1 潜在的线性模型

1.1 因子分析-Factor analysis

混合模型的一个问题是，它们仅使用单个潜在变量来生成观测值。特别是，每次观察只能来自 K 个原型中的一个。可以将混合模型视为使用 K 个隐藏二进制变量，表示集群标识的一个热编码。但由于这些变量相互排斥，模型的代表性仍然有限。

另一种方法是使用实值潜变量向量,$\mathbf{z}_i \in \mathbb{R}^L$。使用前最简单的是高斯函数（稍后我们将考虑其他选择）：

$$p(\mathbf{z}_i) = \mathcal{N}(\mathbf{z}_i|\boldsymbol{\mu}_0, \sum_0) \tag{12.1}$$

如果观察也是连续的，那么 $\mathbf{x}_i \in \mathbb{R}^D$，我们可以使用高斯函数表示可能性。与线性回归一样，我们将假设平均值是（隐藏）输入的线性函数，从而产生:

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

其中，$\mathbf{W}$ 是 $D \times L$ 矩阵，称为因子加载矩阵，$\boldsymbol{\Psi}$ 是 $D \times D$ 协方差矩阵。我们将 $\boldsymbol{\Psi}$ 视为对角，因为模型的整个要点是“强制” $\mathbf{z}_i$ 解释相关性，而不是“烘焙”到观测值的协方差。这种整体模型称为因子分析或 FA。$\boldsymbol{\Psi} = \sigma^2\mathbf{I}$ 的特殊情况称为概率主成分分析或 PPCA。这个名字的原因稍后将变得明显。生成过程，其中 $L = 1$ $D = 2$，$\boldsymbol{\Psi}$ 是对角的，如图 12.1 所示。我们取一个各向同性高斯“喷雾罐”，沿 $\mathbf{w}z_i + \mu$ 定义的 1d 线滑动。这会在 2d 中产生一个延迟（因此相关）高斯。

1.1.1 FA 是 MVN 的低秩参数化

FA 可以被认为是一种使用少量参数在 $\mathbf{x}$ 上指定关节密度模型的方法。要看到这一点，请注意，从等式 4.126 中，诱导边际分布 $p(\mathbf{x}_i|\theta)$ 是高斯分布：

$$p(\mathbf{x}_i|\theta) = \int \mathcal{N}(\mathbf{x}_i|\mathbf{W}\mathbf{z}_i + \boldsymbol{\mu}, \boldsymbol{\Psi})\mathcal{N}(\mathbf{z}_i|\boldsymbol{\mu}_0, \sum_0)d\mathbf{z}_i \tag{12.3}$$

$$= \mathcal{N}(\mathbf{x}_i|\mathbf{W}\boldsymbol{\mu}_0 + \boldsymbol{\mu}, \boldsymbol{\Psi} + \mathbf{W}\sum_0 \mathbf{W}^T) \tag{12.4}$$

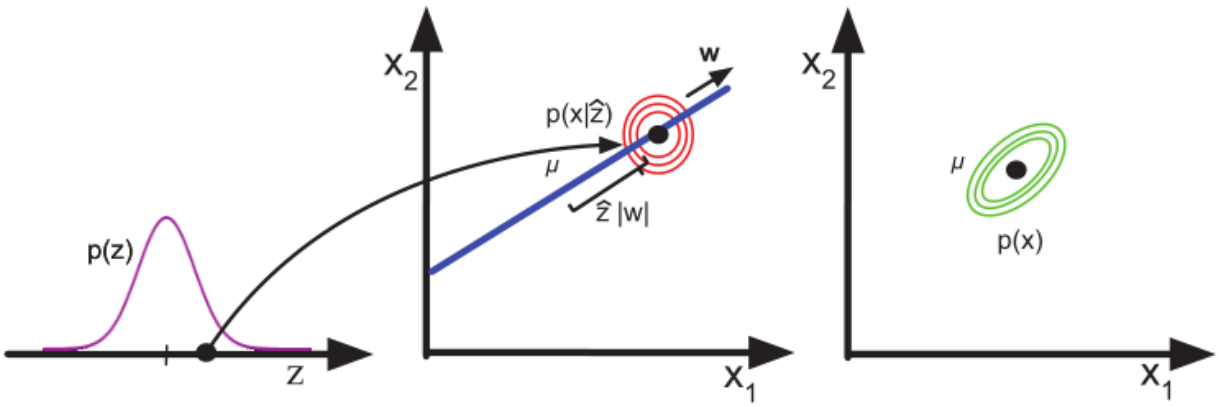


图 12.1: PPCA 生成过程的图示，其中 L=1 个潜在维度生成 D=2 个观察维度。基于（Bishop 2006b）的图 12.9。

由此我们可以看出，我们可以在不损失通用性的情况下设置 $\boldsymbol{\mu}_0 = 0$，因为我们总是可以将 $\mathbf{W}\boldsymbol{\mu}_0$ 吸收到 $\boldsymbol{\mu}$ 中。类似地，我们可以设置 $\sum_0 = \mathbf{I}$ 而不丧失通用性，因为我们可以通过定义新的权重矩阵 $\widetilde{\mathbf{W}} = \mathbf{W}\sum_0^{-1/2}$ 来“模拟”相关实验, 然后我们发现:

$$cov[\mathbf{x}|\boldsymbol{\theta}] = \widetilde{\mathbf{W}}^T + \mathbb{E}[\boldsymbol{\epsilon}\boldsymbol{\epsilon}^T] = (\mathbf{W}\sum_0^{-1/2})\sum_0(\mathbf{W}\sum_0^{-1/2})^T + \boldsymbol{\Psi} = \mathbf{W}\mathbf{W}^T + \boldsymbol{\Psi} \tag{12.5}$$

因此，我们看到 FA 使用低秩分解近似可见向量的协方差矩阵:

$$\mathbf{C} \triangleq cov[\mathbf{x}] = \mathbf{W}\mathbf{W}^T + \boldsymbol{\Psi} \tag{12.6}$$

这仅使用 O(LD) 参数，这允许在具有 $O(D^2)$ 参数的全协方差高斯和具有 O(D) 参数的对角协方差之间进行灵活的折衷。注意，如果我们不将 $\boldsymbol{\Psi}$ 限制为对角，我们可以很容易地将 $\boldsymbol{\Psi}$ 设置为全协方差矩阵；然后我们可以设置 $\mathbf{W} = 0$，在这种情况下，不需要潜在因素。

1.1.2 潜在因素推断

虽然 FA 可以被认为只是定义 $\mathbf{x}$ 上密度的一种方法，但它经常被使用，因为我们希望潜在因子 $\mathbf{z}$ 将揭示数据的有趣之处。为此，我们需要计算潜在因素的后验值。我们可以使用高斯的贝叶斯规则来给出:

$$p(\mathbf{z}_i|\mathbf{x}_i, \boldsymbol{\theta}) = \mathcal{N}(\mathbf{z}_i|\mathbf{m}_i, \sum_i) \tag{12.7}$$

$$\sum_i \triangleq (\sum_0^{-1} + \mathbf{W}^T\boldsymbol{\Psi}^{-1}\mathbf{W})^{-1} \tag{12.8}$$

$$\mathbf{m}_i \triangleq \sum_i(\mathbf{W}^T\boldsymbol{\Psi}^{-1}(\mathbf{x}_i - \boldsymbol{\mu}) + \sum_0^{-1}\boldsymbol{\mu}_0) \tag{12.9}$$

注意，在 FA 模型中，$\sum_i$ 实际上独立于 i，因此我们可以用 $\sum$ 表示它。计算该矩阵需要 $O(L^3 + L^2D)$ 时间，计算每个 $\mathbf{m}_i = \mathbb{E}[\mathbf{z}_i|\mathbf{x}_i, \boldsymbol{\theta}]$ 需要 $O(L^2 + LD)$ 时间。$\mathbf{m}_i$ 有时被称为潜在分数或潜在因子。

让我们举一个简单的例子，基于（Shalizi 2009）。我们考虑了一个由 $D = 11$ 变量和 $N = 387$ 个案例组成的数据集，这些案例描述了汽车的各个方面，例如发动机尺寸、气缸数量、每加仑英里数 (MPG)、价格等。我们首先拟合了一个 $L = 2$ 维模型。我们可以将 $\mathbf{m}_i$ 分数绘制为 $\mathbb{R}^2$ 中的点，以可视化数据，如图 12.2 所示。

为了更好地理解潜在因素的“意义”，我们可以将对应于每个特征维度的单位向量 $\mathbf{e}_1 = (1, 0, ..., 0), \mathbf{e}_2 = (0, 1, ..., 0)$ 等投影到低维空间。这些在图 12.2 中显示为蓝线；这被称为双时隙。我们看到横轴代表价格，对应于标记为“经销商”和“零售”的功能，右侧是昂贵的汽车。纵轴表示燃油效率 (以 MPG 为单位) 与尺寸的关系：重型车辆效率较低，较高，而轻型车辆效率较高，较低。我们可以通过点击一些点来“验证”这种解释，并在训练集中找到最接近的样本，然后打印它们的名称，如图 12.2 所示。然而，正如我们在第 12.1.3 节中讨论的那样，一般来说，解释潜变量模型充满了困难。

1.1.3 不可辨识性

就像混合模型一样，FA 也是无法识别的。为了说明这一点，假设 $\mathbf{R}$ 是一个任意正交旋转矩阵，满足 $\mathbf{R}\mathbf{R}^T = \mathbf{I}$。让我们定义 $\widetilde{\mathbf{W}} = \mathbf{W}\mathbf{R}$。那么可能修正矩阵的函数与未修正矩阵的函数相同，因为:

$$cov[\mathbf{x}] = \widetilde{\mathbf{W}}\mathbb{E}[\mathbf{z}\mathbf{z}^T]\widetilde{\mathbf{W}}^T + \mathbb{E}[\boldsymbol{\epsilon}\boldsymbol{\epsilon}^T] \tag{12.10}$$

$$= \mathbf{W}\mathbf{R}\mathbf{R}^T\mathbf{W}^T + \boldsymbol{\Psi} = \mathbf{W}\mathbf{W}^T + \boldsymbol{\Psi} \tag{12.11}$$

几何上，将 $\mathbf{W}$ 乘以正交矩阵就像在生成 $\mathbf{x}$ 之前旋转 $\mathbf{z}$；但由于 $\mathbf{z}$ 是从各向同性高斯中提取的，因此这对可能性没有影响。因此，我们无法唯一识别 $\mathbf{W}$，因此也无法唯一识别潜在因素。

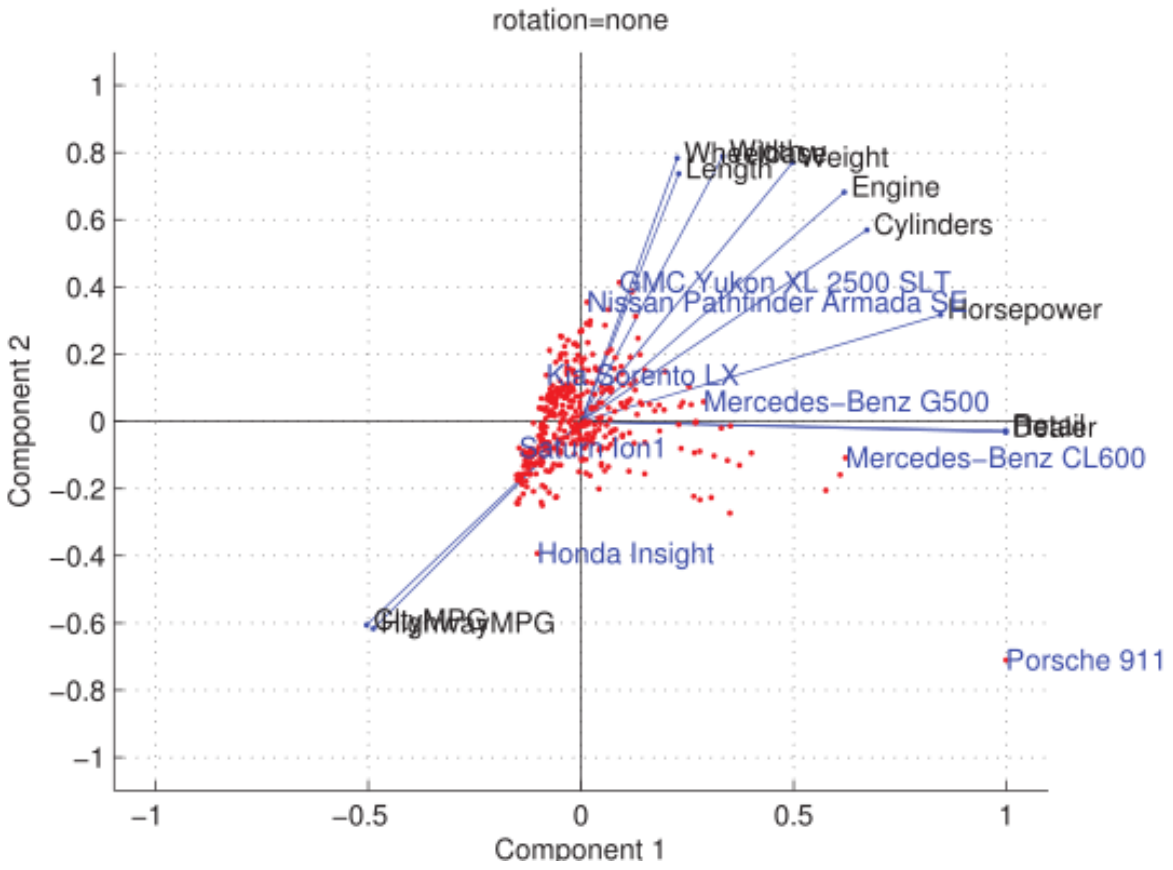


图 12.2: 基于因子分析的 2004 年 cars 数据的 2D 投影。蓝色文字是与某些选定点对应的汽车名称。图由 faBiplotDemo 生成。

为了确保唯一的解决方案，我们需要删除 $L(L-1)/2$ 自由度，因为这是大小为 $L \times L$ [1] 的正交矩阵的总数，FA 模型具有 $D + LD - L(L-1)/2$ 自由参数（不包括平均值），其中第一项来自 $\boldsymbol{\Psi}$。显然，我们要求它小于或等于 $D(D+1)/2$, 这是无约束 (但对称) 协方差矩阵中的参数数。这给了我们一个关于 L 的上界，如下：

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

例如，$D = 6$ 意味着 $L \leqslant 3$. 但我们通常从不选择这个上限，因为它会导致过度拟合（见第 12.3 节关于如何选择 L 的讨论）。不幸的是，即使我们设置 $L < L_{max}$，我们仍然无法唯一识别参数，因为旋转模糊仍然存在。不可识别性不会影响模型的预测性能。然而，它确实会影响载荷矩阵，从而影响潜在因素的解释。由于因子分析通常用于揭示数据中的结构，因此需要解决这个问题。以下是一些常用的解决方案：

- **强迫 W 正交**：可能是可识别性问题的最干净的解决方案是强迫 **W** 正交，并通过减少相应潜在因子的方差来对列排序。这是主成分分析采用的方法，我们将在第 12.2 节中讨论。结果不一定更容易解释，但至少是独特的。
- **W 强制为下三角形**：将 **W** 强制为下三角形是实现可识别性的一种方法，这在贝叶斯社区（例如，(Lopes and West 2004)）中很流行，其目的是确保第一个可见特征仅由第一个潜在因子生成，第二个可见特

[1] 要看到这一点，请注意 $L-1$ 在 R 中的自由参数在第一列中（因为列向量必须归一化为单位长度），有 $L-2$ 第二列中的 2 个自由参数（必须与第一列正交），依此类推。

征仅由前两个潜在因子生成，等等。例如，如果 $L = 3$ $D = 4$，则相应的因子载荷矩阵由下式给出：

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

对于 $j = 1 : L$，我们还要求 $w_{jj} > 0$。该约束矩阵中的参数总数为 $D + DL - L(L-1)/2$，等于唯一可识别参数的数量。这种方法的缺点是，前 L 个可见变量被称为创始人变量，是影响解读的潜在因素，因此必须慎重选择。

- **稀疏性促进权重先验** ：不是预先指定 $\mathbf{W}$ 中的哪些条目为零，我们可以使用 ℓ_1 正则化（Zou 等人，2006），ARD（Bishop 1999；Archambeau 和 Bach 2008），或 spike 和 slab 先验（Rattray 等人，2009）。这称为稀疏因子分析。这并不一定能确保唯一的 MAP 估计，但它确实鼓励可解释的解决方案。见第 13.8 节。

- **选择信息旋转矩阵**: 有多种启发式方法试图找到旋转矩阵 $\mathbf{R}$，旋转矩阵 $\mathbf{R}$ 可用于修改 $\mathbf{W}$（以及潜在因子），从而尝试增加可解释性，通常是通过鼓励它们（近似）稀疏。一种流行的方法是 varimax（Kaiser 1958）。

- **对潜在因素使用非高斯预设值**: 在第 12.6 节中，我们将讨论用非高斯分布代替 $p(\mathbf{z}_i)$ 如何使我们有时能够唯一地识别 $\mathbf{W}$ 以及潜在因素。这种技术被称为独立分量分析。

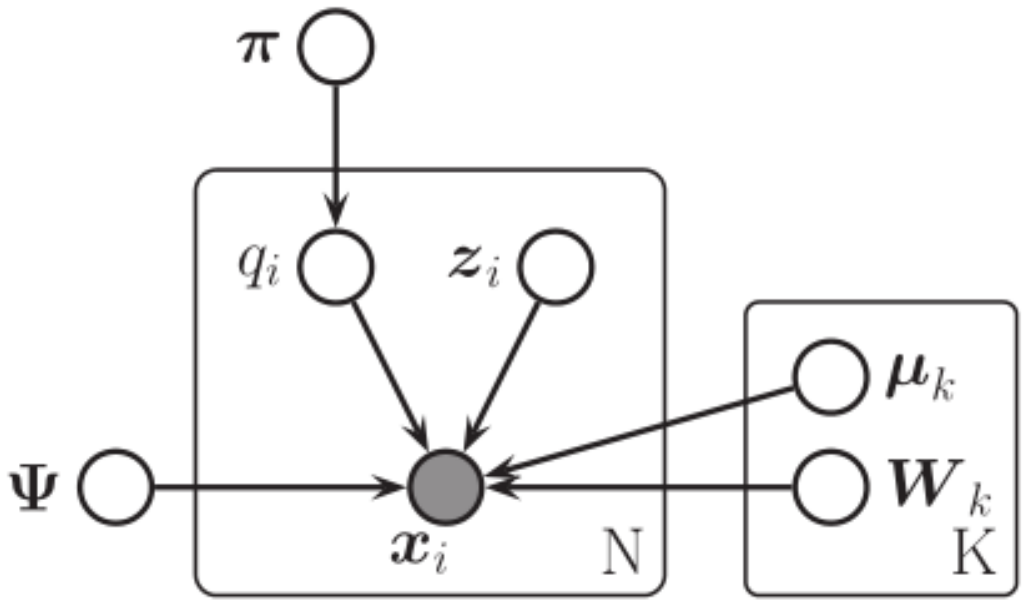


图 12.3: 混合因子分析仪作为 DGM。

1.1.4 因素分析器的混合物

FA 模型假设数据存在于低维线性流形上。事实上，大多数数据最好用某种形式的低维曲线流形建模。我们可以用分段线性流形近似曲线流形。这表明了以下模型：让维数 L_k 的第 k 个线性子空间由 $\mathbf{W}_k$ 表示，

$k = 1 : \mathbf{K}$。假设我们有一个潜在指示符 $q_i \in \{1, ..., K\}$ 指定我们应该使用哪个子空间来生成数据。然后，我们从高斯先验中采样 $\mathbf{z}_i$，并将其通过 $\mathbf{W}_k$ 矩阵（其中 $k = q_i$），然后添加噪声。更准确地说，模型如下：

