|^2 \quad (12.26)$$

其中 $\hat{\mathbf{x}}_i = \mathbf{W}\mathbf{z}_i$ ，并满足 **W 正交归一**的约束。等价地，我们可以把目标写为

$$J(\mathbf{W}, \mathbf{Z}) = \|\mathbf{X} - \mathbf{WZ}^\top\|_F^2 \quad (12.27)$$

其中 \mathbf{Z} 为 $N \times L$ 矩阵，其每一行是 \mathbf{z}_i ；而 $\|\mathbf{A}\|_F$ 是矩阵 \mathbf{A} 的弗罗贝尼乌斯范数，定义为

$$\|\mathbf{A}\|_F = \sqrt{\sum_{i=1}^m \sum_{j=1}^n a_{ij}^2} = \sqrt{\text{tr}(\mathbf{A}^\top \mathbf{A})} = \|\mathbf{A}(:,)\|_2. \quad (12.28)$$

最优解可通过令 $\hat{\mathbf{W}} = \mathbf{V}_L$ 获得，其中 \mathbf{V}_L 由**经验协方差矩阵**

$$\hat{\Sigma} = \frac{1}{N} \sum_{i=1}^N \mathbf{x}_i \mathbf{x}_i^\top$$

(为书写简便，假设 \mathbf{x}_i 已零均值) 最大的 L 个特征值对应的特征向量组成。此外，数据的最优低维编码为

$$\hat{\mathbf{z}}_i = \mathbf{W}^\top \mathbf{x}_i,$$

即把数据**正交投影**到这些特征向量张成的列空间上。

在 $D = 2, L = 1$ 的例子中, 如图 12.5(a)。对角直线是向量 \mathbf{w}_1 ; 它称为**第一主成分** (主方向)。数据点 $\mathbf{x}_i \in \mathbb{R}^2$ 被正交投影到这条直线上, 得到 $\mathbf{z}_i \in \mathbb{R}$ 。这就是对数据的最佳一维近似。(图 12.5(b) 我们稍后讨论。) 一般地, 高维数据难以可视化, 但如果数据碰巧是一组图像, 则很容易做到。图 12.6 展示了前三个主方向 (重塑为图像) 以及用不同数量的基本向量对某一幅图像的重构。(如何选择 L 见第 11.5 节。) 下面我们将展示: **主方向就是数据方差最大的方向**。这意味着 PCA 可能被“误导”: 若某些方向的方差大只是因为量纲/量程不同。图 12.7(a) 给出了一个例子: 纵轴 (体重) 的取值范围比横轴 (身高) 大很多, 导致拟合出的直线看起来“不自然”。因此, 标准做法是**先把数据标准化**, 或者等价地, 使用**相关系数矩阵**而非协方差矩阵。其好处见图 12.7(b)。

通俗解释

1) PCA 与 PPCA 是什么关系?

- **FA 约束**: 把 FA 的噪声设成各向同性 $\Psi = \sigma^2 \mathbf{I}$, 并强制 \mathbf{W} 的列互相正交且单位长度。
- **当 $\sigma^2 \rightarrow 0$** (没有噪声) 时, 就只剩“把数据投到一个 L 维正交子空间上并最小化重构误差”——这就是**经典 PCA**。
- **当 $\sigma^2 > 0$** 时就是**PPCA**: 仍是线性子空间, 但带各向同性噪声的概率模型。它的重构**不是正交投影**, 而是会**朝均值收缩** (图 12.5(b) 叉号更靠近红星)。

2) 定理在说什么 (用最小二乘的语言)

- 我们要找 L 个互相垂直的“方向” \mathbf{w}_j , 让所有样本被这 L 个方向线性组合后, **尽量还原**原始样本 (最小平方误差)。
- **答案**: 取样本协方差矩阵 $\hat{\Sigma}$ 的**前 L 个特征向量**; 把每个样本 \mathbf{x} 用 $\mathbf{z} = \mathbf{W}^\top \mathbf{x}$ 表示 (这就是把 \mathbf{x} 正交投影到该子空间上)。
- 这样得到的是**最能保留方差**的方向: 第一主成分是方差最大的方向, 第二主成分在与第一正交的前提下方差次大,因此它也是让重构误差最小的方向集。

3) 图 12.5 的直观

- **PCA (a)** : 蓝线段与直线垂直, 因为是**正交投影**。
- **PPCA (b)** : 叉号不在垂线脚上, 而**更靠近均值**。原因: PPCA 的生成模型含噪声, 最优重构是后验均值, 它在子空间上, 但会因噪声而向均值收缩。

4) 图 12.6 的“eigendigits”

- 把每张 28×28 的“3”当作 D -维向量。PCA 找出能最好重构这批“3”的几个“模板”（主基）。
- 用前 $2/10/100$ 个主基重构：基越多，细节越清楚；全部基（等于原维数）时重构几乎与原图一致。

5) 为什么要标准化（图 12.7）

- PCA 会偏向方差大的方向。如果体重的数值范围远大于身高，第一主成分几乎沿“体重轴”，看起来“不合理”。
 - 解决：**标准化**每个特征（减均值、除以标准差），或者等价地用**相关矩阵**做 PCA。右图显示标准化后，主成分更符合“身高一体重共同变化”的直觉。
-

要点小结

- **PCA 是什么：**在所有 L 维正交子空间中，找那个能**最小化重构误差**（或等价：**最大化投影方差**）的子空间； \mathbf{W} 取协方差矩阵的前 L 个特征向量，编码为 $\mathbf{z} = \mathbf{W}^\top \mathbf{x}$ （正交投影）。
- **PPCA 是什么：**FA 的特例 ($\Psi = \sigma^2 \mathbf{I}$ 、 \mathbf{W} 正交)。 $\sigma^2 \rightarrow 0 \rightarrow$ 经典 PCA； $\sigma^2 > 0 \rightarrow$ 概率版 PCA，重构点**向均值收缩**，投影不必正交到原始点。
- **几何直觉：**PCA 选“变化最大”的方向；第一主成分方差最大、第二在正交条件下次大，依此类推。
- **实践细节：**不同量纲会误导 PCA，先**标准化**或用**相关矩阵**；图像等高维数据中，主基（eigenfaces/eigendigits）可作为低维表示与压缩重构。
- **图示理解：**
 - 图 12.5：PCA 正交投影 vs PPCA 的收缩重构。
 - 图 12.6：主基及用少/多基的重构效果。
 - 图 12.7：标准化能让主成分符合直觉。

翻译并解释以下内容，翻译不能有遗漏，解释请通俗易懂且详尽，因为我的数学基础不太好，最后总结要点。

12.2.2 Proof

Proof. We use $w_j \in RD$ to denote the j 'th principal

direction, $x_i \in RD$ to denote the i 'th high-dimensional observation, $z_i \in RL$ to denote the i 'th low-dimensional representation, and $\tilde{z}_j \in RN$ to denote the j 'th component of all the low-dimensional vectors.

Let us start by estimating the best 1d solution, $w_1 \in RD$, and the corresponding projected points $\tilde{z}_1 \in RN$. We will find the remaining bases w_2, w_3 , etc. later. The reconstruction error is given by

$$J(w_1, z_1) = \frac{1}{N} \sum_{i=1}^N \|x_i - z_i w_1\|^2 = \frac{1}{N} \sum_{i=1}^N (x_i - z_i w_1)^T (x_i - z_i w_1) \quad (12.29)$$

$$= \frac{1}{N} \sum_{i=1}^N [x_i^T x_i - 2z_i w_1^T x_i + z_i^2 w_1^T w_1] \quad (12.30)$$

$$= \frac{1}{N} \sum_{i=1}^N [x_i^T x_i - 2z_i w_1^T x_i + z_i^2] \quad (12.31)$$

since $w_1^T w_1 = 1$ (by the orthonormality assumption).

Taking derivatives wrt $z_i w_1$ and equating to zero gives

$$\partial \partial z_i w_1 J(w_1, z_1) = \frac{1}{N} [-2w_1^T x_i + 2z_i] = 0 \Rightarrow z_i w_1 = w_1^T x_i \quad (12.32)$$

So the optimal reconstruction weights are obtained by orthogonally projecting the data onto the first principal direction, w_1 (see Figure 12.5(a)). Plugging back in gives

$$J(w_1) = \frac{1}{N} \sum_{i=1}^N [x_i^T x_i - z_i^2] = \text{const} - \frac{1}{N} \sum_{i=1}^N z_i^2 \quad (12.33)$$

Now the variance of the projected coordinates is given by

$$\text{var} [\tilde{z}_1] = E \tilde{z}_1^2 - (E \tilde{z}_1)^2 = \frac{1}{N} \sum_{i=1}^N z_i^2 - 0 \quad (12.34)$$

since

$$E z_i = E x_i^T w_1 = E x_i^T w_1 = 0 \quad (12.35)$$

because the data has been centered. From this, we see that minimizing the reconstruction error is equivalent to maximizing the variance of the projected data, i.e.,

$$\arg \min w_1 J(w_1) = \arg \max w_1 \text{var} [\tilde{z}_1] \quad (12.36)$$

This is why it is often said that PCA finds the directions of maximal variance. This is called the analysis view of PCA.

The variance of the projected data can be written as

$$\frac{1}{N} \sum_{i=1}^N z_i^2 = \frac{1}{N} \sum_{i=1}^N w_1^T x_i x_i^T w_1 = w_1^T \hat{\Sigma} w_1 \quad (12.37)$$

where $\hat{\Sigma} = \frac{1}{N} \sum_{i=1}^N x_i x_i^T$ is the empirical covariance matrix (or correlation matrix if the data is standardized).

We can trivially maximize the variance of the projection (and hence minimize the reconstruction error) by letting $\|w_1\| \rightarrow \infty$, so we impose the constraint $\|w_1\| = 1$ and instead maximize

$$J^* (w_1) = w_1^T \hat{\Sigma} w_1 + \lambda (w_1^T w_1 - 1) \quad (12.38)$$

where λ is the Lagrange multiplier. Taking derivatives and

equating to zero we have

$$\partial \partial w_1 J^*(w_1) = 2 \Sigma w_1 - 2 \lambda_1 w_1 = 0 \quad (12.39)$$

$$\Sigma w^T w_1 = \lambda_1 w_1 \quad (12.40)$$

Hence the direction that maximizes the variance is an eigenvector of the covariance matrix. Left multiplying by w_1 (and using $w_1^T w_1 = 1$) we find that the variance of the projected data is

$$w_1^T \Sigma w_1 = \lambda_1 \quad (12.41)$$

Since we want to maximize the variance, we pick the eigenvector which corresponds to the largest eigenvalue. Now let us find another direction w_2 to further minimize the reconstruction error, subject to $w_1^T w_2 = 0$ and $w_2^T w_2 = 1$. The error is

$$J(w_1, z_1, w_2, z_2) = \frac{1}{N} \sum_{i=1}^N \|x_i - z_1 w_1 - z_2 w_2\|^2 \quad (12.42)$$

Optimizing wrt w_1 and z_1 gives the same solution as before. Exercise 12.4 asks you to show that $\partial J / \partial z_2 = 0$ yields $z_2 = w_2^T x_i$. In other words, the second principal encoding is gotten by projecting onto the second principal direction. Substituting in yields

$$J(w_2) = \frac{1}{N} \sum_{i=1}^N [x_i^T x_i - w_1^T x_i w_1 - w_2^T x_i w_2] = \text{const} - w_2^T \Sigma w_2 \quad (12.43)$$

Dropping the constant term and adding the constraints yields

$$J^*(w_2) = -w_2^T \Sigma w_2 + \lambda_2 (w_2^T w_2 - 1) + \lambda_1 (w_2^T w_1 - 0) \quad (12.44)$$

Exercise 12.4 asks you to show that the solution is given by the eigenvector with the second largest eigenvalue: $\Sigma w^T w_2 = \lambda_2 w_2$ (12.45)

The proof continues in this way. (Formally one can use induction.)

下面先逐字完整翻译原文（含公式，不删不漏），然后给出通俗解释，最后总结要点。

（记号： w_j 第 j 个主方向； x_i 第 i 个高维样本； z_i 第 i 个低维坐标； $\tilde{z}_j = [z_{1j}, \dots, z_{Nj}]^\top$ 是所有样本在第 j 个主成分上的坐标向量。）

逐字翻译

12.2.2 证明

证明。 用 $w_j \in \mathbb{R}^D$ 表示第 j 个主方向，用 $x_i \in \mathbb{R}^D$ 表示第 i 个高维观测，用 $z_i \in \mathbb{R}^L$ 表示第 i 个低维表示，用 $\tilde{z}_j \in \mathbb{R}^N$ 表示 $[z_{1j}, \dots, z_{Nj}]$ ，即所有低维向量的第 j 个分量。

先来估计最佳的一维解 $\mathbf{w}_1 \in \mathbb{R}^D$ 及其对应的投影点 $\tilde{\mathbf{z}}_1 \in \mathbb{R}^N$ 。其余的基 $\mathbf{w}_2, \mathbf{w}_3, \dots$ 之后再找。重构误差为

$$J(\mathbf{w}_1, \mathbf{z}_1) = \frac{1}{N} \sum_{i=1}^N \|\mathbf{x}_i - z_{i1}\mathbf{w}_1\|^2 = \frac{1}{N} \sum_{i=1}^N (\mathbf{x}_i - z_{i1}\mathbf{w}_1)^\top (\mathbf{x}_i - z_{i1}\mathbf{w}_1) \quad (12.29)$$

$$= \frac{1}{N} \sum_{i=1}^N [\mathbf{x}_i^\top \mathbf{x}_i - 2z_{i1}\mathbf{w}_1^\top \mathbf{x}_i + z_{i1}^2 \mathbf{w}_1^\top \mathbf{w}_1] \quad (12.30)$$

$$= \frac{1}{N} \sum_{i=1}^N [\mathbf{x}_i^\top \mathbf{x}_i - 2z_{i1}\mathbf{w}_1^\top \mathbf{x}_i + z_{i1}^2] \quad (12.31)$$

因为 $\mathbf{w}_1^\top \mathbf{w}_1 = 1$ (由正交归一假设)。对 z_{i1} 求导并令其为零得到

$$\frac{\partial}{\partial z_{i1}} J(\mathbf{w}_1, \mathbf{z}_1) = \frac{1}{N} [-2\mathbf{w}_1^\top \mathbf{x}_i + 2z_{i1}] = 0 \Rightarrow z_{i1} = \mathbf{w}_1^\top \mathbf{x}_i. \quad (12.32)$$

所以最佳的重构权重是把数据正交投影到第一个主方向 \mathbf{w}_1 上 (见图 12.5(a))。代回可得

$$J(\mathbf{w}_1) = \frac{1}{N} \sum_{i=1}^N [\mathbf{x}_i^\top \mathbf{x}_i - z_{i1}^2] = \text{const} - \frac{1}{N} \sum_{i=1}^N z_{i1}^2. \quad (12.33)$$

现在，投影坐标的方差为

$$\text{var}[\tilde{\mathbf{z}}_1] = E[\tilde{\mathbf{z}}_1^2] - (E[\tilde{\mathbf{z}}_1])^2 = \frac{1}{N} \sum_{i=1}^N z_{i1}^2 - 0, \quad (12.34)$$

因为

$$E[z_{i1}] = E[\mathbf{x}_i^\top \mathbf{w}_1] = E[\mathbf{x}_i]^\top \mathbf{w}_1 = 0 \quad (12.35)$$

(数据已中心化)。由此可见，最小化重构误差等价于最大化投影数据的方差，即

$$\arg \min_{\mathbf{w}_1} J(\mathbf{w}_1) = \arg \max_{\mathbf{w}_1} \text{var}[\tilde{\mathbf{z}}_1]. \quad (12.36)$$

这就是常说的“PCA 找到方差最大的方向”。这被称为 PCA 的分析视角。投影数据的方差可写为

$$\frac{1}{N} \sum_{i=1}^N z_{i1}^2 = \frac{1}{N} \sum_{i=1}^N \mathbf{w}_1^\top \mathbf{x}_i \mathbf{x}_i^\top \mathbf{w}_1 = \mathbf{w}_1^\top \hat{\Sigma} \mathbf{w}_1, \quad (12.37)$$

其中 $\hat{\Sigma} = \frac{1}{N} \sum_{i=1}^N \mathbf{x}_i \mathbf{x}_i^\top$ 是经验协方差矩阵 (若数据已标准化则为相关矩阵)。

若令 $\|\mathbf{w}_1\| \rightarrow \infty$ 就能“轻易”最大化投影方差（从而最小化重构误差），因此我们加入约束 $\|\mathbf{w}_1\| = 1$ ，并转而最大化

$$\tilde{J}(\mathbf{w}_1) = \mathbf{w}_1^\top \hat{\Sigma} \mathbf{w}_1 + \lambda_1 (\mathbf{w}_1^\top \mathbf{w}_1 - 1), \quad (12.38)$$

其中 λ_1 是拉格朗日乘子。求导并令其为零得

$$\frac{\partial}{\partial \mathbf{w}_1} \tilde{J}(\mathbf{w}_1) = 2\hat{\Sigma} \mathbf{w}_1 - 2\lambda_1 \mathbf{w}_1 = 0 \quad (12.39)$$

$$\hat{\Sigma} \mathbf{w}_1 = \lambda_1 \mathbf{w}_1. \quad (12.40)$$

因此，使方差最大的方向是协方差矩阵的**特征向量**。左乘 \mathbf{w}_1^\top （并用到 $\mathbf{w}_1^\top \mathbf{w}_1 = 1$ ）可得投影数据的方差为

$$\mathbf{w}_1^\top \hat{\Sigma} \mathbf{w}_1 = \lambda_1. \quad (12.41)$$

因为我们要**最大化**方差，所以取**最大特征值**对应的特征向量。

现在再找另一个方向 \mathbf{w}_2 进一步减小重构误差，满足 $\mathbf{w}_1^\top \mathbf{w}_2 = 0$ 且 $\mathbf{w}_2^\top \mathbf{w}_2 = 1$ 。误差为

$$J(\mathbf{w}_1, \mathbf{z}_1, \mathbf{w}_2, \mathbf{z}_2) = \frac{1}{N} \sum_{i=1}^N \|\mathbf{x}_i - z_{i1} \mathbf{w}_1 - z_{i2} \mathbf{w}_2\|^2. \quad (12.42)$$

对 $\mathbf{w}_1, \mathbf{z}_1$ 优化给出与先前相同的解。练习 12.4 要你证明 $\frac{\partial J}{\partial \mathbf{z}_2} = 0$ 导出 $z_{i2} = \mathbf{w}_2^\top \mathbf{x}_i$ 。换句话说，第二主编码仍然是投到第二主方向上。代入得到

$$J(\mathbf{w}_2) = \frac{1}{N} \sum_{i=1}^N [\mathbf{x}_i^\top \mathbf{x}_i - \mathbf{w}_1^\top \mathbf{x}_i \mathbf{x}_i^\top \mathbf{w}_1 - \mathbf{w}_2^\top \mathbf{x}_i \mathbf{x}_i^\top \mathbf{w}_2] = \text{const} - \mathbf{w}_2^\top \hat{\Sigma} \mathbf{w}_2. \quad (12.43)$$

丢掉常数项并加上约束，得到

$$\tilde{J}(\mathbf{w}_2) = -\mathbf{w}_2^\top \hat{\Sigma} \mathbf{w}_2 + \lambda_2 (\mathbf{w}_2^\top \mathbf{w}_2 - 1) + \lambda_{12} (\mathbf{w}_2^\top \mathbf{w}_1 - 0). \quad (12.44)$$

练习 12.4 要你证明其解由**次大特征值**对应的特征向量给出：

$$\hat{\Sigma} \mathbf{w}_2 = \lambda_2 \mathbf{w}_2. \quad (12.45)$$

证明以同样方式继续（严格说可用数学归纳法）。

通俗解释（一步步讲明白）

1. 目标

我们要找一个方向 \mathbf{w}_1 （一条直线），把点 \mathbf{x}_i 投到这条直线上成为 $z_{i1} \mathbf{w}_1$ ，让**平方重**

构误差最小。

2. 先对投影系数求最优

在固定 \mathbf{w}_1 下, 对每个 z_{i1} 求导 \rightarrow 得到

$$z_{i1} = \mathbf{w}_1^\top \mathbf{x}_i$$

——这正是“垂直投影长度”。所以最佳重构就是把点正交投影到直线。

3. 最小误差 \Leftrightarrow 最大方差

把上面的最优 z_{i1} 代回, 误差就变成“常数 - 投影方差”。因为数据已零均值, 投影的一阶矩为 0, 所以

$$\text{误差越小} \Leftrightarrow \sum z_{i1}^2 \text{ 越大} \Leftrightarrow \text{投影方差越大}.$$

这解释了“PCA 找最大方差方向”。

4. 避免把 \mathbf{w}_1 放大

若不限制 $\|\mathbf{w}_1\|$, 把它变很大就能让方差无限大, 所以必须加约束 $\|\mathbf{w}_1\| = 1$ 。

5. 拉格朗日乘子 \rightarrow 特征向量

带约束最大化 $\mathbf{w}_1^\top \hat{\Sigma} \mathbf{w}_1$ 的一阶条件给出

$$\hat{\Sigma} \mathbf{w}_1 = \lambda_1 \mathbf{w}_1,$$

这表示 \mathbf{w}_1 必须是协方差矩阵的特征向量; 投影方差等于对应的特征值 λ_1 。要最大方差, 就选最大特征值的特征向量。

6. 第二主成分

再找一个与 \mathbf{w}_1 正交的方向 \mathbf{w}_2 , 同理可得它也是 $\hat{\Sigma}$ 的特征向量, 并且为第二大特征值对应的那个。之后的主成分同理 (归纳)。

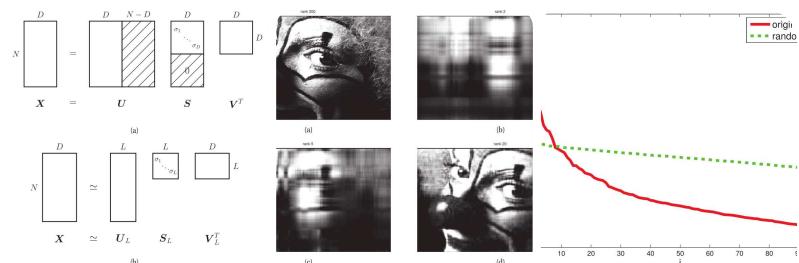
直观记忆:

- **编码:** $\hat{\mathbf{z}}_i = \mathbf{W}^\top \mathbf{x}_i$ (正交投影长度)。
- **方差:** 第 j 个主成分的方差 = 对应特征值 λ_j 。
- **顺序:** 按特征值从大到小取特征向量。

要点小结

- **PCA 的目标:** 在单位范数约束下, 找方向 \mathbf{w} 使得投影后的方差最大, 等价于重构误差最小。
- **最优投影系数:** $z_{ij} = \mathbf{w}_j^\top \mathbf{x}_i$ (把点正交投影到方向 \mathbf{w}_j)。
- **核心结论:** 主方向 \mathbf{w}_j 是样本协方差矩阵 $\hat{\Sigma}$ 的特征向量; 该方向上的方差等于对应特征值。

- 第一主成分：最大特征值的特征向量；
- 第二主成分：与第一正交、且对应第二大特征值；依此类推。
- **前提：**数据需居中（零均值）；不然投影方差和推导需先减去均值。
- **直觉：**PCA 把数据“压到”一个低维正交子空间上，保证保留的总方差（信息量）最多，因此重构最小化。



翻译并解释以下内容，翻译不能有遗漏，解释请通俗易懂且详尽，因为我的数学基础不太好，最后总结要点。

Figure 12.8 (a) SVD decomposition of non-square matrices $X = USVT$. The shaded parts of S , and all the off-diagonal terms, are zero. The shaded entries in U and S are not computed in the economy-sized version, since they are not needed. (b) Truncated SVD approximation of rank L .

Figure 12.9 Low rank approximations to an image. Top left: The original image is of size 200×320 , so has rank 200. Subsequent images have ranks 2, 5, and 20.

Figure 12.10 First 50 log singular values for the clown image (solid red line), and for a data matrix obtained by randomly shuffling the pixels (dotted green line).

12.2.3 Singular value decomposition (SVD)

We have defined the solution to PCA in terms of eigenvectors of the covariance matrix. However, there is another way to obtain the solution, based on the singular value decomposition, or SVD. This basically generalizes the notion of eigenvectors from square matrices to any kind of matrix. In particular, any (real) $N \times D$ matrix X can be decomposed as follows

$$3456X N \times D = U 3456 N \times N 3456S N \times D VT 3456 D \times D \quad (12.46)$$

where U is an $N \times N$ matrix whose columns are

orthonormal (so $U^T U = I_N$), V is $D \times D$ matrix whose rows and columns are orthonormal (so $V^T V = V V^T = I_D$), and S is a $N \times D$ matrix containing the $r = \min(N, D)$ singular values $\sigma_i \geq 0$ on the main diagonal, with 0s filling the rest of the matrix. The columns of U are the left singular vectors, and the columns of V are the right singular vectors. See Figure 12.8(a) for an example.

Since there are at most D singular values (assuming $N > D$), the last $N - D$ columns of U are irrelevant, since they will be multiplied by 0. The economy sized SVD, or thin SVD, avoids computing these unnecessary elements. Let us denote this decomposition by $U^* S^* V^*$. If $N > D$, we have

$$X = U^* S^* V^* \quad (12.47)$$

as in Figure 12.8(a). If $N < D$, we have

$$X = U^* S^* V^* \quad (12.48)$$

Computing the economy-sized SVD takes $O(ND\min(N, D))$ time (Golub and van Loan 1996, p254). The connection between eigenvectors and singular vectors is the following. For an arbitrary real matrix X , if $X = USV^T$, we have

$$X^T X = V^T S^* U^* U S V^T = V^T S^* S V = V^T D^2 V \quad (12.49)$$

where $D = S^* S$ is a diagonal matrix containing the squares of the singular values. Hence

$$(X^T X)V = D^2 V \quad (12.50)$$

so the eigenvectors of $X^T X$ are equal to V , the right singular vectors of X , and the eigenvalues of $X^T X$ are equal to D^2 , the squared singular values. Similarly

$$X^T X = U^* S^* V^T V^* S^* U^* = U^* S^* S U \quad (12.51)$$

$$(X^T X)U = S^* S U = D^2 U \quad (12.52)$$

so the eigenvectors of $X^T X$ are equal to U , the left singular vectors of X . Also, the eigenvalues of $X^T X$ are equal to the squared singular values. We can summarize all this as follows:

$$U = \text{evec}(X^T X), V = \text{evec}(X^T X), S^2 = \text{eval}(X^T X) = \text{eval}(X^T X) \quad (12.53)$$

Since the eigenvectors are unaffected by linear scaling of a matrix, we see that the right singular vectors of X are equal to the eigenvectors of the empirical covariance $\hat{\Sigma}$. Furthermore, the eigenvalues of $\hat{\Sigma}$ are a scaled version of the squared singular values. This means we can perform

PCA using just a few lines of code (see `pcaPmtk`).

However, the connection between PCA and SVD goes deeper. From Equation 12.46, we can represent a rank r matrix as follows:

$$X = \sigma_1 (|u_1|) - v_1^T - + \dots + \sigma_r (|u_r|) - v_r^T - \quad (12.54)$$

If the singular values die off quickly as in Figure 12.10, we can produce a rank L approximation to the matrix as follows:

$$X \approx U(:,1:L) S(1:L,1:L) V(:,1:L) \quad (12.55)$$

This is called a truncated SVD (see Figure 12.8(b)). The total number of parameters needed to represent an $N \times D$ matrix using a rank L approximation is

$$NL + LD + L = L(N + D + 1) \quad (12.56)$$

As an example, consider the 200×320 pixel image in Figure 12.9(top left). This has 64,000 numbers in it. We see that a rank 20 approximation, with only $(200+320+1) \times 20 = 10,420$ numbers is a very good approximation. One can show that the error in this approximation is given by

$$\|X - X_L\|_F \approx \sigma_{L+1} \quad (12.57)$$

Furthermore, one can show that the SVD offers the best rank L approximation to a matrix (best in the sense of minimizing the above Frobenius norm). Let us connect this back to PCA. Let $X = USV^T$ be a truncated SVD of X.

We know that $W^T = V$, and that $Z = X^T W$, so

$$Z^T = USV^T = US \quad (12.58)$$

Furthermore, the optimal reconstruction is given by $\hat{X} = Z^T W^T$, so we find

$$\hat{X} = USV^T \quad (12.59)$$

This is precisely the same as a truncated SVD approximation! This is another illustration of the fact that PCA is the best low rank approximation to the data.

下面先逐字完整翻译原文（含图注与公式，毫无遗漏），随后给出通俗解释，最后总结要点。

（记号：粗体为矩阵/向量； $\|\cdot\|_F$ 为弗罗贝尼乌斯范数；rank=秩。）

逐字翻译

图 12.8 (a) 非方阵的 SVD 分解 $X = USV^T$ 。S 中阴影部分以及所有非对角元素都为 0。在“经济型 (economy-sized) ”版本中，U 与 S 里被阴影的那些项不会被计算，因为不需

要它们。 (b) 秩为 L 的截断 SVD 近似。

图 12.9 对一幅图像的低秩近似。左上：原图像大小为 200×320 ，因此秩为 200。后续三幅图的秩分别为 2、5、20。

图 12.10 小丑图像的前 50 个奇异值的对数（红色实线），以及将像素随机打乱后得到的数据矩阵的对应曲线（绿色虚线）。

12.2.3 奇异值分解 (SVD)

我们已用协方差矩阵的特征向量来定义 PCA 的解。然而还有另一种得到解的方法，基于**奇异值分解** (SVD)。它把“特征向量”的概念从方阵推广到了任意矩阵。具体地，任何（实的） $N \times D$ 矩阵 \mathbf{X} 都可以分解为

$$\underbrace{\mathbf{X}}_{N \times D} = \underbrace{\mathbf{U}}_{N \times N} \underbrace{\mathbf{S}}_{N \times D} \underbrace{\mathbf{V}^T}_{D \times D} \quad (12.46)$$

其中 \mathbf{U} 是 $N \times N$ 的矩阵，其列向量两两正交归一（所以 $\mathbf{U}^T \mathbf{U} = \mathbf{I}_N$ ）； \mathbf{V} 是 $D \times D$ 的矩阵，其行与列都正交归一（所以 $\mathbf{V}^T \mathbf{V} = \mathbf{V} \mathbf{V}^T = \mathbf{I}_D$ ）； \mathbf{S} 是 $N \times D$ 的矩阵，在主对角线上放着 $r = \min(N, D)$ 个奇异值 $\sigma_i \geq 0$ ，其余元素为 0。 \mathbf{U} 的列称为**左奇异向量**， \mathbf{V} 的列称为**右奇异向量**。示意见图 12.8(a)。

由于最多只有 D 个奇异值（假设 $N > D$ ）， \mathbf{U} 的最后 $N - D$ 列是无关的，因为它们会乘以 0。**经济型 SVD** (thin SVD) 避免计算这些无用部分。记该分解为 $\hat{\mathbf{U}} \hat{\mathbf{S}} \hat{\mathbf{V}}^T$ 。若 $N > D$ ，则

$$\underbrace{\mathbf{X}}_{N \times D} = \underbrace{\hat{\mathbf{U}}}_{N \times D} \underbrace{\hat{\mathbf{S}}}_{D \times D} \underbrace{\hat{\mathbf{V}}^T}_{D \times D} \quad (12.47)$$

如图 12.8(a)。若 $N < D$ ，则

$$\underbrace{\mathbf{X}}_{N \times D} = \underbrace{\hat{\mathbf{U}}}_{N \times N} \underbrace{\hat{\mathbf{S}}}_{N \times N} \underbrace{\hat{\mathbf{V}}^T}_{N \times D} \quad (12.48)$$

计算经济型 SVD 的时间为 $O(ND \min(N, D))$ (Golub and van Loan 1996, 第 254 页)。特征向量与奇异向量之间的联系如下。对任意实矩阵 \mathbf{X} ，若 $\mathbf{X} = \mathbf{U} \mathbf{S} \mathbf{V}^T$ ，则

$$\mathbf{X}^T \mathbf{X} = \mathbf{V} \mathbf{S}^T \mathbf{U}^T \mathbf{U} \mathbf{S} \mathbf{V}^T = \mathbf{V} (\mathbf{S}^T \mathbf{S}) \mathbf{V}^T = \mathbf{V} \mathbf{D} \mathbf{V}^T \quad (12.49)$$

其中 $\mathbf{D} = \mathbf{S}^2$ 是对角阵，元素是奇异值的平方。因此

$$(\mathbf{X}^T \mathbf{X}) \mathbf{V} = \mathbf{V} \mathbf{D} \quad (12.50)$$

也就是说， $\mathbf{X}^\top \mathbf{X}$ 的特征向量等于 \mathbf{V} (\mathbf{X} 的右奇异向量)，其特征值等于 \mathbf{D} (奇异值的平方)。类似地，

$$\mathbf{X}\mathbf{X}^\top = \mathbf{U}\mathbf{S}\mathbf{V}^\top \mathbf{V}\mathbf{S}^\top \mathbf{U}^\top = \mathbf{U}(\mathbf{S}\mathbf{S}^\top)\mathbf{U}^\top \quad (12.51)$$

$$(\mathbf{X}\mathbf{X}^\top)\mathbf{U} = \mathbf{U}(\mathbf{S}\mathbf{S}^\top) = \mathbf{U}\mathbf{D} \quad (12.52)$$

所以 $\mathbf{X}\mathbf{X}^\top$ 的特征向量等于 \mathbf{U} (\mathbf{X} 的左奇异向量)，其特征值也等于奇异值的平方。可总结为

$$\mathbf{U} = \text{evec}(\mathbf{X}\mathbf{X}^\top), \quad \mathbf{V} = \text{evec}(\mathbf{X}^\top\mathbf{X}), \quad \mathbf{S}^2 = \text{eval}(\mathbf{X}\mathbf{X}^\top) = \text{eval}(\mathbf{X}^\top\mathbf{X}) \quad (12.53)$$

由于线性缩放不会改变特征向量，我们看到： \mathbf{X} 的右奇异向量与经验协方差 $\hat{\Sigma}$ 的特征向量相同；而 $\hat{\Sigma}$ 的特征值是奇异值平方的一个按比例缩放版。这意味着我们只需几行代码就能用 SVD 实现 PCA (见 `pcaPmtk`)。不过 PCA 与 SVD 的联系更深入。由式 (12.46)，一个秩为 r 的矩阵可以写成

$$\mathbf{X} = \sigma_1 \underbrace{\mathbf{u}_1}_{\text{列}} \underbrace{\mathbf{v}_1^\top}_{\text{行}} + \dots + \sigma_r \mathbf{u}_r \mathbf{v}_r^\top \quad (12.54)$$

如果奇异值像图 12.10 那样快速衰减，我们可以给出秩为 L 的近似：

$$\mathbf{X} \approx \mathbf{U}_{:,1L} \mathbf{S}_{1L,1L} \mathbf{V}_{:,1L}^\top \quad (12.55)$$

这称为截断 SVD (见图 12.8(b))。用秩 L 的近似来表示一个 $N \times D$ 矩阵，所需参数总数为

$$NL + LD + L = L(N + D + 1). \quad (12.56)$$

例如：考虑图 12.9 左上角的 200×320 像素图像。它包含 64,000 个数。可以看到，秩 20 的近似只用 $(200 + 320 + 1) \times 20 = 10,420$ 个数，就已是非常好的近似。可以证明该近似的误差为

$$\|\mathbf{X} - \mathbf{X}_L\|_F \approx \sigma_{L+1}. \quad (12.57)$$

进一步还能证明：SVD 给出了矩阵的最佳秩 L 近似 (“最佳”的含义是最小化上式的弗罗贝尼乌斯范数)。把它和 PCA 联系起来：设 $\mathbf{X} = \mathbf{U}\mathbf{S}\mathbf{V}^\top$ 是 \mathbf{X} 的截断 SVD。我们知道 $\hat{\mathbf{W}} = \mathbf{V}$ ，且 $\hat{\mathbf{Z}} = \mathbf{X}\hat{\mathbf{W}}$ ，因此

$$\hat{\mathbf{Z}} = \mathbf{U}\mathbf{S} \quad (12.58)$$

另外，最优重构为 $\hat{\mathbf{X}} = \hat{\mathbf{Z}}\hat{\mathbf{W}}^\top$ ，于是得到

$$\hat{\mathbf{X}} = \mathbf{U}\mathbf{S}\mathbf{V}^\top. \quad (12.59)$$

这与截断 SVD 近似完全一致！这再次说明：PCA 就是对数据进行最佳低秩近似。

通俗解释

SVD 是什么？

把任意矩阵 \mathbf{X} 拆成“左方向” \mathbf{U} 、“强度” \mathbf{S} （对角上的非负数）和“右方向” \mathbf{V} 。直觉上， $\sigma_i \mathbf{u}_i \mathbf{v}_i^\top$ 是一张“透明底片”，若把前 L 张叠起来，就能近似原图（式 12.54、12.55）。

经济型 (thin) SVD：

当 $N > D$ 时，只保留 \mathbf{U} 的前 D 列与 $D \times D$ 的 \mathbf{S} ——后面的列反正会被 0 消掉，不算也行。反之 $N < D$ 类似处理。

与特征分解的关系：

$\mathbf{X}^\top \mathbf{X}$ 的特征向量 = \mathbf{V} ，特征值 = σ_i^2 ； $\mathbf{X} \mathbf{X}^\top$ 的特征向量 = \mathbf{U} 。因此对居中后的数据矩阵做 SVD， \mathbf{V} 就是 PCA 的主方向， σ_i^2/N 与 PCA 的特征值成比例。

为何截断 SVD 有用？

如果奇异值快速下降（图 12.10 红线），说明数据“本质维度”低，用前 L 个就能很好地还原（图 12.9：秩 20 已很清晰）。而随机打乱像素（绿色虚线）不会有这种快速下降——表示没有低维结构。

参数节省：

原矩阵 $N \times D$ 需要 ND 个数；秩 L 近似只需 $L(N + D + 1)$ 个数，通常大幅压缩。

与 PCA 完全等价：

- 编码： $\hat{\mathbf{Z}} = \mathbf{X} \hat{\mathbf{W}} = \mathbf{U} \mathbf{S}$ 。
 - 重构： $\hat{\mathbf{X}} = \hat{\mathbf{Z}} \hat{\mathbf{W}}^\top = \mathbf{U} \mathbf{S} \mathbf{V}^\top$ 。
- 这正是截断 SVD。因此 PCA = 用 SVD 找最佳低秩子空间。

要点小结

- SVD 分解： $\mathbf{X} = \mathbf{U} \mathbf{S} \mathbf{V}^\top$ ， \mathbf{U}, \mathbf{V} 正交， \mathbf{S} 对角放奇异值 $\sigma_i \geq 0$ 。
- thin/economy SVD：只保留必要列，复杂度 $O(ND \min(N, D))$ 。
- 与特征分解：

$$\mathbf{X}^\top \mathbf{X} = \mathbf{V} \mathbf{S}^2 \mathbf{V}^\top,$$

$$\mathbf{X} \mathbf{X}^\top = \mathbf{U} \mathbf{S}^2 \mathbf{U}^\top.$$

⇒ 右奇异向量 = 协方差矩阵的特征向量； σ_i^2 与特征值成比例。
- 截断 SVD (秩 L 近似)： $\mathbf{X}_L = \mathbf{U}_{:,1:L} \mathbf{S}_{1:L,1:L} \mathbf{V}_{:,1:L}^\top$ 。参数量 = $L(N + D + 1)$ ，误差 $\approx \sigma_{L+1}$ 。

- 与 PCA 的等价:** PCA 的编码/重构正好给出截断 SVD，因此 PCA 是最佳低秩近似。
- 实践指引:** 看奇异值衰减曲线判断可压缩性；图像/文本等大矩阵常用秩几十的近似即可兼顾质量与压缩。

翻译并解释以下内容，翻译不能有遗漏，解释请通俗易懂且详尽，因为我的数学基础不太好，最后总结要点。

12.2.4 Probabilistic PCA

We are now ready to revisit PPCA. One can show the following remarkable result.

Theorem 12.2.2 ((Tipping and Bishop 1999)). Consider a factor analysis model in which $\Psi = \sigma^2 I$ and W is orthogonal. The observed data log likelihood is given by

$$\log p(X|W, \sigma^2) = -N/2 \ln |C| - 1/2 \sum_{i=1}^N x_i^T C^{-1} x_i = -N/2 \ln |C| + \text{tr}(C^{-1} \Sigma) \quad (12.60)$$

where $C = WWT + \sigma^2 I$ and $S = 1/N \sum_{i=1}^N x_i x_i^T = (1/N)XTX$. (We are assuming centered data, for notational simplicity.) The maxima of the log-likelihood are given by

$$\hat{W} = V(\Lambda - \sigma^2 I)^{-1/2} R \quad (12.61)$$

where R is an arbitrary $L \times L$ orthogonal matrix, V is the $D \times L$ matrix whose columns are the first L eigenvectors of S , and Λ is the corresponding diagonal matrix of eigenvalues. Without loss of generality, we can set $R = I$. Furthermore, the MLE of the noise variance is given by

$$\hat{\sigma^2} = 1/D - \sum_{j=L+1}^D \lambda_j \quad (12.62)$$

which is the average variance associated with the discarded dimensions. Thus, as $\sigma^2 \rightarrow 0$, we have $\hat{W} \rightarrow V$, as in classical PCA. What about \hat{Z} ? It is easy to see that the posterior over the latent factors is given by

$$p(\hat{z}_i|x_i, \hat{\theta}) = N(\hat{z}_i|\hat{F}^{-1} \hat{W} x_i, \hat{\sigma}^2 \hat{F}^{-1}) \quad (12.63)$$

$$\hat{F} = \hat{W}^T \hat{W} + \hat{\sigma}^2 I \quad (12.64)$$

(Do not confuse $F = WTW + \sigma^2 I$ with $C = WWT + \sigma^2 I$.)

Hence, as $\sigma^2 \rightarrow 0$, we find $\hat{W} \rightarrow V$, $\hat{F} \rightarrow I$ and $\hat{z}_i \rightarrow Vx_i$.

Thus the posterior mean is obtained by an orthogonal projection of the data onto the column space of V , as in classical PCA. Note, however, that if $\sigma^2 >$, the posterior mean is not an orthogonal projection, since it is shrunk somewhat towards the prior mean, as illustrated in Figure 12.5(b). This sounds like an undesirable property, but it

means that the reconstructions will be closer to the overall data mean, $\hat{\mu} = \mathbf{x}$.

已思考 32s

下面先逐字完整翻译原文（含公式，不删不漏），再给出通俗解释，最后总结要点。

逐字翻译

12.2.4 概率 PCA (Probabilistic PCA, PPCA)

我们现在可以回到 PPCA 了。可以证明如下令人惊讶的结果。

定理 12.2.2 (Tipping 和 Bishop, 1999)。

考虑一个因子分析模型，其中 $\Psi = \sigma^2 I$ 且 W 为正交 (orthogonal)。观测数据的对数似然为

$$\log p(\mathbf{X} | \mathbf{W}, \sigma^2) = -\frac{N}{2} \ln |\mathbf{C}| - \frac{1}{2} \sum_{i=1}^N \mathbf{x}_i^\top \mathbf{C}^{-1} \mathbf{x}_i = -\frac{N}{2} \ln |\mathbf{C}| + \text{tr}(\mathbf{C}^{-1} \hat{\Sigma}) \quad (12.60)$$

其中 $\mathbf{C} = \mathbf{W}\mathbf{W}^\top + \sigma^2 I$ ，且 $S = \frac{1}{N} \sum_{i=1}^N \mathbf{x}_i \mathbf{x}_i^\top = (1/N)\mathbf{X}^\top \mathbf{X}$ 。（为书写简洁，假设数据已中心化。）

使对数似然达到极大的 \mathbf{W} 为

$$\hat{\mathbf{W}} = \mathbf{V}(\Lambda - \sigma^2 I)^{1/2} \mathbf{R} \quad (12.61)$$

其中 \mathbf{R} 是任意的 $L \times L$ 正交矩阵， \mathbf{V} 是 $D \times L$ 矩阵，其列为 S 的前 L 个特征向量， Λ 是相应的特征值对角矩阵。在不失一般性的情况下，可令 $\mathbf{R} = I$ 。进一步，噪声方差的极大似然估计为

$$\hat{\sigma}^2 = \frac{1}{D-L} \sum_{j=L+1}^D \lambda_j \quad (12.62)$$

它等于被丢弃维度所对应的**平均方差**。因此，当 $\sigma^2 \rightarrow 0$ 时，有 $\hat{\mathbf{W}} \rightarrow \mathbf{V}$ ，与经典 PCA 相同。

那 $\hat{\mathcal{Z}}$ 呢？容易看出，给定参数的潜在因子的后验为

$$p(z_i | \mathbf{x}_i, \hat{\theta}) = \mathcal{N}(z_i | \hat{F}^{-1} \hat{\mathbf{W}}^\top \mathbf{x}_i, \hat{\sigma}^2 \hat{F}^{-1}) \quad (12.63)$$

$$\hat{F} = \hat{\mathbf{W}}^\top \hat{\mathbf{W}} + \hat{\sigma}^2 I \quad (12.64)$$

(不要把 $F = W^\top W + \sigma^2 I$ 与 $C = WW^\top + \sigma^2 I$ 混淆。) 因此, 当 $\sigma^2 \rightarrow 0$ 时, 有 $\hat{W} \rightarrow V$ 、 $\hat{F} \rightarrow I$ 且 $\hat{z}_i \rightarrow V^\top x_i$ 。也就是说, 后验均值由把数据**正交投影**到 V 的列空间得到, 和经典 PCA 一样。

但是需要注意, 如果 $\sigma^2 > 0$, 后验均值**不是正交投影**, 因为它会向先验均值稍微**收缩** (见图 12.5(b))。这听起来好像不好, 但它意味着重构将更靠近整体数据的均值 $\hat{\mu} = \bar{x}$ 。

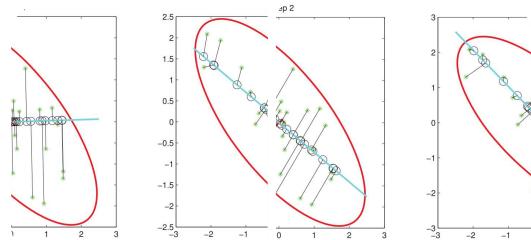
通俗解释

- **模型设定:** PPCA 就是 FA 的特例: 噪声各向同性 $\Psi = \sigma^2 I$, 并把载荷矩阵 W 约束为列正交。整个数据的协方差被分解为“低秩部分 WW^\top ”+“各向同性噪声 $\sigma^2 I$ ”。
 - **极大似然解长什么样:**
 - 把经验协方差 S 做特征分解, 取前 L 个特征向量组成 V , 对应特征值为 Λ 。
 - 最优的 W 不是直接等于 V , 而是**把每个主方向按 $\sqrt{\lambda_j - \sigma^2}$ 缩放** (再允许一个任意的旋转 R)。当噪声趋近于 0 时, 这个缩放因子趋于 $\sqrt{\lambda_j}$, 于是 $\hat{W} \rightarrow V$ ——回到经典 PCA。
 - 最优的噪声方差 $\theta^2 = \text{被丢弃的那些特征值的平均值}$ (第 $L+1$ 到第 D 个)。直觉: 未解释的方差就归到“均匀噪声”。
 - **如何得到低维表示 z :**
 - 后验均值 $\hat{z}_i = \hat{F}^{-1} \hat{W}^\top x_i$, 其中 $\hat{F} = \hat{W}^\top \hat{W} + \theta^2 I$ 。
 - 当噪声很小 ($\sigma^2 \rightarrow 0$) 时, $\hat{F} \rightarrow I$, 所以 $\hat{z}_i \rightarrow V^\top x_i$: 这正是**PCA 的正交投影坐标**。
 - 当噪声大于 0 时, \hat{F}^{-1} 会把系数缩小一些, 相当于**岭回归/权重衰减**: 投影不是垂直落下, 而是被“拉回”到整体均值, 更稳健、抗噪。
 - **关于旋转 R :**
 R 体现了**不可识别的旋转自由度** (第 12.1.3 节提过); 为了固定唯一解可以取 $R = I$ 。
 - **为什么“收缩”不一定坏:**
数据含噪时, PCA 的硬投影可能过拟合个别点; PPCA 的后验均值向均值回缩, 重构会更贴近整体分布 (更稳健)。
-

要点小结

- **PPCA = FA ($\Psi = \sigma^2 I$ 、 W 正交)**, 其极大似然解:
 - $\hat{W} = V(\Lambda - \sigma^2 I)^{1/2} R$ (V 为协方差前 L 个特征向量)。

- $\hat{\sigma}^2 = \frac{1}{D-L} \sum_{j=L+1}^D \lambda_j$ (丢弃维度的平均方差)。
- **与 PCA 关系:** 当 $\sigma^2 \rightarrow 0$ 时, $\hat{W} \rightarrow V$, $\hat{z}_i \rightarrow V^\top x_i$: 回到经典 PCA 的正交投影。
- **后验与重构:**
 $p(z_i | x_i) = N(\hat{F}^{-1} \hat{W}^\top x_i, \sigma^2 \hat{F}^{-1})$;
若 $\sigma^2 > 0$, \hat{z}_i 会向均值收缩 \rightarrow 重构更靠近整体均值, 表现更稳健。
- **旋转不唯一:** 出现任意正交 R (旋转模糊); 通常取 $R = I$ 以固定解。



翻译并解释以下内容，翻译不能有遗漏，解释请通俗易懂且详尽，因为我的数学基础不太好，最后总结要点。

Figure 12.11 Illustration of EM for PCA when $D=2$ and $L=1$. Green stars are the original data points, black circles are their reconstructions. The weight vector w is represented by blue line. (a) We start with a random initial guess of w . The E step is represented by the orthogonal projections. (b) We update the rod w in the M step, keeping the projections onto the rod (black circles) fixed. (c) Another E step. The black circles can 'slide' along the rod, but the rod stays fixed. (d) Another M step. Based on Figure 12.12 of (Bishop 2006b). Figure generated by pcaEmStepByStep.