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

$$p(\mathbf{z}_i|\boldsymbol{\theta}) = \mathcal{N}(\mathbf{z}_i|\mathbf{0}, \mathbf{I}) \tag{12.15}$$

$$p(q_i|\boldsymbol{\theta}) = Cat(q_i|\boldsymbol{\pi}) \tag{12.16}$$

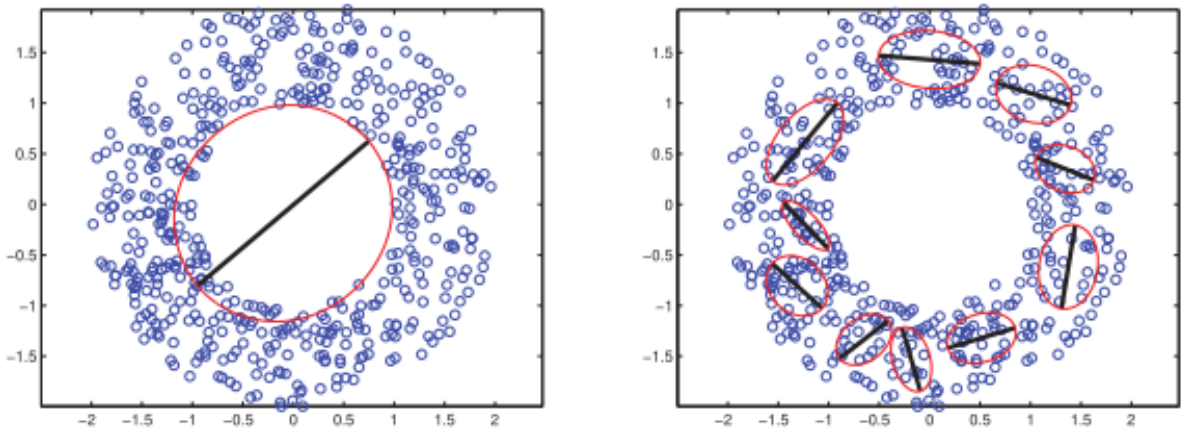


图 12.4: 拟合数据集的 1d PPCA 的混合物，$K = 1, 10$。图由 mixPpcaDemoNetlab 生成。

这被称为**因子分析仪的混合物**（MFA）（Hinton 等人，1997 年）。CI 假设如图 12.3 所示。

另一种考虑该模型的方法是将其视为高斯混合的低秩版本。特别是，该模型需要 $O(KLD)$ 参数，而不是全协方差高斯混合所需的 $O(KD^2)$ 参数。这可以减少过度装配。事实上，MFA 是高维实值数据的良好通用密度模型。

1.1.5 因素分析模型的 EM

利用第 4 章的结果，可以直接推导出一种 EM 算法来拟合 FA 模型。只需再多做一点工作，我们就可以适应混合 FA。下面我们陈述了没有证据的结果。推导见（Ghahramani 和 Hinton 1996a）；然而，如果你想精通数学，自己推导这些方程是一个有用的练习。

为了获得单因素分析仪的结果，只需在以下等式中设置 $r_{ic} = 1$ 和 $c = 1$。在第 12.2.5 节中，我们将看到在拟合 PPCA 模型时产生的这些方程的进一步简化，结果将具有特别简单和优雅的解释。

在 E 步骤中，我们用以下方法计算数据点 i 的集群 c 的后验责任。

$$r_{ic} \triangleq p(q_i = c|\mathbf{x}_i, \boldsymbol{\theta}) \propto \pi_c \mathcal{N}(\mathbf{x}_i|\boldsymbol{\mu}_c, \mathbf{W}_c\mathbf{W}_c^T + \boldsymbol{\Psi}) \tag{12.17}$$

$\mathbf{z}_i$ 的条件后验值由以下公式得出：

$$p(\mathbf{z}_i|\mathbf{x}_i, q_i = c, \boldsymbol{\theta}) = \mathcal{N}(\mathbf{z}_i|\mathbf{m}_{ic}, \sum_{ic}) \tag{12.18}$$

$$\sum_{ic} \triangleq (\mathbf{I}_L + \mathbf{W}_c^T\boldsymbol{\Psi}_c^{-1}\mathbf{W}_c)^{-1} \tag{12.19}$$

$$\mathbf{m}_{ic} \triangleq \sum_{ic}(\mathbf{W}_c^T\boldsymbol{\Psi}_c^{-1}(\mathbf{x}_i - \boldsymbol{\mu}_c)) \tag{12.20}$$

在 M 步骤中，最简单的是同时估计 $\boldsymbol{\mu}_c$ 和 $\mathbf{W}_c$，通过定义 $\widetilde{\mathbf{W}}_c = (\mathbf{W}_c, \boldsymbol{\mu}_c), \widetilde{\mathbf{z}} = (\mathbf{z}, 1)$, 此外，定义：

$$\mathbf{b}_{ic} \triangleq \mathbb{E}[\widetilde{\mathbf{z}}|\mathbf{x}_i, q_i = c] = [\mathbf{m}_{ic}; 1] \tag{12.21}$$

$$\mathbf{C}_{ic} \triangleq \mathbb{E}[\widetilde{\mathbf{z}}\widetilde{\mathbf{z}}^T|\mathbf{x}_i, q_i = c] = \begin{pmatrix} \mathbb{E}[\mathbf{z}\mathbf{z}^T|\mathbf{x}_i, q_i = c] & \mathbb{E}[\mathbf{z}|\mathbf{x}_i, q_i = c] \\ \mathbb{E}[\mathbf{z}|\mathbf{x}_i, q_i = c]^T & 1 \end{pmatrix} \tag{12.22}$$

然后 M 步骤如下:

$$\hat{\tilde{W}}_c = \left[\sum_i r_{ic}\mathbf{x}_i\mathbf{b}_{ic}^T\right]\left[\sum_i r_{ic}\mathbf{C}_{ic}\right]^{-1} \tag{12.23}$$

$$\hat{\boldsymbol{\Psi}} = \frac{1}{N}diag\left\{\sum_i r_{ic}\left(\mathbf{x}_i - \hat{\tilde{W}}_c\mathbf{b}_{ic}\right)\mathbf{x}_i^T\right\} \tag{12.24}$$

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

请注意，这些更新是针对“vanilla”EM 的。基于 ECM 的该算法的更快版本如（Zhao 和 Yu 2008）所述。

1.1.6 缺失数据的 FA 模型拟合

在许多应用程序中，例如协同过滤，我们会丢失数据。EM 方法拟合 FA/PPCA 模型的一个优点是很容易扩展到这种情况。然而，如果有大量缺失数据，则过度拟合可能是一个问题。因此，进行 MAP 估计或使用贝叶斯推理很重要。详见（Ilin 和 Raiko 2010）。

1.2 主成分分析（PCA）

考虑 FA 模型，其中我们约束 $\boldsymbol{\Psi} = \sigma^2\mathbf{I}$，$\mathbf{W}$ 为正交。可以证明（Tipping 和 Bishop 1999），作为 $\sigma^2 \to 0$ 时，该模型简化为经典（**非概率**）**主成分分析（PCA）**，也称为 **arhunen-Loeve 变换**。$\sigma^2 > 0$ 的版本称为**概率主成分分析（PPCA）**（Tipping 和 Bishop 1999），或**合理主成分分析**（Roweis 1997）。（从不同角度独立得出了一个等效结果，见（Moghaddam 和 Pentland 1995）

为了理解这个结果，我们首先必须学习经典主成分分析。然后，我们将主成分分析连接到奇异值分解。最后我们再来讨论 PPCA。

1.2.1 经典 PCA：定理陈述

在下面的定理中总结了经典主成分分析 (PCA) 的**综合观点**。

My Theorem 1.2.1. *假设我们想要找到 L 线性基向量的正交集 $\mathbf{w}_j \in \mathbb{R}^D$ 和相应的分数 $\mathbf{z}_i \in \mathbb{R}^D$，使平均重建误差最小化,*

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

其中 $\hat{\mathbf{x}}_i = \mathbf{W}\mathbf{z}_i$，受 $\mathbf{W}$ 是正交的约束。等价地，我们可以将此目标写为：

$$J(\mathbf{W}, \mathbf{Z}) = \parallel \mathbf{X} - \mathbf{W}\mathbf{Z}^T \parallel_F^2 \tag{12.27}$$

其中 $\mathbf{Z}$ 是一个 $N \times L$ 矩阵，其中 $\mathbf{z}_i$ 在其行中,$\parallel \mathbf{A} \parallel$ 是矩阵 $\mathbf{A}$ 的 **Frobenius 范数**，定义如下：

$$\parallel \mathbf{A} \parallel_F = \sqrt{\sum_{i=1}^m\sum_{j=1}^n a_{ij}^2} = \sqrt{tr(\mathbf{A}^T\mathbf{A})} = \parallel \mathbf{A}(:) \parallel_2 \tag{12.28}$$

通过设置 $\hat{\mathbf{W}} = \mathbf{V}_L$ 获得最优解，其中 $\mathbf{V}_L$ 包含具有经验协方差矩阵最大特征值的 L 个特征向量 $\hat{\sum} = \frac{1}{N}\sum_{i=1}^N \mathbf{x}_i\mathbf{x}_i^T$（为了符号简单起见，我们假设 $\mathbf{x}_i$ 具有零均值。）此外，数据的最佳低维编码由 $\hat{\mathbf{z}}_i = \mathbf{W}^T\mathbf{x}_i$ 给出，这是数据在特征向量跨越的列空间上的正交投影。

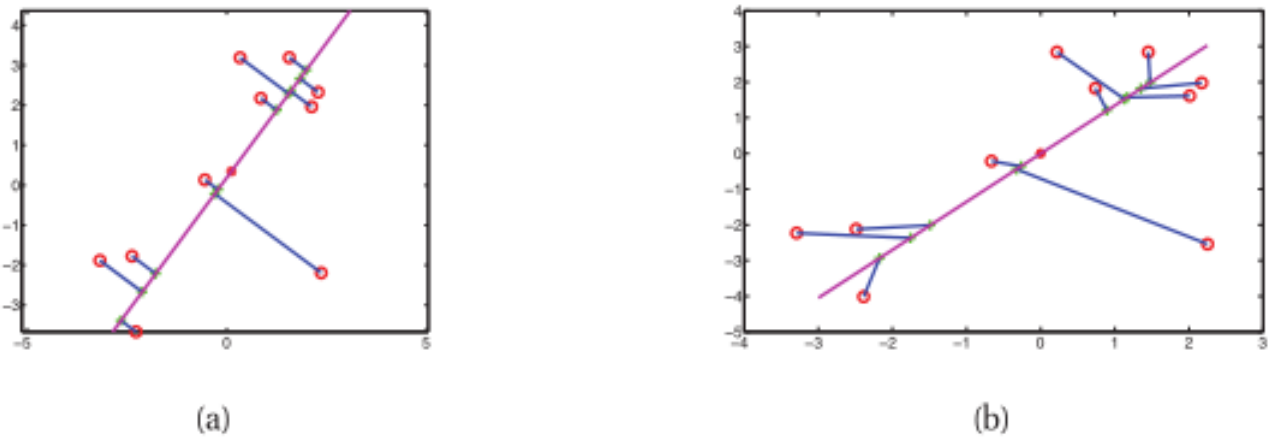


图 12.5 在 D=2 和 L=1 的情况下，PCA 和 PPCA 的说明。圆圈是原始数据点，交叉点是重构的数据。红星是数据的平均值。(a) PCA。这些点被正交地投射到线上。图由 pcaDemo2d 生成。(b) PPCA。投影不再是正交的了。重构的数据向数据平均值缩减（红星）。基于（Nabney 2001）的图 7.6。图由 ppcaDemo2d 生成。

图 12.5（a）中显示了 $D = 2$ 和 $L = 1$ 的示例。对角线是矢量 $\mathbf{w}_1$；这被称为第一主分量或主方向。数据点 $\mathbf{x}_i \in \mathbb{R}^2$ 正交投影到这条线上，得到 $z_i \in \mathbb{R}$、这是对数据的最佳一维近似。（稍后我们将讨论图 12.5（b）

一般来说，很难将高维数据可视化，但如果这些数据恰好是一组图像，就很容易做到。一组图像，就很容易做到这一点。图 12.6 显示了前三个主向量，重塑为图像，以及使用不同数量的基向量重建特定图像。（我们在第 11.5 节中讨论了如何选择 L。）

下面我们将表明，主方向是数据显示最大方差的方向。这意味着主成分分析可能仅仅因为测量尺度而被方差高的方向“误导”。图 12.7（a）显示了一个示例，其中纵轴（重量）使用的范围比横轴（高度）大，导致线条看起来有些“不自然”。因此，标准做法是首先标准化数据，或等效地使用相关矩阵而不是协方差矩阵。从图 12.7（b）中可以明显看出这一点的好处。

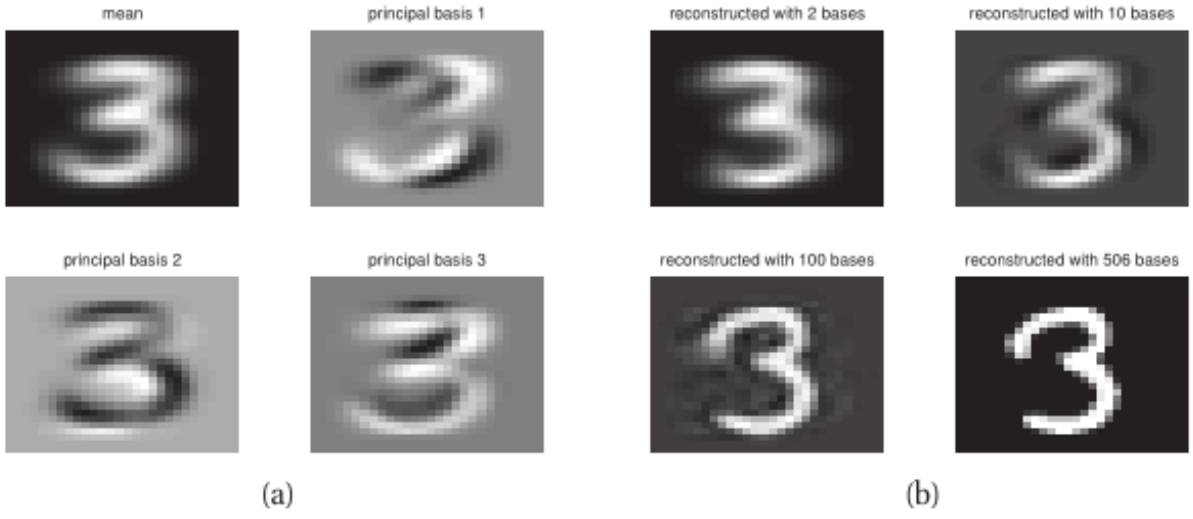


图 12.6:（a）基于数字 3 的 25 个图像（来自 MNIST 数据集）的平均值和前三个 PC 基向量（特征数字）。（b）基于 2、10、100 和所有基向量重建图像。由 pcaImageDemo 生成的图。

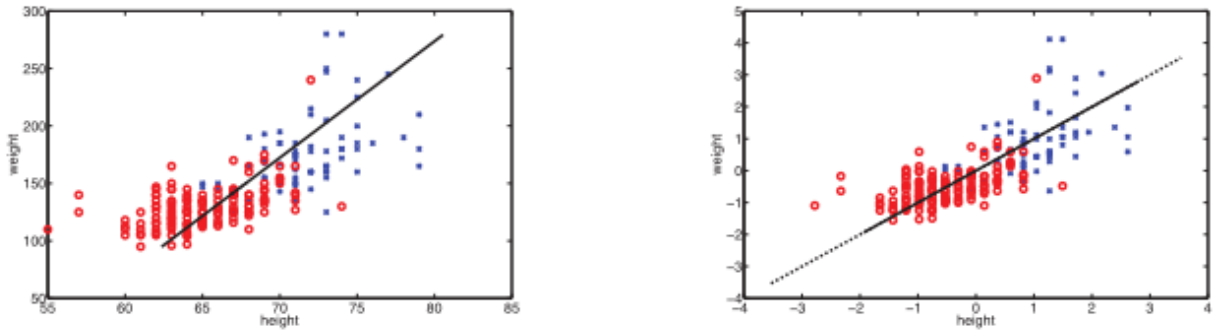


图 12.7: 标准化对应用于身高/体重数据集的主成分分析的影响。左：原始数据的主成分分析。右：标准化数据的主成分分析。pcaDemoHeightWeight 生成的图。

1.2.2 证明 *

证明: 我们使用 $\mathbf{w}_j \in \mathbb{R}^D$ 表示第 j 个主方向，$\mathbf{x}_i \in \mathbb{R}^D$ 表示第 i 个高维观测，$\mathbf{z}_i \in \mathbb{R}^L$ 表示第 i 个低维表示，和 $\tilde{\mathbf{z}}_j \in \mathbb{R}^N$ 表示 $[z_{1j}, ..., z_{Nj}]$，它是所有低维向量的第 j 个分量。

让我们从估计最佳 1d 解，$\mathbf{w}_1 \in \mathbb{R}^D$，以及相应的投影点 $\tilde{\mathbf{z}}_1 \in \mathbb{R}^N$ 开始。我们将在后面找到其余的基数 $\mathbf{w}_2, \mathbf{w}_3$ 等。重建误差由以下公式给出:

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

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

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

$$\tag{1}$$

由于 $\mathbf{w}_1^T\mathbf{w}_1 = 1$（根据正交性假设）。取导数 $wrt\ z_{i1}$ 并等于零，得到:

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

因此，通过将数据正交投影到第一主方向 $\mathbf{w}_1$ 上获得最佳重建权重（见图 12.5（a））。插回会得到：

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

现在，投影坐标的方差由以下公式给出:

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

由于:

$$\mathbb{E}[z_{i1}] = \mathbb{E}[\mathbf{x}_i^T\mathbf{w}_1] = \mathbb{E}[\mathbf{x}_i]^T\mathbf{w}_1 = 0 \tag{12.35}$$

因为数据已经被居中了。由此我们可以看出，最小化重建误差相当于使投影数据的方差最大化，即:

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

这就是为什么人们经常说主成分分析可以找到最大方差的方向。这称为 PCA 的**分析视图**。

预测数据的方差可以写成:

$$\frac{1}{N}\sum_{i=1}^{N} z_{i1}^2 \;=\; \frac{1}{N}\sum_{i=1}^{N} \mathbf{w}_1^T\hat{\sum}\mathbf{w}_1 \tag{12.37}$$

其中 $\hat{\sum} \;=\; \frac{1}{N}\sum_{i=1}^{N}\sum_i \mathbf{x}_i\mathbf{x}_i^T$ 是经验协方差矩阵（如果数据标准化，则为相关矩阵）。

我们可以通过让 $||\mathbf{w}_1|| \;\rightarrow\; \infty$ 使投影的方差最小化（从而最小化重建误差），因此我们施加约束 $||\mathbf{w}_1|| = 1$，代替最大化。

$$\widetilde{J}(\mathbf{w}_1) \;=\; \mathbf{w}_1^T\hat{\sum}\mathbf{w}_1 \;+\; \lambda_1(\mathbf{w}_1^T\mathbf{w}_1 - 1) \tag{12.38}$$

其中 λ_1 是拉格朗日乘子。取导数并等于零, 我们有:

$$\frac{\partial}{\partial \mathbf{w}_1}\widetilde{J}(\mathbf{w}_1) \;=\; 2\hat{\sum}\mathbf{w}_1 - 2\lambda_1\mathbf{w}_1 \;=\; 0 \tag{12.39}$$

$$\hat{\sum}\mathbf{w}_1 \;=\; \lambda_1\mathbf{w}_1 \tag{12.40}$$

因此，方差最大化的方向是协方差矩阵的特征向量。左乘以 $\mathbf{w}_1$（使用 $\mathbf{w}_1^T\mathbf{w}_1 = 1$），我们发现投影数据的方差为:

$$\mathbf{w}_1^T\hat{\sum}\mathbf{w}_1 \;=\; \lambda_1 \tag{12.41}$$

由于我们想要最大化方差，我们选择对应于最大特征值的特征向量。现在，让我们找到另一个方向 $\mathbf{w}_2$，以进一步最小化重建误差，前提是 $\mathbf{w}_1^T\mathbf{w}_2 = 0$ 和 $\mathbf{w}_2^T\mathbf{w}_2 = 1$。错误是:

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

优化 wrt $\mathbf{w}_1$ 和 $\mathbf{z}_1$ 得到了与之前相同的解决方案。练习 12.4 要求你展示 $\frac{\partial J}{\partial \mathbf{z}_2} = 0$ 产生 $z_{i2} = \mathbf{w}_2^T\mathbf{x}_i$。换句话说，通过投影到第二主方向来获得第二主编码。代入后得到的结果是:

$$J(\mathbf{w}_2) \;=\; \frac{1}{n}\sum_{i=1}^{N}[\mathbf{x}_i^T\mathbf{x}_i - \mathbf{w}_1^T\mathbf{x}_i\mathbf{x}_i^T\mathbf{w}_1 - \mathbf{w}_2^T\mathbf{x}_i\mathbf{x}_i^T\mathbf{w}_2] = const - \mathbf{w}_2^T\hat{\sum}\mathbf{w}_2 \tag{12.43}$$

去掉常数项并添加约束得到:

$$\widetilde{J}(\mathbf{w}_2) \;=\; -\mathbf{w}_2^T\hat{\sum}\mathbf{w}_2 + \lambda_2(\mathbf{w}_2^T\mathbf{w}_2 - 1) + \lambda_{12}(\mathbf{w}_2^T\mathbf{w}_1 - 0) \tag{12.44}$$

练习 12.4 要求你证明解是由具有第二大特征值的特征向量给出的最大的特征值:

$$\hat{\sum}\mathbf{w}_2 = \lambda_2\mathbf{w}_2 \tag{12.45}$$

证明继续以这种方式进行。（在形式上，可以使用归纳法。）

1.2.3 奇异值分解（SVD）

我们根据协方差矩阵的特征向量定义了 PCA 的解。然而，还有另一种获得解的方法，即基于**奇异值分解的**或者 **SVD**。这基本上将特征向量的概念从平方矩阵推广到任何类型的矩阵。

特别是，任何（实）$N \times D$ 矩阵 $\mathbf{X}$ 可以分解如下:

$$\underbrace{\mathbf{X}}_{N\times D} \;=\; \underbrace{\mathbf{U}}_{N\times N}\underbrace{\mathbf{S}}_{N\times D}\underbrace{\mathbf{V}^T}_{D\times D} \tag{12.46}$$

其中 $\mathbf{U}$ 是一个 $N \times N$ 矩阵，其列是正交的（即：$\mathbf{U}^T\mathbf{U} = \mathbf{I}_N$），$\mathbf{V}$ 是 $D \times D$ 矩阵，其行和列是正交的（即:$\mathbf{V}^T\mathbf{V} = \mathbf{V}\mathbf{V}^T = \mathbf{I}_D$），$\mathbf{S}$ 是一个 $N \times D$ 矩阵，包含 $r = min(N, D)$ **奇异值** $\sigma_i \geq 0$ 主对角线上的 0，0 填充矩阵的其余部分。$\mathbf{U}$ 的列是左奇异向量，$\mathbf{V}$ 的列是右奇异向量。示例见图 12.8（a）。

由于最多有 D 个奇异值（假设 $N > D$），$N - D$ 的最后 $\mathbf{U}$ 列是不相关的，因为它们将被乘以 0。**经济型 SVD 或薄型 SVD** 避免了计算这些不必要的元素。让我们用 $\hat{\mathbf{U}}\hat{\mathbf{S}}\hat{V}$ 表示这个分解。如果 $N > D$，我们有:

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

如图 12.8（a）所示。如果 $N < D$，则我们得到:

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

计算经济型 SVD 需要 $O(N\ Dmin(N, D))$ 时间（Golub 和 van Loan 1996，第 254 页）。

特征向量和奇异向量之间的联系如下。对于任意实矩阵 $\mathbf{X}$，如果 $\mathbf{X} = \mathbf{USV}^T$，我们有:

$$\mathbf{X}^T\mathbf{X} \ = \ \mathbf{VS}^T\mathbf{U}^T\mathbf{USV}^T \ = \ \mathbf{V}(\mathbf{S}^T\mathbf{S})\mathbf{V}^T \ = \ \mathbf{VDV}^T \tag{12.49}$$

其中，$\mathbf{D} = \mathbf{S}^2$ 是包含平方奇异值的对角矩阵。因此:

$$(\mathbf{X}^T\mathbf{X})\mathbf{V} \ = \ \mathbf{VD} \tag{12.50}$$

所以 $\mathbf{XX}^T$ 的特征向量等于 $\mathbf{V}$，$\mathbf{X}$ 的右奇异向量，$\mathbf{XX}^T$ 的特征值等于 $\mathbf{D}$，奇异值的平方。类似地:

$$\mathbf{XX}^T \ = \ \mathbf{USV}^T\mathbf{VS}^T\mathbf{U}^T \ = \ \mathbf{U}(\mathbf{SS}^T)\mathbf{U}^T \tag{12.51}$$

$$(\mathbf{XX}^T)\mathbf{U} \ = \ \mathbf{U}(\mathbf{SS}^T) \ = \ \mathbf{UD} \tag{12.52}$$

所以 $\mathbf{XX}^T$ 的特征向量等于 $\mathbf{U}$，$\mathbf{X}$ 的左奇异向量。同样，$\mathbf{XX}^T$ 的特征值等于奇异值的平方。我们可以总结如下：

$$\mathbf{U} \ = \ evec(\mathbf{XX}^T), \mathbf{V} \ = \ evec(\mathbf{X}^T\mathbf{X}), \mathbf{S}^2 \ = \ eval(\mathbf{XX}^T) \ = \ eval(\mathbf{X}^T\mathbf{X}) \tag{12.53}$$

由于特征向量不受矩阵线性缩放的影响，我们可以看到 $\mathbf{S}$ 的右奇异向量等于经验协方差 $\hat{\sum}$ 的特征向量。此外，$\hat{\sum}$ 的特征值是平方奇异值的比例形式。这意味着我们只需要几行代码就可以执行 PCA（参见 pcaPmtk）。

然而，PCA 和 SVD 之间的联系更深入。根据等式 12.46，我们可以表示秩 r 矩阵，如下所示：

$$\mathbf{X} = \sigma_1 \begin{pmatrix} | \\ \mathbf{u}_1 \\ | \end{pmatrix} (- \ \mathbf{v}_1^T \ -) + \cdots + \sigma_r \begin{pmatrix} | \\ \mathbf{u}_r \\ | \end{pmatrix} (- \ \mathbf{v}_r^T \ -) \tag{12.54}$$

如果奇异值如图 12.10 所示快速消失，我们可以生成矩阵的秩 L 近似值，如下所示：

$$\mathbf{X} \approx \mathbf{U}_{:,1:L}\ \mathbf{S}_{1:L,1:L}\ \mathbf{V}_{:,1:L}^T \tag{12.55}$$

这称为**截断奇异值分解**（见图 12.8（b））。使用秩 L 近似表示 $N \times D$ 矩阵所需的参数总数为:

$$NL + LD + L = L(N + D + 1) \tag{12.56}$$

例如，考虑图 12.9（左上）中的 200×320 像素图像。这里面有 64000 个数字。我们看到秩近似 20，只有 $(200 + 320 + 1)20 = 10420$ 个数字是一个非常好的近似。

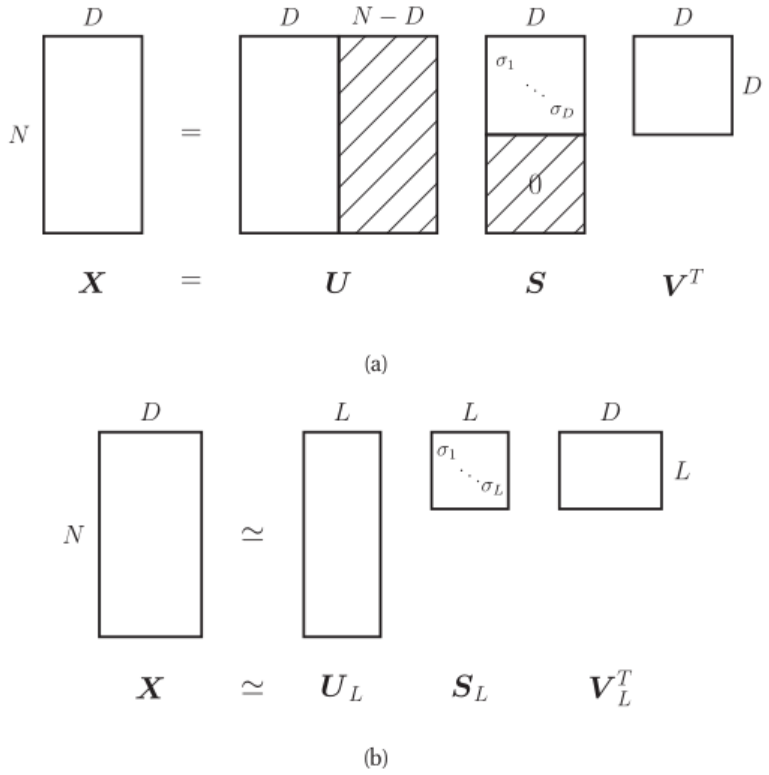


图 12.8:（a）非平方矩阵的奇异值分解 $\mathbf{X} = \mathbf{USV}^T$。**S** 的阴影部分和所有非对角项均为零。**U** 和 **S** 中的阴影条目不在经济型版本中计算，因为它们不需要。（b）秩 L 的截断奇异值分解近似。

可以证明，该近似中的误差由以下公式得出：

$$\| \mathbf{X} - \mathbf{X}_L \|_F \approx \sigma_{L+1} \tag{12.57}$$

此外，可以证明奇异值分解为矩阵提供了最佳秩 L 近似（在最小化上述 Frobenius 范数的意义上最佳）。

让我们将其连接回 PCA。设 $\mathbf{X} = \mathbf{USV}^T$ 是 **X** 的截断奇异值分解。我们知道 $\hat{\mathbf{W}} = \mathbf{V}$，并且 $\hat{\mathbf{Z}} = \mathbf{X}\hat{\mathbf{W}}$，即：

$$\hat{\mathbf{Z}} = \mathbf{USV}^T\mathbf{V} = \mathbf{US} \tag{12.58}$$

此外，最优重构由 $\hat{\mathbf{X}} = \mathbf{Z}\hat{\mathbf{W}}^T$，我们得到：

$$\hat{\mathbf{X}} = \mathbf{USV}^T \tag{12.59}$$

这与截断奇异值分解近似完全相同！这是另一个例子，说明 PCA 是数据的最佳低秩近似。

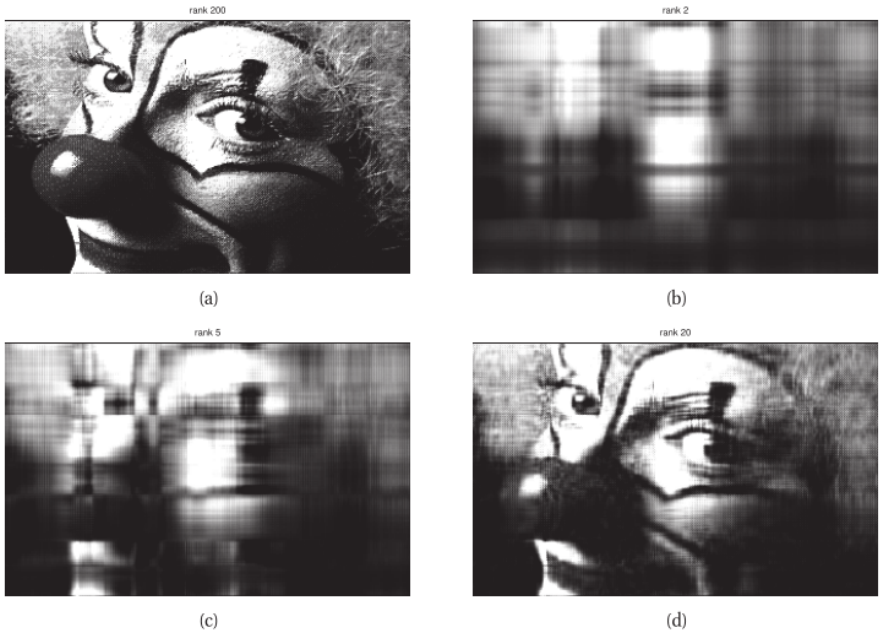


图 12.9: 图像的低秩近似。左上角：原始图像的大小为 200×320，秩为 200。后续图像的秩为 2、5 和 20。由 svdImageDemo 生成的图。

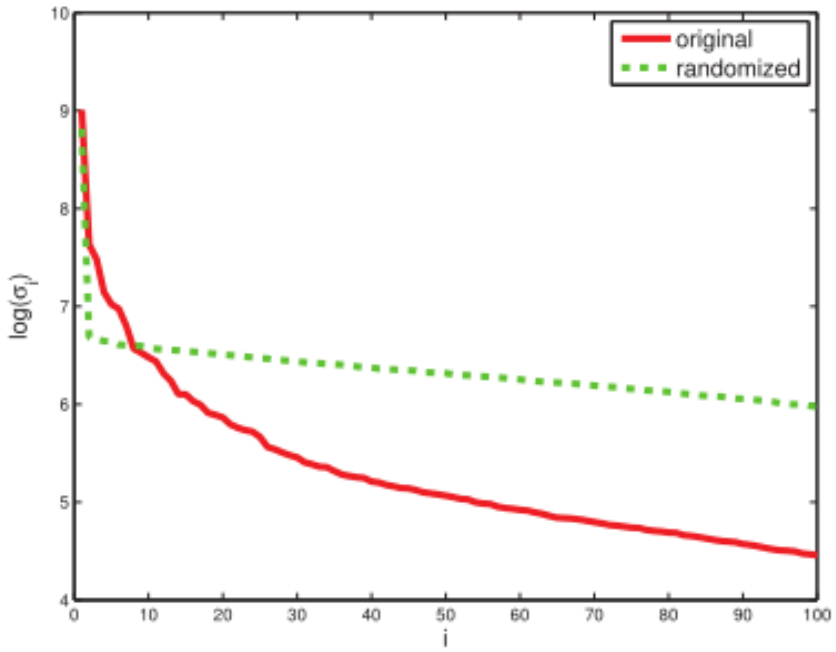


图 12.10: 粗略的图像的前 50 个对数奇异值（红色实线）和通过随机洗牌像素获得的数据矩阵（绿色虚线）。由 svdImageDemo 生成的图。

1.2.4 概率主成分分析

我们现在准备再次观测 PPCA。可以显示以下显著结果。

My Theorem 1.2.2. （(*Tipping* 和 *Bishop*，*1999* 年)）。考虑一个因子分析模型，其中 $\boldsymbol{\Psi} = \sigma^2\mathbf{I}$，$\mathbf{W}$ 是正交的。观察到的数据对数似然由以下公式得出：

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

其中 $\mathbf{C} = \mathbf{W}\mathbf{W}^T + \sigma^2\mathbf{I}$ 和 $\mathbf{S} = \frac{1}{N}\sum_{i=1}^{N}\mathbf{x}_i\mathbf{x}_i^T = (1/N)\mathbf{X}^T\mathbf{X}$。（为了符号简单起见，我们假设数据居中。）对数似然的最大值由下式得出：

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

其中 $\mathbf{R}$ 是任意 $L \times L$ 正交矩阵，$\mathbf{V}$ 是 $D \times L$ 矩阵，其列是 $\mathbf{S}$ 的前 L 个特征向量，Λ 是特征值的对应对角矩阵。在不损失一般性的情况下，我们可以设置 $\mathbf{R} = \mathbf{I}$。此外，噪声方差的最大似然估计由下式得出：

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

这是与被抛弃的维度相关的平均方差。

因此，作为 $\sigma^2 \to 0$，与经典的 PCA 相同。$\hat{\mathbf{Z}}$ 呢？很容易看出，潜在因素的后验值由以下公式得出：

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

$$\hat{\mathbf{F}} \quad \triangleq \quad \hat{\mathbf{W}}^T\hat{\mathbf{W}} + \hat{\sigma}^2\mathbf{I} \tag{12.64}$$

（不要将 $\mathbf{F} = \mathbf{W}^T\mathbf{W} + \sigma^2\mathbf{I}$ 与 $\mathbf{C} = \mathbf{W}\mathbf{W}^T + \sigma^2\mathbf{I}$ 混淆。）因此，作为 $\sigma^2 \to 0$，我们找到 $\mathbf{W} \hat{\to} \mathbf{V}$、$\hat{\mathbf{F}} \to \mathbf{I}$ 和 $\hat{\mathbf{z}_i} \to \mathbf{V}^T\mathbf{x}_i$。因此，后验平均值是通过将数据正交投影到 $\mathbf{V}$ 的列空间来获得的，与经典 PCA 中一样。

然而，请注意，如果 $\sigma^2 >$，则后验平均值不是正交投影，因为它向前验平均值收缩了一些，如图 12.5（b）所示。这听起来像是一个不受欢迎的特性，但这意味着重建将更接近整体数据平均值，$\hat{\boldsymbol{\mu}} = \bar{\mathbf{x}}$。

1.2.5 PCA 的 EM 算法

虽然拟合主成分分析模型的通常方法使用特征向量方法或奇异值分解，但我们也可以使用 EM，这将证明具有一些优势，我们将在下面讨论。主成分分析的 EM 依赖于主成分分析的概率公式。然而，该算法继续在零噪声限制下工作，$\sigma^2 = 0$，如（Roweis 1997）所示。

设 $\widetilde{\mathbf{Z}}$ 是一个 $L \times N$ 矩阵，沿其列存储后验均值（低维表示）。类似地，将原始数据沿其列存储。根据等式 12.63，当 $\sigma^2 = 0$ 时，我们有:

$$\widetilde{\mathbf{Z}} = (\mathbf{W}^T\mathbf{W})^{-1}\mathbf{W}^T\widetilde{\mathbf{X}} \tag{12.65}$$

这构成了 E 步骤。请注意，这只是数据的正交投影。

根据等式 12.23，第 M 步长由下式得出：

$$\hat{\mathbf{W}} \quad = \quad \left[\sum_i \mathbf{x}_i\mathbb{E}[\mathbf{z}_i]^T\right]\left[\sum_i \mathbb{E}[\mathbf{z}_i]\mathbb{E}[\mathbf{z}_i]^T\right]^{-1} \tag{12.66}$$

在这里，我们利用了当 $\sigma^2 = 0$ 时 $\sum = cov[\mathbf{z}_i|\mathbf{x}_i,\theta] = 0\mathbf{I}$ 的事实。值得将此表达式与多输出线性回归的最大似然估计进行比较（方程 7.89），其形式为 $\mathbf{W} = (\sum_i \mathbf{y}_i\mathbf{x}_i^T)(\sum_i \mathbf{x}_i\mathbf{x}_i^T)^{-1}$. 因此，我们看到 M 步类似于线性回归，其中我们用潜在变量的预期值替换观察到的输入。

总之，以下是整个算法：

- **第 E 步** $\widetilde{\mathbf{Z}} = (\mathbf{W}^T\mathbf{W})^{-1}\mathbf{W}^T\widetilde{\mathbf{X}}$
- **第 M 步** $\mathbf{W} = \widetilde{\mathbf{X}}\widetilde{\mathbf{Z}}^T(\widetilde{\mathbf{Z}}\widetilde{\mathbf{Z}}^T)^{-1}$

（Tipping 和 Bishop 1999）表明，EM 算法的唯一稳定不动点是全局最优解。也就是说，EM 算法收敛到一个解，其中 **W** 跨越与第一个 L 特征向量定义的线性子空间相同的线性子空间。然而，如果我们希望 **W** 是正交的，并且以特征值的降序包含特征向量，我们必须正交化得到的矩阵（这可以非常便宜地完成）。或者，我们可以修改 EM 以直接给出主基（Ahn 和 Oh 2003）。

该算法在 $D = 2$ 和 $L = 1$ 的情况下具有简单的物理类比（Roweis 1997）。考虑 $\mathbb{R}^2$ 中由弹簧连接到刚性杆的一些点，其方向由向量 **w** 定义。z_i 是第 i 个弹簧连接到杆的位置。在 E 步骤中，我们固定杆，让附着点四处滑动，以最小化弹簧能量（与残差平方和成比例）。在 M 步骤中，我们固定附着点，让杆旋转，以最小化弹簧能量。请参见图 12.11 得到解释。

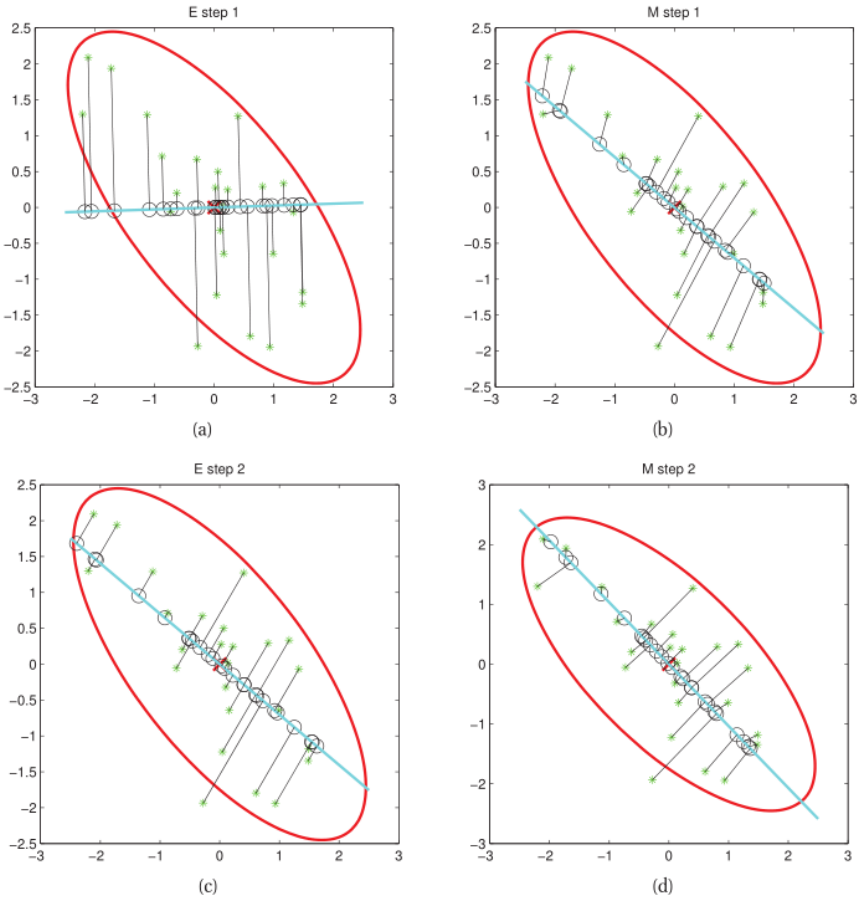


图 12.11: $D = 2$ 和 $L = 1$ 时 PCA 的 EM 图示。绿星是原始数据点，黑圈是其重建。权重向量 w 由蓝线表示。（a）我们从 w 的随机初始猜测开始。E 步由正交投影表示。（b）我们在 M 步骤中更新杆 w，保持杆上的投影（黑色圆圈）固定。（c）另一个 E 步骤。黑色圆圈可以沿杆“滑动”，但杆保持固定。（d）另一个 M 步骤。基于（Bishop 2006b）的图 12.12。pcaEmStepByStep 生成的图。