12.2.5 EM algorithm for PCA

Although the usual way to fit a PCA model uses eigenvector methods, or the SVD, we can also use EM, which will turn out to have some advantages that we discuss below. EM for PCA relies on the probabilistic formulation of PCA. However the algorithm continues to work in the zero noise limit, $\sigma^2 = 0$, as shown by (Roweis 1997). Let \tilde{Z} be a $L \times N$ matrix storing the posterior means (low-dimensional representations) along its columns. Similarly, let $X^* = XT$ store the original data along its columns. From Equation 12.63, when $\sigma^2 = 0$, we

have

$$\tilde{Z} = (\tilde{W}\tilde{W})^{-1}\tilde{W}\tilde{X} \quad (12.65)$$

This constitutes the E step. Notice that this is just an orthogonal projection of the data. From Equation 12.23, the M step is given by

$$\hat{W} = i \sum_i E[\tilde{z}_i]T i E[\tilde{z}_i]T - 1 \quad (12.66)$$

where we exploited the fact that $\Sigma = \text{cov}[\tilde{z}_i | x_i, \theta] = \sigma^2 I$ when $\sigma^2 = 0$. It is worth comparing this expression to the MLE for multi-output linear regression (Equation 7.89), which has the form $\hat{W} = (\sum_i y_i x_i T i)(\sum_i x_i x_i T i)^{-1}$. Thus we see that the M step is like linear regression where we replace the observed inputs by the expected values of the latent variables. In summary, here is the entire algorithm:

- Estep $\tilde{Z} = (\tilde{W}\tilde{W})^{-1}\tilde{W}\tilde{X}$
- Mstep $\hat{W} = \tilde{X}\tilde{Z}T(\tilde{Z}\tilde{Z}T)^{-1}$

(Tipping and Bishop 1999) showed that the only stable fixed point of the EM algorithm is the globally optimal solution. That is, the EM algorithm converges to a solution where \hat{W} spans the same linear subspace as that defined by the first L eigenvectors. However, if we want \hat{W} to be orthogonal, and to contain the eigenvectors in descending order of eigenvalue, we have to orthogonalize the resulting matrix (which can be done quite cheaply).

Alternatively, we can modify EM to give the principal basis directly (Ahn and Oh 2003). This algorithm has a simple physical analogy in the case $D=2$ and $L=1$ (Roweis 1997). Consider some points in R^2 attached by springs to a rigid rod, whose orientation is defined by a vector w . Let z_i be the location where the i 'th spring attaches to the rod. In the E step, we hold the rod fixed, and let the attachment points slide around so as to minimize the spring energy (which is proportional to the sum of squared residuals). In the M step, we hold the attachment points fixed and let the rod rotate so as to minimize the spring energy. See Figure 12.11 for an illustration. Apart from this pleasing intuitive interpretation, EM for PCA has the following advantages over eigenvector methods:

- EM can be faster. In particular, assuming $N,D \ll L$, the dominant cost of EM is the projection operation in the E step, so the overall time is $O(TLN^2)$, where T is the number of iterations. (Roweis 1997) showed experimentally that the number of iterations is usually very small (the mean was 3.6), regardless of N or D . (This results depends on

the ratio of eigenvalues of the empirical covariance matrix.) This is much faster than the $O(\min(ND^2, DN^2))$ time required by straightforward eigenvector methods, although more sophisticated eigenvector methods, such as the Lanczos algorithm, have running times comparable to EM.

- EM can be implemented in an online fashion, i.e., we can update our estimate of W as the data streams in.
- EM can handle missing data in a simple way (see Section 12.1.6).
- EM can be extended to handle mixtures of PPCA/ FA models.
- EM can be modified to variational EM or to variational Bayes EM to fit more complex models.

已思考 25s

下面先逐字完整翻译原文（含图注与公式，不删不漏），再给出通俗解释，最后总结要点。

逐字翻译

图 12.11 当 $D = 2$ 、 $L = 1$ 时，用 EM 做 PCA 的示意。绿色星号是原始数据点，黑色空心圆是它们的重构。权重向量 w 用蓝色直线表示。(a) 从 w 的一个随机初值开始。E 步由**正交投影**表示。(b) 在 M 步中我们更新“杆” w ，同时保持投到杆上的投影（黑色圆点）不变。(c) 另一次 E 步。黑色圆点可以沿着杆“滑动”，但杆保持不动。(d) 另一次 M 步。基于 (Bishop 2006b) 图 12.12。图由 `pcaEmStepByStep` 生成。

12.2.5 PCA 的 EM 算法

虽然拟合 PCA 的常规方法是用**特征向量**或 SVD，我们也可以使用 EM，其会带来一些优点（见下文）。PCA 的 EM 依赖于 PCA 的概率表述。然而，该算法在**零噪声极限** $\sigma^2 = 0$ 下仍能工作 (Roweis 1997 所示)。令 \tilde{Z} 为一个 $L \times N$ 矩阵，把每个样本的**后验均值**（低维表示）按列存放。类似地，令 $\tilde{X} = X^\top$ ，把原始数据也按列存放。由式 (12.63)，当 $\sigma^2 = 0$ 时，有

$$\tilde{Z} = (W^\top W)^{-1} W^\top \tilde{X} \quad (12.65)$$

这就构成了**E 步**。注意这只是把数据做**正交投影**。

由式 (12.23)，**M 步**为

$$\hat{W} = \left(\sum_i x_i E[z_i]^\top \right) \left(\sum_i E[z_i] E[z_i]^\top \right)^{-1} \quad (12.66)$$

这里利用了当 $\sigma^2 = 0$ 时 $\Sigma = \text{cov}[z_i | x_i, \theta] = 0 \cdot I$ 的事实。值得把该式与**多输出线性回归**的极大似然解（式 7.89）比较： $W = (\sum_i y_i x_i^\top) (\sum_i x_i x_i^\top)^{-1}$ 。因此可见，M 步就像线性回归，只不过把**观测到的输入换成了潜变量的期望值**。

小结一下，整个算法是：

- **E 步** $\tilde{Z} = (W^\top W)^{-1} W^\top \tilde{X}$
- **M 步** $W = \tilde{X} \tilde{Z}^\top (\tilde{Z} \tilde{Z}^\top)^{-1}$

Tipping 和 Bishop (1999) 表明：该 EM 算法的**唯一稳定的不动点就是全局最优解**。也就是说，EM 收敛到的 W 与由前 L 个**特征向量**定义的线性子空间张成相同的空间。不过，如果我们希望 W 本身是**正交的**，并且列按特征值从大到小排列，就需要对得到的矩阵做一次**正交化**（成本很低）。或者可以修改 EM 以直接得到主基 (Ahn and Oh 2003)。

当 $D = 2, L = 1$ 时，该算法有一个简单的物理类比 (Roweis 1997)：设 \mathbb{R}^2 里的一些点各自用**弹簧拴在一根刚性杆上**，杆的方向由向量 w 定义， z_i 是第 i 根弹簧在杆上的连接位置。**E 步**：保持杆不动，让连接点沿杆滑动，以**最小化弹簧能量**（与残差平方和成正比）。**M 步**：保持连接点不动，让杆旋转以最小化弹簧能量。见图 12.11。

除了这个直观解释外，PCA 的 EM 相比特征向量法还有以下优点：

- **可能更快。** 假设 $N, D \gg L$ ，EM 的主要成本来自 E 步的投影运算，因此总时间为 $O(TLND)$ ，其中 T 是迭代次数。Roweis (1997) 实验表明 T 通常很小（平均 3.6），与 N, D 无关（该结果与经验协方差的特征值之比有关）。这比朴素特征向量法需要的 $O(\min(ND^2, DN^2))$ 要快得多；但更高级的特征向量算法（如 Lanczos）运行时间可与 EM 相当。
- **可在线实现。** 数据流入时可持续更新 W 。
- **能简单处理缺失数据**（见 12.1.6）。
- **可扩展到 PPCA/FA 的混合模型。**
- **可改成变分 EM 或变分贝叶斯 EM 来拟合更复杂的模型。**

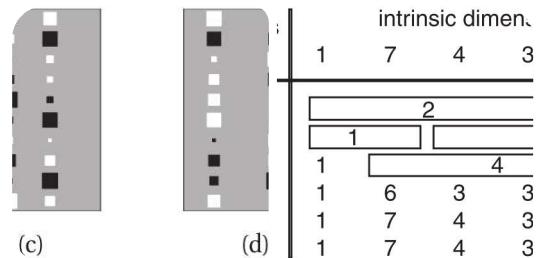
通俗解释（一步到位）

- **把 PCA 当成“有概率的线性生成模型”：**在噪声 $\sigma^2 \rightarrow 0$ 时，PPCA 退化成经典 PCA。EM 就在这个模型上交替优化：

- **E 步：**在当前“杆” W 下，把每个点正交投影到杆上，得到该点的低维坐标 \tilde{Z} （式 12.65）。
 - **M 步：**把这些低维坐标当作“输入”，做一次多输出线性回归来更新 W （式 12.66）。直觉上就是“固定投影点，转动杆，使总距离最小”。
 - **它为什么会收敛到正确的子空间？**
因为唯一稳定的不动点就是由前 L 个主成分张成的空间；EM 的每一步都在降低重构误差（等价地增加投影方差），因此会到达该空间。
 - **为什么图 12.11 里点会‘滑动’？**
E 步固定杆，沿杆滑动的最佳位置是垂足；M 步固定垂足，再旋转杆，使所有垂线的总长度最短。几次交替后，杆就对准主方向。
 - **实际使用的好处**
当维度 D 与样本数 N 很大而所需主成分 L 很小（常见的“高维低秩”情形），EM 每次只做矩阵乘法与小矩阵求逆，通常迭代数很少，因此很省时、省内存；还能在线更新，还能自然处理缺失值，并且易于扩展到混合/贝叶斯版本。
 - **小注意**
EM 得到的 W 不一定本身就正交、有序；想要与“标准 PCA 基”一致，可在收敛后做一次正交化与按方差排序（很快）。
-

要点小结

- **PCA 的 EM 形式 ($\sigma^2 = 0$) :**
 - **E 步：** $\tilde{Z} = (W^\top W)^{-1} W^\top \tilde{X}$ （把数据正交投影到当前子空间）。
 - **M 步：** $W = \tilde{X} \tilde{Z}^\top (\tilde{Z} \tilde{Z}^\top)^{-1}$ （像线性回归，用 \tilde{Z} 作为“输入”）。
- **收敛性：** 唯一稳定不动点是全局最优子空间（与前 L 个特征向量张成相同）；若要正交且有序，收敛后再正交化/排序或用改进的 EM。
- **直观类比：** 点—弹簧—刚性杆；E 步滑动垂足，M 步转动杆；反复直到能量最小。
- **复杂度与优势：** 总时间 $O(TLND)$ ，迭代次数少；可在线、能处理缺失、可扩展到混合/变分/贝叶斯版本；大规模低秩场景常比直接特征分解更高效。



翻译并解释以下内容，翻译不能有遗漏，解释请通俗易懂且详尽，因为我的数学基础不太好，最后总结要点。

Figure 12.12 Illustration of estimating the effective dimensionalities in a mixture of factor analysers using VBEM. The blank columns have been forced to 0 via the ARD mechanism. The data was generated from 6 clusters with intrinsic dimensionalities of 7, 4, 3, 2, 2, 1, which the method has successfully estimated. Source: Figure 4.4 of (Beal 2003). Used with kind permission of Matt Beal.

Figure 12.13 We show the estimated number of clusters, and their estimated dimensionalities, as a function of sample size. The VBEM algorithm found two different solutions when $N = 8$. Note that more clusters, with larger effective dimensionalities, are discovered as the sample sizes increases. Source: Table 4.1 of (Beal 2003). Used with kind permission of Matt Beal.

12.3 Choosing the number of latent dimensions

In Section 11.5, we discussed how to choose the number of components K in a mixture model. In this section, we discuss how to choose the number of latent dimensions L in a FA/PCA model.

12.3.1 Model selection for FA/PPCA

If we use a probabilistic model, we can in principle compute $L^* = \operatorname{argmax}_L p(L|D)$. However, there are two problems with this. First, evaluating the marginal likelihood for LVMs is quite difficult. In practice, simple approximations, such as BIC or variational lower bounds (see Section 21.5), can be used (see also (Minka 2000a)). Alternatively, we can use the cross-validated likelihood as a performance measure, although this can be slow, since it requires fitting each model F times, where F is the number of CV folds.

The second issue is the need to search over a potentially

large number of models. The usual approach is to perform exhaustive search over all candidate values of L. However, sometimes we can set the model to its maximal size, and then use a technique called automatic relevancy determination (Section 13.7), combined with EM, to automatically prune out irrelevant weights. This technique will be described in a supervised context in Chapter 13, but can be adapted to the (M)FA context as shown in (Bishop 1999; Ghahramani and Beal 2000).

Figure 12.12 illustrates this approach applied to a mixture of FAs fit to a small synthetic dataset. The figures visualize the weight matrices for each cluster, using Hinton diagrams, where the size of the square is proportional to the value of the entry in the matrix.² We see that many of them are sparse. Figure 12.13 shows that the degree of sparsity depends on the amount of training data, in accord with the Bayesian Occam's razor. In particular, when the sample size is small, the method automatically prefers simpler models, but as the sample size gets sufficiently large, the method converges on the "correct" solution, which is one with 6 subspaces of dimensionality 1, 2, 2, 3, 4 and 7.

Although the ARD/ EM method is elegant, it still needs to perform search over K. This is done using "birth" and "death" moves (Ghahramani and Beal 2000). An alternative approach is to perform stochastic sampling in the space of models. Traditional approaches, such as (Lopes and West 2004), are based on reversible jump MCMC, and also use birth and death moves. However, this can be slow and difficult to implement. More recent approaches use non-parametric priors, combined with Gibbs sampling, see e.g., (Paisley and Carin 2009).

下面先逐字完整翻译原文（含图注，不漏译），再给出通俗解释，最后总结要点。

逐字翻译

图 12.12 用 VBEM (变分贝叶斯 EM) 在“因子分析器混合” (MFA) 中估计**有效维度数**的示意。空白的列通过 ARD 机制被强制为 0。数据由 6 个簇生成，它们的**内在维度**分别为 7、4、3、2、2、1，而该方法已成功地估计出了这些值。来源：(Beal 2003) 图 4.4。经 Matt Beal 许可使用。

图 12.13 展示了估计到的簇的数量以及它们的估计维度数, 随样本规模的变化情况。当 $N = 8$ 时, VBEM 算法找到了两个不同的解。注意: 随着样本数增加, 会发现更多的簇, 而且它们的**有效维度**也更大。来源: (Beal 2003) 表 4.1。经 Matt Beal 许可使用。

12.3 选择潜在维度的个数

在第 11.5 节, 我们讨论了如何在混合模型中选择成分数 K 。本节讨论如何在 FA/PCA 模型中选择潜在维度数 L 。

12.3.1 FA/PPCA 的模型选择

如果使用概率模型, 原则上可以计算

$$L^* = \arg \max_L p(L \mid D).$$

但这有两个问题。**第一**, 对潜变量模型来说, **边缘似然**很难精确计算。在实践中, 可以使用一些简单近似, 例如 BIC 或**变分下界** (见第 21.5 节; 也可见 Minka 2000a)。另一种办法是把**交叉验证的对数似然**作为性能指标, 不过这会较慢, 因为需要对每个模型做 F 次拟合 (F 是折数)。

第二个问题是: 需要在可能很多的模型上进行搜索。通常的做法是对 L 的所有候选值做**穷举**。但有时可以把模型设成**最大尺寸**, 然后用一种叫**自动相关性确定 (ARD)** (第 13.7 节) 的方法, 配合 EM, 自动剪掉不相关的权重。该技术会在第 13 章以监督学习的背景讲解, 但也可改造到 (M) FA 的情境中 (Bishop 1999; Ghahramani 和 Beal 2000)。

图 12.12 展示了把这种方法应用到一个合成数据集上、拟合 MFA 的结果。图中用 **Hinton 图**来可视化每个簇的权重矩阵: 矩阵中条目的大小用方块面积表示。我们看到, 很多矩阵是**稀疏的**。图 12.13 说明稀疏度依赖于训练数据的多少, 这与**贝叶斯的奥卡姆剃刀**一致。具体来说, 当样本数很少时, 该方法会自动偏好**更简单的模型**; 而当样本数足够大时, 方法会收敛到“正确”解——即 6 个子空间, 其维度分别为 1、2、2、3、4 和 7。

尽管 ARD/EM 方法很优雅, 它仍然需要对 K 做搜索。这可以通过“**出生/死亡**”移动完成 (Ghahramani 和 Beal 2000)。另一类做法是在模型空间中进行**随机采样**。传统方法 (如 Lopes 和 West 2004) 基于**可逆跳跃 MCMC**, 同样使用出生/死亡移动, 但往往较慢且实现困难。近年来的做法使用**非参数先验**并结合 Gibbs 采样, 例如 (Paisley 和 Carin 2009)。

通俗解释 (尽量白话)

1. 问题是什么？

在 FA/PPCA/MFA 里，我们得决定**潜在维度 L** （每个簇里低维子空间的维数）以及有无多簇时的**簇数 K** 。 L 选小了表示力不够，选大了容易过拟合。

2. 理论上最严格的方法

直接比较不同 L 的模型谁的**后验概率 $p(L \mid \text{数据})$ **最大。但这需要计算“**边缘似然**”，对带潜变量的模型来说通常很难。

3. 工程上常用的替代

- **BIC/变分下界**：给出便宜的近似打分来比较不同 L 。
- **交叉验证**：看不同 L 在验证集上的对数似然/误差，谁好选谁，但要多次训练，比较慢。

4. 不用穷举所有 L 的思路

先把模型做“大一点”（例如设 L 很大、或 MFA 中每个簇给较大的 L_k ），然后用 ARD（一种带超参数的先验/正则化）让不重要的列**自动缩到 0**。这会把“有效维度”学出来。

- 图 12.12 里：每列是一维因子（载荷向量）。被 ARD 压到 0 的列显示成**空白列**，剩下的列数就是**估计的维度**。数据真相是 6 个簇的内在维度 = 7, 4, 3, 2, 2, 1，算法学到了几乎一样的答案。
- 图 12.13 里：样本多了以后，算法发现的簇数更多，**每个簇的维度也更大**——因为数据足够支撑更复杂的结构；样本少时，算法更“保守”（奥卡姆剃刀：简单优先）。

5. 还要不要搜 K ？

ARD 能自动决定每个簇里“保留多少维”，但簇的个数 K 仍要决定。

- 一种办法是**出生/死亡操作**：尝试增加或删除一个簇，配合 EM 看对数似然/下界是否提升。
- 另一种是直接在模型空间做**采样**：可逆跳跃 MCMC（理论漂亮，但慢、实现难）；或用**非参数先验**（比如 Dirichlet 进程、Beta 过程等）+ **Gibbs 采样**，让模型“自己决定”需要多少簇/多少因子。

6. Hinton 图是啥？

是把矩阵画成小方块网格，**方块面积 \approx 数值绝对值，颜色/方向 \approx 正负号**。一眼就能看稀疏与强弱。

要点小结

- **目标**：为 FA/PPCA/MFA 选择潜在维度 L （和可能的簇数 K ）。

- **严格方案**: 最大化 $p(L \mid \text{数据})$, 但边缘似然难算; 常用 BIC/变分下界 或 交叉验证 近似。
- **ARD+EM 自适应法**: 把模型设大, 用 ARD 自动把不重要的因子收缩为 0, 得到每个簇的**有效维度**; 与奥卡姆剃刀一致——数据少 \Rightarrow 简单模型, 数据多 \Rightarrow 更复杂模型。
- **图示理解**:
 - 图 12.12: 空白列=被剪掉的因子; 成功恢复真维度 7, 4, 3, 2, 2, 1。
 - 图 12.13: 样本越多, 发现的簇越多、维度越大。
- **搜索簇数 K** : 可用**出生/死亡步骤或采样方法** (可逆跳跃 MCMC; 或**非参数先验 +Gibbs**)。
- **实践建议**: 若可行, 先用 VBEM+ARD 省心估维; 再用 BIC/下界/交叉验证核实; 样本不大时保持模型简单, 避免过拟合。

↳ 我们得决定**潜在维度 LLL** (每个簇里低维子空间的维数)

这里的“低维子空间”以及“维数”指的是什么?

已思考 18s

好问题! 这里的“**低维子空间**”和“**维数**”是指在 FA / PPCA / PCA / MFA 这些模型里, 用来近似 (或生成) 高维数据的那块“更小的线性空间”。

通俗解释

- 你的观测向量 x 在一个很高的维度里 (比如 $D = 784$ 的手写数字图像)。但这些数据**并不在整个 \mathbb{R}^D 里到处乱跑**, 而是主要集中在某个“薄薄的片子”上。
- 这块“片子”就是**低维子空间** (更准确地说是**仿射子空间**: 一块经过平移的线性子空间)。

数学写法:

$$S = \{ \mu + Wz : z \in \mathbb{R}^L \}$$

这里 $W \in \mathbb{R}^{D \times L}$ 的每一列是一个方向 (基向量), μ 是均值 (把子空间从原点平移到数据所在的位置), z 是**低维坐标** (潜变量)。

- **维数**就是这块空间里**相互独立方向的个数**, 也就是上式里 z 的长度 L (若 W 满秩则“有效维数”=rank(W) = L)。

直观:

- $L = 1$: 一条线;

- $L = 2$: 一张平面;
 - $L = 3$: 三维的“体积”;
- 这些都嵌在高维的观测空间里 (维度 D 远大于它们)。
- 在这些模型里每个样本都近似满足

$$x \approx \mu + Wz + \varepsilon,$$

其中 ε 是小噪声 (PPCA/FA 取高斯噪声)。也就是说, **主要变化沿着 W 的列空间进行** (低维子空间), 而噪声是“厚度”。

放到具体方法里怎么理解

- **PCA/PPCA (单个子空间)**
选出 L 个主成分向量 (\hat{W} 的列), 它们张成的就是那块**低维子空间**; 维数就是你保留的主成分个数 L 。把点投影到这里得到低维坐标 z 。
- **FA (因子分析)**
同样是 $x = \mu + Wz + \varepsilon$ 。这里“低维子空间”是 $\text{col}(W)$; 维数 L 等于潜在因子个数。
- **MFA (因子分析混合)**
有多个簇, 每个簇 k 有自己的 W_k, μ_k 。于是数据被多块**局部低维子空间**拼起来近似 (弯曲流形的分段线性近似)。每个簇的维数是 L_k 。

小例子

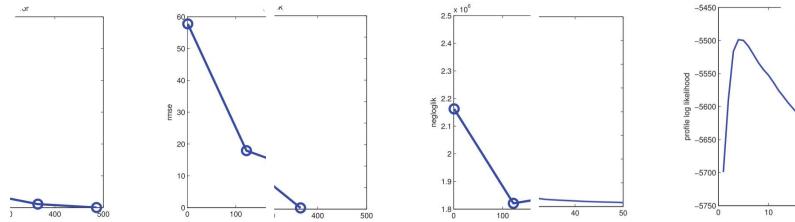
- 身高-体重 ($D = 2$): 若几乎线性相关, 最佳子空间是一条**直线** ($L = 1$)。
- 3D 点云大多落在某个平面上: 用**平面** ($L = 2$) 就足以描述主要变化。
- 手写“3” ($D = 784$): 前 $L = 50$ 个“eigendigits”就能重构大部分细节 \Rightarrow 有效维度大约 50。

常见说法对照

- 维数 $L \approx$ 潜在因子数 \approx 主成分个数 \approx 本征/内在维度 (intrinsic dimensionality) \approx 自由度。
- 低维子空间 $= \mu + \text{span}(\text{columns of } W)$ 。

要点小结

- 低维子空间: 数据主要所在的那块 (仿射) 线性空间, $\{\mu + Wz\}$ 。
- 维数: 这块空间的独立方向数量, 等于潜在变量维度 L (或 $\text{rank}(W)$)。
- 在 PCA/PPCA/FA 中: W 的列空间就是子空间; 在 MFA 中: 每个簇各有一个子空间; L (或 L_k) 是要选择/学习的超参数, 表示数据的内在复杂度。
- 物理直觉: 高维里的“薄片”, 沿片子方向变化 (由 z 控制), 片子的厚度是噪声。



翻译并解释以下内容，翻译不能有遗漏，解释请通俗易懂且详尽，因为我的数学基础不太好，最后总结要点。

Figure 12.14 Reconstruction error on MNIST vs number of latent dimensions used by PCA. (a) Training set. (b) Test set. Figure generated by pcaOverfitDemo.

Figure 12.15 Negative log likelihood on MNIST vs number of latent dimensions used by PPCA. (a) Training set. (b) Test set. Figure generated by pcaOverfitDemo.

Figure 12.16 (a) Scree plot for training set, corresponding to Figure 12.14(a). (b) Profile likelihood. Figure generated by pcaOverfitDemo.

12.3.2 Model selection for PCA

Since PCA is not a probabilistic model, we cannot use any of the methods described above. An obvious proxy for the likelihood is the reconstruction error:

$$E(D, L) = \frac{1}{|D|} \sum_{i \in D} \|x_i - \hat{x}_i\|^2 \quad (12.67)$$

In the case of PCA, the reconstruction is given by $\hat{x}_i = Wz_i + \mu$, where $z_i = WT(x_i - \mu)$ and W and μ are estimated from D_{train} .

Figure 12.14(a) plots $E(D_{train}, L)$ vs L on the MNIST training data in Figure 12.6. We see that it drops off quite quickly, indicating that we can capture most of the empirical correlation of the pixels with a small number of factors, as illustrated qualitatively in Figure 12.6. Exercise 12.5 asks you to prove that the residual error from only using L terms is given by the sum of the discarded eigenvalues:

$$E(D_{train}, L) = \sum_{j=L+1}^D \lambda_j \quad (12.68)$$

Therefore an alternative to plotting the error is to plot the retained eigenvalues, in decreasing order. This is called a scree plot, because “the plot looks like the side of a mountain, and ‘scree’ refers to the debris fallen from a mountain and lying at its base”.³ This will have the same

shape as the residual error plot. A related quantity is the fraction of variance explained, defined as

$$F(D_{\text{train}}, L) = \sum_{j=1}^L \lambda_j / \max_j \lambda_j = 1 - \lambda_j (12.69)$$

This captures the same information as the scree plot.

Of course, if we use $L = \text{rank}(X)$, we get zero reconstruction error on the training set. To avoid overfitting, it is natural to plot reconstruction error on the test set. This is shown in Figure 12.14(b). Here we see that the error continues to go down even as the model becomes more complex! Thus we do not get the usual U-shaped curve that we typically expect to see.

What is going on? The problem is that PCA is not a proper generative model of the data. It is merely a compression technique. If you give it more latent dimensions, it will be able to approximate the test data more accurately. By contrast, a probabilistic model enjoys a Bayesian Occam's razor effect (Section 5.3.1), in that it gets "punished" if it wastes probability mass on parts of the space where there is little data. This is illustrated in Figure 12.15, which plots the negative log likelihood, computed using PPCA, vs L . Here, on the test set, we see the usual U-shaped curve. These results are analogous to those in Section 11.5.2, where we discussed the issue of choosing K in the K-means algorithm vs using a GMM.

12.3.2.1 Profile likelihood

Although there is no U-shape, there is sometimes a "regime change" in the plots, from relatively large errors to relatively small. One way to automate the detection of this is described in (Zhu and Ghodsi 2006). The idea is this. Let λ_k be some measure of the error incurred by a model of size k , such that $\lambda_1 \geq \lambda_2 \geq \dots \geq \lambda_{L_{\text{max}}}$. In PCA, these are the eigenvalues, but the method can also be applied to K-means. Now consider partitioning these values into two groups, depending on whether $k < L$ or $k > L$, where L is some threshold which we will determine. To measure the quality of L , we will use a simple change-point model, where $\lambda_k \sim N(\mu_1, \sigma^2)$ if $k \leq L$, and $\lambda_k \sim N(\mu_2, \sigma^2)$ if $k > L$. (It is important that σ^2 be the same in both models, to prevent overfitting in the case where one regime has less data than the other.) Within each of the two regimes, we assume the λ_k are iid, which is obviously incorrect, but is adequate for our present purposes. We can fit this model for each $L = 1: L_{\text{max}}$ by partitioning the data and

computing the MLEs, using a pooled estimate of the variance:

$$\mu_1(L) = k \leq L \lambda k L, \mu_2(L) = k > L \lambda k N - L \quad (12.70)$$

$$\sigma^2(L) = k \leq L (\lambda k - \mu_1(L))^2 + k > L (\lambda k - \mu_2(L))^2 N \quad (12.71)$$

We can then evaluate the profile log likelihood

$$(L) = \sum_{k=1}^L \log N(\lambda k | \mu_1(L), \sigma^2(L)) + K \sum_{k=L+1}^{N-1} \log N(\lambda k | \mu_2(L), \sigma^2(L)) \quad (12.72)$$

Finally, we choose $L^* = \arg \max (L)$. This is illustrated in Figure 12.16. On the left, we plot the scree plot, which has the same shape as in Figure 12.14(a). On the right, we plot the profile negloglik likelihood. Rather miraculously, we see a fairly well-determined peak.

下面先完整翻译原文（含图注与公式，不漏译），再给出通俗解释，最后列要点小结。

逐字翻译

图 12.14 用 PCA 的潜在维度个数与在 MNIST 上的重构误差关系。(a) 训练集。 (b) 测试集。
图由 `pcaOverfitDemo` 生成。

图 12.15 用 PPCA 的潜在维度个数与在 MNIST 上的**负对数似然**关系。(a) 训练集。 (b) 测试集。
图由 `pcaOverfitDemo` 生成。

图 12.16 (a) 与图 12.14(a) 相对应的训练集**碎石图** (scree plot) 。(b) **轮廓似然** (profile likelihood) 。图由 `pcaOverfitDemo` 生成。

12.3.2 选择 PCA 的模型

由于 PCA 不是一个概率模型，我们无法使用上面那些方法。一个显而易见的“似然替代物”是**重构误差**：

$$E(D, L) = \frac{1}{|D|} \sum_{x_i \in D} \|x_i - \hat{x}_i\|^2 \quad (12.67)$$

在 PCA 的情形下，重构为 $\hat{x}_i = Wz_i + \mu$ ，其中 $z_i = W^\top(x_i - \mu)$ ，而 W, μ 用训练集 D_{train} 估计。

图 12.14(a) 把 $E(D_{\text{train}}, L)$ 随 L 的变化画在了图 12.6 的 MNIST 训练数据上。可以看到它下降得很快，说明只用少量因子就能捕捉大部分像素间的经验相关性，这在图 12.6 里也有

定性展示。练习 12.5 让你证明：只用 L 个项（主成分）时的残差误差等于被丢弃的特征值之和：

$$E(\mathbf{D}_{\text{train}}, L) = \sum_{j=L+1}^D \lambda_j \quad (12.68)$$

因此，绘制误差的一个替代方法是把保留的特征值按从大到小画出来。这叫碎石图（scree plot），因为“曲线看起来像一座山的侧面，‘scree’ 指堆在山脚的碎石”。它与残差误差图具有相同形状。一个相关量是解释方差占比：

$$F(\mathbf{D}_{\text{train}}, L) = \frac{\sum_{j=1}^L \lambda_j}{\sum_{j=1}^{L_{\max}} \lambda_j} \quad (12.69)$$

它包含与碎石图相同的信息。

当然，如果我们取 $L = \text{rank}(X)$ ，就能在训练集上得到零重构误差。为避免过拟合，自然应当在测试集上画重构误差。图 12.14(b) 就是这样。这里可以看到：**即使模型越来越复杂，误差仍在下降！** 因而我们没有得到通常所见的 U 形曲线。

这是为什么？问题在于 PCA 不是数据的一个**真正的生成模型**。它只是一个**压缩技术**。如果你给它更多潜在维度，它在测试数据上也能更精确地近似。相反，概率模型具有**贝叶斯奥卡姆剃刀效应**（第 5.3.1 节）：如果它把概率质量浪费在几乎没有数据的区域，就会受到“惩罚”。图 12.15 做了说明：它画的是使用 PPCA 计算的**负对数似然**与 L 的关系。在测试集上，我们看到了通常的 U 形曲线。这些结果与第 11.5.2 节的类比类似：那里我们讨论了在 K-means 中选择 K 与在 GMM 中选择 K 的区别。

12.3.2.1 轮廓似然 (Profile likelihood)

尽管没有 U 形，但曲线中有时会出现从“相对大误差”到“相对小误差”的**“相变/阶段切换”。一种自动检测的方法见 (Zhu 和 Ghodsi 2006)。想法如下。令 λ_k 是某个大小为 k 的模型所导致的误差度量，且 $\lambda_1 \geq \lambda_2 \geq \dots \geq \lambda_{L_{\max}}$ 。在 PCA 中，这些就是特征值**，不过该方法也能用于 K-means。现在把这些值按 k 是否小于阈值 L 分成两组。为了度量这个 L 的质量，我们用一个简单的**变点模型**：

当 $k \leq L$ 时 $\lambda_k \sim N(\mu_1, \sigma^2)$ ，当 $k > L$ 时 $\lambda_k \sim N(\mu_2, \sigma^2)$ 。（关键是两边的方差 σ^2 相同，以避免当某一边数据少时发生过拟合。）在两个阶段内，我们假设 λ_k 独立同分布——这当然不严格，但对当前目的足够。对每个 $L = 1:L_{\max}$ ，我们通过分组并计算 MLE 来拟合模型，使用**合并方差估计**：

$$\mu_1(L) = \frac{\sum_{k \leq L} \lambda_k}{L}, \quad \mu_2(L) = \frac{\sum_{k > L} \lambda_k}{N - L} \quad (12.70)$$

$$\sigma^2(L) = \frac{\sum_{k \leq L} (\lambda_k - \mu_1(L))^2 + \sum_{k > L} (\lambda_k - \mu_2(L))^2}{N} \quad (12.71)$$

然后计算轮廓对数似然

$$\ell(L) = \sum_{k=1}^L \log N(\lambda_k | \mu_1(L), \sigma^2(L)) + \sum_{k=L+1}^K \log N(\lambda_k | \mu_2(L), \sigma^2(L)) \quad (12.72)$$

最后选择 $L^* = \arg \max_L \ell(L)$ 。图 12.16 给出了示例。左图是碎石图（与图 12.14(a) 形状相同），右图是轮廓对数似然。几乎“奇迹般地”，我们看到了一个相当清晰的峰值。

通俗解释（尽量白话）

1) 用什么来挑 L ？

- PCA 不是概率模型 → 不能直接比“似然”。
- 最直接的替代物：**重构误差**（把样本投到前 L 个主成分再还原，和原样本的差）。训练集上必然随 L 下降；测试集上，PCA 也会**持续下降**（图 12.14(b)），所以看不出“最佳 L ”的拐点——越大越好，像压缩码率越高图越清晰。

2) 为什么 PPCA 会出现 U 形？

- PPCA 是**概率模型**，会对“浪费容量”的复杂模型进行**惩罚**（奥卡姆剃刀）。
- 维度太小 → 拟合差，负对数似然大；维度适中 → 拟合好；维度太大 → 参数多，给“空白区域”也分了概率 → 测试似然又变差。于是得到**U 形**（图 12.15）。

3) 还能看什么图？

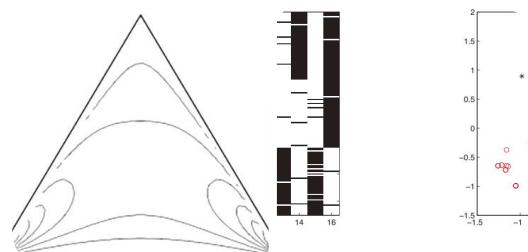
- **碎石图** (scree)：把**特征值**从大到小画出来；残差误差 = **被丢弃的特征值之和**，所以二者形状一样。
- **解释方差占比**：前 L 个特征值的和 / 全部特征值的和，常用来报告“保留了多少信息”。

4) 如何自动找“拐点/相变”？— 轮廓似然

- 把按大小排好序的 λ_k (PCA 中为特征值) 当作两段**高斯**数据：前半段均值 μ_1 、后半段均值 μ_2 、**共同方差** σ^2 。
- 对每个可能的切分点 L 计算对应的 μ_1, μ_2, σ^2 与**对数似然**，取**最大的**那个 L 作为估计。
- 直观：前几维“信息大、误差大”，后面“尾巴小、误差小”，两段均值不同；选一个切分点让两段各自更像同方差高斯 → 就是“相变点”。
- 图 12.16(b) 的峰值就给出了一个**明确的** L 。

要点小结

- PCA 选 L :
 - 训练误差、测试误差都会随 L 下降 (PCA 只是压缩/投影，不会自然出现 U 形)，因此单看误差难以给出“最佳 L ”。
 - 残差 = 被丢弃特征值之和；碎石图/解释方差占比可直观查看“信息保留量”。
- PPCA 选 L : 看** (负) 对数似然常出现U形**；中间的最小值给出合适的 L (奥卡姆剃刀作用)。
- 轮廓似然法 (Zhu & Ghodsi, 2006)：把按序特征值分成两段同方差高斯，逐一试 L 并最大化对数似然 → 自动找出“相变”点，常与碎石图拐点一致。
- 实务建议：若能用概率版 (PPCA/FA)，优先用验证集似然或变分/BIC挑 L ；若仅有 PCA，可结合碎石图 + 解释方差占比 + 轮廓似然做决定，并用下游任务 (分类/回归表现) 再验证。



翻译并解释以下内容，翻译不能有遗漏，解释请通俗易懂且详尽，因为我的数学基础不太好，最后总结要点。

Figure 12.17 Some examples of the logistic normal distribution defined on the 3d simplex. (a) Diagonal covariance and non-zero mean. (b) Negative correlation between states 1 and 2. (c) Positive correlation between states 1 and 2. Source: Figure 1 of (Blei and Lafferty 2007). Used with kind permission of David Blei.

Figure 12.18 Left: 150 synthetic 16 dimensional bit vectors. Right: the 2d embedding learned by binary PCA, using variational EM. We have color coded points by the identity of the true “prototype” that generated them. Figure generated by binaryFaDemoTipping.

12.4 PCA for categorical data

In this section, we consider extending the factor analysis model to the case where the observed data is categorical

rather than real-valued. That is, the data has the form $y_{ij} \in \{1, \dots, C\}$, where $j = 1: R$ is the number of observed response variables. We assume each y_{ij} is generated from a latent variable $z_i \in RL$, with a Gaussian prior, which is passed through the softmax function as follows:

$$(z_i) = N(0, I) \quad (12.73)$$

$$p(y_{ir}|z_i, \theta) = R r=1 \text{Cat}(y_{ir}|S(W^T r z_i + w_0 r)) \quad (12.74)$$

where $W_r \in RL \times M$ is the factor loading matrix for response j , and $w_0 r \in RM$ is the offset term for response r , and $\theta = (W_r, w_0 r) R r=1$. (We need an explicit offset term, since clamping one element of z_i to 1 can cause problems when computing the posterior covariance.) As in factor analysis, we have defined the prior mean to be $m_0 = 0$ and the prior covariance $V_0 = I$, since we can capture non-zero mean by changing $w_0 j$ and non-identity covariance by changing W_r . We will call this categorical PCA. See Chapter 27 for a discussion of related models.

It is interesting to study what kinds of distributions we can induce on the observed variables by varying the parameters. For simplicity, we assume there is a single ternary response variable, so y_i lives in the 3d probability simplex. Figure 12.17 shows what happens when we vary the parameters of the prior, m_0 and V_0 , which is equivalent to varying the parameters of the likelihood, W_1 and $w_0 1$. We see that this can define fairly complex distributions over the simplex. This induced distribution is known as the logistic normal distribution (Aitchison 1982). We can fit this model to data using a modified version of EM. The basic idea is to infer a Gaussian approximation to the posterior $p(z_i|y_i, \theta)$ in the E step, and then to maximize θ in the M step. The details for the multiclass case, can be found in (Khan et al. 2010) (see also Section 21.8.1.1). The details for the binary case for the sigmoid link can be found in Exercise 21.9, and for the probit link in Exercise 21.10.

One application of such a model is to visualize high dimensional categorical data. Figure 12.18(a) shows a simple example where we have 150 6-dimensional bit vectors. It is clear that each sample is just a noisy copy of one of three binary prototypes. We fit a 2d catFA to this model, yielding approximate MLEs $\hat{\theta}$. In Figure 12.18(b), we plot $E + z_i|x_i, \hat{\theta}$. We see that there are three distinct clusters, as is to be expected.

In (Khan et al. 2010), we show that this model outperforms finite mixture models on the task of imputing missing entries in design matrices consisting of real and categorical data. This is useful for analysing social science survey data, which often has missing data and variables of mixed type.

已思考 24s

下面先完整翻译原文（含图注与公式，不漏译），然后给出通俗解释，最后总结要点。

逐字翻译

图 12.17 一些定义在三维单纯形 (3d simplex) 上的逻辑正态分布的例子。(a) 协方差为对角阵且均值非零。(b) 状态 1 与状态 2 之间负相关。(c) 状态 1 与状态 2 之间正相关。来源：Blei & Lafferty (2007) 图 1。经 David Blei 授权使用。

图 12.18 左：150 个合成的 16 维比特向量。右：用**二值 PCA** (variational EM 估计) 学习到的二维嵌入。我们按生成这些点的真实“原型”的身份对点做了颜色标记。图由 `binaryFaDemoTipping` 生成。

12.4 用于类别型数据的 PCA

本节把因子分析模型推广到观测数据是**类别型**而不是实数型的情形。也就是说，数据为 $y_{ij} \in \{1, \dots, C\}$ ，其中 $j = 1:R$ 是观测响应变量的个数。我们假设每个 y_{ij} 都由一个**潜变量** $z_i \in \mathbb{R}^L$ 生成，潜变量具有高斯先验，并通过 softmax 函数映射，具体为：

$$p(z_i) = N(0, I) \quad (12.73)$$

$$p(y_i | z_i, \theta) = \prod_{r=1}^R \text{Cat}(y_{ir} | S(W_r^\top z_i + w_{0r})) \quad (12.74)$$

其中 $W_r \in \mathbb{R}^{L \times M}$ 是响应 r 的因子载荷矩阵， $w_{0r} \in \mathbb{R}^M$ 是响应 r 的偏置项， $\theta = (W_r, w_{0r})_{r=1}^R$ 。（我们需要显式的偏置项，因为把 z_i 的某个分量“钉死”为 1 在计算后验协方差时会造成问题。）与因子分析一样，我们把先验均值设为 $m_0 = 0$ 、先验协方差设为 $V_0 = I$ ，因为**非零均值**可以靠改变 w_{0j} 表达，**非单位协方差**可以靠改变 W_r 表达。我们把这种模型称作**类别型 PCA**。相关模型见第 27 章。

研究“改变参数能在观测变量上诱导出什么样的分布”是很有趣的。为简单起见，假定只有一个三分类的响应变量，于是 y_i 位于**三维概率单纯形**上。图 12.17 显示了当我们改变先验

参数 m_0 与 V_0 (等价于改变似然参数 W_1 与 w_{01}) 时会发生什么。可以看到，这能在单纯形上定义相当复杂的分布。这个诱导分布称为**逻辑正态分布** (Aitchison, 1982)。

我们可以用**改进版 EM**来拟合该模型。基本思想是：在**E 步里**对后验 $p(z_i | y_i, \theta)$ 做**高斯近似**；在**M 步里**最大化 θ 。多分类的细节见 Khan et al. (2010) (亦见 21.8.1.1 节)。二分类情形下，sigmoid 链接的细节见练习 21.9，probit 链接见练习 21.10。

这种模型的一个应用是**可视化高维类别数据**。图 12.18(a) 给出了一个简单例子：我们有 150 个 16 (文中原文作 6) 维比特向量。显然，每个样本只是三个二值“原型”之一的带噪拷贝。我们拟合一个二维的 catFA (类别版因子分析)，得到近似极大似然 $\hat{\theta}$ 。在图 12.18(b) 中，我们画出了 $E[z_i | x_i, \hat{\theta}]$ 。可以看到，正如预期，出现了 3 个明显的簇。

在 Khan et al. (2010) 中，我们展示了该模型在“对包含实值和类别变量的设计矩阵进行**缺失值填补**”这一任务上优于有限混合模型。这对于分析**社会科学调查数据**很有用，因为此类数据经常有缺失，且变量类型混杂。

通俗解释

1) 为什么要“类别版”的 PCA?

传统 PCA/FA 适合**连续数据**。遇到**离散/多分类变量** (例如问卷题、是否点击等)，直接做 PCA 不合适。做法是引入连续的**潜在因子** z ，让它们通过一堆**共享的线性函数** $W_r^T z + w_{0r}$ 再经 softmax (或 sigmoid/probit) 变成类别概率，从而生成观测的分类 y 。这就像“很多多项逻辑回归**共享同一组隐变量**”。

2) 三角形里的等高线是啥？

三角形是“3 类概率的所有可能取值”的空间 (每个点是 (p_1, p_2, p_3) ，三者相加为 1)。把高斯 z 经过 softmax 后在三角形里诱导出一个分布，这就是**逻辑正态分布**：

- 调整均值/协方差 (或等价地调整 W, w_0) 会让概率**偏向某一角**、或沿着某条边扩散、或让两类正/负相关 (图 12.17 a/b/c)。

3) 怎么训练？为什么要“近似”？

softmax 与高斯先验**不共轭**，后验 $p(z | y)$ 没有闭式解 → 用**变分 EM**：

- **E 步**：用一个高斯去近似 $p(z_i | y_i)$ (求它的均值与协方差)。
 - **M 步**：用这份近似后验来最大化参数 W, w_0 的期望对数似然。
- 二分类时可用常见的技巧 (如 Jaakkola–Jordan 上界、或 probit 的解析期望等)。

4) 它能做什么？

- **可视化/降维**：把高维的 0/1 或多分类向量嵌入到 2D/3D (图 12.18(b) 出现 3 个簇，与 3 个原型对应)。

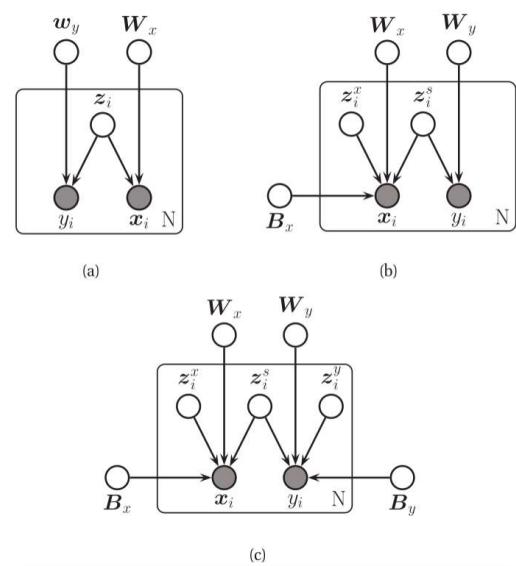
- **缺失值填补**: 因为共享的隐因子把各题目(列)“串”起来, 能根据其他答案推测缺失条目, 通常比“每列独立”或“简单混合模型”更好。

5) 与“混合模型”相比?