除了这种令人愉快的直观解释外，主成分分析的 EM 与特征向量方法相比具有以下优势：

- EM 可以更快。特别是，假设 $N, D \gg L$,EM 的主要成本是 E 步骤中的投影操作，因此总时间为 $O(TLND)$, 其中 T 是迭代次数。（Roweis 1997）实验表明，无论 N 或 D 如何，迭代次数通常非常小（平均值为 3.6）。（这一结果取决于经验协方差矩阵的特征值之比。）这比直接特征向量方法所需的 $O(min(ND^2, DN^2))$ 时间快得多，尽管更复杂的特征向量方法，如 Lanczos 算法，其运行时间与 EM 相当。
- EM 可以以在线方式实现，也就是说，我们可以在数据流入时更新我们对 **W** 的估计数据流。

- EM 可以以简单的方式处理缺失数据（见第 12.1.6 节）。
- EM 可以扩展到处理 PPCA/FA 模型的混合物。
- EM 可以修改为变分 EM 或变分贝叶斯 EM，以适应更复杂的模型。

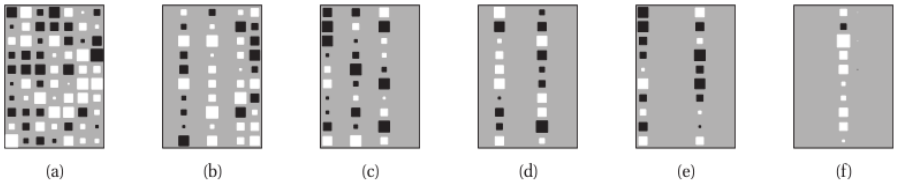


图 12.12: 使用 VBEM 估计混合因子分析仪中有效维数的图示。已通过 ARD 机制将空白列强制为 0。数据来自 6 个内部维数为 7、4、3、2、2、1 的聚类，该方法已成功估计。资料来源：图 4.4（Beal 2003）。经马特 • 比尔许可使用。

1.3 选择潜在维度的数量

在第 11.5 节中，我们讨论了如何在混合模型中选择组分 K 的数量。在本节中，我们讨论了如何在 FA/PCA 模型中选择潜在维数 L 的数量。

1.3.1 FA/PPCA 的模型选择

如果我们使用概率模型，原则上我们可以计算 $L^* = argmax_L p(L|D)$。然而，这有两个问题。首先，评估 LVMs 的边际可能性相当困难。在实践中，可以使用简单近似，例如 BIC 或变分下限（见第 21.5 节）（另请参见（Minka 2000a））。或者，我们可以使用交叉验证的可能性作为性能度量，尽管这可能很慢，因为它需要拟合每个模型 F 次，其中 F 是 CV 折叠的数量。

第二个问题是需要搜索潜在的大量模型。通常的方法是对 L 的所有候选值进行穷举搜索。然而，有时我们可以将模型设置为其最大大小，然后使用称为自动相关性确定（第 13.7 节）的技术，结合 EM，自动删除不相关的权重。该技术将在第 13 章的监督上下文中描述，但可以适用于 (M)FA 上下文，如（Bishop 1999；Ghahramani 和 Beal 2000）所示。

number of points per cluster	intrinsic dimensionalities					
	1	7	4	3	2	2
8	2				1	
8	1		2			
16	1	4				2
32	1	6	3	3	2	2
64	1	7	4	3	2	2
128	1	7	4	3	2	2

图 12.13: 显示了作为样本量函数的聚类估计数及其估计维数。当 $N = 8$ 时，VBEM 算法找到了两个不同的解。注意，随着样本量的增加，发现了更多具有更大有效维数的簇。资料来源：表 4.1（Beal 2003）。经马特 • 比尔善意许可使用。

图 12.12 说明了这种方法适用于适合小型合成数据集的混合 FA。这些图使用 **Hinton 图**可视化了每个簇的权重矩阵，其中平方的大小与矩阵[2]中条目的值成比例。我们看到其中许多是稀疏的。图 12.13 显示

[2]杰夫 • 辛顿是托伦托大学计算机科学的英语教授。

稀疏度取决于训练数据量，符合贝叶斯 Occam 剃刀。特别是，当样本量较小时，该方法自动倾向于使用更简单的模型，但当样本量足够大时，该方法收敛于“正确”解，即具有维度 1、2、2、3、4 和 7 的 6 个子空间的解。

虽然 ARD/EM 方法很优雅，但它仍然需要在 K 上执行搜索。这是使用“出生”和“死亡”移动完成的（Ghahramani 和 Beal 2000）。另一种方法是在模型空间中进行随机抽样。传统方法，例如（Lopes 和 West 2004），基于可逆跳跃 MCMC，也使用出生和死亡移动。然而，这可能很慢，很难实现。最近的方法使用非参数先验，结合吉布斯采样，例如（Paisley 和 Carin，2009）。

1.3.2 主成分分析的模型选择

由于主成分分析不是概率模型，我们不能使用上述任何方法。可能性的一个明显代表是重建误差：

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

在主成分分析的情况下，重构由 $\hat{\mathbf{x}}_i = \mathbf{W}\mathbf{z}_i + \boldsymbol{\mu}$, 其中,$\mathbf{z}_i = \mathbf{W}^T(\mathbf{x}_i - \boldsymbol{\mu})$,$\mathbf{W}$ 和 $\boldsymbol{\mu}$ 由 $\mathcal{D}_{train}$ 估算。

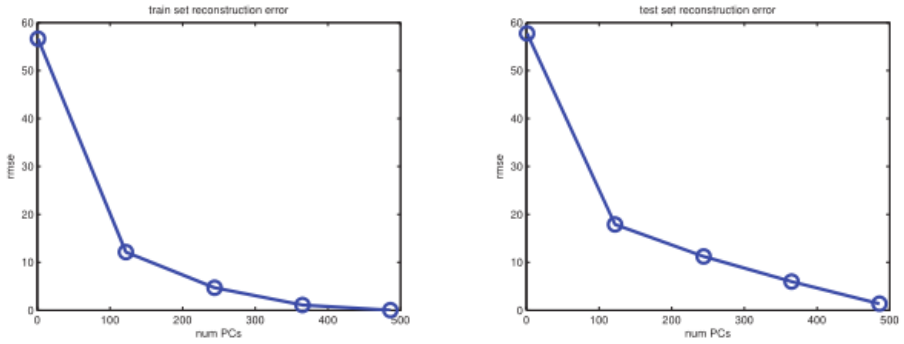


图 12.14: MNIST 的重建误差与主成分分析使用的潜在维数的数量。(a) 训练集。(b) 测试集。图由 pcaOverfitDemo 生成。

图 12.14（a）绘制了图 12.6 中 MNIST 训练数据的 $E(\mathcal{D}_{train}, L)$ 与 L。我们看到它下降得很快，这表明我们可以用少量因子捕捉像素的大部分经验相关性，如图 12.6 定性所示。

练习 12.5 要求您证明仅使用 L 项的残余误差由丢弃的特征值之和得出：

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

因此，替代绘制误差的方法是绘制保留的特征值，以递减的方式顺序。这被称为 **scree 图**，因为” 该图看起来像一座山的侧面，而’scree’ 是指从山上掉下来的、躺在山脚下的碎石。指的是从山上掉下来的、躺在山脚下的碎片”。[3]形状与残余误差图相同。

相关量是解释的**方差分数**，定义为:

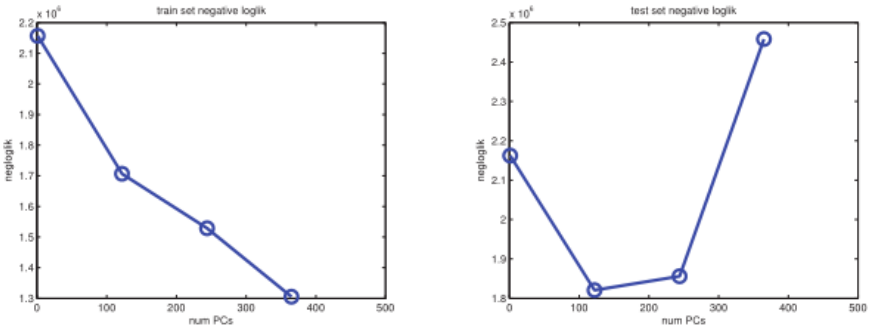
$$F(\mathcal{D}_{train}, L) = \frac{\sum_{j=1}^{L} \lambda_j}{\sum_{j'=1}^{L_{max}} \lambda_{j'}} \tag{12.69}$$

这与 scree 绘图捕获的信息相同。

当然，如果我们使用 L= 秩（$\mathbf{X}$），我们在训练集上得到零重建误差。为了避免过度拟合，在测试集上绘制重建误差是很自然的。如图 12.14（b）所示。在这里，我们看到，即使模型变得更复杂，误差仍在继续下降！因此，我们没有得到通常我们期望看到的 U 形曲线。

[3]来自：http://janda.org/workshop/factoranalysis/SPSSrun/SPSS08.htm.

怎么回事？问题是主成分分析不是一个合适的数据生成模型。这只是一种压缩技术。如果你给它更多的潜在维度，它将能够更准确地近似测试数据。相比之下，概率模型具有贝叶斯-奥卡姆剃刀效应（第 5.3.1 节），即如果在数据较少的空间部分浪费概率质量，则会受到“惩罚”。这如图 12.15 所示，图中绘制了使用 PPCA 计算的负对数似然与 L。在这里，在测试集上，我们看到了常见的 U 形曲线。



(a) (b)

图 12.15:MNIST 的负对数似然与 PPCA 使用的潜在维度数。(a) 训练集。(b) 测试集。图由 pcaOverfitDemo 生成。

这些结果与第 11.5.2 节中的结果类似，在第 11.5.2 节中，我们讨论了在 K-means 算法中选择 K 与使用 GMM 的问题。

12.3.2.1 Profile likelihood-轮廓可能性

虽然没有 U 型曲线，但图中有时会出现“制度更迭”，从相对较大的误差到相对较小的误差。一种自动检测这种情况的方法在（zhu 和 Ghodsi 2006）中描述。想法是这样的。设 λ_k 是大小为 k 的模型产生的误差的某种度量，使得 $\lambda_1 \geq \lambda_2 \geq ... \geq \lambda_{L_{max}}$。在主成分分析中，这些是特征值，但该方法也可以应用于 K 均值。现在考虑将这些值分为两组，这取决于 $k < L$ 还是 $k > L$，我们将确定一个阈值。为了测量 L 的质量，我们将使用一个简单的变点模型，其中如果 $k \leq L$，$\lambda_k \sim \mathcal{N}(\mu_2, \sigma^2)$ 如果 $k > L$。（重要的是 σ^2 在两个模型中相同，以防止在一个区域的数据少于另一个区域的情况下过度拟合。）在这两个区域中，我们假设 λ_k 是 iid，这显然是不正确的，但足以满足我们目前的目的。我们可以通过对数据进行分区并计算最大似然估计（MLE），使用方差的汇总估计，为每个 $L = 1 : L_{max}$ 拟合该模型：

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

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

然后，我们可以评估**剖面对数的可能性**.

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

最后，我们选择 $L^* = arg\ max\ \ell(L)$。如图 12.16 所示。在左边，我们绘制了树状图，其形状与图 12.14（a）中的形状相同。在右边，我们绘制了剖面图。

1.4 分类数据的 PCA

在本节中，我们考虑将因子分析模型扩展到观察数据为分类数据而非实值数据的情况。也就是说，数据的形式为 $y_{ij} \in \{1, ..., C\}$，其中 $j = 1 : R$ 是观察到的响应变量的数量。我们假设每个 y_{ij} 由一个潜在变

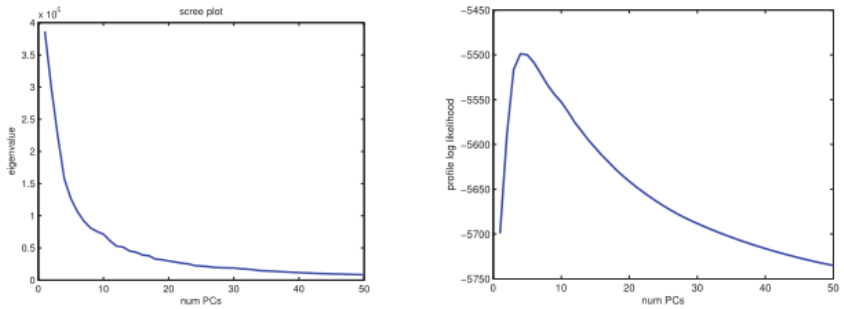


图 12.16: (a) 对应于图 12.14(a) 的训练集的屏幕图。(b) 轮廓可能性。图由 pcaOverfitDemo 生成。

量 $\mathbf{z}_i \in \mathbb{R}^L$ 生成，具有高斯先验，其通过 softmax 函数如下：

$$p(\mathbf{z}_i) = \mathcal{N}(0, \mathbf{I}) \tag{12.73}$$

$$p(\mathbf{y}_i|\mathbf{z}_i, \boldsymbol{\theta}) = \prod_{r=1}^{R} Cat(y_{ir}|S(\mathbf{W}_r^T \mathbf{z}_i + \mathbf{w}_{0r})) \tag{12.74}$$

其中 $\mathbf{W}_r \in \mathbb{R}^{L \times M}$ 是响应 j 的因子载荷矩阵,$\mathbf{w}_{0r} \in \mathbb{R}^R$ 是响应 r 的偏移项和 $\boldsymbol{\theta} = (\mathbf{W}_r, \mathbf{w}_{0r})_{r=1}^R$。（我们需要一个明确的偏移项，因为将 $\mathbf{z}_i$ 的一个元素设置为 1 可能会在计算后验协方差时引起问题。）与因子分析一样，我们将先验均值定义为 $\mathbf{m_0}$，先验协方差 $\mathbf{V}_0 = \mathbf{I}$，因为我们可以通过改变 w_{0j} 来捕捉非零均值，通过改变 $\mathbf{W}_r$ 来捕捉非同一协方差。我们将其称为**分类主成分分析**。有关相关模型的讨论，请参阅第 27 章。

有趣的是，研究通过改变参数，我们可以在观察变量上归纳出什么样的分布。为简单起见，我们假设存在单个三元响应变量，因此 y_i 存在在三维概率单纯形中。图 12.17 显示了当我们改变先验参数 $\mathbf{m}_0$ 和 $\mathbf{V}_0$ 时会发生什么，这相当于改变似然参数 $\mathbf{W}_1$ 和 $\mathbf{w}_{01}$。我们看到，这可以定义单纯形上相当复杂的分布。这种诱导分布称为**逻辑正态分布**（Aitchison 1982）。

我们可以使用 EM 的改进版本将该模型拟合到数据中。基本思想是在 E 步中推断后验 $p(\mathbf{z}_i|\mathbf{y}_i, \boldsymbol{\theta})$ 的高斯近似，然后在 M 步中最大化 $\boldsymbol{\theta}$。多类案例的详细信息见（Khan 等人，2010）（另见第 21.8.1.1 节）。sigmoid 链路的二进制情况的详细信息可以在练习 21.9 中找到，probit 链路的详细信息可以在练习 21.10 中找到。

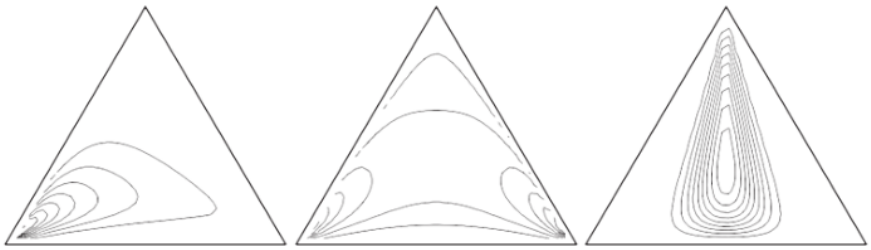


图 12.17: 在三维单线上定义的逻辑正态分布的一些示例。(a) 对角协方差和非零均值。(b) 状态 1 和 2 之间的负相关性。(c) 状态 1 和 2 之间正相关。资料来源：图 1（Blei 和 Lafferty 2007）。经 David Blei 善意许可使用。

这种模型的一个应用是可视化高维分类数据。图 12.18(a) 显示了一个简单的示例，其中我们有 150 个 6 维位向量。很明显，每个样本只是三个二进制原型之一的嘈杂副本。我们将 2d catFA 拟合到该模型，得到近似的 MLEs$\hat{\boldsymbol{\theta}}$。在图 12.18（b）中，我们绘制了 $\mathbb{E}[\mathbf{z}_i|\mathbf{x}_i, \hat{\boldsymbol{\theta}}]$。正如预期的那样，我们看到有三个不同的集群。

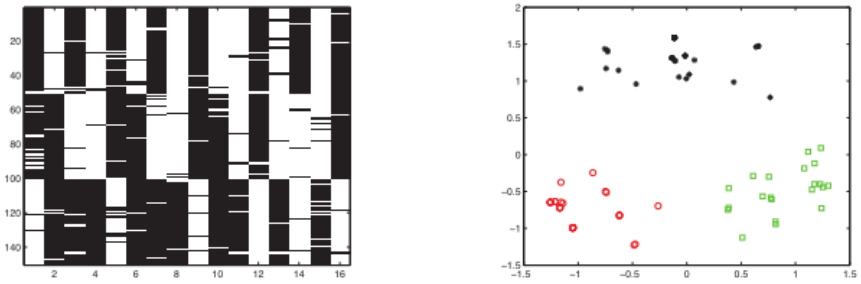


图 12.18: 左：150 个合成 16 维位向量。右图：通过二进制 PCA 学习的 2d 嵌入，使用变分 EM。我们根据生成它们的真实“原型”的身份获得颜色编码点。由 binaryFaDemoTipping 生成的图。

在（Khan 等人，2010 年）中，我们表明，该模型在填充由真实数据和分类数据组成的设计矩阵中的缺失项方面优于有限混合模型。这有助于分析社会科学调查数据，因为这些数据往往缺少数据和混合类型的变量。

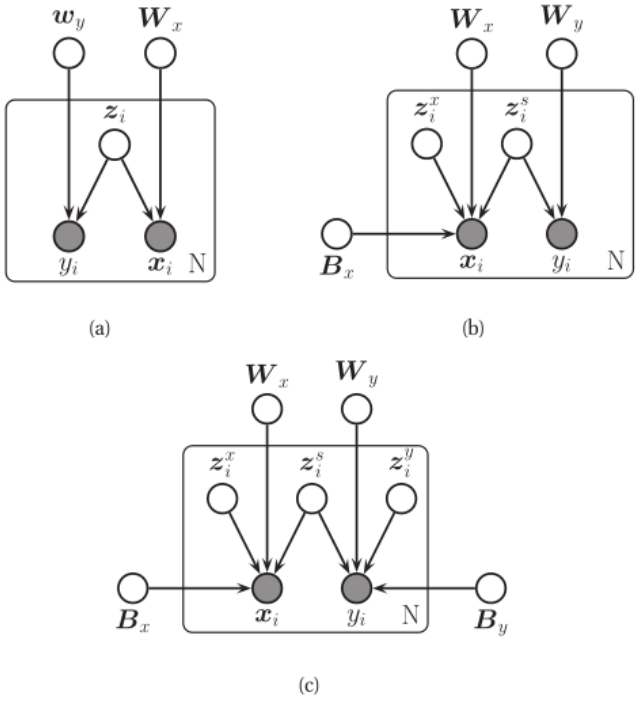


图 12.19: 成对数据的高斯潜在因子模型。(a) 有监督的 PCA。(b) 偏最小二乘法。(c) 典范相关分析。

1.5 成对和多视图数据的 PCA

通常会有一对相关数据集，例如基因表达和基因拷贝数，或用户的电影分级和电影评论。很自然地，我们希望将这些结合在一起，形成一个低维嵌入。这是数据融合的一个例子。在某些情况下，我们可能希望通过低维“瓶颈”从另一个元素 $\mathbf{x}_{i2}$ 预测配对中的一个元素，例如 $\mathbf{x}_{i1}$。

下面，我们将在 (Virtanen 2010) 的介绍之后讨论这些任务的各种潜在高斯模型。对于 $m = 1 : M$，模型很容易从成对推广到数据集 $\mathbf{x}_{im}$，。我们重点讨论了 $\mathbf{x}_{im} \in \mathbb{R}^{D_m}$ 的情况。在这种情况下，联合分布是多元高斯分布，因此我们可以使用 EM 或 Gibbs 采样拟合模型。

正如我们在第 27.2.2 节中所解释的那样，我们可以通过使用指数族作为响应分布而不是高斯分布来概括模型，以处理离散和计数数据。然而，这将需要在 E 步骤中使用近似推理（或对 MCMC 的类似修改）。

1.5.1 监督 PCA（潜在因子回归）

考虑以下模型，如图 12.19（a）所示：

$$p(\mathbf{z}_i) = \mathcal{N}(\mathbf{0}, \mathbf{I}_L) \tag{12.75}$$

$$p(y_i|\mathbf{z}_i) = \mathcal{N}(\mathbf{w}_y^T \mathbf{z}_i + \mu_y, \sigma_y^2) \tag{12.76}$$

$$p(\mathbf{x}_i|\mathbf{z}_i) = \mathcal{N}(\mathbf{W}_x \mathbf{z}_i + \boldsymbol{\mu}_x, \sigma_x^2 \mathbf{I}_D) \tag{12.77}$$

在（Yu 等人，2006 年）中，这被称为**监督 PCA**。在（West 2003）中，这被称为**贝叶斯因子回归**。该模型类似于主成分分析，只是在学习低维嵌入时考虑了目标变量 y_i。由于该模型是联合高斯模型，我们有:

$$y_i|\mathbf{x}_i \sim \mathcal{N}(\mathbf{x}_i^T \mathbf{w}, \sigma_y^2 + \mathbf{w}_y^T \mathbf{C} \mathbf{w}_y) \tag{12.78}$$

其中 $\mathbf{w} = \boldsymbol{\Psi}^{-1} \mathbf{W}_x \mathbf{C} \mathbf{w}_y, \boldsymbol{\Psi} = \sigma_x^2 \mathbf{I}_D$, 和 $\mathbf{C}^{-1} = \mathbf{I} + \mathbf{W}_x^T \boldsymbol{\Psi}^{-1} \mathbf{W}_x$。因此，虽然这是 $(y_i, \mathbf{x}_i)$ 的联合密度模型，但我们可以推断出隐含的条件分布。

我们现在展示了一个与 Zellner 先验的有趣联系。假设 $p(\mathbf{w}_y) = \mathcal{N}(\mathbf{0}, \frac{1}{g} \sum^2)$，并且假设 $\mathbf{X} = \mathbf{R}\mathbf{V}^T$ 是 $\mathbf{X}$ 的奇异值分解，其中 $\mathbf{V}^T \mathbf{V} = \mathbf{I}$ 和 $\mathbf{R}^T \mathbf{R} = \sum^2 = diag(\sigma_j^2)$ 包含平方奇异值。然后可以证明（West 2003）

$$p(\mathbf{w}) = \mathcal{N}(\mathbf{0}, g\mathbf{V}^{-T} \sum\nolimits^{-\mathbf{2}} \mathbf{V}^{-1}) = \mathcal{N}(\mathbf{0}, g(\mathbf{X}^T \mathbf{X})^{-1}) \tag{12.79}$$

因此，$\mathbf{w}$ 对 $\mathbf{X}$ 的先验依赖性来自于这样一个事实，即 $\mathbf{w}$ 是通过 $\mathbf{X}$ 和 $\mathbf{y}$ 的联合模型间接推导出来的。

上述讨论集中于回归。（Guo 2009）将 CCA 推广到指数族，如果 $\mathbf{x}_i$ 和 y_i 是离散的，则更合适。虽然我们无法再以闭合形式计算条件 $p(y_i|\mathbf{x}_i, \theta)$，但该模型具有与回归情况类似的解释，即我们通过潜在的“瓶颈”预测响应。

压缩 $\mathbf{x}_i$ 来预测 $\mathbf{y}_i$ 的基本思想可以用信息论来表述。特别是，我们可能想找到一个编码分布 $p(\mathbf{z}|\mathbf{x})$，使我们最小化

$$\mathbb{I}(\mathbf{X}; \mathbf{Z}) - \beta \mathbb{I}(\mathbf{X}; \mathbf{Y}) \tag{12.80}$$

其中 $\beta \geq 0$ 是控制压缩和预测精度之间权衡的参数。这被称为**信息瓶颈**（Tishby 等人，1999 年）。通常 Z 被认为是离散的，如在聚类中。然而，在高斯情况下，IB 与 CCA 密切相关（Chechik 等人，2005 年）。

我们可以很容易地将 CCA 推广到 y_i 是要预测的反应的向量的情况，如在多标签分类中。(Ma 等人，2008；Williamson 和 Ghahramani，2008) 用这个模型来进行协作过滤，目标是预测 $y_{ij} \in \{1, ..., 5\}$，即人的评分 i 对电影 j 的评价，其中的” 侧面信息”x_i 采取了 i 的朋友名单的形式。这种方法背后的直觉是这种方法背后的直觉是，了解你的朋友是谁，以及所有其他用户的评级以及所有其他用户的评分，应该有助于预测你会喜欢哪些电影。一般来说，任何任务相关的环境任务是相关的，都可以从 CCA 中受益。一旦我们采用了概率论的观点，各种扩展是直接的。例如，我们可以很容易地推广到半监督式的的情况下，我们不能对所有的 i 都观察到 $\mathbf{y}_i$（Yu 等人，2006）。

12.5.1.1 判别监督主成分分析

该模型的一个问题是，它在预测输入 $\mathbf{x}_i$ 和输出 $\mathbf{y}_i$ 上的权重相同。这可以通过使用以下形式的加权目标来部分缓解（Rish 等人，2008 年）：

$$\ell(\boldsymbol{\theta}) = \prod_i p(\mathbf{y}_i|\boldsymbol{\eta}_{iy})^{\alpha_y} p(\mathbf{x}_i|\boldsymbol{\eta}_{ix})^{\alpha_x} \tag{12.81}$$

其中 α_m 控制数据源的相对重要性，$\boldsymbol{\eta}_{im} = \mathbf{W}_m \mathbf{z}_i$。对于高斯数据，我们可以看到 α_m 仅控制噪声方差：

$$\ell(\boldsymbol{\theta}) \propto \prod_i exp(-\frac{1}{2}\alpha_x ||\mathbf{x}_i^T - \boldsymbol{\eta}_{ix}||^2) exp(-\frac{1}{2}\alpha_y ||\mathbf{y}_i^T - \boldsymbol{\eta}_{iy}||^2) \tag{12.82}$$

这种解释更普遍地适用于指数族。然而，请注意，很难估计 α_m 参数，因为改变它们会改变似然的归一化常数。我们给出了一种替代方法，以在下面更重地加权 $\mathbf{y}$。

1.5.2 偏最小二乘法

偏最小二乘法（PLS）(Gustafsson 2001；Sun 等人，2009）是监督主成分分析的一种不对称或更具“判别性”的形式。关键思想是允许输入特征中的一些（co）方差由其自身的子空间 $\mathbf{z}_i^x$ 解释，并允许子空间 $\mathbf{z}_i^s$ 的其余部分在输入和输出之间共享。模型具有以下形式:

$$
\begin{aligned}
p(\mathbf{z}_i) &= \mathcal{N}(\mathbf{z}_i^s|\mathbf{0},\mathbf{I}_{L_s})\mathcal{N}(\mathbf{z}_i^x|\mathbf{0},\mathbf{I}_{L_x}) && (12.83)\\
p(\mathbf{y}_i|\mathbf{z}_i) &= \mathcal{N}(\mathbf{W}_y\mathbf{z}_i^s+\boldsymbol{\mu}_y,\sigma^2\mathbf{I}_{D_y}) && (12.84)\\
p(\mathbf{x}_i|\mathbf{z}_i) &= \mathcal{N}(\mathbf{W}_x\mathbf{z}_i^s+\mathbf{B}_x\mathbf{z}_i^x+\boldsymbol{\mu}_x,\sigma^2\mathbf{I}_{D_x}) && (12.85)
\end{aligned}
$$

见图 12.19（b）。可见变量上的相应诱导分布具有以下形式：

$$
p(\mathbf{v}_i|\boldsymbol{\theta})=\int \mathcal{N}(\mathbf{v}_i|\mathbf{W}\mathbf{z}_i+\boldsymbol{\mu},\sigma^2\mathbf{I})\mathcal{N}(\mathbf{z}_i|\mathbf{0},\mathbf{I})d\mathbf{z}_i=\mathcal{N}(\mathbf{v}_i|\boldsymbol{\mu},\mathbf{W}\mathbf{W}^T+\sigma^2\mathbf{I}) \tag{12.86}
$$

其中 $\mathbf{v}_i=(\mathbf{x}_i;\mathbf{y}_i),\boldsymbol{\mu}=(\boldsymbol{\mu}_y;\boldsymbol{\mu}_x)$ 和

$$
\begin{aligned}
\mathbf{W} &= \begin{pmatrix}\mathbf{W}_y & 0\\ \mathbf{W}_x & \mathbf{B}_x\end{pmatrix} && (12.87)\\
\mathbf{W}\mathbf{W}^T &= \begin{pmatrix}\mathbf{W}_y\mathbf{W}_y^T & \mathbf{W}_x\mathbf{W}_x^T\\ \mathbf{W}_x\mathbf{W}_x^T & \mathbf{W}_x\mathbf{W}_x^T+\mathbf{B}_x\mathbf{B}_x^T\end{pmatrix} && (12.88)
\end{aligned}
$$

我们应该选择足够大的 L，以使共享子空间不捕捉协变量特异性变化。

使用指数族可以很容易地将该模型推广到离散数据（Virtanen 2010）。

1.5.3 典型相关分析

典型相关分析（CCA）类似于 PLS 的对称无监督版本：它允许每个视图都有自己的“私有”子空间，但也有一个共享子空间。如果我们有两个观察变量，$\mathbf{x}_i$ 和 $\mathbf{y}_i$，那么我们有三个潜在变量,$\mathbf{z}_i^s\in\mathbb{R}^{L_0}$ 是共享的，$\mathbf{z}_i^x\in\mathbb{R}^{L_x}$ 和 $\mathbf{z}_i^y\in\mathbb{R}^{L_y}$ 这是私人的。我们可以将模型编写如下（Bach 和 Jordan 2005）：

$$
\begin{aligned}
p(\mathbf{z}_i) &= \mathcal{N}(\mathbf{z}_i^s|\mathbf{0},\mathbf{I}_{L_s})\mathcal{N}(\mathbf{z}_i^x|\mathbf{0},\mathbf{I}_{L_x})\mathcal{N}(\mathbf{z}_i^y|\mathbf{0},\mathbf{I}_{L_y}) && (12.89)\\
p(\mathbf{x}_i|\mathbf{z}_i) &= \mathcal{N}(\mathbf{x}_i|\mathbf{B}_x\mathbf{z}_i^x+\mathbf{W}_x\mathbf{z}_i^s+\boldsymbol{\mu}_x,\sigma^2\mathbf{I}_{D_x}) && (12.90)\\
p(\mathbf{y}_i|\mathbf{z}_i) &= \mathcal{N}(\mathbf{y}_i|\mathbf{B}_y\mathbf{z}_i^y+\mathbf{W}_y\mathbf{z}_i^s+\boldsymbol{\mu}_y,\sigma^2\mathbf{I}_{D_y}) && (12.91)
\end{aligned}
$$

见图 12.19（c）。相应的观察到的联合分布具有以下形式

$$
p(\mathbf{v}_i|\boldsymbol{\theta})=\int \mathcal{N}(\mathbf{v}_i|\mathbf{W}\mathbf{z}_i+\boldsymbol{\mu},\sigma^2\mathbf{I})\mathcal{N}(\mathbf{z}_i|\mathbf{0},\mathbf{I})d\mathbf{z}_i=\mathcal{N}(\mathbf{v}_i|\boldsymbol{\mu},\mathbf{W}\mathbf{W}^T+\sigma^2\mathbf{I}_D) \tag{12.92}
$$

其中:

$$
\begin{aligned}
\mathbf{W} &= \begin{pmatrix}\mathbf{W}_x & \mathbf{B}_x & 0\\ \mathbf{W}_y & 0 & \mathbf{B}_y\end{pmatrix} && (12.93)\\
\mathbf{W}\mathbf{W}^T &= \begin{pmatrix}\mathbf{W}_x\mathbf{W}_x^T+\mathbf{B}_x\mathbf{B}_x^T & \mathbf{W}_x\mathbf{W}_y^T\\ \mathbf{W}_y\mathbf{W}_y^T & \mathbf{W}_y\mathbf{W}_y^T+\mathbf{B}_y\mathbf{B}_y^T\end{pmatrix} && (12.94)
\end{aligned}
$$

可以使用 EM 计算该模型的最大似然估计。（Bach 和 Jordan 2005）表明，得到的最大似然估计等价于（旋转和缩放）经典的非概率视图。然而，概率观点的优点很多：我们可以简单地推广到 $M>2$ 个观察变量；我们可以创建 CCA 的混合物（Viinikanoja 等人，2010 年）；我们可以使用 ARD 创建 CCA 的稀疏版本（Archambeau 和 Bach 2008）; 我们可以推广到指数族（Klami 等人，2010）; 我们可以对参数进行贝叶斯推断（WQang2007；Klami 和 Kaski 2008）；我们可以处理 $\mathbf{W}$ 和 $\mathbf{B}$ 的非参数稀疏性提升先验（Rai 和 Daume 2009）；等等。

1.6 独立分量分析（ICA）

考虑以下情况。你在一个拥挤的房间里，很多人在说话。我们的耳朵基本上就像两个麦克风，它们在听房间里不同语音信号的线性组合。我们的目标是将混合信号分解为其组成部分。这被称为**鸡尾酒会问题**，是**盲信号分离（BSS）或盲源分离**的一个示例，其中“盲”表示我们对信号源“一无所知”。除了在声学信号处理中的明显应用外，在分析 EEG 和 MEG 信号、财务数据和任何其他潜在源或因素以线性方式混合在一起的数据集（不一定是暂时的）时，也会出现这个问题。

我们可以将问题形式化如下。让 $\mathbf{x}_t \in \mathbb{R}^D$ 是“时间”t 时传感器上的观察信号,$\mathbf{z}_t \in \mathbb{R}^L$ 是源信号的矢量。我们假设

$$\mathbf{x}_t = \mathbf{W}\mathbf{z}_t + \epsilon_t \tag{12.95}$$

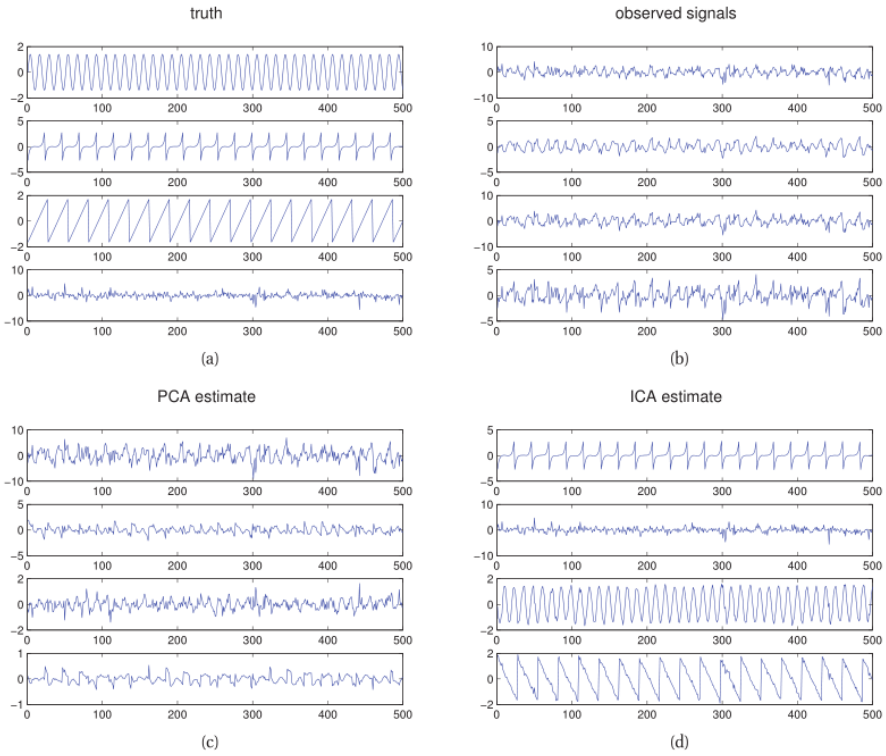


图 12.20: ICA 应用于 4d 源信号的 500 个 iid 样本的图示。(a) 潜在信号。(b) 观察结果。(c) 主成分分析估计。(d) ICA 估计。图由 icaDemo 生成，由 Aapo Hyvarinen 编写。

其中 $\mathbf{W}$ 是 $D \times L$ 矩阵，并且 $\epsilon_t \sim \mathcal{N}(\mathbf{0}, \boldsymbol{\Psi})$。在本节中，我们将每个时间点视为一个独立的观察值，即我们不建模时间相关性（因此我们可以用 i 代替 t 指数，但我们坚持用 t 与许多独立分量分析文献一致）。目标是推断源信号 $p(\mathbf{z}_t|\mathbf{x}_t, \boldsymbol{\theta})$，如图 12.20 所示。在这种情况下，$\mathbf{W}$ 被称为**混合矩阵**。如果 $L = D$（信源数量 = 传感器数量），则它将是一个方阵。为了简单起见，我们通常假设噪声级 $|\boldsymbol{\Psi}|$ 为零。

到目前为止，该模型与因子分析（如果没有噪声，则为 PCA）相同，但我们通常不需要 $\mathbf{W}$ 的正交性。然而，我们将对 $p(\mathbf{z}_t)$ 使用不同的先验。在主成分分析中，我们假设每个信源是独立的，并且具有高斯分布

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

现在，我们将放松这个高斯假设，并让源分布为任何非高斯分布

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

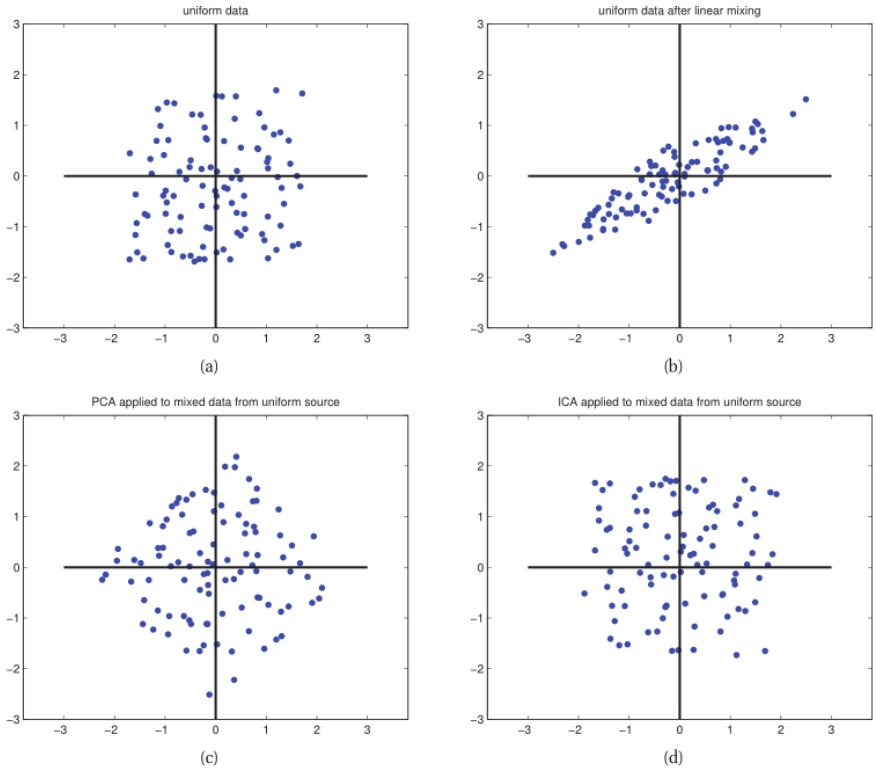


图 12.21: 对均匀分布的 2d 源信号的 100 个 iid 样本应用独立分量分析和主成分分析的图示。（a）潜在信号。（b）观察结果。（c）主成分分析估计。（d）ICA 估计。图由 icaDemoUniform 生成，由 Aapo Hyvarinen 编写。