混合模型把每个样本归到某个原型(簇)。类别型PCA/FA则让每个样本由**连续因子**线性生成, 再经过非线性(softmax)映射, 能刻画“在多个方向上连续变化”的结构, 常常更节省参数、泛化更好。

要点小结

- **类别型PCA(catFA)**: $z \sim N(0, I)$; 每个响应 r 的类别 $y_{ir} \sim \text{Cat}(\text{softmax}(W_r^\top z_i + w_{0r}))$ 。
- **逻辑正态分布**: 把高斯经 softmax 映射到概率单纯形得到的分布; 能表达复杂相关结构(图 12.17)。
- **训练**: 用**变分EM**。E步以高斯近似后验 $p(z | y)$, M步更新 W, w_0 。二分类可用 sigmoid/probit 链接的专门技巧。
- **用途**:
 - 高维类别数据的**可视化/降维** (“二值/多项式PCA”) ;
 - **缺失值填补**, 尤其在混合类型(数值+类别)的调查/问卷数据中表现优良。
- **直觉**: 很多“共享的逻辑回归”由同一个低维连续因子驱动 → 用少量因子刻画大量类别变量的共变结构。



翻译并解释以下内容, 翻译不能有遗漏, 解释请通俗易懂且详尽, 因为我的数学基础不太好, 最后总结要点。

Figure 12.19 Gaussian latent factor models for paired data. (a) Supervised PCA. (b) Partial least squares. (c) Canonical correlation analysis.

12.5 PCA for paired and multi-view data

It is common to have a pair of related datasets, e.g., gene expression and gene copy number, or movie ratings by users and movie reviews. It is natural to want to combine these together into a low-dimensional embedding. This is an example of data fusion. In some cases, we might want to predict one element of the pair, say x_{i1} , from the other one, x_{i2} , via the low-dimensional “bottleneck”.

Below we discuss various latent Gaussian models for these tasks, following the presentation of (Virtanen 2010). The models easily generalize from pairs to sets of data, x_{im} , for $m = 1: M$. We focus on the case where $x_{im} \in \mathbb{R}^{D_m}$. In this case, the joint distribution is multivariate Gaussian, so we can easily fit the models using EM, or Gibbs sampling.

We can generalize the models to handle discrete and count data by using the exponential family as a response distribution instead of the Gaussian, as we explain in Section 27.2.2. However, this will require the use of approximate inference in the E step (or an analogous modification to MCMC).

12.5.1 Supervised PCA (latent factor regression)

Consider the following model, illustrated in Figure 12.19(a):

$$p(z_i) = N(0, I_L) \quad (12.75)$$

$$p(y_i|z_i) = N(w^T y_i + \mu_y, \sigma^2_y) \quad (12.76)$$

$$p(x_i|z_i) = N(Wx_i + \mu_x, \sigma^2_x I_D) \quad (12.77)$$

In (Yu et al. 2006), this is called supervised PCA. In (West 2003), this is called Bayesian factor regression. This model is like PCA, except that the target variable y_i is taken into account when learning the low dimensional embedding.

Since the model is jointly Gaussian, we have

$$y_i|x_i \sim N(x_i^T w + w^T C y_i) \quad (12.78)$$

where $w = \Psi^{-1} W x_i$, $\Psi = \sigma^2_x I_D$, and $C = I + W^T W$. So although this is a joint density model of (y_i, x_i) , we can infer the implied conditional distribution. We now show an interesting connection to Zellner’s g-prior. Suppose $p(w) = N(0, 1/g\Sigma_2)$, and let $X = RVT$ be the SVD of X , where $VT = I$ and $RTR^T = \Sigma_2 = \text{diag}(\sigma_j^2)$ contains the

squared singular values. Then one can show (West 2003) that

$$p(w) = N(0, gV - T\Sigma - 2V - 1) = N(0, g(XTX)^{-1}) \quad (12.79)$$

So the dependence of the prior for w on X arises from the fact that w is derived indirectly by a joint model of X and y .

The above discussion focussed on regression. (Guo 2009) generalizes CCA to the exponential family, which is more appropriate if x_i and/or y_i are discrete. Although we can no longer compute the conditional $p(y_i|x_i, \theta)$ in closed form, the model has a similar interpretation to the regression case, namely that we are predicting the response via a latent "bottleneck". The basic idea of compressing x_i to predict y_i can be formulated using information theory. In particular, we might want to find an encoding distribution $p(z|x)$ such that we minimize

$$I(X; Z) - \beta I(X; Y) \quad (12.80)$$

where $\beta \geq 0$ is some parameter controlling the tradeoff between compression and predictive accuracy. This is known as the information bottleneck (Tishby et al. 1999). Often Z is taken to be discrete, as in clustering. However, in the Gaussian case, IB is closely related to CCA (Chechik et al. 2005).

We can easily generalize CCA to the case where y_i is a vector of responses to be predicted, as in multi-label classification. (Ma et al. 2008; Williamson and Ghahramani 2008) used this model to perform collaborative filtering, where the goal is to predict $y_{ij} \in \{1, \dots, 5\}$, the rating person i gives to movie j , where the "side information" x_i takes the form of a list of i 's friends. The intuition behind this approach is that knowledge of who your friends are, as well as the ratings of all other users, should help predict which movies you will like. In general, any setting where the tasks are correlated could benefit from CCA. Once we adopt a probabilistic view, various extensions are straightforward. For example, we can easily generalize to the semi-supervised case, where we do not observe y_i for all i (Yu et al. 2006).

12.5.1.1 Discriminative supervised PCA

One problem with this model is that it puts as much weight on predicting the inputs x_i as the outputs y_i . This can be partially alleviated by using a weighted objective of the following form (Rish et al. 2008):

$$l(\theta) = \prod_i p(y_i | \eta_i y) \alpha_y p(x_i | \eta_i x) \alpha_x \quad (12.81)$$

where the α_m control the relative importance of the data sources, and $\eta_{im} = W_m z_i$. For Gaussian data, we can see that α_m just controls the noise variance:

$$l(\theta) \propto \prod_i \exp(-\frac{1}{2} \alpha_x \|x_i - \eta_i x\|^2) \exp(-\frac{1}{2} \alpha_y \|y_i - \eta_i y\|^2) \quad (12.82)$$

This interpretation holds more generally for the exponential family. Note, however, that it is hard to estimate the α_m parameters, because changing them changes the normalization constant of the likelihood. We give an alternative approach to weighting y more heavily below.

下面先完整翻译原文（含图注与公式，不删不漏），接着给出通俗解释，最后总结要点。

逐字翻译

图 12.19 成对数据 (paired data) 的高斯潜因子模型。 (a) 监督式 PCA。(b) 偏最小二乘 (PLS)。 (c) 典型相关分析 (CCA)。

12.5 成对与多视图数据的 PCA

现实中经常会遇到一对相关的数据集，例如“基因表达量与基因拷贝数”、或“用户的电影评分与电影的文字评论”。很自然地，我们希望把它们结合起来，映射到一个低维嵌入中——这就是数据融合的一个例子。在某些场景下，我们还希望通过低维“瓶颈”用一个元素（如 x_{i1} ）去预测另一个元素（如 x_{i2} ）。

下面我们按照 (Virtanen 2010) 的讲法，讨论若干用于这些任务的潜在高斯模型。这些模型很容易从“一对”推广到“一组”数据 $\{x_i^{(m)}\}_{m=1}^M$ 。我们聚焦于 $x_i^{(m)} \in \mathbb{R}^{D_m}$ 的情形。这时联合分布是多元高斯，因此可用 EM 或 Gibbs 采样轻松拟合。

把模型推广到离散/计数数据也很容易：把高斯响应换成指数族响应分布即可（第 27.2.2 节会解释）。但这样一来 E 步就需要近似推断（或在 MCMC 中做相应修改）。

12.5.1 监督式 PCA (潜因子回归)

考虑图 12.19(a) 中的模型：

$$p(z_i) = \mathcal{N}(0, I_L) \quad (12.75)$$

$$p(y_i | z_i) = \mathcal{N}(w_y^\top z_i + \mu_y, \sigma_y^2) \quad (12.76)$$

$$p(x_i | z_i) = \mathcal{N}(W_x z_i + \mu_x, \sigma_x^2 I_D) \quad (12.77)$$

在 (Yu et al., 2006) 中这叫**监督式 PCA**；在 (West, 2003) 中这叫**贝叶斯因子回归**。该模型与 PCA 类似，但在学习低维嵌入时把**目标变量** y_i 也考虑进去。由于模型是**联合高斯**，我们有

$$y_i | x_i \sim N(x_i^\top w, \sigma_y^2 + w_y^\top C w_y) \quad (12.78)$$

其中 $w = \Psi^{-1} W_x C w_y$ 、 $\Psi = \sigma_x^2 I_D$ 、 $C^{-1} = I + W_x^\top \Psi^{-1} W_x$ 。因此，尽管模型给出了 (y_i, x_i) 的**联合密度**，我们依然能推出隐含的**条件分布**。

下面给出与 Zellner 的 g-先验之间一个有趣的联系。设 $p(w_y) = N(0, \frac{1}{g} \Sigma^2)$ ，并令 $X = RV^\top$ 是 X 的 SVD，其中 $V^\top V = I$ ，且 $R^\top R = \Sigma^2 = \text{diag}(\sigma_j^2)$ 放着**奇异值的平方**。则可证明 (West 2003)：

$$p(w) = N(0, g V^{-\top} \Sigma^{-2} V^{-1}) = N(0, g (X^\top X)^{-1}) \quad (12.79)$$

也就是说， w 的先验依赖于 X 的事实，来源于： w 是由 X 与 y 的**联合模型**间接导出的。

以上讨论聚焦于**回归**。(Guo, 2009) 将 CCA 推广到了**指数族**，这更适合 x_i 与/或 y_i 为离散量的情形。虽然这时我们不能再闭式地计算 $p(y_i | x_i, \theta)$ ，但其解释仍与回归情形类似：我们是通过一个**潜在的“瓶颈”**来预测响应。用信息论可以刻画“压缩 x 以预测 y ”的思想，具体是寻找编码分布 $p(z | x)$ 以最小化

$$I(X; Z) - \beta I(Z; Y), \quad \beta \geq 0 \quad (12.80)$$

这叫做**信息瓶颈** (Tishby et al., 1999)。常见做法令 Z 为离散量（如聚类）。不过在**高斯**情形下，IB 与 CCA 密切相关 (Chechik et al., 2005)。

我们可以很容易把 CCA 推广到“ y_i 是一个**向量**”的情形（多标签分类）。(Ma et al., 2008; Williamson & Ghahramani, 2008) 将该模型用于**协同过滤**：目标是预测 $y_{ij} \in \{1, \dots, 5\}$ （用户 i 对电影 j 的评分），而“侧信息” x_i 是用户 i 的**好友列表**。直觉是：知道你的朋友是谁，以及其他用户的评分，都有助于预测你会喜欢哪些电影。一般来说，只要**任务之间存在相关性**，就能从 CCA 中获益。采用**概率化视角**后，各种扩展都很容易。例如很容易推广到**半监督**情形（不是所有 i 都观测到 y_i ）(Yu et al., 2006)。

12.5.1.1 判别式监督 PCA

该模型的一个问题是：它对预测输入 x_i 与输出 y_i 给予了**同等权重**。可通过下式的**加权目标部分缓解** (Rish et al., 2008)：

$$\ell(\theta) = \prod_i p(y_i | \eta_i^y)^{\alpha_y} p(x_i | \eta_i^x)^{\alpha_x} \quad (12.81)$$

其中 α_m 控制各数据源的相对重要性， $\eta_i^{(m)} = W_m z_i$ 。对高斯数据，可见 α_m 只是**控制噪声方差**：

$$\ell(\theta) \propto \prod_i \exp\left(-\frac{1}{2}\alpha_x \|x_i^\top - \eta_i^x\|^2\right) \exp\left(-\frac{1}{2}\alpha_y \|y_i^\top - \eta_i^y\|^2\right) \quad (12.82)$$

这一解释对指数族也更普遍地成立。不过要注意， α_m **很难估计**，因为改变它们会改变似然的**归一化常数**。下面我们会给出另一种让 y 权重更大的做法。

通俗解释

这节在讲什么？

当我们手里有**两种视角**（或多种视角）的数据，比如（图片，文字描述）或（用户评分，用户社交关系），希望**共同学习一个低维表示** z ，既能概括 x ，又能帮助预测 y 。图 12.19 里的三个图就是常见的**高斯潜因子**版本：

- (a) **监督式 PCA (又名因子回归)**

$z \sim N(0, I)$ ，用同一个 z 生成 x 和 y 。这样学出来的 z 既解释 x ，也与 y 强相关——比单纯 PCA 更“面向任务”。因为都是高斯，推一下就能得到 $p(y | x)$ 的闭式公式（式 12.78），相当于一个从 x 到 y 的**回归器**，其回归系数由联合模型推导出来。

- (b) **PLS (偏最小二乘)**

也是让 x, y 通过共同的低维 z 连接，目标是挑出能**同时解释 x 且对 y 预测力强**的方向；在高斯线性世界里与上面模型密切相关。

- (c) **CCA (典型相关分析)**

寻找两边的线性变换，使得变换后的低维表示**相互最相关**。它强调“跨视角相关性”，而不是单侧的重构误差。

信息瓶颈 (IB) 的联系

IB 的目标是把 x 压到低维 z （小 $I(X; Z)$ ）但又让 z 尽量**保留对 y 的信息**（大 $I(Z; Y)$ ）。在**高斯**情形下，这与 CCA 其实是一回事：CCA 找到的就是能最大化相关性的线性低维表示。

g-先验的联系

从监督式 PCA 的联合高斯推导出 $p(y | x)$ 后，可以看到**回归系数** w 的先验等价于 $w \sim N(0, g(X^\top X)^{-1})$ ——这正是贝叶斯线性回归里著名的 Zellner g-prior。直觉：因为 w 是从“ X 与 y 的联合模型”间接得到的，先验自然会依赖 X 的协方差结构。

为什么要“判别式”的监督 PCA？

上面的联合模型同时重构 x 与 y ，如果我们只关心**预测 y** ，就希望给 y **更高权重**。式 (12.81)–(12.82) 就是一个简单的做法：把两边似然分别乘上指数权重 α_x, α_y （在高斯下等价于改变噪声方差）。但 α 的估计不太方便，因为会影响归一化常数，所以文献还会用其他办法专门偏向 y 。

推广与应用

- 可以把 y 设为向量（多标签），或把多个视角扩展到 $M > 2$ 。
- 结合侧信息（如用户好友关系）做协同过滤；也能做半监督（不是所有样本都标了 y ）。
- 若数据是离散/计数型，可用指数族版本，但 E 步要做近似推断。

要点小结

- 目标：**把多视角/成对数据通过一个低维 z 融合起来，既能描述数据结构，又能做预测。
- 三大经典模型（高斯版）：**
 - 监督式 PCA/因子回归：** z 同时生成 x, y ；能得到 $p(y | x)$ 的闭式回归（式 12.78）；与 g-prior 有关联（式 12.79）。
 - PLS：** 选能解释 x 且对 y 预测力强的方向。
 - CCA：** 最大化两视角低维表示的相关性；在高斯下与信息瓶颈相关。
- 判别式加权：**用 α_x, α_y 给 x, y 的似然加权（式 12.81–12.82），等价于调整噪声方差；但 α 不易估计。
- 扩展：**多视角 ($M > 2$)、半监督、指数族（离散/计数）版本、EM 或 Gibbs 拟合。
- 应用：**多模态融合、协同过滤、带侧信息的预测、多标签学习等。

翻译并解释以下内容，翻译不能有遗漏，解释请通俗易懂且详尽，因为我的数学基础不太好，最后总结要点。

12.5.2 Partial least squares

The technique of partial least squares (PLS) (Gustafsson 2001; Sun et al. 2009) is an asymmetric or more "discriminative" form of supervised PCA. The key idea is to allow some of the (co)variance in the input features to be explained by its own subspace, zx_i , and to let the rest of the subspace, zs_i , be shared between input and output.

The model has the form

$$p(z_i) = N(zs_i | 0, I_{Ls})N(zx_i | 0, I_{Lx}) \quad (12.83)$$

$$p(y_i | z_i) = N(Wyzs_i + \mu_y, \sigma^2 I_{Dy}) \quad (12.84)$$

$$p(x_i | z_i) = N(Wxzs_i + Bzx_i + \mu_x, \sigma^2 I_{Dx}) \quad (12.85)$$

See Figure 12.19(b). The corresponding induced distribution on the visible variables has the form
 $p(v_i|\theta) = N(v_i|Wz_i + \mu, \sigma^2 I)N(z_i|0, I)dz_i = N(v_i|\mu, WWT + \sigma^2 I)$ (12.86)

where $v_i = (x_i; y_i)$, $\mu = (\mu_y; \mu_x)$ and
 $W = Wy 0 Wx Bx$ (12.87)

$WWT = WyWT y WxWT x WxWT x WxWT x + BxBT x$
(12.88)

We should choose L large enough so that the shared subspace does not capture covariatespecific variation. This model can be easily generalized to discrete data using the exponential family (Virtanen 2010).

12.5.3 Canonical correlation analysis

Canonical correlation analysis or CCA is like a symmetric unsupervised version of PLS: it allows each view to have its own “private” subspace, but there is also a shared subspace. If we have two observed variables, x_i and y_i , then we have three latent variables, $z_s i \in RL_0$ which is shared, $z_x i \in RL_x$ and $z_y i \in RL_y$ which are private. We can write the model as follows (Bach and Jordan 2005):

$$p(z_i) = N(z_s i | 0, I_{L_s})N(z_x i | 0, I_{L_x})N(z_y i | 0, I_{L_y}) \quad (12.89)$$

$$p(x_i|z_i) = N(x_i|Bx z_s i + Wx z_x i + \mu_x, \sigma^2 I D_x) \quad (12.90)$$

$$p(y_i|z_i) = N(y_i|By z_s i + Wy z_x i + \mu_y, \sigma^2 I D_y) \quad (12.91)$$

See Figure 12.19(c). The corresponding observed joint distribution has the form

$$p(v_i|\theta) = N(v_i|Wz_i + \mu, \sigma^2 I)N(z_i|0, I)dz_i = N(v_i|\mu, WWT + \sigma^2 I D) \quad (12.92)$$

where

$$W = Wx Bx 0 Wy 0 By \quad (12.93)$$

$$WWT = WxWT x + BxBT x WxWT y WyWT y WyWT y + ByBT y \quad (12.94)$$

One can compute the MLE for this model using EM. (Bach and Jordan 2005) show that the resulting MLE is equivalent (up to rotation and scaling) to the classical, non-probabilistic view. However, the advantages of the probabilistic view are many: we can trivially generalize to $M > 2$ observed variables; we can create mixtures of CCA (Viinikanoja et al. 2010); we can create sparse versions of CCA using ARD (Archambeau and Bach 2008); we can generalize to the exponential family (Klami et al. 2010); we can perform Bayesian inference of the parameters (Wang 2007; Klami and Kaski 2008); we can handle non-

parametric sparsity-promoting priors for W and B (Rai and Daume 2009); and so on.

下面先逐字完整翻译原文（含公式与图指代，不漏译），再给出通俗解释，最后总结要点。

逐字翻译

12.5.2 偏最小二乘 (Partial least squares, PLS)

偏最小二乘 (PLS) (Gustafsson 2001; Sun 等 2009) 是一种**不对称的**、或者说更“**判别式**”的监督式 PCA。关键思想是：允许输入特征中的一部分（协）方差由它**自己的子空间** z_i^x 来解释，而把其余的子空间 z_i^s **在输入和输出之间共享**。模型形式为

$$p(z_i) = N(z_i^s | 0, I_{L_s}) N(z_i^x | 0, I_{L_x}) \quad (12.83)$$

$$p(y_i | z_i) = N(W_y z_i^s + \mu_y, \sigma^2 I_{D_y}) \quad (12.84)$$

$$p(x_i | z_i) = N(W_x z_i^s + B_x z_i^x + \mu_x, \sigma^2 I_{D_x}) \quad (12.85)$$

见图 12.19(b)。对可见变量所诱导的分布为

$$p(v_i | \theta) = \int N(v_i | W z_i + \mu, \sigma^2 I) N(z_i | 0, I) dz_i = N(v_i | \mu, WW^\top + \sigma^2 I) \quad (12.86)$$

其中 $v_i = (x_i; y_i)$ 、 $\mu = (\mu_y; \mu_x)$ ，并且

$$W = \begin{pmatrix} W_y & 0 \\ W_x & B_x \end{pmatrix} \quad (12.87)$$

$$WW^\top = \begin{pmatrix} W_y W_y^\top & W_x W_y^\top \\ W_y W_x^\top & W_x W_x^\top + B_x B_x^\top \end{pmatrix} \quad (12.88)$$

我们应当把 L 选得足够大，使得共享子空间**不会把仅属于协变量的变动也“吸”进来**。该模型可用指数族把数据推广到离散情形 (Virtanen 2010)。

12.5.3 典型相关分析 (Canonical correlation analysis, CCA)

典型相关分析 (CCA) 就像是**对称的、无监督的** PLS：它允许每个视角都有自己的“**私有**”子空间，同时还存在一个**共享**子空间。若我们有两个观测变量 x_i 和 y_i ，那么就有三个潜变量：共享的 $z_i^s \in \mathbb{R}^{L_0}$ ，以及私有的 $z_i^x \in \mathbb{R}^{L_x}$ 、 $z_i^y \in \mathbb{R}^{L_y}$ 。模型可写为 (Bach 和 Jordan 2005)：

$$p(z_i) = \mathcal{N}(z_i^s | 0, I_{L_s}) \mathcal{N}(z_i^x | 0, I_{L_x}) \mathcal{N}(z_i^y | 0, I_{L_y}) \quad (12.89)$$

$$p(x_i | z_i) = \mathcal{N}(x_i | B_x z_i^x + W_x z_i^s + \mu_x, \sigma^2 I_{D_x}) \quad (12.90)$$

$$p(y_i | z_i) = \mathcal{N}(y_i | B_y z_i^y + W_y z_i^s + \mu_y, \sigma^2 I_{D_y}) \quad (12.91)$$

见图 12.19(c)。相应的观测联合分布为

$$p(v_i | \theta) = \int \mathcal{N}(v_i | Wz_i + \mu, \sigma^2 I) \mathcal{N}(z_i | 0, I) dz_i = \mathcal{N}(v_i | \mu, WW^\top + \sigma^2 I_D, 92) \quad (12.92)$$

其中

$$W = \begin{pmatrix} W_x & B_x & 0 \\ W_y & 0 & B_y \end{pmatrix} \quad (12.93)$$

$$WW^\top = \begin{pmatrix} W_x W_x^\top + B_x B_x^\top & W_x W_y^\top \\ W_y W_x^\top & W_y W_y^\top + B_y B_y^\top \end{pmatrix} \quad (12.94)$$

可以用 EM 来计算该模型的极大似然估计。Bach 和 Jordan (2005) 表明：所得的 MLE 与经典的**非概率版结果等价**（差一个旋转与尺度）。不过**概率视角**的好处很多：我们可以直接推广到 $M > 2$ 个观测变量；可以做 CCA 的**混合模型** (Viinikanoja 等 2010)；可以用 ARD 做**稀疏 CCA** (Archambeau 和 Bach 2008)；可以推广到**指数族** (Klami 等 2010)；可以对参数做**贝叶斯推断** (Wang 2007; Klami 与 Kaski 2008)；可以对 W 与 B 使用**非参数的促稀疏先验** (Rai 与 Daume 2009)；等等。

通俗解释

PLS：更偏“预测 y ”的监督式降维

- 有两个数据块：输入 x 和输出 y 。
- 设一个低维向量 z 拆成两部分：
 - **共享部分** z^s ：同时影响 x 和 y ；
 - x 的**私有部分** z^x ：只影响 x ，解释与 y 无关的那部分方差。
- 于是：

$$y \approx W_y z^s + \text{噪声}$$
 (只看共享信息来预测 y) ;

$$x \approx W_x z^s + B_x z^x + \text{噪声}$$
 (既有共享，也有只属于 x 的成分) 。
- 这就是“**不对称**”：我们专门为 x 留了私有维度，让模型把与 y 无关的 x 变化丢到 z^x 里，避免污染共享表示 z^s 。
- 实操要点：把共享维度 L_s 选得**够大**，否则会把本应在 z^x 里的“ x -专属变化”硬塞到共享里，削弱对 y 的预测力。

- 可推广到**离散/计数**数据：把高斯输出换成指数族（如 softmax、Poisson），思路不变，只是推断要用近似。

CCA：对称的“找两边最相关”的表示

- 现在两边都拥有**私有子空间** (z^x, z^y) 和一个**共享子空间** z^s 。
 - 目标不是直接最小化重构误差或预测误差，而是让两边通过 z^s **尽量相关**。
 - 高斯版 CCA 的概率模型如式 (12.89)–(12.94)。用 EM 拟合后，得到的解与经典 CCA (线性代数那套) **等价**。
 - 为什么用概率版？**
 - 自然扩展到多视角 $M > 2$ 。
 - 能做**混合 CCA** (多簇/非线性分段)。
 - 能做**稀疏 CCA** (通过 ARD 把不重要的载荷压到 0)。
 - 能换成**指数族**处理离散数据。
 - 能做**贝叶斯估计**与**不确定性量化**，或用**非参数先验**获得更强的稀疏性/自适应性。
-

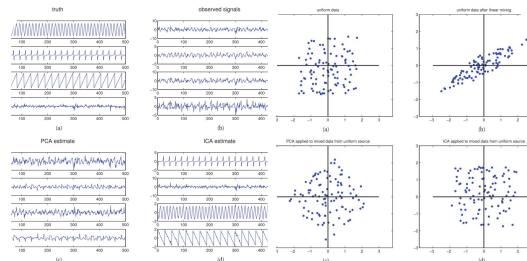
小例子直觉

- 电影场景： x =“用户的好友网络特征”， y =“对电影的评分”。
 - PLS**: 共享因子 z^s 抓住“社交相关的口味”， z^x 则吸收与评分无关但解释网络结构的变化。我们关心预测 y ，所以建模是**不对称的**。
 - CCA**: 在“用户侧特征”和“评分向量”两边各自做线性投影，要求两个投影“**最相关**”，强调**跨视角的共同结构**，**是对称的**。
-

要点小结

- PLS (判别、非对称)**
 - 结构： y 仅由共享因子 z^s 生成； x 由共享 $z^s +$ 私有 z^x 生成。
 - 目的：把与 y 无关的 x 变化隔离到私有部分，提高**对 y 的预测力**。
 - 选型：共享维度要足够；可扩到指数族。
- CCA (对称、无监督)**
 - 结构：两边都有私有与共享因子；通过共享因子实现**最大相关**。
 - 概率版优点：可做多视角、混合、稀疏、指数族与贝叶斯版本等；EM 拟合；与经典 CCA 等价 (差旋转/缩放)。
- 矩阵块形式**

- PLS 的 W, WW^\top 如式 (12.87)–(12.88)。
- CCA 的 W, WW^\top 如式 (12.93)–(12.94)。
- **何时用谁**
 - 更关心预测 y : 用 PLS / 监督式 PCA。
 - 更关心**发现两视角的共同结构、做对齐/检索**: 用 CCA。



翻译并解释以下内容，翻译不能有遗漏，解释请通俗易懂且详尽，因为我的数学基础不太好，最后总结要点。

Figure 12.20 Illustration of ICA applied to 500 iid samples of a 4d source signal. (a) Latent signals. (b) Observations. (c) PCA estimate. (d) ICA estimate. Figure generated by icaDemo, written by Aapo Hyvärinen.

Figure 12.21 Illustration of ICA and PCA applied to 100 iid samples of a 2d source signal with a uniform distribution. (a) Latent signals. (b) Observations. (c) PCA estimate. (d) ICA estimate. Figure generated by icaDemoUniform, written by Aapo Hyvärinen.

12.6 Independent Component Analysis (ICA) Consider the following situation. You are in a crowded room and many people are speaking. Your ears essentially act as two microphones, which are listening to a linear combination of the different speech signals in the room. Your goal is to deconvolve the mixed signals into their constituent parts. This is known as the cocktail party problem, and is an example of blind signal separation (BSS), or blind source separation, where “blind” means we know “nothing” about the source of the signals. Besides the obvious applications to acoustic signal processing, this problem also arises when analysing EEG and MEG signals, financial data, and any other dataset (not necessarily temporal) where latent sources or factors get mixed together in a linear way.

We can formalize the problem as follows. Let $\mathbf{x}_t \in \mathcal{R}^D$ be the observed signal at the sensors at "time" t , and $\mathbf{z}_t \in \mathcal{R}^L$ be the vector of source signals. We assume that

$$\mathbf{x}_t = \mathbf{W}\mathbf{z}_t + \mathbf{t} \quad (12.95)$$

where \mathbf{W} is an $D \times L$ matrix, and $\mathbf{t} \sim N(0, \Psi)$. In this section, we treat each time point as an independent observation, i.e., we do not model temporal correlation (so we could replace the t index with i , but we stick with t to be consistent with much of the ICA literature). The goal is to infer the source signals, $p(\mathbf{z}_t | \mathbf{x}_t, \Theta)$, as illustrated in Figure 12.20. In this context, \mathbf{W} is called the mixing matrix. If $L = D$ (number of sources = number of sensors), it will be a square matrix. Often we will assume the noise level, $|\Psi|$, is zero, for simplicity. So far, the model is identical to factor analysis (or PCA if there is no noise, except we don't in general require orthogonality of \mathbf{W}). However, we will use a different prior for $p(\mathbf{z}_t)$. In PCA, we assume each source is independent, and has a Gaussian distribution

$$p(\mathbf{z}_t) = \prod_{j=1}^L N(z_{tj} | 0, 1) \quad (12.96)$$

We will now relax this Gaussian assumption and let the source distributions be any non-Gaussian distribution

$$p(\mathbf{z}_t) = \prod_{j=1}^L p_j(z_{tj}) \quad (12.97)$$

Without loss of generality, we can constrain the variance of the source distributions to be 1, because any other variance can be modelled by scaling the rows of \mathbf{W} appropriately. The resulting model is known as independent component analysis or ICA.

The reason the Gaussian distribution is disallowed as a source prior in ICA is that it does not permit unique recovery of the sources, as illustrated in Figure 12.20(c). This is because the PCA likelihood is invariant to any orthogonal transformation of the sources \mathbf{z}_t and mixing matrix \mathbf{W} . PCA can recover the best linear subspace in which the signals lie, but cannot uniquely recover the signals themselves.

To illustrate this, suppose we have two independent sources with uniform distributions, as shown in Figure 12.21(a). Now suppose we have the following mixing matrix

$$\mathbf{W} = \begin{pmatrix} 2 & 3 \\ 2 & 1 \end{pmatrix} \quad (12.98)$$

Then we observe the data shown in Figure 12.21(b) (assuming no noise). If we apply PCA followed by scaling to this, we get the result in Figure 12.21(c). This

corresponds to a whitening of the data. To uniquely recover the sources, we need to perform an additional rotation. The trouble is, there is no information in the symmetric Gaussian posterior to tell us which angle to rotate by. In a sense, PCA solves “half” of the problem, since it identifies the linear subspace; all that ICA has to do is then to identify the appropriate rotation. (Hence we see that ICA is not that different from methods such as varimax, which seek good rotations of the latent factors to enhance interpretability.)

Figure 12.21(d) shows that ICA can recover the source, up to a permutation of the indices and possible sign change. ICA requires that W is square and hence invertible. In the non-square case (e.g., where we have more sources than sensors), we cannot uniquely recover the true signal, but we can compute the posterior $p(z_t|x_t, W^\top)$, which represents our beliefs about the source. In both cases, we need to estimate W as well as the source distributions p_j . We discuss how to do this below.

下面先完整翻译原文（含图注与公式，不漏译），再给出通俗解释，最后总结要点。

逐字翻译

图 12.20：把 ICA 应用于一个 4 维源信号的 500 个独立同分布样本的示意。(a) 潜在信号。(b) 观测到的信号。(c) PCA 的估计。(d) ICA 的估计。图由 Aapo Hyvärinen 编写的 `icaDemo` 生成。

图 12.21：把 ICA 与 PCA 应用于一个**二维**、服从**均匀分布**的源信号的 100 个独立同分布样本的示意。(a) 潜在信号。(b) 观测值。(c) PCA 的估计。(d) ICA 的估计。图由 Aapo Hyvärinen 编写的 `icaDemoUniform` 生成。

12.6 独立成分分析 (ICA)

考虑如下情形：你在一个嘈杂的房间里，很多人同时说话。你的两只耳朵本质上就是两个麦克风，它们接收到的是房间里不同语音信号的**线性组合**。你的目标是把混合后的信号“解卷积”，分离出各自的成分。这被称为**鸡尾酒会问题**，是**盲信号分离** (BSS，或**盲源分离**) 的一个例子，其中“盲”表示我们对信号源几乎“一无所知”。除了明显的声学信号处理应用

外，这个问题在分析 EEG/MEG 脑电、金融数据，以及任何（不一定是时间序列的）**潜在源/因子被线性混合**的数据集时都会出现。

我们把问题形式化如下。设 $x_t \in \mathbb{R}^D$ 是在“时间” t 传感器处观测到的信号， $z_t \in \mathbb{R}^L$ 是源信号向量。假设

$$x_t = Wz_t + \varepsilon_t \quad (12.95)$$

其中 W 是 $D \times L$ 的矩阵， $\varepsilon_t \sim N(0, \Psi)$ 。本节把每个时刻看作**独立观测**，即不建模时间相关性（因此也可以把下标 t 换成 i ，但为了与 ICA 文献保持一致，我们继续用 t ）。目标是**推断源信号** $p(z_t | x_t, \theta)$ ，如图 12.20 所示。在这个语境下， W 称为**混合矩阵**。若 $L = D$ （源的个数 = 传感器个数），则 W 为方阵。为简单起见，常假定噪声 $|\Psi| = 0$ 。到目前为止，这个模型与**因子分析**是一样的（若无噪声也等价于 PCA，只是一般不要求 W 正交）。不过，我们会对 $p(z_t)$ 采用不同的先验：在 PCA 中，假设每个源独立且服从高斯分布

$$p(z_t) = \prod_{j=1}^L N(z_{tj} | 0, 1) \quad (12.96)$$

而现在我们放宽高斯假设，让源的分布是**任意的非高斯分布**

$$p(z_t) = \prod_{j=1}^L p_j(z_{tj}) \quad (12.97)$$

不失一般性，可把源的方差约束为 1，因为其它方差可以通过适当缩放 W 的行来表示。由此得到的模型称为**独立成分分析 (ICA)**。

在 ICA 中**不允许**把源先验设为高斯，原因是这会导致**无法唯一恢复源**，如图 12.20(c) 所示。这是因为 PCA 的似然对源 z_t 与混合矩阵 W 的任意**正交变换**不变。PCA 能恢复信号所处的“最佳线性子空间”，但**不能唯一恢复信号本身**。

为说明这一点，设有两个独立源，均为**均匀分布**，如图 12.21(a)。再设混合矩阵为

$$W = \begin{pmatrix} 2 & 3 \\ 2 & 1 \end{pmatrix} \quad (12.98)$$

在无噪声下，就得到图 12.21(b) 的观测数据。若对其应用 PCA（再配合缩放），得到图 12.21(c) 的结果，这等价于**白化数据**。要**唯一恢复源**，还需要额外做一次**旋转**。问题在于，**对称的高斯后验**并不会告诉我们应该旋到哪个角度。从某种意义上说，PCA 只解了“**一半**”的问题——它找到了线性子空间；剩下 ICA 要做的就是找**合适的旋转**。（因此 ICA 与如 varimax 之类试图“寻找更可解释旋转”的方法并不那么不同。）

图 12.21(d) 表明 ICA 可以恢复源（除了可能的重新排序与符号翻转这类不可辨识性）。标准 ICA 要求 W 是方阵并且可逆。非方情形（例如源比传感器多）下，无法唯一恢复真实信号，但可以计算后验 $p(z_t | x_t, \hat{W})$ ，表征我们对源的信念。无论哪种情形，都需要同时估计 W 以及各个源的分布 p_j 。下面将讨论如何做到这一点。

通俗解释

1) ICA 在做什么？

把观测到的“混合声”拆回若干“独立的原声”。数学上就是：观测 x 是若干独立源 z 的线性组合 $x = Wz$ （可能带少量噪声），我们要找分离矩阵把它们拆开。

2) 为什么 PCA 不行？

PCA 只关心**方差最大方向**，找到的是一个“好看的坐标系”。但只要源是**高斯**，无论怎么旋转，联合分布都还是高斯且对称，因此**没有依据**确定“对的角度”。所以 PCA 最多把数据“白化”（去相关、单位方差），仍然有一个**不确定的旋转**没有解决；这就是图 12.21(c) 的样子。

3) ICA 的关键假设

- **独立性**：各个源彼此独立（式 12.97 的乘积形式）。
- **非高斯性**：至少大多数源不是高斯的（“非高斯 + 独立”使得旋转能被唯一确定——这是 ICA 可辨识性的关键定理思路）。常见源分布如尖峰-厚尾（超高斯）、均匀、拉普拉斯等。
- **（通常）无噪声或小噪声**：很多算法默认噪声很小，以简化。

4) 图里的直观

- **图 12.20**：真实 4 个源（正弦、脉冲、锯齿、噪声） \rightarrow 线性混合后看不出原貌；
 - 用 PCA 分离：得到的分量仍相互混杂（看不出原始形状）；
 - 用 ICA：几乎把 4 个源（顺序、正负号可能不同）都找了回来。
- **图 12.21**：二维均匀源 \rightarrow 混合后形成一条“斜的椭圆云”；
 - **PCA+白化**：把云团“圆起来”但还没转到正确角度；
 - ICA：继续转一个角度，使两个分量再次独立，恢复原来的均匀方形云团（除了排列与符号）。

5) 关于方阵与非方阵

- **方阵（传感器=源）**：在非高斯独立假设下，可在尺度/排列/符号不确定性外唯一恢复源。

- **非方阵（源多或源少）**：无法一一对应地完全恢复；但能给出**后验分布**，表达“哪个源组合更可能”。

要点小结

- ICA 模型： $x = Wz + \varepsilon$, $\varepsilon \sim N(0, \Psi)$; 源独立且**非高斯**: $p(z) = \prod_j p_j(z_j)$ 。
- 与 PCA/FA 的区别：PCA 假设源是**高斯** → 只定子空间，**旋转不唯一**；ICA 用**非高斯性**来**固定旋转**，因此能（在尺度/符号/排列以外）恢复源。
- 直觉：PCA“白化”= 解了一半；ICA 再“找角度”= 解完。
- 可辨识性：通常要求 W 方阵可逆；非方阵时只能做**近似/后验分离**。
- 应用：语音分离（鸡尾酒会）、脑电/脑磁、金融时间序列、以及任何“独立非高斯源线性混合”的数据。

翻译并解释以下内容，翻译不能有遗漏，解释请通俗易懂且详尽，因为我的数学基础不太好，最后总结要点。

12.6.1 Maximum likelihood estimation

In this section, we discuss ways to estimate square mixing matrices W for the noise-free ICA model. As usual, we will assume that the observations have been centered; hence we can also assume z is zero-mean. In addition, we assume the observations have been whitened, which can be done with PCA. If the data is centered and whitened, we have $E xx^T = I$. But in the noise free case, we also have $\text{cov}[x] = E xx^T = WE zz^T W^T = WWT$ (12.99)

Hence we see that W must be orthogonal. This reduces the number of parameters we have to estimate from D^2 to $D(D - 1)/2$. It will also simplify the math and the algorithms. Let $V = W^{-1}$; these are often called the recognition weights, as opposed to W , which are the generative weights.⁴ Since $x = Wz$, we have, from Equation 2.89,

$$p(x|Wz) = p(z|Wz) | \det(Wz)| = p(z|Vx) | \det(V)| \quad (12.100)$$

Hence we can write the log-likelihood, assuming T iid samples, as follows:

$$\sum_{t=1}^T \log p(x_t|V) = \sum_{t=1}^T \log |\det(V)| + \sum_{t=1}^T \sum_{j=1}^D \log p_j(v_j^T z_t)$$

(12.101)

where v_j is the j 'th row of V . Since we are constraining V to be orthogonal, the first term is a constant, so we can drop it. We can also replace the average over the data with an expectation operator to get the following objective

$$NLL(V) = \sum_{j=1}^J E[G_j(z_j)] \quad (12.102)$$

where $z_j = v_j^T x$ and $G_j(z) = -\log p_j(z)$. We want to minimize this subject to the constraint that the rows of V are orthogonal. We also want them to be unit norm, since this ensures that the variance of the factors is unity (since, with whitened data, $E[v_j^T x] = \|v_j\|^2$), which is necessary to fix the scale of the weights. In otherwords, V should be an orthonormal matrix. It is straightforward to derive a gradient descent algorithm to fit this model; however, it is rather slow. One can also derive a faster algorithm that follows the natural gradient; see e.g., (MacKay 2003, ch 34) for details. A popular alternative is to use an approximate Newton method, which we discuss in Section 12.6.2. Another approach is to use EM, which we discuss in Section 12.6.3.

下面先把原文完整翻译（含公式，不漏译），再给出通俗解释，最后要点小结。

逐字翻译

12.6.1 极大似然估计

本节讨论在**无噪声** ICA 模型里，**对方阵**混合矩阵 W 的估计方法。像往常一样，我们假定观测已经**中心化**（减去均值）；因此也可以假定 z 的均值为 0。除此之外，我们还假定观测已经经过**白化**（可用 PCA 完成）。如果数据已中心化并白化，则有 $E[xx^T] = I$ 。而在无噪声情形，还成立

$$\text{cov}[x] = E[xx^T] = W E[zz^T] W^T = WW^T \quad (12.99)$$

因此可知 W 必须是**正交的**。这把待估参数个数从 D^2 降到了 $D(D - 1)/2$ 。它也会让数学与算法更简单。令 $V = W^{-1}$ ；它们常被称为**识别权重** (recognition weights)，与之相对， W 被称为**生成权重**。⁴ 因为 $x = Wz$ ，由式 (2.89) 得

$$p_x(Wz_t) = p_z(z_t) |\det(W^{-1})| = p_z(Vx_t) |\det(V)| \quad (12.100)$$

于是（假设有 T 个独立同分布样本）对数似然可写成

$$\frac{1}{T} \log p(\mathbf{D} | V) = \log |\det(V)| + \frac{1}{T} \sum_{j=1}^L \sum_{t=1}^T \log p_j(v_j^\top x_t) \quad (12.101)$$

其中 v_j 是 V 的第 j 行。由于我们把 V 约束为正交矩阵，第一项是常数，可忽略。我们也可把对数据的平均改写成期望算子，从而得到如下目标函数

$$\text{NLL}(V) = \sum_{j=1}^L \mathbb{E}[G_j(z_j)] \quad (12.102)$$

其中 $z_j = v_j^\top x$ ，且 $G_j(z) = -\log p_j(z)$ 。我们希望在“ V 的各行两两正交”的约束下**最小化**该目标。同时还希望各行是**单位范数**，因为这能保证因子的方差为 1（在白化数据下， $\mathbb{E}[v_j^\top x] = \|v_j\|^2$ ），这对于固定权重的尺度是必要的。换句话说， V 应当是一个**正交归一（正交且单位长度）**的矩阵。

为该模型推导一个**梯度下降**算法是直截了当的；然而它比较慢。也可以推导一个沿**自然梯度**前进的更快算法；细节见 (MacKay 2003, 第34章)。一种流行的替代方案是使用**近似牛顿法**，我们在第 12.6.2 节讨论。另一种方法是使用 EM，见第 12.6.3 节。

通俗解释

- **先做两件预处理**

1. **中心化**：把每个传感器的均值减掉，这样信号“围绕 0 摆动”。
2. **白化**：把各维度做线性变换，使得协方差变成单位阵 I 。这意味着各维之间不相关、每维方差为 1。

- **为什么白化后 W 必须正交？**

无噪声时 $x = Wz$ ，而 z 的协方差（在 ICA 里按约定）是 I 。于是 $\text{cov}(x) = W \text{cov}(z) W^\top = WW^\top$ 。
但白化又要求 $\text{cov}(x) = I$ 。因此 $WW^\top = I$ ，即 W 是**正交矩阵**（列向量两两正交且单位长度）。

好处：要估计的自由参数锐减（从 D^2 到 $D(D-1)/2$ ），计算更稳更快。

- **用“变量代换公式”写似然**

设 $V = W^{-1}$ （通常也叫“解混/分离矩阵”）：

由密度的变量代换 $p_x(x) = p_z(Vx) |\det(V)|$ 。

把每个源的密度写成 $p_j(\cdot)$ 的乘积，就得到式 (12.101) 的**平均对数似然**。

- **为什么能把 $\log |\det(V)|$ 扔掉？**

因为我们已把 V 约束为**正交**（正交矩阵的行列式为 ± 1 ），该项是**常数**，对最优化无影响。

- **目标函数的直观**

定义 $G_j(z) = -\log p_j(z)$ (即“负对数密度”)；我们最小化 $\sum_j \text{E}[G_j(v_j^\top x)]$ 。

这等价于“让每个投影 $v_j^\top x$ 更像它应当来自的非高斯源分布 p_j ”。

同时加上**正交** + **单位长度约束**，保证源的方差被规范化为 1 (避免“谁变大谁变小”的尺度不确定性)。

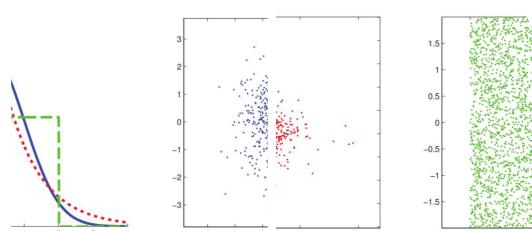
- **怎么求解？**

- **朴素的梯度下降**可行但慢。
 - **自然梯度** (考虑了矩阵在流形上的几何) 更快，是经典 ICA 算法 (如 Amari) 采用的思想。
 - **近似牛顿法** (下一节) 是实践中常用的快速独立成分分析 (FastICA) 的思想。
 - 也可以用**EM** (再下一节)，把“非高斯”通过辅助变量或近似来处理。
-

要点小结

- 在**无噪声 + 白化**条件下， $\text{cov}(x) = I = WW^\top \Rightarrow W$ **正交**；自由度从 D^2 降到 $D(D-1)/2$ 。
- 令 $V = W^{-1}$ (分离矩阵)，用变量代换可得平均对数似然：

$$\frac{1}{T} \log p(\mathbf{D}|V) = \log |\det V| + \frac{1}{T} \sum_{j,t} \log p_j(v_j^\top x_t)$$
 正交约束下第一项为常数，可忽略。
- **最小化**
 $\text{NLL}(V) = \sum_j \text{E}[-\log p_j(v_j^\top x)]$ ，并约束 V **正交归一** (行向量两两正交且单位范数)。
- 求解策略：**普通梯度下降** (慢) → **自然梯度** (更快) → **近似牛顿/FastICA** (12.6.2) → **EM** (12.6.3)。
- 这些步骤的本质：在白化后的空间里，寻找一个**旋转** V ，使每个投影都尽可能符合其假定的**非高斯独立**源分布。



翻译并解释以下内容，翻译不能有遗漏，解释请通俗易懂且详尽，因为我的数学基础不太好，最后总结要点。

Figure 12.22 Illustration of Gaussian, sub-Gaussian (uniform) and super-Gaussian (Laplace) distributions in 1d and 2d. Figure generated by subSuperGaussPlot, written by Kevin Swersky.