在不丧失一般性的情况下，我们可以将源分布的方差限制为 1，因为任何其他方差都可以通过适当地缩放 $\mathbf{W}$ 的来建模。由此产生的模型称为**独立分量分析（ICA）**。

高斯分布在独立分量分析中不允许作为信源先验的原因是，它不允许信源的唯一恢复，如图 12.20（c）所示。这是因为 PCA 似然对源 $\mathbf{z}_t$ 和混合矩阵 $\mathbf{W}$ 的任何正交变换都是不变的。PCA 可以恢复信号所在的最佳线性子空间，但不能唯一地恢复信号本身。

为了说明这一点，假设我们有两个均匀分布的独立源，如图 12.21（a）所示。现在假设我们有以下混合矩阵

$$\mathbf{W} = \begin{pmatrix} 2 & 3 \\ 2 & 1 \end{pmatrix} \tag{12.98}$$

然后，我们观察图 12.21（b）中所示的数据（假设没有噪声）。如果我们应用 PCA，然后对其进行缩放，我们得到的结果如图 12.21（c）所示。这对应于数据的白化。为了唯一地恢复信源，我们需要执行额外的旋转。问题是，对称高斯后验中没有信息告诉我们旋转的角度。在某种意义上，主成分分析解决了问题的“一半”，因为它识别了线性子空间；ICA 所要做的就是识别适当的旋转。（因此，我们发现独立分量分析与 varimax 等方法没有太大不同，后者寻求潜在因子的良好旋转，以增强可解释性。）

图 12.21（d）表明，独立分量分析可以恢复信源，直至索引排列和可能的符号变化。独立分量分析要求 $\mathbf{W}$ 是平方的，因此是可逆的。在非平方情况下（例如，我们的信源比传感器多），我们无法唯一地恢复真实信号，但我们可以计算后验 $p(\mathbf{z}_t|\mathbf{x}_t, \hat{W})$，它表示我们对信源的信任。在这两种情况下，我们需要估计 $\mathbf{W}$ 以及源分布 p_j。我们在下面讨论如何做到这一点。

1.6.1 最大似然估计

在本节中，我们讨论了无噪声 ICA 模型的平方混合矩阵 $\mathbf{W}$ 的估计方法。像往常一样，我们假设观察结果已经居中；因此，我们也可以假设 $\mathbf{z}$ 是零均值。此外，我们假设观察值经过白化处理，这可以通过主成

分分析完成。

如果数据居中并白化处理，我们有 $\mathbb{E}[\mathbf{x}\mathbf{x}^T] = \mathbf{I}$。但在无噪声的情况下，我们也有

$$cov[\mathbf{x}] = \mathbb{E}[\mathbf{x}\mathbf{x}^T] = \mathbf{W}\mathbb{E}[\mathbf{z}\mathbf{z}^T]\mathbf{W}^T = \mathbf{W}\mathbf{W}^T \tag{12.99}$$

因此我们看到 $\mathbf{W}$ 必须是正交的。这将我们必须估计的参数数量从 D^2 减少到 $D(D-1)/2$. 它还将简化数学和算法。

设 $\mathbf{V} = \mathbf{W}^{-1}$, 这些通常被称为**识别权重**，而 $\mathbf{W}$ 是**生成权重**[4]。

由于 $\mathbf{x} = \mathbf{W}\mathbf{z}$，我们从等式 2.89 中得到:

$$p_x(\mathbf{W}\mathbf{z}_t) = p_z(\mathbf{z}_t)|det(\mathbf{W}^{-1})| = p_z(\mathbf{V}\mathbf{x}_t)|det(\mathbf{V})| \tag{12.100}$$

因此，假设 T iid 样本，我们可以写出对数似然，如下所示：

$$\frac{1}{T}\log p(\mathcal{D}|\mathbf{V}) = \log|det(\mathbf{V})| + \frac{1}{T}\sum_{j=1}^{L}\sum_{t=1}^{T}\log p_j(\mathbf{v}_j^T\mathbf{x}_t) \tag{12.101}$$

其中，$\mathbf{v}_j$ 是 $\mathbf{V}$ 的第 j 行。由于我们将 $\mathbf{V}$ 约束为正交，第一项是常数，因此我们可以去掉它。我们还可以用期望值替换数据的平均值，以实现以下目标

$$NLL(\mathbf{V}) = \sum_{j=1}^{L}\mathbb{E}[\mathbf{G}_j(\mathbf{z}_j)] \tag{12.102}$$

其中 $z_j = \mathbf{v}_j^T\mathbf{x}$ 和 $G_j(z) \triangleq -\log p_j(z)$。我们希望最小化这个主题，使 $\mathbf{V}$ 的行正交。我们还希望它们是单位范数，因为这确保了因子的方差是统一的（因为，对于白化数据，$\mathbb{E}[\mathbf{v}_j^T\mathbf{x}] = \| \mathbf{v}_j \|^2$，这是固定权重比例所必需的。换句话说，$\mathbf{V}$ 应该是正交矩阵。

推导出适合该模型的梯度下降算法很简单；然而，这相当缓慢。我们还可以推导出一种遵循自然梯度的更快算法；详见（MacKay 2003，第 34 章）。一种流行的替代方法是使用近似牛顿法，我们在第 12.6.2 节中讨论了该方法。另一种方法是使用 EM，我们将在第 12.6.3 节中讨论。

1.6.2 FastICA 算法

现在，我们描述了基于（Hyvarinen 和 Oja 2000）的**快速 ICA 算法**，我们将展示用于拟合 ICA 模型的近似牛顿法。

为了简单起见，我们最初假设只有一个潜在因素。此外，我们最初假设所有信源分布都是已知的并且是相同的，因此我们可以只写 $G(z) = -\log p(z)$。设 $g(z) = \frac{d}{dz}G(z)$ 约束目标及其梯度和 Hessian 由下式给出：

$$f(\mathbf{v}) = \mathbb{E}[G(\mathbf{v}^T\mathbf{x})] + \lambda(1 - \mathbf{v}^T\mathbf{v}) \tag{12.103}$$

$$\nabla f(\mathbf{v}) = \mathbb{E}[\mathbf{x}g(\mathbf{v}^T\mathbf{x})] - \beta\mathbf{v} \tag{12.104}$$

$$\mathbf{H}(\mathbf{v}) = \mathbb{E}[\mathbf{x}\mathbf{x}^T g'(\mathbf{v}^T\mathbf{x})] - \beta\mathbf{I} \tag{12.105}$$

其中 $\beta = 2\lambda$ 是拉格朗日乘子。让我们进行近似

$$\mathbb{E}[\mathbf{x}\mathbf{x}^T g'(\mathbf{v}^T\mathbf{x})] \approx \mathbb{E}[\mathbf{x}\mathbf{x}^T]\mathbb{E}[g'(\mathbf{v}^T\mathbf{x})] = \mathbb{E}[g'(\mathbf{v}^T\mathbf{x})] \tag{12.106}$$

这使得 Hessian 函数非常容易反转，从而产生以下牛顿更新：

$$\mathbf{v}^* \triangleq \mathbf{v} - \frac{\mathbb{E}[\mathbf{x}g(\mathbf{v}^T\mathbf{x})] - \beta\mathbf{v}}{\mathbb{E}[g'(\mathbf{v}^T\mathbf{x})] - \beta} \tag{12.107}$$

[4]在文献中，通常用 $\mathbf{A}$ 表示生成权重，用 $\mathbf{W}$ 表示识别权重，但我们试图与本章前面使用的符号保持一致。

可以用以下方式重写

$$\mathbf{v}^* \triangleq \mathbb{E}[\mathbf{x}g(\mathbf{v}^T\mathbf{x})] - \mathbb{E}[g'(\mathbf{v}^T\mathbf{x})]\mathbf{v} \tag{12.108}$$

（在实践中，可以用训练集的 Monte Carlo 估计值代替期望值，这提供了一种有效的在线学习算法。）执行此更新后，应使用

$$\mathbf{v}^{new} \triangleq \frac{\mathbf{v}^*}{||\mathbf{v}^*||} \tag{12.109}$$

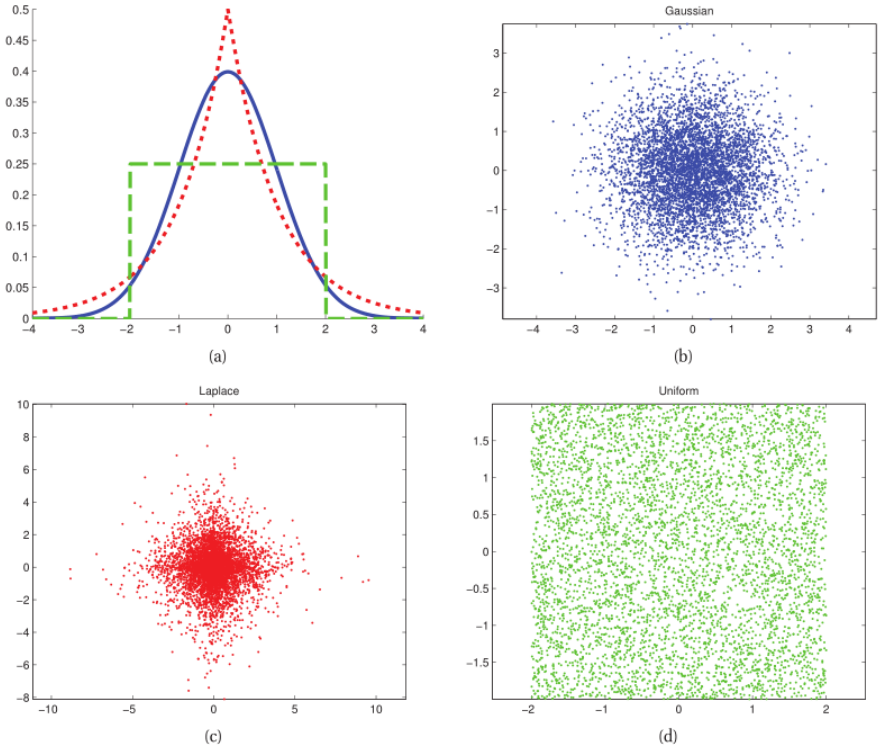


图 12.22: 1d 和 2d 中高斯、亚高斯（均匀）和超高斯（拉普拉斯）分布的图示。图由凯文 • 斯沃斯基（Kevin-Swersky）编写的 subSuperGaussPlot 生成。

迭代该算法直到收敛。（由于 v 的符号模糊性，$\mathbf{v}$ 的值可能不会收敛，但该向量定义的方向应收敛，因此可以通过监测 $|\mathbf{v}^T\mathbf{v}^{new}|$ 来评估收敛性，该值应接近 1。）

由于目标不是凸的，因此存在多个局部最优解。我们可以利用这个结果来学习多个不同的权重向量或**特征**。我们可以依次学习特征，然后投影出位于由早期特征定义的子空间中的 $\mathbf{v}_j$ 部分，或者我们可以并行学习它们，并行正交化 $\mathbf{V}$。后一种方法通常是首选的，因为与 PCA 不同，特征不以任何方式排序。因此，第一个特征并不比第二个特征“更重要”，因此最好对称地处理它们。

12.6.2.1 模拟源密度

到目前为止，我们假设 $G(z) = -\log p(z)$ 是已知的。什么样的模型可能是合理的信号先验？我们知道使用高斯（对应于 G 的二次函数）是行不通的。所以我们需要某种非高斯分布。通常，有几种非高斯分布，例如：

- **超高斯分布**这些分布在平均值处有一个大尖峰，因此（为了确保单位方差）有重尾。拉普拉斯分布是一个典型的例子。见图 12.22。形式上，如果 $kurt(z) > 0$，我们称分布为**超高斯或 leptokurtic**（“lepto”来自希腊语，表示“瘦”），其中 kurt（z）是分布的**峰度**，定义如下：

$$kurt(z) \triangleq \frac{\mu_4}{\sigma^4} - 3 \tag{12.110}$$

其中 σ 是标准差，μ_k 是第 k 个中心力矩，或关于平均值的力矩：

$$\mu_k \triangleq \mathbb{E}[(\mathbf{X} - \mathbb{E}[\mathbf{X}])^k] \tag{12.111}$$

（因此 $\mu_1 = \mu$ 是平均值，$\mu_2 = \sigma^2$ 是方差。）通常在峰度定义中减去 3，使高斯变量的峰度等于零。

- **亚高斯分布亚高斯或平谷分布**（“平谷”来自希腊语，表示“宽”）具有负峰度。这些分布比高斯分布平坦得多。均匀分布是一个典型的例子。见图 12.22。
- **偏态分布**“非高斯”的另一种方式是不对称。其中一个度量是**偏度**，由

$$skew(z) \triangleq \frac{\mu_3}{\sigma^3} \tag{12.112}$$

（右）偏态分布的一个例子是伽马分布（见图 2.9）。

当我们观察许多自然信号（如图像和语音）的经验分布时，当通过某些线性滤波器时，它们往往是超高斯的。这一结果既适用于在大脑某些部位发现的线性滤波器，如初级视觉皮层中发现的简单细胞，也适用于信号处理中使用的线性滤波器，如小波变换。因此，使用独立分量分析对自然信号建模的一个明显选择是拉普拉斯分布。对于均值零和方差 1，其对数 pdf 由以下公式得出：

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

由于拉普拉斯先验在原点不可微，因此更常见的是使用其他更平滑的超高斯分布。一个例子是逻辑分布。对应的对数 pdf，对于平均值为零，方差为 1 的情况（因此 $\mu = 0, s = \frac{\sqrt{3}}{\pi}$），由以下公式得出：

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

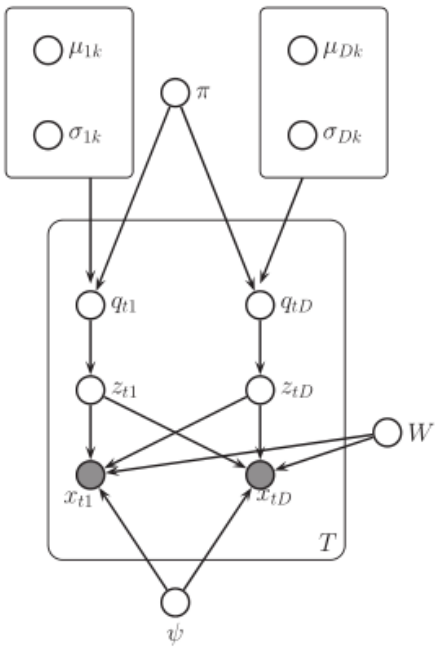


图 12.23: 使用单变量高斯混合建模源分布（的独立因子分析模型（Moulines 等人，1997；Attias 1999））。

估计 $G(Z) = -\log p(z)$ 在开创性论文（Pham 和 Garrat 1997）中进行了讨论。然而，当通过最大似然法拟合独立分量分析时，知道信源分布的准确形状并不重要（尽管知道它是亚高斯分布还是超高斯分布很重要）。因此，通常只使用 $G(z) = \sqrt{z}$ 或 $G(z) = \log cosh(z)$，而不是上面更复杂的表达式。

1.6.3 使用 EM

假设 $G(z)$ 的特定形式或 $p(z)$ 的等效形式的替代方法是使用灵活的非参数密度估计器，例如（单变量）高斯的混合物：

$$p(q_j = k) = \pi_k \tag{12.115}$$

$$p(z_j|q_j = k) = \mathcal{N}(\mu_{j,k}, \sigma^2_{j,k}) \tag{12.116}$$

$$p(\mathbf{x}|\mathbf{z}) = \mathcal{N}(\mathbf{W}\mathbf{z}, \boldsymbol{\Psi}) \tag{12.117}$$

这种方法是在（Moulines 等人，1997 年；Attias 1999 年）中提出的，相应的图形模型如图 12.23 所示。

对于这个模型，有可能推导出一个精确的 EM 算法。关键的观察是，可以通过对 $\mathbf{q}_t$ 变量的所有 K^L 组合求和来精确计算 $\mathbb{E}[\mathbf{z}_t|\mathbf{x}_t, \boldsymbol{\theta}]$，其中 K 是每个源的混合成分的数量。(如果这个方法太昂贵，可以使用变异平均场近似法（Attias 1999））。然后，我们可以通过对 $\mathbb{E}[\mathbf{z}_t]$ 进行标准的 GMM 拟合来估计所有的源分布。当源的 GMMs 已知时，我们可以非常容易地计算边际 $p_j(z_j)$，使用

$$p_j(z_j) = \sum_{k=1}^{K} \pi_{j,k} \mathcal{N}(z_j|\mu_{j,k}, \sigma^2_{j,k}) \tag{12.118}$$

给定 p_j，我们可以使用独立分量分析算法来估计 $\mathbf{W}$。当然，这些步骤应该交错进行。详情见（Attias 1999）。

1.6.4 其他估算原则 *

使用与最大似然法不同的方法估计独立分量分析模型的参数是很常见的。我们将在下面回顾其中一些方法，因为它们为独立分量分析提供了更多的见解。然而，我们也将看到，这些方法实际上毕竟等价于最大似然法。我们的介绍基于（Hyvarinen 和 Oja 2000）。

12.6.4.1 最大限度地提高非高斯性

独立分量分析的早期方法是找到矩阵 $\mathbf{V}$，使分布 $\mathbf{z} = \mathbf{V}\mathbf{x}$ 尽可能远离高斯分布。（统计学中有一种相关方法称为投影寻踪。）非高斯性的一个度量是峰度，但这可能对异常值敏感。另一个度量是**负熵**，定义为

$$negentropy(z) \triangleq \mathbb{H}(\mathcal{N}(\mu, \sigma^2)) - \mathbb{H}(z) \tag{12.119}$$

其中 $\mu = \mathbb{E}[z]$ 和 $\sigma^2 = var[z]$。由于高斯分布是最大熵分布，因此该测度总是非负的，对于高度非高斯的分布，该测度会变大。

我们可以将目标定义为最大化

$$J(\mathbf{V}) = \sum_j negentropy(z_j) = \sum_j \mathbb{H}(\mathcal{N}(\mu_j, \sigma^2_j)) - \mathbb{H}(z_j) \tag{12.120}$$

其中 $\mathbf{z} = \mathbf{V}\mathbf{x}$。如果我们将 $\mathbf{V}$ 固定为正交，并且如果我们将数据白化处理，$\mathbf{z}$ 的协方差将是独立于 $\mathbf{V}$ 的 $\mathbf{I}$，因此第一项是常数。因此

$$J(\mathbf{V}) = \sum_j -\mathbb{H}(z_j) + const = \sum_j \mathbb{E}[\log p(z_j)] + const \tag{12.121}$$

我们看到它等于（符号变化和无关常数）等式 12.102 中的似然对数。

12.6.4.2 最大限度地减少相互信息

一组随机变量的依赖性的一个衡量标准是**多信息的**。

$$\mathbf{I}(\mathbf{z}) \triangleq \mathbb{KL}\left(p(\mathbf{z})||\prod_j p(z_j)\right) = \sum_j \mathbb{H}(z_j) - \mathbb{H}(\mathbf{z}) \tag{12.122}$$

我们希望尽量减少这种情况，因为我们正在努力寻找独立的组件。换句话说，我们希望得到联合分布的最佳因子近似值。

既然 $\mathbf{z} = \mathbf{V}\mathbf{x}$，那么

$$I(\mathbf{z}) = \sum_j \mathbb{H}(z_j) - \mathbb{H}(\mathbf{V}\mathbf{x}) \tag{12.123}$$

如果我们将 $\mathbf{V}$ 约束为正交，我们可以去掉最后一项，因为 $\mathbb{H}(\mathbf{V}\mathbf{x}) = \mathbb{H}(\mathbf{x})$ 因为乘以 $\mathbf{V}$ 不会改变分布的形状），而 $\mathbb{H}(\mathbf{x})$ 是一个常数，仅由经验分布决定。因此我们有 $I(\mathbf{z}) = \sum_j \mathbb{H}(z_j)$。最小化这一点相当于最大化负熵，这相当于最大似然。