12.6.2 The FastICA algorithm

We now describe the fast ICA algorithm, based on (Hyvärinen and Oja 2000), which we will show is an approximate Newton method for fitting ICA models. For simplicity of presentation, we initially assume there is only one latent factor. In addition, we initially assume all source distributions are known and are the same, so we can just write $G(z) = -\log p(z)$. Let $g(z) = d dz G(z)$. The constrained objective, and its gradient and Hessian, are given by

$$f(v) = E G(v^T x) + \lambda(1 - v^T v) \quad (12.103)$$

$$\nabla f(v) = E x g(v^T x) - \beta v \quad (12.104)$$

$$H(v) = E x x^T g(v^T x) - \beta I \quad (12.105)$$

where $\beta = 2\lambda$ is a Lagrange multiplier. Let us make the approximation

$$E x x^T g(v^T x) \approx E x x^T E g(v^T x) = E g(v^T x) \quad (12.106)$$

This makes the Hessian very easy to invert, giving rise to the following Newton update:

$$v^* = v - E x g(v^T x) - \beta v [E g(v^T x)] - \beta \quad (12.107)$$

One can rewrite this in the following way

$$v^* = E x g(v^T x) - E g(v^T x) v \quad (12.108)$$

(In practice, the expectations can be replaced by Monte Carlo estimates from the training set, which gives an efficient online learning algorithm.) After performing this update, one should project back onto the constraint surface using

$$v_{\text{new}} = v^* / \|v^*\| \quad (12.109)$$

One iterates this algorithm until convergence. (Due to the sign ambiguity of v , the values of v may not converge, but the direction defined by this vector should converge, so one can assess convergence by monitoring $|v^T v_{\text{new}}|$, which should approach 1.) Since the objective is not convex, there are multiple local optima. We can use this fact to learn multiple different weight vectors or features. We can either learn the features sequentially and then project out the part of v_j that lies in the subspace defined

by earlier features, or we can learn them in parallel, and orthogonalize V in parallel. This latter approach is usually preferred, since, unlike PCA, the features are not ordered in any way. So the first feature is not “more important” than the second, and hence it is better to treat them symmetrically.

12.6.2.1 Modeling the source densities

So far, we have assumed that $G(z) = -\log p(z)$ is known. What kinds of models might be reasonable as signal priors? We know that using Gaussians (which correspond to quadratic functions for G) won’t work. So we want some kind of non-Gaussian distribution. In general, there are several kinds of non-Gaussian distributions, such as the following:

- Super-Gaussian distributions These are distributions which have a big spike at the mean, and hence (in order to ensure unit variance) have heavy tails. The Laplace distribution is a classic example. See Figure 12.22.

Formally, we say a distribution is super-Gaussian or leptokurtic (“lepto” coming from the Greek for “thin”) if $kurt(z) > 0$, where $kurt(z)$ is the kurtosis of the distribution, defined by

$$kurt(z) = \mu_4 \sigma^4 - 3 \quad (12.110)$$

where σ is the standard deviation, and μ_k is the k'th central moment, or moment about the mean:

$$\mu_k = E(X - E[X])^k \quad (12.111)$$

(So $\mu_1 = \mu$ is the mean, and $\mu_2 = \sigma^2$ is the variance.) It is conventional to subtract 3 in the definition of kurtosis to make the kurtosis of a Gaussian variable equal to zero.

- Sub-Gaussian distributions A sub-Gaussian or platykurtic (“platy” coming from the Greek for “broad”) distribution has negative kurtosis. These are distributions which are much flatter than a Gaussian. The uniform distribution is a classic example. See Figure 12.22.
- Skewed distributions Another way to “be non-Gaussian” is to be asymmetric. One measure of this is skewness, defined by

$$skew(z) = \mu_3 \sigma^3 \quad (12.112)$$

An example of a (right) skewed distribution is the gamma distribution (see Figure 2.9). When one looks at the empirical distribution of many natural signals, such as images and speech, when passed through certain linear filters, they tend to be very super-Gaussian. This result

holds both for the kind of linear filters found in certain parts of the brain, such as the simple cells found in the primary visual cortex, as well as for the kinds of linear filters used in signal processing, such as wavelet transforms. One obvious choice for modeling natural signals with ICA is therefore the Laplace distribution. For mean zero and variance 1, this has a log pdf given by

$$\log p(z) = -\sqrt{2|z|} - \log(\sqrt{2}) \quad (12.113)$$

Since the Laplace prior is not differentiable at the origin, it is more common to use other, smoother super-Gaussian distributions. One example is the logistic distribution. The corresponding log pdf, for the case where the mean is zero and the variance is 1 (so $\mu = 0$ and $s = \sqrt{3\pi}$), is given by the following:

$$\log p(z) = -2 \log \cosh(\pi/2 \sqrt{3} z) - \log(4\sqrt{3\pi}) \quad (12.114)$$

Various ways of estimating $G(z) = -\log p(z)$ are discussed in the seminal paper (Pham and Garrat 1997). However, when fitting ICA by maximum likelihood, it is not critical that the exact shape of the source distribution be known (although it is important to know whether it is sub or super Gaussian). Consequently, it is common to just use $G(z) = \sqrt{|z|}$ or $G(z) = \log \cosh(z)$ instead of the more complex expressions above.

下面先**完整翻译**原文（含图注与公式，不漏译），再给出**通俗解释**，最后**总结要点**。

逐字翻译

图 12.22 在 1 维与 2 维中演示了高斯 (Gaussian)、亚高斯/次高斯 (uniform, 均匀分布) 与超高斯 (Laplace, 拉普拉斯) 分布。图由 Kevin Swersky 编写的 `subSuperGaussPlot` 生成。

12.6.2 FastICA 算法

下面我们描述基于 (Hyvärinen and Oja 2000) 的快速 ICA (FastICA) 算法，我们将展示它是拟合 ICA 模型的一种**近似牛顿法**。为便于讲解，先假设只有一个潜因子。此外，先假设**所有源分布已知且相同**，因此我们可直接写 $G(z) = -\log p(z)$ 。令 $g(z) = \frac{d}{dz} G(z)$ 。带约束的目标函数及其梯度、海森矩阵为

$$f(v) = E[G(v^T x)] + \lambda(1 - v^T v) \quad (12.103)$$

$$\nabla f(v) = E[x g(v^T x)] - \beta v \quad (12.104)$$

$$H(v) = E [xx^\top g'(v^\top x)] - \beta I \quad (12.105)$$

其中 $\beta = 2\lambda$ 是拉格朗日乘子。做如下近似

$$E [xx^\top g'(v^\top x)] \approx E[xx^\top] E[g'(v^\top x)] = E[g'(v^\top x)] \quad (12.106)$$

这使得海森矩阵很容易求逆，从而得到如下牛顿更新：

$$v^{*} = v - \frac{E[xg(v^\top x)] - \beta v}{E[g'(v^\top x)] - \beta} \quad (12.107)$$

可将其改写为

$$v^{*} = E[xg(v^\top x)] - E[g'(v^\top x)] v \quad (12.108)$$

(实际中，可用训练集的蒙特卡罗平均来近似这些期望，从而得到高效的**在线学习算法**。) 做完这一步后，应当用

$$v_{\text{new}} = \frac{v^{*}}{\|v^{*}\|} \quad (12.109)$$

把向量投回到约束曲面上。重复上述迭代直至收敛。（由于 v 存在**符号不确定性**，向量的数值可能不收敛，但其方向应当收敛，因此可用 $|v^\top v_{\text{new}}|$ 接近 1 来判断收敛。）因为目标函数**非凸**，会有多个局部最优。我们可以利用这一点去学习多个不同的权向量或“特征”。要么**顺序**学习，再把第 j 个向量在先前特征所张成的子空间内的部分**投影出去**；要么**并行**学习，并在并行过程中对 V 做**正交化**。通常更偏向后者，因为与 PCA 不同，ICA 的特征没有固定顺序，第一条并不比第二条“更重要”，因此对它们**对称对待更好**。

12.6.2.1 源密度的建模

到目前为止我们假设 $G(z) = -\log p(z)$ 是已知的。那么作为信号先验，哪些模型是合理的呢？我们知道使用高斯（对应 G 为二次函数）是行不通的，所以需要某种**非高斯分布**。通常有以下几类：

- **超高斯 (Super-Gaussian) 分布**

这类分布在均值处有一个很大的尖峰，因此（在保持单位方差的同时）具有**重尾**。拉普拉斯分布是经典例子，见图 12.22。形式上，若

$$\text{kurt}(z) = \frac{\mu_4}{\sigma^4} - 3 > 0 \quad (12.110)$$

则称为**超高斯或尖峰厚尾/高峰度 (leptokurtic)** 分布，其中 σ 是标准差， μ_k 是第 k 阶中心矩（关于均值的矩）：

$$\mu_k = E[(X - E[X])^k] \quad (12.111)$$

(因此 $\mu_1 = \mu$ 是均值, $\mu_2 = \sigma^2$ 是方差。) 定义峰度时约定减去 3, 使高斯变量的峰度为 0。

- 次高斯/亚高斯 (Sub-Gaussian) 分布

次高斯或低峰度 (platykurtic) 分布的峰度为负, 它们比高斯更平坦。均匀分布是经典例子, 见图 12.22。

- 偏斜 (Skewed) 分布

另一种“非高斯”的方式是不对称。其度量之一是偏度:

$$\text{skew}(z) = \frac{\mu_3}{\sigma^3} \quad (12.112)$$

一个 (右) 偏分布的例子是Gamma 分布 (见图 2.9)。

对许多自然信号 (如图像与语音) 做某些线性滤波后得到的经验分布, 往往是强超高斯的。这对来自大脑某些区域 (如初级视皮层简单细胞) 的线性滤波器以及信号处理中常用的滤波器 (如小波变换) 都成立。因此, 用 ICA 建模自然信号, 一个显而易见的选择是拉普拉斯分布。在均值为 0、方差为 1 的情况下, 其对数密度为

$$\log p(z) = -\sqrt{2} |z| - \log(\sqrt{2}) \quad (12.113)$$

由于拉普拉斯先验在原点不可导, 更常见的是使用其他更光滑的超高斯分布。例如逻辑斯蒂 (logistic) 分布。当均值为 0、方差为 1 (因此 $\mu = 0, s = \sqrt{3}/\pi$) 时, 其对数密度为

$$\log p(z) = -2 \log \cosh\left(\frac{\pi}{2\sqrt{3}} z\right) - \log\left(\frac{4\sqrt{3}}{\pi}\right) \quad (12.114)$$

关于估计 $G(Z) = -\log p(z)$ 的各种方法, 可见奠基性论文 (Pham and Garrat 1997)。不过, 在用极大似然拟合 ICA 时, 源分布的精确形状并不关键 (虽然知道它是次高斯还是超高斯很重要)。因此, 实践中常直接采用更简单的 $G(z) = \sqrt{z}$ 或 $G(z) = \log \cosh(z)$, 而不是上面更复杂的表达式。

通俗解释

FastICA 的核心思想

- 在白化后的空间里, 我们要找一个单位向量 v , 使得投影 $z = v^\top x$ 的分布尽量非高斯 (与假定的先验 $p(z)$ 更匹配)。
- 通过把目标写成 $f(v) = E G(v^\top x)$ 并加上 “ $\|v\| = 1$ ” 的约束, 用近似牛顿法做迭代更新。
- 关键近似: 白化后 $E[xx^\top] = I$, 把 $E[xx^\top g'(v^\top x)]$ 近似为 $E[xx^\top]E[g'(v^\top x)] = E[g'(v^\top x)]$, 海森矩阵就很容易“反掉”, 得到封闭形式的更新式

(12.108)。

- 每次更新后做**归一化** (式 12.109)，保证 $\|\nu\| = 1$ 。

多个成分怎么学？

- 顺序法：**学一个 ν 就把它在后续里“投影出去”（去相关）。
- 并行法：**同时学多个向量，并在每次迭代后把矩阵 V 做**正交化**。

实用上更推荐并行法，因为 ICA 的分量没有“主次之分”，应对称处理。

源分布怎么选？

- 只要非高斯即可：**
 - 超高斯（尖峰厚尾，如拉普拉斯、逻辑斯蒂）；
 - 次高斯（很平，如均匀）；
 - 或者偏斜（如 Gamma）。
 - 经验上，自然信号投影后通常是**超高斯**，所以常用 $G(z) = \log \cosh(z)$ 等光滑的“对比函数”。
-

要点小结

- FastICA = 近似牛顿法：**在白化空间中最小化 $E G(\nu^\top x)$ 并保持 $\|\nu\| = 1$ ，更新式为 $\nu^{*\top} = E[x g(\nu^\top x)] - E[g'(\nu^\top x)] \nu$ ，随后**归一化**。
- 收敛判断：**用 $|\nu^\top \nu_{\text{new}}| \rightarrow 1$ ；目标非凸，存在多解。
- 学习多个成分：**顺序+投影消去，或并行+正交化（常用）。
- 非高斯先验：**
 - 超高斯**（峰度 > 0 ，如 Laplace、logistic）；
 - 次高斯**（峰度 < 0 ，如 uniform）；
 - 偏斜**（非对称，偏度 μ_3/σ^3 ）。

实务中不必精确知道形状，常用 $G(z) = \log \cosh(z)$ 等简单对比函数即可。
- 图 12.22 直观：**高斯呈“鼓肚”圆云；拉普拉斯有尖峰重尾；均匀分布是“扁平方砖”。这些形状差异正是 ICA 能区分并“对准”源方向的依据。

ICA可以理解为把观测到的混合声拆回若干独立的原声，那和“ICA的分量没有主次之分，应对称处理”有什么关联，我该如何理解

已思考 16s

很好这点抓得准：ICA = 把混合信号拆回若干“独立的原声”。

那“ICA 的分量没有主次之分，应对称处理”指的就是：在标准 ICA 里，这些被分离出来的“原声”没有天然的先后、重要性或排名。原因有三：

为什么没有主次顺序？

1. 目标函数是完全对称的

以 FastICA 为例，优化的是

$$\sum_{j=1}^L E[G(v_j^\top x)].$$

这是对每个分量的对称求和；把任意两个分量 j 和 k 对调，目标值不变。

⇒ 数学上没有“第1个”比“第2个”更重要的依据。

2. 模型具有置换与符号不变性

若把分离矩阵 V 的行（或 W 的列）做任意置换，并对任一行乘 ± 1 ，得到的分量集合在似然/独立性上完全等价。

⇒ 只能确定“是哪几个源”，不能确定“它们的顺序和正负号”。

3. 与 PCA 的对比

PCA 的成分按方差大小（特征值）天然排序，第一主成分解释的方差最多；

但 ICA 的准则是非高斯性/独立性，没有“解释方差最多”的排序标准。

这跟算法实践有什么关系？

- 并行（对称）学习更合适

FastICA 常用“对称式”更新：同时更新所有 v_j ，再做一次正交化。因为分量没有主次，这样的处理与目标一致，也不依赖初始化的先后。

- 顺序（deflation）学习也能用，但“谁先谁后”只是人为选择，不同初始化/顺序可能得到不同排列或把误差不均地留给后面的分量；这不是分量真的“更重要”，只是算法路径差异。

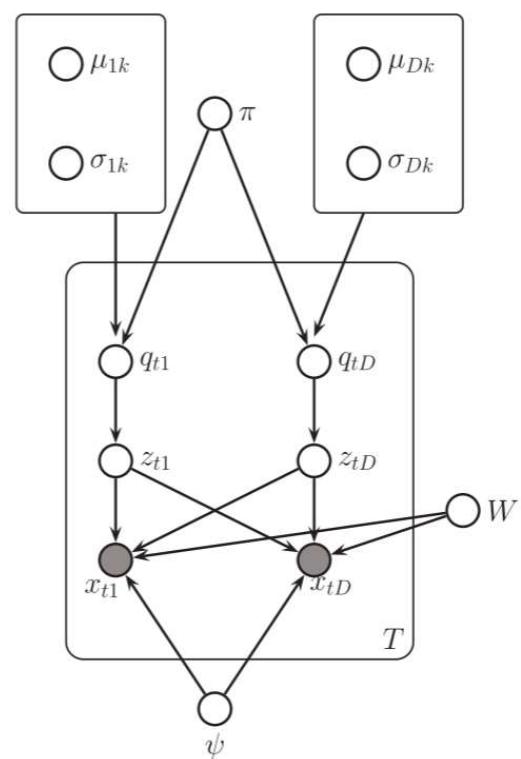
什么时候会出现“排序”？

只有当你额外引入标准时才会有：

比如按方差（能量）、峰度 ($|\text{kurtosis}|$)、与某个监督信号的相关性（“监督式 ICA/PLS/CCA”）、信噪比等外部准则对分量事后排序。这些排序不是 ICA 自带的，而是任务需求决定的。

一句话总结

- 纯粹的 ICA 只关心把源**独立地**分离出来；其数学与几何结构对各分量**完全对称**，并且**对置换/符号不变**，因此**没有天然的主次顺序**。
- 实作上就用**并行更新 + 正交化**的对称式 FastICA；若用顺序法，要知道“谁先被找出”不代表它“更重要”。
- 需要排序时，请**自定义标准**（方差、峰度、与目标的相关性等）再对分量进行排名。



翻译并解释以下内容，翻译不能有遗漏，解释请通俗易懂且详尽，因为我的数学基础不太好，最后总结要点。

Figure 12.23 Modeling the source distributions using a mixture of univariate Gaussians (the independent factor analysis model of (Moulines et al. 1997; Attias 1999)).

12.6.3 Using EM

An alternative to assuming a particular form for $G(z)$, or equivalently for $p(z)$, is to use a flexible non-parametric density estimator, such as a mixture of (uni-variate) Gaussians:

$$p(q_j = k) = \pi_k \quad (12.115)$$

$$p(z_j | q_j = k) = N(\mu_j, k, \sigma^2_j, k) \quad (12.116)$$

$$p(x|z) = N(Wz, \Psi) \quad (12.117)$$

This approach was proposed in (Moulines et al. 1997; Attias 1999), and the corresponding graphical model is shown in Figure 12.23. It is possible to derive an exact EM algorithm for this model. The key observation is that it is possible to compute $E[z_t|x_t, \theta]$ exactly by summing over all KL combinations of the q_t variables, where K is the number of mixture components per source. (If this is too expensive, one can use a variational mean field approximation (Attias 1999).) We can then estimate all the source distributions in parallel by fitting a standard GMM to $E[z_t]$. When the source GMMs are known, we can compute the marginals $p_j(z_j)$ very easily, using

$$p_j(z_j) = \sum_{k=1}^K \pi_{j,k} N(z_j | \mu_{j,k}, \sigma_{j,k}^2) \quad (12.118)$$

Given the p_j 's, we can then use an ICA algorithm to estimate W . Of course, these steps should be interleaved. The details can be found in (Attias 1999).

下面先把原文完整翻译（含公式与图注），再给出通俗解释，最后要点小结。

逐字翻译

图 12.23 用一元高斯混合来建模源分布 (Moulines 等, 1997; Attias, 1999 的独立因子分析模型)。

12.6.3 使用 EM

与其为 $G(z)$ (等价于 $p(z)$) 预先假定一个具体形式，不如使用一个灵活的**非参数**密度估计器，例如** (一维) 高斯混合**：

$$p(q_j = k) = \pi_k \quad (12.115)$$

$$p(z_j | q_j = k) = N(\mu_{j,k}, \sigma_{j,k}^2) \quad (12.116)$$

$$p(x | z) = N(Wz, \Psi) \quad (12.117)$$

这种做法由 (Moulines et al., 1997; Attias, 1999) 提出，相应的图模型见图 12.23。可以为该模型推导出**精确的 EM 算法**。关键观察是：我们可以通过对所有 q_t 变量的 K^L 种组合求和，精确地计算 $E[z_t | x_t, \theta]$ ，其中 K 是每个源的混合分量数。（如果这太贵，可以使用**变分均值场近似** (Attias 1999)。）之后，我们就能**并行**地估计所有源的分布：对 $E[z_t]$ 拟合一个标准的 GMM。

当各源的 GMM 已知时，可以用

$$p_j(z_j) = \sum_{k=1}^K \pi_{j,k} N(z_j | \mu_{j,k}, \sigma_{j,k}^2) \quad (12.118)$$

很容易地得到各边缘分布 $p_j(z_j)$ 。给定这些 p_j ，接着用某个 ICA 算法来估计 W 。当然，上述步骤应当**交替进行**。细节见 (Attias 1999)。

通俗解释

这节讲的是：**不要为源的形状 $p(z)$ 先拍脑袋定一个固定分布**（比如 Laplace 或 logistic），而是让数据自己来告诉我们它长什么样。做法是把**每个独立源 z_j** 的分布都用**一维高斯混合 (GMM) **来拟合——这非常灵活，既能表示尖峰厚尾（超高斯）、也能表示扁平（次高斯），甚至能表示多峰或偏斜。

- 变量含义：

- q_j 是源 z_j 选择了哪个高斯分量 (1...K) 的**指示变量**；
- π_k 是分量权重； $\mu_{j,k}, \sigma_{j,k}^2$ 是第 j 个源在第 k 个分量的均值与方差；
- 观测模型仍是线性高斯： $x | z \sim N(Wz, \Psi)$ 。

- **为什么能用 EM？**

EM 的 E 步需要算后验的**期望** $E[z_t | x_t]$ 。给定混合分配 $q_t = (q_{t1}, \dots, q_{tL})$ ， z_t 与 x_t 都是高斯，计算容易；但 q_t 有 K^L 种组合，所以精确求和是**指数级**，只适合 L 或 K 很小的情况。大一点时，就用**变分均值场**把 $p(q_t | x_t)$ 近似为可分的 $\prod_j r_{tjk}$ 。

- **M 步怎么做？**

1. 有了 $E[z_t]$ 及与分量相关的期望，就像在普通 GMM 里那样，更新 $\pi_{j,k}, \mu_{j,k}, \sigma_{j,k}^2$ ——等价于“对估计出的源样本做 GMM 拟合”。
2. 有了各源的边缘 $p_j(z_j)$ (式 12.118)，我们就知道了 ICA 里的“源先验”。在这个先验下，再去更新/估计混合矩阵 W (比如用 FastICA 或其它最大似然法)。
3. 重复 1)-2) 直到收敛。

这就是所谓的 **Independent Factor Analysis (独立因子分析, IFA)**：把“独立的源 + 每个源是高斯混合”与“线性混合”结合起来。

- **直观理解：**

你可以把它想成三步反复做：

- (分离) **先根据当前的 W 把每条观测“解混”出对源的估计；
- (学形状) **把这些估计当作“源样本”，给每个源单独用 GMM 学它真实的形状；
- (再分离) **有了更准的源形状，再去更新分离矩阵 W ；循环往复，互相提升。

- **优缺点：**

- 优点：源分布更灵活（可多峰/重尾/偏斜），分离通常更好；统一在 EM 框架里。
- 缺点：精确 E 步需要对 K^L 个状态求和，可能很贵；需要变分近似；仍有局部最优。

要点小结

- 本节提出用**一维高斯混合**来刻画每个独立源的分布： $q_j \sim \text{Cat}(\pi)$, $z_j | q_j \sim N(\mu_{j,k}, \sigma_{j,k}^2)$, 观测 $x | z \sim N(Wz, \Psi)$ 。
- 这是 **Independent Factor Analysis (IFA)**；可推导出基于 EM 的学习：
 - **E 步：**计算 $E[z_t | x_t]$ （精确枚举 K^L 或用**变分均值场**）。
 - **M 步：**
 1. 对 $E[z_t]$ 拟合每个源的 GMM，更新 $\pi_{j,k}, \mu_{j,k}, \sigma_{j,k}^2$ ；
 2. 给定新的 $p_j(z_j)$ ，用 ICA 方法更新 W 。
- 这两步**交替**进行直到收敛；优点是先验灵活、能拟合复杂源形状；代价是计算量可能较大，需要近似。

翻译并解释以下内容，翻译不能有遗漏，解释请通俗易懂且详尽，因为我的数学基础不太好，最后总结要点。

12.6.4 Other estimation principles

It is quite common to estimate the parameters of ICA models using methods that seem different to maximum likelihood. We will review some of these methods below, because they give additional insight into ICA. However, we will also see that these methods in fact are equivalent to maximum likelihood after all. Our presentation is based on (Hyvärinen and Oja 2000).