12.6.4.3 最大化相互信息（infomax）

与其试图最小化 $\mathbf{z}$ 分量之间的互信息，不如想象一个神经网络，其中 $\mathbf{x}$ 是输入，$y_i = \phi(\mathbf{v}_j^T) + \epsilon$ 是噪声输出，其中 ϕ 是一些非线性标量函数，和 $\epsilon \sim \mathcal{N}(0, 1)$。试图最大化通过该系统的信息流似乎是合理的，这一原则被称为 **infomax**。（Bell 和 Sejnowski 1995）。也就是说，我们希望最大化 $\mathbf{y}$(内部神经表示) 和 $\mathbf{x}$（观察到的输入信号）之间的互信息。我们有 $\Pi(\mathbf{x}; \mathbf{y}) = \mathbb{H}(\mathbf{y}) - \mathbb{H}(\mathbf{y}|\mathbf{x})$，如果我们假设噪声具有恒定方差，则后一项是恒定的。可以证明，我们可以近似前一项，如下所示

$$\mathbb{H}(\mathbf{y}) = \sum_{j=1}^{L} \mathbb{E}[\log \phi'(\mathbf{v}_j^T \mathbf{x})] + \log |det(\mathbf{V})| \tag{12.124}$$

其中，像往常一样，如果 $\mathbf{V}$ 是正交的，我们可以去掉最后一项。如果我们将 $\phi(z)$ 定义为 cdf，那么 $\phi'(z)$ 是其 pdf，且上述表达式等效于对数似然。特别是，如果我们使用逻辑非线性，$\phi(z) = sigm(z)$，那么相应的 pdf 是逻辑分布，$\phi'(z) \log cosh(z)$(忽略无关常数)。因此，我们看到 infomax 等价于最大似然。

练习

练习 12.1 FA 的 M 步骤

对于 FA 模型，表明 $\mathbf{W}$ 的 M 步最大似然估计由等式 12.23 给出。

练习 12.2 FA 模型的 MAP 估计

使用参数共轭先验推导 FA 模型的 M 步。

练习 12.3 评估主成分分析适用性的启发式方法

（来源：（出版社 2005 年第 9.8 期）。假设经验协方差矩阵 $\sum$ 具有特征值 $\lambda_1 \geq \lambda_2 \geq ... \geq \lambda_d > 0$。解释为什么评估值的方差 $\sigma^2 = \frac{1}{d}\sum_{i=1}^{d}(\lambda_i - \overline{\lambda})^2$ 是衡量主成分分析是否有助于分析数据的良好指标 σ^2 的值越高，主成分分析越有用）。

练习 12.4 推导第二主成分

a. 设:

$$J(\mathbf{v}_2, \mathbf{z}_2) = \frac{1}{n}\sum_{i=1}^{n}(\mathbf{x}_i - z_{i1}\mathbf{v}_1 - z_{i2}\mathbf{v}_2)^T(\mathbf{x}_i - z_{i1}\mathbf{v}_1 - z_{i2}\mathbf{v}_2) \tag{12.125}$$

证明 $\frac{\partial J}{\partial \mathbf{z}_2} = 0$ 产生 $z_{i2} = \mathbf{v}_2^T \mathbf{x}_i$

b. 证明最小化的 $\mathbf{v}_2$ 的值是

$$\widetilde{J}(\mathbf{v}_2) = -\mathbf{v}_2^T \mathbf{C}\mathbf{v}_2 + \lambda_2(\mathbf{v}_2^T\mathbf{v}_2 - 1) + \lambda_{12}(\mathbf{v}_2^T\mathbf{v}_1 - 0) \tag{121.126}$$

由 $\mathbf{C}$ 的第二大特征值的特征向量给出的。提示：回顾一下，$\mathbf{C}\mathbf{v}_1 = \lambda_1\mathbf{v}_1$ 和 $\frac{\partial \mathbf{x}^T\mathbf{A}\mathbf{x}}{\partial \mathbf{x}} = (\mathbf{A} + \mathbf{A}^T)\mathbf{x}$。

练习 12.5 推导主成分分析的残余误差

a. 证明

$$||\mathbf{x}_i - \sum_{j=1}^{K} z_{ij}\mathbf{v}_j||^2 = \mathbf{x}_i^T\mathbf{x}_i - \sum_{j=1}^{K} \mathbf{v}_j^T\mathbf{x}_i\mathbf{x}_i^T\mathbf{v}_j \tag{12.127}$$

提示：首先考虑 $K=2$ 的情况。使用 $\mathbf{v}_j^T\mathbf{v}_j=1$ 和 $\mathbf{v}_j^T\mathbf{v}_k=0$ 表示 $k\neq j$, 此外，回想一下 $z_{ij}=\mathbf{x}_i^T\mathbf{v}_j$。

b. 现在证明

$$J_K \triangleq \frac{1}{n}\sum_{i=1}^{n}\left(\mathbf{x}_i^T\mathbf{x}_i-\sum_{j=1}^{K}\mathbf{v}_j^T\mathbf{x}_i\mathbf{x}_i^T\mathbf{v}_j\right)=\frac{1}{n}\sum_{i=1}^{n}\mathbf{x}_i^T\mathbf{x}_i-\sum_{j=1}^{K}\lambda_j \tag{12.128}$$

提示：回想一下 $\mathbf{v}_j^T\mathbf{C}\mathbf{v}_j=\lambda_j\mathbf{v}_j^T\mathbf{v}_j=\lambda_j$

c. 如果 $K=d$，则没有截断，因此 $J_d=0$。使用这个证明，仅使用 $K<d$ 项的误差由以下公式得出：

$$J_K=\sum_{j=K+1}^{d}\lambda_j \tag{12.129}$$

提示：将总和 $\sum_{j=1}^{d}\lambda_j$ 划分成 $\sum_{j=1}^{K}\lambda_j$ 和 $\sum_{j=K+1}^{d}\lambda_j$。

练习 12.6 Fisher 线性判别式的推导

证明 $J(\mathbf{w})=\dfrac{\mathbf{w}^T\mathbf{S_B}\mathbf{w}}{\mathbf{w}^T\mathbf{S}_W\mathbf{w}}$ 的最大值由 $\mathbf{S}_B\mathbf{w}=\lambda\mathbf{S}_w\mathbf{w}$ 给出，其中 $\lambda=\dfrac{\mathbf{w}^T\mathbf{S}_B\mathbf{w}}{\mathbf{w}^T\mathbf{S}_W\mathbf{w}}$。提示：回想一下，两个标量之比的导数由 $\frac{d}{dx}\frac{f(x)}{g(x)}=\dfrac{f'g-fg'}{g^2}$ 给出，其中 $f'=\frac{d}{dx}f(x)$ 和 $g'=\frac{d}{dx}g(x)$。此外，还记得 $\frac{d}{dx}\mathbf{x}^T\mathbf{A}\mathbf{x}=(\mathbf{A}+\mathbf{A}^T)\mathbf{x}$。

练习 12.7 通过连续通缩的 PCA

设 $\mathbf{v}_1,\mathbf{v}_2,...,\mathbf{v}_k$ 是最大特征值为 $\mathbf{C}=\frac{1}{n}\mathbf{X}^T\mathbf{X}$ 的前 k 个特征向量，即主基向量。这些满足

$$\mathbf{v}_j^T\mathbf{v}_k=\begin{cases}0 & \text{if } j\neq k\\ 1 & \text{if } j=k.\end{cases} \tag{12.130}$$

我们将构造一种按顺序查找 $\mathbf{v}_j$ 的方法。

如我们在课堂上所示，$\mathbf{v}_1$ 是 $\mathbf{C}$ 的第一个主特征向量，满足 $\mathbf{C}\mathbf{v}_1=\lambda_1\mathbf{v}_1$。现在，将 $\widetilde{\boldsymbol{x}}_i$ 定义为 $\mathbf{x}_1$ 在与 $\mathbf{v}_1$ 正交的空间上的正交投影：

$$\widetilde{\boldsymbol{x}}_i=\mathbf{P}_{\perp\mathbf{v}_1}\mathbf{x}_i=(\mathbf{I}-\mathbf{v}_1\mathbf{v}_1^T)\mathbf{x}_i \tag{12.131}$$

定义 $\widetilde{\boldsymbol{X}}=[\widetilde{\boldsymbol{x}}_1;...;\widetilde{\boldsymbol{x}}_n]$ 为秩 $d-1$ 的**缩减矩阵**，通过从 d 维数据中删除位于第一主方向方向的分量获得：

$$\widetilde{\boldsymbol{X}}=(\mathbf{I}-\mathbf{v}_1\mathbf{v}_1^T)\mathbf{X}=(\mathbf{I}-\mathbf{v}_1\mathbf{v}_1^T)\mathbf{X} \tag{12.132}$$

a. 使用 $\mathbf{X}^T\mathbf{X}\mathbf{v}_1=n\lambda_1\mathbf{v}_1$,(因此 $\mathbf{v}_1^T\mathbf{X}^T\mathbf{X}=n\lambda_1\mathbf{v}_1^T$) 和 $\mathbf{v}_1^T\mathbf{v}_1=1$ 的事实，证明缩减矩阵的协方差由以下公式得出：

$$\widetilde{\boldsymbol{C}}\triangleq\frac{1}{n}\widetilde{\boldsymbol{X}}^T\widetilde{\boldsymbol{X}}=\frac{1}{n}\mathbf{X}^T\mathbf{X}-\lambda_1\mathbf{v}_1\mathbf{v}_1^T \tag{12.133}$$

b. 设 $\mathbf{u}$ 为 $\widetilde{\boldsymbol{C}}$ 的主特征向量。解释为什么 $\mathbf{u}=\mathbf{v}_2$。（你可以假设 $\mathbf{u}$ 是单位范数。）

c. 假设我们有一个简单的方法来寻找一个 pd 矩阵的前导特征向量和特征值，表示为 $[\lambda,\boldsymbol{\mu}]=f(\mathbf{C})$。编写一些伪代码，用于查找仅使用特殊 f 函数和简单向量算法的 $\mathbf{X}$ 的前 K 个主基向量，即您的代码不应使用或 eig 函数。提示：这应该是一个简单的迭代程序，需要 2-3 行代码来编写。输入为 $\mathbf{C}$、K 和函数 f，当 $j=1:K$ 时，输出应为 $\mathbf{v}_j$ 和 λ_j。不要担心语法是否正确。

练习 12.8 潜在语义索引

（来源：de Freitas）。在本练习中，我们研究了一种称为**潜在语义索引**的技术，该技术通过术语矩阵将奇异值分解应用于文档，以创建数据的低维嵌入，该嵌入旨在捕获单词的语义相似性。

文件 lsiDocuments.pdf 包含 9 个不同主题的文件。这些文件中出现的所有 460 个独特的在这些文件中出现的词或术语的列表在 lsiWords.txt 中。一个按术语分类的文件矩阵是在 lsiMatrix.txt。

a. 假设 X 是 LSIMetrix 的转置，因此每列表示一个文档。计算 X 的奇异值分解，并使用前 2 个奇异值/向量对 $\hat{X}$ 进行近似。在 2D 中绘制 9 个文档的低维表示。您应该得到如图 12.24 所示的结果。

b. 考虑找到有关外星人绑架的文件。如果你看一下 lsiWords.txt，这个词有 3 个版本，术语 23（" 被绑架"）、术语 24（" 绑架"）和术语 25（" 绑架"）。假设我们想找到包含" 被绑架" 这个词的文件。文件 2 和 3 包含它。但文件 1 却没有。然而，文件 1 显然与这个主题有关。因此，LSI 也应该找到文档 1。创建

一个包含” 被绑架” 这个词的测试文档 q，并将其投射到二维子空间来制作 $\hat{q}$。现在计算 $\hat{q}$ 和所有文档的低维表示之间的余弦相似度。前三个最接近的匹配是什么?

练习 12.9 FA 模型中的估算

推导 FA 模型的 $p(\mathbf{x}_h|\mathbf{x}_v, \boldsymbol{\theta})$ 表达式。

练习 12.10 有效评估 PPCA 密度

在插入 MLEs 的基础上，利用矩阵反转定理，推导出 PPCA 模型的 $p(\mathbf{x}|\hat{\boldsymbol{W}}, \hat{\sigma}^2)$ 表达式。矩阵反转定理。

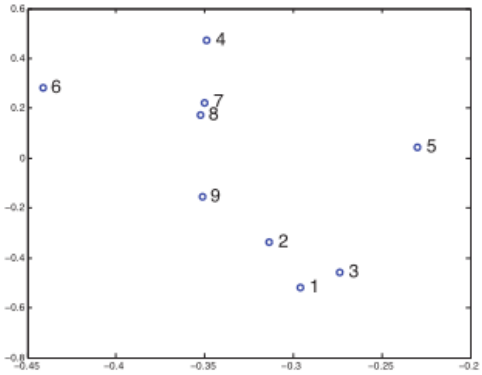


图 12.24: 将 9 份文件投影到二维。lsiCode 生成的图。

练习 12.11 PPCA 与 FA

（来源：练习 14.15（Hastie 等人，2009），由 Hinton 提供。）。从以下模型中生成 200 个观察值，其中 $\mathbf{z}_i \sim \mathcal{N}(\mathbf{0}, \mathbf{I}) : x_{i1} = z_{i1}, x_{i2} = z_{i1} + 0.001 z_{i2}, x_{i3} = 10 z_{i3}$。拟合具有 1 个潜在因子的 FA 和 PCA 模型。因此表明，在主成分分析的情况下，相应的权重向量 $\mathbf{w}$ 与最大方差方向（维数 3）对齐，但在 FA 的情况下与最大相关方向（维数 1+2）对齐。

2 稀疏线性模型

2.1 介绍

我们在第 3.5.4 节中介绍了特征选择的主题，其中我们讨论了寻找与输出具有高互信息的输入变量的方法。这种方法的问题在于，它基于一种短视的策略，一次只看一个变量。如果存在交互效应，则可能会失败。例如，如果 $y = xor(x_1, x_2)$，那么 x_1 和 x_2 本身都不能预测响应，但它们一起可以完美地预测响应。关于这方面的真实例子，请考虑基因关联研究：有时两个基因本身可能是无害的，但当它们同时存在时，会导致隐性疾病（Balding 2006）。

在本章中，我们重点关注使用基于模型的方法一次选择变量集。如果该模型是一个广义线性模型，对于某些链接函数 f，其形式为 $p(y|\mathbf{x}) = p(y|f(\mathbf{w}^T\mathbf{x}))$，那么我们可以通过鼓励权重向量 $\mathbf{w}$ 是**稀疏**的来进行特征选择。即有大量的零。事实证明，这种方法具有显著的计算优势，我们将在下文中看到。

以下是一些特征选择/稀疏性有用的应用程序：

- 在许多问题中，我们有比训练案例 N 更多的维度 D。相应的设计矩阵是短而胖的，而不是高而瘦的。这叫做**小 N，大 D 问题**。随着我们开发更多的高通量测量设备，例如，利用基因微阵列，这变得越来越普遍，测量 $D \sim 10000$ 个基因，但只得到 $N \sim 100$ 个这样的例子。（这可能是即使是我们的数据似乎也在变得越来越胖的时代的一个迹象……）我们可能希望找到能够准确预测响应（例如细胞生长率）的最小特征集，以防止过度拟合，降低构建诊断设备的成本，或帮助科学洞察问题。

- 在第 14 章中，我们将使用以训练示例为中心的基函数，因此 $\boldsymbol{\phi}(\mathbf{x}) = [\kappa(\mathbf{x}, \mathbf{x}_1), ..., \kappa(\mathbf{x}, \mathbf{x}_N)]$，其中 κ 是一个核函数。得到的设计矩阵大小为 $N \times N$。在这种情况下，特征选择相当于选择训练示例的子集，这有助于减少过拟合和计算成本。这被称为稀疏内核机器。

- 在信号处理中，通常用小波基函数表示信号（图像、语音等）。为了节省时间和空间，根据少量此类基函数找到信号的稀疏表示非常有用。这使我们能够从少量测量中估计信号，并对信号进行压缩。详见第 13.8.3 节。

请注意，特征选择和稀疏性主题是当前机器学习/统计中最活跃的领域之一。在本章中，我们只剩下篇幅来概述主要结果。

2.2 贝叶斯变量选择

提出变量选择问题的自然方法如下。如果特征 j “相关”，则设 $\gamma_j = 1$，否则设 $\gamma_j = 0$。我们的目标是计算后验模型

$$p(\gamma|\mathcal{D}) = \frac{e^{-f(\gamma)}}{\sum_{\gamma'} e^{-f(\gamma')}} \tag{13.1}$$

其中 $f(\gamma)$ 是成本函数：

$$f(\gamma) \triangleq -[\log p(\mathcal{D}|\gamma) + \log p(\gamma)] \tag{13.2}$$

例如，假设我们从 $D = 10$ 维线性回归模型 $y_i \sim \mathcal{N}(\mathbf{w}^T\mathbf{x}_i, \sigma^2)$ 中生成 $N = 20$ 个样本，其中，$\mathbf{w}$ 的 $K = 5$ 个元素为非零。特别是，我们使用 $\mathbf{w} = (0.00, -1.67, 0.13, 0.00, 0.00, 1.19, 0.00, -0.04, 0.33, 0.00)$ 和 $\sigma^2 = 1$。我们列举了所有 $2^{10} = 1024$ 个模型，并计算每个模型的 $p(\gamma|\mathcal{D})$（我们给出了下面的等式）。我们按格雷码顺序对模型进行排序，以确保连续向量恰好相差 1 位（原因是计算性的，在第 13.2.3 节中讨论）。

结果位模式集如图 13.1（a）所示。每个模型的成本 $f(\gamma)$，如图 13.1（b）所示。我们看到，这个目标函数非常“颠簸”。如果我们计算模型 $p(\gamma|\mathcal{D})$ 上的后验分布，结果更容易解释。这如图 13.1（c）所示。前 8 模型如下：

model	prob	member
4	0.447	2,
61	0.241	2,6,
452	0.103	2,6,9,
60	0.091	2,3,6
29	0.041	2,5,
68	0.021	2,6,7,
36	0.015	2,5,6,
5	0.010	2,3,

“真实”模型是 {2，3，6，8，9}。然而，与特征 3 和 8 相关的系数非常小（相对于 σ^2）。因此，这些变量更难检测。如果有足够的数据，该方法将收敛于真实模型（假设数据是从线性模型生成的），但对于有限的数据集，通常会存在相当大的后验不确定性。

在大量模型上解释后验数据相当困难，因此我们将寻求各种汇总统计数据。自然模式是后验模式，或 MAP 估计

$$\hat{\boldsymbol{\gamma}} = argmax(\boldsymbol{\gamma}|\mathcal{D}) = argminf(\boldsymbol{\gamma}) \tag{13.3}$$

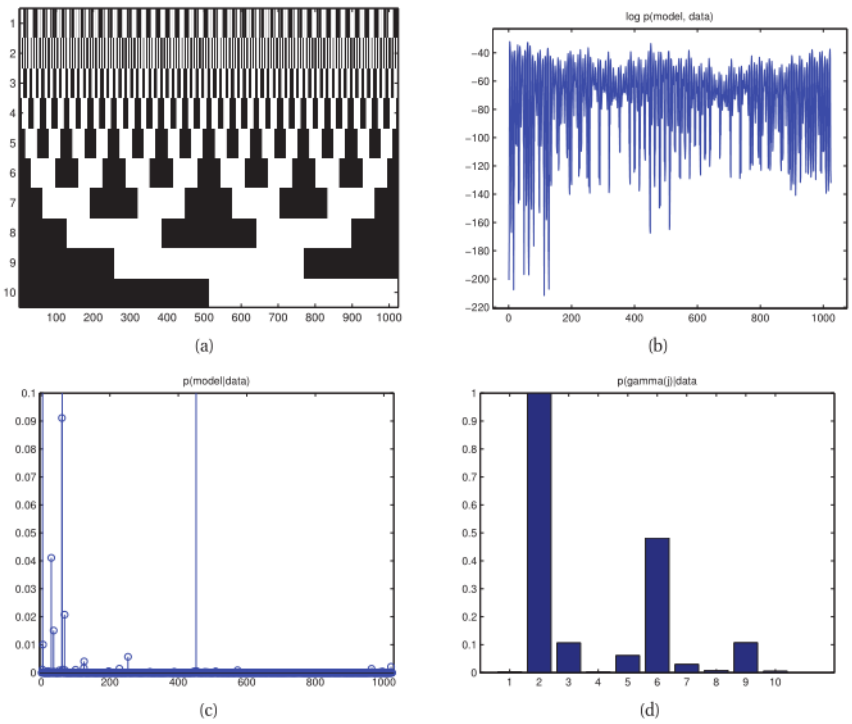


图 13.1:（a）长度为 10 的所有可能位向量按格雷码顺序枚举。（b）所有可能模型的评分函数。（c）所有 1024 个模型的后验。为了清晰起见，垂直比例被截断为 0.1。（d）边际包容概率。图由 linregAllsubsetsGraycodeDemo 生成。