12.6.4.1 Maximizing non-Gaussianity

An early approach to ICA was to find a matrix V such that the distribution $z = Vx$ is as far from Gaussian as possible. (There is a related approach in statistics called projection pursuit.) One measure of non-Gaussianity is kurtosis, but this can be sensitive to outliers. Another measure is the negentropy, defined as

$$\text{negentropy}(z) = H(N(\mu, \sigma^2) - H(z)) \quad (12.119)$$

where $\mu = E[z]$ and $\sigma^2 = \text{var}[z]$. Since the Gaussian is the maximum entropy distribution, this measure is always non-negative and becomes large for distributions that are highly nonGaussian. We can define our objective as maximizing

$$J(V) = j \text{ negentropy}(z) = j H(N(\mu_j, \sigma^2_j) - H(z)) \quad (12.120)$$

where $z = Vx$. If we fix V to be orthogonal, and if we whiten the data, the covariance of z will be I independently of V , so the first term is a constant. Hence

$$J(V) = j - H(z) + \text{const} = j E[\log p(z)] + \text{const} \quad (12.121)$$

which we see is equal (up to a sign change, and irrelevant constants) to the log-likelihood in Equation 12.102.

12.6.4.2 Minimizing mutual information

One measure of dependence of a set of random variables is the multi-information:

$$I(z) = KL(p(z) || \prod_j p(z_j)) = j H(z) - H(z) \quad (12.122)$$

We would like to minimize this, since we are trying to find independent components. Put another way, we want the best possible factored approximation to the joint distribution. Now since $z = Vx$, we have

$$I(z) = j H(z) - H(Vx) \quad (12.123)$$

If we constrain V to be orthogonal, we can drop the last term, since then $H(Vx) = H(x)$ (since multiplying by V does not change the shape of the distribution), and $H(x)$ is a constant which is solely determined by the empirical distribution. Hence we have $I(z) = j H(z)$. Minimizing this is equivalent to maximizing the negentropy, which is equivalent to maximum likelihood.

12.6.4.3 Maximizing mutual information (infomax)

Instead of trying to minimize the mutual information between the components of z , let us imagine a neural network where x is the input and $y_j = \phi(v^T j x) + \epsilon$ is the noisy output, where ϕ is some nonlinear scalar function, and $\epsilon \sim N(0, 1)$. It seems reasonable to try to maximize the information flow through this system, a principle known as infomax. (Bell and Sejnowski 1995). That is, we want to maximize the mutual information between y (the internal neural representation) and x (the observed input signal).

We have $I(x; y) = H(y) - H(y|x)$, where the latter term is constant if we assume the noise has constant variance.

One can show that we can approximate the former term as follows

$$H(y) = \sum_j \log \varphi(v^T_j x) + \log |\det(V)| \quad (12.124)$$

where, as usual, we can drop the last term if V is orthogonal. If we define $\varphi(z)$ to be a cdf, then $\varphi(z)$ is its pdf, and the above expression is equivalent to the log likelihood. In particular, if we use a logistic nonlinearity, $\varphi(z) = \text{sigm}(z)$, then the corresponding pdf is the logistic distribution, and $\log \varphi(z) = \log \cosh(z)$ (ignoring irrelevant constants). Thus we see that infomax is equivalent to maximum likelihood.

下面先把原文完整翻译（含小节标题与公式），再给出通俗解释，最后要点小结。

逐字翻译

12.6.4 其他估计原理

用一些看起来不同于极大似然 (maximum likelihood, ML) 的方法来估计 ICA 模型的参数是很常见的。我们在下面回顾其中若干方法，因为它们能给 ICA 额外的直觉。然而我们也会看到，这些方法事实上最终都等价于极大似然。本节的讲述基于 (Hyvärinen and Oja, 2000)。

12.6.4.1 最大化非高斯性

一种早期的 ICA 做法，是寻找一个矩阵 V ，使得 $z = Vx$ 的分布尽可能偏离高斯。（统计里有个相关方法叫投影寻踪 projection pursuit。）衡量“非高斯性”的一个指标是峰度 (kurtosis)，但它对离群点很敏感。另一个指标是负熵 (negentropy)，定义为

$$\text{negentropy}(z) = H(N(\mu, \sigma^2)) - H(z) \quad (12.119)$$

其中 $\mu = E[z]$, $\sigma^2 = \text{var}[z]$ 。由于高斯在给定均值与方差时熵最大，该指标总是非负的，且对高度非高斯的分布会更大。于是目标可以定义为最大化

$$J(V) = \sum_j \text{negentropy}(z_j) = \sum_j H(N(\mu_j, \sigma_j^2)) - H(z_j), \quad z = Vx \quad (12.120)$$

如果把 V 约束为正交，并把数据白化，那么 z 的协方差不论 V 取何值都为 I ，因此上式第一项是常数。于是

$$J(V) = \sum_j -H(z_j) + \text{const} = \sum_j E[\log p(z_j)] + \text{const} \quad (12.121)$$

这与式 (12.102) 的对数似然仅差一个符号与无关常数，因此是等价的。

12.6.4.2 最小化互信息

衡量一组随机变量依赖性的一个指标是**多信息** (multi-information) :

$$I(z) = \text{KL}(p(z) \parallel \prod_j p(z_j)) = \sum_j H(z_j) - H(z) \quad (12.122)$$

我们希望它越小越好，因为我们要找**独立**成分。换言之，我们希望对联合分布最好的“乘积近似”。

又因为 $z = Vx$,

$$I(z) = \sum_j H(z_j) - H(Vx). \quad (12.123)$$

若把 V 约束为正交，则 $H(Vx) = H(x)$ (乘以正交矩阵不改变分布“形状”)，而 $H(x)$ 由数据决定，是常数。于是 $I(z) = \sum_j H(z_j)$ 。**最小化**它就等价于**最大化负熵**，而这又等价于极大似然。

12.6.4.3 最大化互信息 (infomax)

与其最小化 z 的分量之间的互信息，不如设想一个**神经网络**：输入为 x ，输出为 $y_j = \phi(v_j^\top x) + \epsilon$ ，其中 ϕ 是某个非线性标量函数， $\epsilon \sim N(0, 1)$ 。一个合理原则是**最大化信息流** (infomax; Bell & Sejnowski, 1995)，即最大化内部表示 y 与输入 x 之间的互信息。有 $I(x; y) = H(y) - H(y|x)$ ，若噪声方差固定，则后者是常数。可证明

$$H(y) = \sum_{j=1}^L E[\log \phi'(v_j^\top x)] + \log |\det(V)| \quad (12.124)$$

按惯例若 V 正交，最后一项可略。若把 $\phi(z)$ 取为某个**累积分布函数 (cdf)**，则 $\phi'(z)$ 就是其**概率密度函数 (pdf)**，上式就与对数似然等价。特别地，若用**Logistic 非线性** $\phi(z) = \text{sigm}(z)$ ，其对应 pdf 为 logistic 分布，并且（忽略常数） $\log \phi'(z) = \log \cosh(z)$ 。因此可见 infomax 与极大似然等价。

通俗解释

- **三条路，一条心：**

1. “让投影尽量非高斯”（最大负熵）；
2. “让各分量尽量独立”（最小多信息 = 各分量熵之和最小）；
3. “让神经网络输出携带的信息最多”（infomax）。

这三种口号看似不同，但在**白化 + 正交**的常见设置下，它们都变成了**同一个目标**：最大化 $\sum_j E[\log p(z_j)]$ ，也就是**极大似然**地去拟合“源先验”。

- **为什么会等价？**

- 高斯在同均值方差下熵最大 \Rightarrow “远离高斯” = “增大负熵” = “提高 $E[\log p(z_j)]$ ”。
 - 正交变换不改变熵（对白化数据），所以互信息只剩各分量熵之和；让它最小就是让每个分量尽量“尖/非高斯”，与上面等价。
 - infomax 中若把神经元的非线性选成某个分布的 cdf，其导数就是 pdf；**最大输出熵就转化成了最大似然。**
 - **工程上的启示：**
你可以选择自己更直观/更好实现的准则（负熵、互信息或 infomax），本质上都在做 ML。FastICA 用的 $\log \cosh$ 等“对比函数”正是这么来的。
-

要点小结

- 许多“看起来不同”的 ICA 学习准则（**最大非高斯性、最小互信息、infomax**）在白化与正交约束下都**等价于极大似然**。
- **负熵**: $\text{negentropy}(z) = H(\text{Gaussian with same } \mu, \sigma^2) - H(z) \geq 0$ 。最大化负熵 \Leftrightarrow 最大化 $\sum_j E[\log p(z_j)]$ 。
- **多信息**: $I(z) = \sum_j H(z_j) - H(z)$ 。正交下 $H(z) = H(x)$ 为常数 \Rightarrow 最小化 $I \Leftrightarrow$ 最大化负熵 \Leftrightarrow ML。
- **infomax**: 最大化 $I(x; y) = H(y) - H(y|x)$ 。固定噪声下 $H(y|x)$ 常数，若 ϕ 是某分布的 cdf，则 $H(y)$ 的表达与对数似然相同 \Rightarrow **infomax = ML**。
- 实践中任选其一实现即可，理解这一“等价性”能帮助你在不同文献与算法之间自由切换。

基于本对话里我贴过的整章原文，输出：①知识点总结
(分点+关键词+用途) ②子章节关联 (依赖/对比/递进+关系句，附Mermaid可选) ③本章总结。

已思考 24s

下面基于你贴过的整章内容，给出三个部分的整理：①知识点总结 ②子章节关联 ③本章总结。

① 知识点总结 (分点 + 关键词 + 用途)

1. 因子分析 (FA)

- 关键词: $x = Wz + \mu + \varepsilon$, $\Psi = \text{diag}$, $p(z) = \mathcal{N}(0, I)$, 协方差 $C = WW^\top + \Psi$ (低秩分解), 后验 $p(z|x) = \mathcal{N}(m, \Sigma)$ 。
- 用途: 用少量潜因子重建/建模高维实值数据的相关性; 降维、密度估计、缺失值推断。

2. 不可辨识性 (旋转不变)

- 关键词: $W \rightarrow WR$ (R 为正交矩阵) 似然不变; 上界 L_{\max} ; 约束/稀疏化/旋转 (varimax) /非高斯先验 (通向 ICA)。
- 用途: 理解为何载荷矩阵与因子解释非唯一, 并通过正交化、下三角、稀疏先验等提升可解释性。

3. 因子分析的 EM 与 MFA (因子分析混合)

- 关键词: 软分配 r_{ic} 、每簇 (W_c, μ_c, Ψ) ; “分段线性”近似曲流流形; 参数量 $O(KLD)$ 。
- 用途: 高维实值数据的通用密度建模; 多模态/簇结构的降维与可视化。

4. PCA (主成分) 与 PPCA (概率 PCA)

- 关键词: 最小化重构误差 = 最大方差投影; W 正交, 零噪声极限; PPCA: $\Psi = \sigma^2 I$, MLE $W = V(\Lambda - \sigma^2 I)^{1/2}$, σ^2 为“丢弃特征值”均值; PPCA 后验均值是收缩投影。
- 用途: 无监督降维、压缩、去噪、可视化; PPCA还能给出似然、处理缺失、做模型比较。

5. SVD 与低秩近似

- 关键词: $X = USV^\top$, 截断 SVD 给最优秩- L 近似; PCA = 截断 SVD; 参数量 $L(N+D+1)$ 。
- 用途: 图像/矩阵压缩、加速、数值稳定的 PCA 实现。

6. PCA 的 EM

- 关键词: E: $\tilde{Z} = (W^\top W)^{-1} W^\top \tilde{X}$; M: $W = \tilde{X} \tilde{Z}^\top (\tilde{Z} \tilde{Z}^\top)^{-1}$; “弹簧+杆”比喻; 在线/缺失数据友好。
- 用途: 大样本/流式/缺失场景的主成分学习; 可扩展到混合 PPCA/FA。

7. 模型选择 (选 L)

- 关键词: FA/PPCA 用边际似然/BIC/变分下界/交叉验证/ARD; PCA 用重构误差、碎石图 (scree)、解释方差、profile likelihood。
- 用途: 避免过拟合, 确定合适维数; 在非概率 PCA 中也能“半自动”找拐点。

8. 分类变量的 PCA (catFA / logistic normal)

- 关键词: $z \sim \mathcal{N}(0, I)$, Softmax $p(y|z)$; 诱导 logistic normal 分布; 变分 EM。
- 用途: 对多类别/二元响应的降维与可视化; 混合类型数据的缺失值填补。

9. 配对/多视图数据 (监督 PCA、PLS、CCA)

- 关键词：共享子空间 z^s + 私有子空间 z^x, z^y ；监督 PCA（因子回归）、PLS（输入私有+共享）、CCA（对称共享）；信息瓶颈与扩展到指数族。
- 用途：多模态融合、半监督学习、协同过滤、多标签预测。

10. ICA (独立成分分析)

- 关键词： $x = Wz + \varepsilon$, 非高斯独立先验 $p(z) = \prod_j p_j(z_j)$ ；白化 + 正交约束；FastICA (近似牛顿)、超/次高斯、偏度；IFA (每个源是一维 GMM) + EM；其它准则 (负熵、互信息、Infomax) \equiv ML。
 - 用途：盲源分离 (鸡尾酒会、EEG/MEG、金融)、找到独立的“基音/纹理”；对称学习、无主次排序。
-

② 子章节关联 (依赖 / 对比 / 递进)

- FA → PPCA → PCA (递进)**
FA 是“高斯低秩 + 对角噪声”的生成模型；PPCA 在 FA 上加各向同性噪声，给出闭式 MLE 与似然；PCA 是 PPCA 的零噪声极限与正交约束，目标变成最大方差/最小重构误差。
- PCA ↔ SVD (等价)**
截断 SVD 即最优低秩重构 = PCA；数值实现用 SVD 最稳健。
- PCA 的特性 vs PPCA/FA (对比)**
PCA 不是完整概率模型，测试重构误差不呈 U 型；PPCA/FA 有似然与奥卡姆剃刀，可做模型比较、处理缺失。
- FA ↔ MFA (递进)**
MFA 用多个低维线性子空间近似曲流流形，减少 GMM 的参数并防止过拟合。
- 不可辨识性与可解释性 (依赖)**
FA/PPCA 旋转不变 → 需要正交化/下三角/稀疏先验/选择旋转 (varimax) 改善解释；非高斯先验 (\rightarrow ICA) 可唯一化。
- 监督 PCA / PLS / CCA (对比 + 递进)**
监督 PCA：用 y 引导共享表示；PLS：输入有私有 + 共享；CCA：两边都允许私有，完全对称；均可扩展到指数族并用 EM/Bayes 学习。
- ICA 与 PCA/FA (对比)**
共同点：线性生成、需白化；不同点：PCA/FA 追求方差/高斯似然，ICA 追求独立性/非高斯性；PCA 只定子空间，ICA 还定旋转与源 (到符号与排列)。
- ICA 估计方法 (等价)**
最大非高斯 (负熵) \equiv 最小互信息 \equiv Infomax \equiv 最大似然 (在白化 + 正交下)。

可选 Mermaid (简化版) :

```

flowchart LR
FA --> PPCA
PPCA --> PCA
PCA <--equiv--> SVD
SVD --> |mixture| MFA
FA --> |rotation issue| Identifiability
Identifiability --> |non-Gaussian prior| ICA
ICA --> |FastICA / EM(IFAI)| Estimation
Estimation --> |EM实现| PCA_EM
PCA_EM --> |模型选择| ModelSel_PCA
ModelSel_PCA --> |模型选择/缺失| ModelSel_FA
ModelSel_FA --> CategoricalPCA["Categorical PCA (logistic normal)"]
CategoricalPCA --> FA
FA -.-.> CategoricalPCA
CategoricalPCA --> SupervisedPCA
SupervisedPCA --> PLS
PLS --> CCA
  
```

③ 本章总结（一页话）

本章把**线性潜变量模型**串成一条主线：

- 以 FA 为起点，用低秩 WW^\top 解释高维相关性；其特例 PPCA 给出概率化的 PCA；PCA 则是零噪声极限与几何视角（最大方差/最小重构）。SVD 提供最优化低秩近似与稳定实现，EM 让 PCA/FA 能在线、能处理缺失与混合。
- 为了拟合曲流流形，引入 MFA（线性片段拼接），并讨论非唯一性与旋转选择/稀疏化。
- 选择维数方面，概率模型（PPCA/FA/MFA）可用似然/变分/ARD；非概率 PCA 用重构误差、碎石图与 profile likelihood。
- 将思路推广到**离散变量**（categorical PCA / logistic normal）与**多视图**（监督 PCA、PLS、CCA），实现数据融合与因子回归。
- 若把先验从高斯换成**非高斯独立**，得到 ICA：不仅找子空间，还要找“正确旋转”以分离独立源。学习可用 FastICA（近似牛顿）、EM + 一维 GMM（IFA）等；看似不同的准则（负熵、互信息、Infomax）在白化与正交下都归于**最大似然**。

实践选型口诀：

- 仅为**压缩/可视化** → PCA/SVD；
- 要**似然、缺失、模型比较** → PPCA/FA（或 MFA）；
- 数据是**分类/计数** → categorical PCA / 指数族扩展；
- 多模态/配对预测** → 监督 PCA / PLS / CCA；
- 需要**盲源分离/独立因素** → ICA（FastICA/IFA），注意分量**无主次，对称学习**；
- 都要记得：**标准化、关注旋转不唯一、用合适的维数选择**。

请用通俗易懂的话总结本章，使我能够理解

下面把整章内容用“大白话”串起来，你读完能把每个概念放到一个清晰的地图上。

一句话总览

高维数据看起来复杂，其实常常是被**少量“隐形旋钮”（潜因子）**驱动的。

本章讲的就是：怎么用线性的方式找到这些旋钮、如何估好它们、怎么决定要几个、以及在不同数据类型/任务（连续、离散、配对、多模态、盲源分离）下该用哪一套模型。

1) 因子分析/PPCA/PCA：同一家族的三兄弟

- **因子分析 (FA)**：假设观测 x 是少量因子 z 线性变换后加点噪声得到：

$$x = Wz + \mu + \varepsilon,$$
 其中 ε 的协方差 Ψ 取对角。
 这等价于把总体协方差写成**低秩 + 对角噪声**： $C = WW^\top + \Psi$ 。
 用途：用少量因子解释变量间的相关性、填补缺失值、做密度建模。
 - **PPCA (概率 PCA)**：把 FA 的噪声变成各向同性 $\Psi = \sigma^2 I$ 。它既像 PCA 又是真正的概率模型，有似然、能做模型比较、能处理缺失。
 - **PCA (主成分分析)**：把 $\sigma^2 \rightarrow 0$ 且要求 W 正交，得到经典 PCA——
把数据正交投影到“方差最大的方向”，以最小重构误差。
 - **SVD 是扳手**：PCA 最稳健的算法就是对数据做 SVD；**截断 SVD 就是最优低秩近似。**
 - **EM 版 PCA/FA**：可以不用特征分解，交替两步：
 E 步算“在当前杆上每点的投影”（像**弹簧拉杆**的比喻），M 步调杆的方向。优点：可在线、能处理缺失、可扩展到混合模型。
-

2) 旋转不唯一：为什么“解释”会摇摆？

- 对 FA/PPCA，只要把 W 右乘任意正交矩阵 R （即**旋转潜因子**），**似然不变**。
 \Rightarrow **载荷矩阵和因子并不唯一**，解释会随旋转变化。
 - 常见应对：
 - 选定规范（如让 W 正交或下三角）、
 - 加**稀疏先验/正则**（让载荷多为 0 更好解读），
 - 做**可解释的旋转**（如 varimax），
 - 或干脆用**非高斯先验**（通向 ICA，可唯一化到符号/顺序之外）。
-

3) 曲面怎么办？——因子分析的混合 (MFA)

- 现实数据常在**弯曲的低维流形**上。
MFA = “用很多**低维小平面**拼出一条曲面”：每个簇一套 (W_c, μ_c, Ψ) 。

参数量是 $O(KLD)$, 比 K 个满协方差高斯 ($O(KD^2)$) 省得多。

好处：通用的高维密度模型、可视化友好。

4) 选几个因子（维数 L ）？

- FA/PPCA: 有似然, 可用 BIC/变分下界/交叉验证, 也可用 ARD 自动把无用因子“削成 0”。
- PCA: 不是概率模型, 常用
 - 重构误差或碎石图 (特征值从大到小),
 - 解释方差比例,
 - profile likelihood 自动找“拐点”。

注意: PCA 在测试集的重构误差往往一直降, 不一定有 U 型曲线; 而 PPCA 的测试对数似然会呈 U 型, 更便于选 L 。

5) 非连续/多视图的扩展

- 分类 (离散) 数据的 PCA: 把因子 z 过 softmax 生成类别, 诱导出 logistic-normal 分布; 用变分 EM 训练。
用途: 对问卷/打标等分类数据降维、填补缺失。
 - 配对/多模态
 - 监督 PCA (因子回归) : 学一个潜表示同时解释 x 和要预测的 y 。
 - PLS: 输入 x 有私有子空间 + 共享子空间; 共享的那部分帮助预测 y 。
 - CCA: 两边都允许私有, 同时有共享, 完全对称。
它们都能做进指数族/变分/贝叶斯; 适用于数据融合、协同过滤、多标签。
-

6) ICA: 把混合声音拆回“原声”

- 假设 $x = Wz + \varepsilon$, 但源 z 的每一维是独立、非高斯 (可以厚尾/扁平/偏斜)。
先白化数据, W 可视作正交矩阵; PCA 只定“平面”, ICA 还要定旋转, 才能把源分开 (到符号和顺序之外的唯一性)。
- 学习方法
 - FastICA: 近似牛顿法, 使用 $\log \cosh(z)$ 等“对比函数”。
 - IFA (独立因子分析) : 每个源是一维 GMM, 用 EM 交替学源分布与 W 。
 - 看似不同的原则 (最大非高斯性/ 最小互信息/ Infomax) 在白化+正交下其实都≈ 极大似然。

- **分量没有主次**: 同时学、再做正交化，事后按任务需要再排序（如方差、峰度、与标签相关性等）。
-

7) 什么时候用哪一个? (小抄)

- 只想压缩/可视化/去噪: PCA (配 SVD)。
 - 要概率、要似然、要处理缺失: PPCA/FA; 弯曲结构: MFA。
 - 类别/计数型数据: 分类版 PCA (logistic/probit 链接)。
 - 数据融合/用 x 预测 y : 监督 PCA;
更偏预测: PLS;
对称相关性分析: CCA。
 - 盲源分离/独立要素 (语音、EEG、金融因子) : ICA (FastICA/IFA)。
-

8) 常见坑 & 小贴士

- **标准化很关键**: 不同量纲会误导 PCA/FA。
 - **旋转不唯一**: 别把某个方向当“真理”，除非加了约束/先验。
 - **维数选择**: PCA 用碎石图/拐点；概率模型用似然/BIC/ARD。
 - **ICA 分量无主次**: 并行更新、事后按需求排序。
 - **SVD 是工程友好型实现**, 数值稳定、易截断做低秩。
-

最终心法

把复杂高维看成“少量旋钮 + 噪声”。
PCA/PPCA/FA 找**旋钮**并做**线性重建**；
MFA 用多根“小直线”**拼曲面**；
CCA/PLS/监督 PCA 把**多源/监督信息**也塞过“瓶颈”；
ICA 在非高斯前提下把**旋钮变成彼此独立的原声**。
选择哪一个，就看数据类型与任务目标。