然而，该模式往往不能代表全部后部质量（见第 5.2.1.3 节）。一个更好的总结是**中值模型**（Barbieri and Berger 2004; Carvahlo and Lawrence 2007）。计算时使用

$$\hat{\boldsymbol{\gamma}} = \{j : p(\gamma_j = 1|\mathcal{D}) > 0.5\} \tag{13.4}$$

这需要计算后边缘包含概率 $p(\gamma_j = 1|\mathcal{D})$。如图 13.1（d）所示。我们看到，该模型确信包含变量 2 和 6；如果我们将决策阈值降低到 0.1，我们还将添加 3 和 9。然而，如果我们想要“捕获”变量 8，我们将产生两个误报（5 和 7）。这种权衡第 5.7.2.1 节将更详细地讨论假阳性和假阴性之间的权衡。

上面的例子说明了变量选择的“黄金标准”：问题足够小（只有 10 个变量），我们能够精确计算完整的后验值。当然，变量选择在维数较大的情况下最有用。由于存在 2^D 可能模型（位向量），因此通常不可能计算全后验概率，甚至难以找到总结，例如映射估计或边缘包含概率。因此，我们将在本章的大部分时间集中于算法加速。但是在我们这样做之前，我们将解释如何在上面的例子中计算 $p(\gamma|\mathcal{D})$。

2.2.1 尖峰和板模型

后面的公式为：

$$p(\gamma|\mathcal{D}) \propto p(\gamma)p(\mathcal{D}|\gamma) \tag{13.5}$$

我们首先考虑先验，然后考虑可能性。

通常在位向量上使用以下先验：

$$p(\gamma) = \prod_{j=1}^{D} Ber(\gamma_j|\pi_0) = \pi^{||\gamma||_0}(1-\pi_0)^{D-||\gamma||_0} \tag{13.6}$$

其中，π_0 是一个特征相关的概率，$||\gamma||_0 = \sum_{j=1}^{D}\gamma_j$ 是 ℓ_0 的**伪正态**。也就是向量中非零元素的数量。为了与后来的模型进行比较，将对数先验写成以下形式是很有用的为了与后面的模型相比较，把对数先验写成以下样子是很有用的。

$$\begin{aligned}
\log p(\gamma|\pi_0) &= ||\gamma||_0 \log \pi_0 + (D - ||\gamma||_0)\log(1-\pi_0) && (13.7)\\
&= ||\gamma||_0(\log \pi_0 - \log(1-\pi_0)) + const && (13.8)\\
&= -\lambda||\gamma||_0 + const && (13.9)
\end{aligned}$$

其中 $\lambda \triangleq \log\dfrac{1-\pi_0}{\pi_0}$ 控制模型的稀疏性。

我们可以把这个可能性写成如下:

$$p(\mathcal{D}|\gamma) = p(\mathbf{y}|\mathbf{X}, \boldsymbol{\gamma}) = \iint p(\mathbf{y}|\mathbf{X}, \mathbf{w}, \boldsymbol{\gamma})p(\mathbf{w}|\gamma, \sigma^2)p(\sigma^2)d\mathbf{w}d\sigma^2 \tag{13.10}$$

为了简化符号，我们假设响应居中（即 $\overline{y} = 0$），因此可以忽略任何偏移项 μ。

我们现在讨论先验 $p(\mathbf{w}|\gamma, \sigma^2)$。如果 $\gamma_j = 0$，特征 j 是无关的，因此我们期望 $w_j = 0$。如果 $\gamma_j = 1$，我们期望 w_j 是非零的。如果我们标准化输入，一个合理的先验是 $\mathcal{N}(0, \sigma^2\sigma_w^2)$，其中 σ_w^2 控制我们预期与相关变量相关的系数的大小（由整体噪声级 σ^2 缩放）。我们可以总结如下：

$$p(w_j|\sigma^2, \gamma_j) = \begin{cases} \delta_0(w_j) & \text{if } \gamma_j = 0 \\ \mathcal{N}(w_j|0, \sigma^2\sigma_w^2) & \text{if } \gamma_j = 1 \end{cases} \tag{13.11}$$

第一项是原点的“尖峰”。像 $\sigma \to \infty$, 分布 $p(w_j|\gamma_j = 1)$ 接近均匀分布，可以将其视为恒定高度的“板”。因此，这被称为**钉板模型**（Mitchell 和 Beauchamp 1988）。

我们可以从模型中删除系数 w_j，其中 $w_j = 0$，因为在先验条件下，它们被钳制为零。因此，等式 13.10 变为以下（假设高斯似然）：

$$p(\mathcal{D}|\gamma) = \iint \mathcal{N}(\mathbf{y}|\mathbf{X}_\gamma\mathbf{w}_\gamma, \sigma^2\mathbf{I}_N)\mathcal{N}(\mathbf{w}_\gamma|\mathbf{0}_{D_\gamma}, \sigma^2\sigma_w^2\mathbf{I}_{D_\gamma})p(\sigma^2)d\mathbf{w}_\gamma d\sigma^2 \tag{13.12}$$

其中 $D_\gamma = ||\gamma||_0$ 是 γ 中非零元素的数量。在接下来的内容中，我们将通过定义任意正定矩阵 $\sum_\gamma$ [5]的形式 $p(\mathbf{w}|\gamma, \sigma^2) = \mathcal{N}(\mathbf{w}_\gamma|\mathbf{0}_{D_\gamma}, \sigma\sum_{\boldsymbol{\gamma}})$ 的先验来稍微概括这一点。

考虑到这些先验，我们现在可以计算边际似然。如果噪声方差已知，我们可以将边际似然（使用等式 13.151）记下如下：

$$p(\mathcal{D}|\boldsymbol{\gamma}^2) = \int \mathcal{N}(\mathbf{y}|\mathbf{X}_\gamma\mathbf{w}_\gamma, \sigma^2\mathbf{I})\mathcal{N}(\mathbf{w}_\gamma|\mathbf{0}, \sigma^2\sum\nolimits_\gamma)d\mathbf{w}_\gamma = \mathcal{N}(\mathbf{y}|\mathbf{0}, \mathbf{C}_\gamma) \tag{13.13}$$

[5]通常使用 $\sum_\gamma g(\mathbf{X}_\gamma^T\mathbf{X}_\gamma)^{-1}$ 形式的 g 先验第 7.6.3.1 节解释的原因（另见练习 13.4）。已经提出了各种设置 g 的方法，包括交叉验证、经验贝叶斯（Minka 2000b；George 和 Foster 2000）、层次贝叶斯（Liang 等人，2008）等。

$$\mathbf{C}_\gamma \triangleq \sigma^2 \mathbf{X}_\gamma \sum_\gamma \mathbf{X}_\gamma^T + \sigma^2 \mathbf{I}_N \tag{13.14}$$

如果噪声未知，我们可以对其进行先验分析并将其积分。通常使用 $p(\sigma^2) = IG(\sigma^2|a_\sigma, b_\sigma)$ 关于设置 a、b 的一些指南见（Kohn 等人，2001 年）。如果我们使用 $a = b = 0$，我们恢复 Jeffrey 先验，$p(\sigma^2) \propto \sigma^{-2}$ 当我们积分出噪声时，我们得到以下更复杂的边际似然表达式（Brown 等人，1998）：

$$p(\mathcal{D}|\gamma) = \iint p(\mathbf{y}|\gamma, \mathbf{w}_\gamma, \sigma^2) p(\mathbf{w}_\gamma|\gamma, \sigma^2) p(\sigma^2) d\mathbf{w}_\gamma d\sigma^2 \tag{13.15}$$

$$\propto |\mathbf{X}_\gamma^T \mathbf{X}_\gamma + \sum_\gamma^{-1}|^{-\frac{1}{2}} |\sum_\gamma|^{-\frac{1}{2}} (2b_\sigma + S(\gamma))^{-(2a_\sigma + N - 1)/2} \tag{13.16}$$

其中 $S(\gamma)$ 是 RSS：

$$S(\gamma) \triangleq \mathbf{y}^T \mathbf{y} - \mathbf{y}^T \mathbf{X}_\gamma (\mathbf{X}_\gamma^T \mathbf{X}_\gamma + \sum_\gamma^{-1})^{-1} \mathbf{X}_\gamma^T \mathbf{y} \tag{13.17}$$

另见练习 13.4。

当无法以闭合形式计算边际似然时（例如，如果我们使用的是逻辑回归或非线性模型），我们可以使用具有以下形式的 BIC 进行近似

$$\log p(\mathcal{D}|\gamma) \approx \log p(\mathbf{y}|\mathbf{X}, \hat{\mathbf{w}}_\gamma, \hat{\sigma}^2) - \frac{||\gamma||_0}{0} \log N \tag{13.18}$$

其中 $\hat{\mathbf{w}}_\gamma$ 是基于 $\mathbf{X}_\gamma$ 的 ML 或 MAP 估计，$||\gamma||_0$ 0 是模型的“自由度”（Zou 等人，2007）。加上对数先验，总体目标变成了

$$\log p(\gamma|\mathcal{D}) \approx \log p(\mathbf{y}|\mathbf{X}, \hat{\mathbf{w}}_\gamma, \hat{\sigma}^2) - \frac{||\gamma||_0}{0} \log N - \lambda ||\gamma||_0 + const \tag{13.19}$$

我们看到有两个复杂度惩罚：一个来自 BIC 近似的边际似然，另一个来自 $p(\gamma)$ 的先验。边际似然产生的，另一个是由 $p(\gamma)$ 的先验产生的。很明显，这两个参数可以合并为一个整体的复杂性参数，我们用 γ 来表示。

2.2.2 从伯努利-高斯模型到 ℓ_0 正则化

有时使用的另一个模型（例如（Kuo 和 Mallick 1998；Zhou 等人 2009；Soussen 等人 2010））如下：

$$y_i|\mathbf{x}_i, \mathbf{w}, \gamma, \sigma^2 \sim \mathcal{N}(\sum_j \gamma_j w_j x_{ij}, \sigma^2) \tag{13.20}$$

$$\gamma_j \sim Ber(\pi_0) \tag{13.21}$$

$$w_j \sim \mathcal{N}(0, \sigma_w^2) \tag{13.22}$$

在信号处理文献中（例如（Soussen 等人，2010）），这被称为**伯努利高斯**模型，尽管我们也可以称其为**二进制掩码**模型，因为我们可以将 γ_j 变量视为“掩盖”权重 w_j。

与尖峰和板模型不同，我们不整合“无关”系数；它们总是存在的。此外，二元掩码模型的形式为 $\gamma_j \rightarrow \mathbf{y} \leftarrow w_j$，而尖峰和平板模型的形式为 $\gamma_j \rightarrow w_j \rightarrow \mathbf{y}$, 在二元掩码模型中，只能从似然中识别乘积 $\gamma_j w_j$。

该模型的一个有趣方面是，它可以用于推导在（非贝叶斯）子集选择文献中广泛使用的目标函数。首先，注意联合先验具有以下形式：

$$p(\gamma, \mathbf{w}) \propto \mathcal{N}(\mathbf{w}|\mathbf{0}, \sigma_w^2 \mathbf{I}) \pi_0^{||\gamma||_0} (1 - \pi_0)^{D - ||\gamma||_0} \tag{13.23}$$

因此，缩放后的非正态化负对数后验的形式为：

$$f(\gamma, \mathbf{w}) \triangleq -2\sigma^2 \log p(\gamma, \mathbf{w}, \mathbf{y}|\mathbf{X}) = ||\mathbf{y} - \mathbf{X}(\gamma .* \mathbf{w})||^2 \tag{2}$$

$$+ \frac{\sigma^2}{\sigma_w^2} ||\mathbf{w}||^2 + \lambda ||\gamma||_0 + const \tag{13.24}$$

其中：

$$\lambda \triangleq 2\sigma^2 \log(\frac{1 - \pi_0}{\pi_0}) \tag{13.25}$$

让我们把 $\mathbf{w}$ 分成两个子向量，$\mathbf{w}_{-\gamma}$ 和 $\mathbf{w}_\gamma$，分别由 γ 的零和非零项索引。因为 $\mathbf{X}(\gamma .* \mathbf{w}) = \mathbf{X}_\gamma \mathbf{w}_\gamma$，我们可以设置 $\mathbf{w}_{-\gamma} = 0$.

现在考虑 $\sigma_w^2 \to \infty$ 的情况, 因此，我们不正则化非零权重（因此没有来自边缘似然或其 BIC 近似的复杂度惩罚）。在这种情况下，目标变为

$$f(\gamma, \mathbf{w}) = ||\mathbf{y} - \mathbf{X}_\gamma \mathbf{w}_\gamma||_2^2 + \lambda ||\gamma||_0 \tag{13.26}$$

这与上述 BIC 目标类似。

我们可以将相关变量集定义为 $\mathbf{w}$ 的支持集或非零项集，而不是跟踪位向量 γ。然后我们可以将上述等式重写为：

$$f(\mathbf{w}) = ||\mathbf{y} - \mathbf{X}\mathbf{w}||_2^2 + \lambda ||\mathbf{w}||_0 \tag{13.27}$$

这被称为 ℓ_0 **正则化**。我们已经将离散的优化问题（对 $\gamma \in \{0,1\}^D$ 转化为连续问题（在 $\mathbf{w} \in \mathbb{R}^D$ 上）；然而 ℓ_0 伪规范使目标非常不平滑。目标非常不平滑，所以这仍然很难优化。我们将在本章后面讨论不同的解决方案我们将在本章的其余部分讨论不同的解决方案。

2.2.3 算法

由于存在 2^D 模型，我们无法探索全后验，也无法找到全局最优模型。相反，我们将不得不求助于某种形式的启发式。我们将讨论的所有方法都涉及在模型空间中搜索，并评估每个点的成本 $f(\gamma)$。这需要在每一步拟合模型（即计算 $argmax\ p(\mathcal{D}|\mathbf{w})$ 或评估其边际似然 (即计算 $\int p(\mathcal{D}|\mathbf{w})p(\mathbf{w})d\mathbf{w}$) 这有时被称为**封装方法**，因为我们将搜索最佳模型（或一组好模型）“包装”在通用模型拟合过程周围。

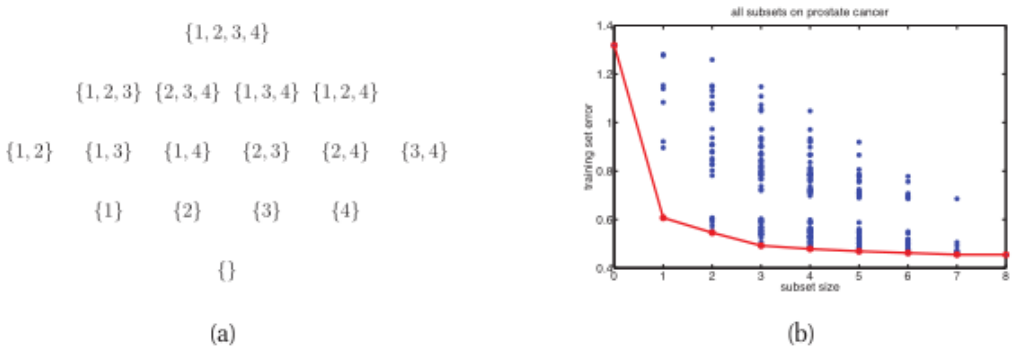


图 13.2: (a) 1, 2, 3, 4 的子集网格。(b) 残余平方和与子集大小的关系，在前列腺癌数据集。下方的包络是任何给定大小的集合所能达到的最佳 RSS。基于 (Hastie et al. 2001) 的图 3.5。图由 prostateSubsets 生成。

为了使包装方法有效，重要的是我们可以快速评估一些新模型的得分函数，γ'，考虑到先前模型的分数，γ。只要我们能够有效地更新计算 $f(\gamma)$ 所需的足够统计信息，就可以做到这一点。如果 γ' 仅在一位上与 γ 不同（对应于添加或删除单个变量），并且假设 $f(\gamma)$ 仅取决于通过 $\mathbf{X}_\gamma$ 的数据。在这种情况下，我们可以使用秩一矩阵更新/下降来有效地计算 $\mathbf{X}_{\gamma'}^T \mathbf{X}_{\gamma}$ 来自 $\mathbf{X}_\gamma^T \mathbf{X}_\gamma$。这些更新通常应用于 $\mathbf{X}$ 的 QR 分解。有关详细信息，请参见（Miller 2002；Schniter 等人，2008）。

13.2.3.1 贪婪搜索

假设我们想要找到地图模型。如果我们使用 ℓ_0-正则化目标在等式 13.27 中，我们可以利用最小二乘法的性质来推导各种有效的贪心正向搜索方法，其中一些我们总结如下。有关更多详细信息，请参见（Miller 2002；Soussen 等人，2010）。

- **单一最佳替换**最简单的方法是使用贪心爬山，在每个步骤中，我们将当前模型的邻域定义为通过翻转一位 γ 可以达到的所有模型，即，对于每个变量，如果它当前不在模型中，我们考虑添加它，如果它当前在模型中，我们考虑删除它。在（Soussen 等人，2010 年）中，他们将其称为单一最佳替换（SBR）。由于我们希望得到稀疏解，我们可以从空集 $\gamma = \mathbf{0}$ 开始。我们基本上是在子集的晶格中移动，如图 13.2（a）所示。我们继续添加或删除，直到无法改善为止。

- **正交最小二乘法**如果我们在等式 13.27 中设置 $\gamma = 0$，那么没有复杂度惩罚，就没有理由执行删除步骤。在这种情况下，SBR 算法等效于**正交最小二乘法**（Chen 和 Wigger 1995），而正交最小二乘法又等效于贪婪**正向选择**。在该算法中，我们从空集开始，在每一步添加最佳特征。误差将随着 $||\gamma||_0$ 单调下降，如图 13.2（b）所示。我们可以选择下一个最好的功能 j^* 通过求解将 γ_t 添加到当前集合

$$j^* = arg \min_{j \notin \gamma_t} \min_{\mathbf{w}} ||\mathbf{y} - (\mathbf{X}_{\gamma_t \cup j})\mathbf{w}||^2 \tag{13.28}$$

  然后，我们通过设置 $\gamma^{(t+1)} = \gamma^{(t)} \cup \{j^*\}$ 来更新活动集. 为了选择下一个要在步骤 t 中添加的特征，我们需要求解第 t 步的 $D - D_t$ 最小二乘问题，其中 $D_t = |\gamma_t|$ 是当前活动集的基数。选择了要添加的最佳特征后，我们需要解决一个额外的最小二乘问题来计算 $\mathbf{w}_{t+1}$）。

- **正交匹配追求**正交最小二乘有点昂贵。一种简化方法是将当前权重“冻结”为其当前值，然后通过求解选择下一个要添加的特征

$$j^* = arg \min_{j \notin \gamma_t} \min_{\beta} ||\mathbf{y} - \mathbf{X}\mathbf{w}_t - \beta \mathbf{x}_{:,j}||^2 \tag{13.29}$$

  这种内部优化很容易求解：我们只需设置 $\beta = \mathbf{x}_{:,j}^T \mathbf{r}_t / ||\mathbf{x}_{:,j}||^2$ 其中 $\mathbf{r}_t = \mathbf{y} - \mathbf{X}\mathbf{w}_t$ 是当前残差向量。如果列是单位范数，我们有

$$j^* = arg\ max \mathbf{x}_{:,j}^T \mathbf{r}_t \tag{13.30}$$

  所以我们只是寻找与当前残差最相关的列。然后，我们更新活动集，并使用 $\mathbf{X}_{\gamma_{t+1}}$ 计算新的最小二乘估计 $\mathbf{w}_{t+1}$。这种方法称为**正交匹配追踪或 OMP**（Mallat 等人，1994）。每次迭代只需要一次最小二乘计算，因此比正交最小二乘法更快，但不太准确（Blumensath 和 Davies 2007）。

- **匹配追求**更激进的近似方法是贪婪地添加与当前残差最相关的特征。这被称为**匹配追求**（Mallat 和 Zhang 1993）。这也相当于一种称为最小二乘增压的方法（第 16.4.6 节）。

- **向后选择**向后选择从模型中的所有变量（所谓的**饱和模型**）开始，然后在每一步删除最差的变量。这相当于从晶格顶部向下执行贪婪搜索。这比自底向上搜索可以得到更好的结果，因为关于是否保留变量的决定是在可能依赖于它的所有其他变量的上下文中作出的。然而，这种方法通常不适用于大型问题，因为饱和模型的拟合成本太高。

- **FoBa**（Zhang 2008）的**前后向算法**与上述单一最佳替换算法类似，只是在选择下一步时使用了类似 OMP 的近似值。（Moghaddam 等人，2008 年）中描述了类似的“双通道”算法。

- **贝叶斯匹配追踪**（Schniter 等人，2008 年）的算法与 OMP 相似，只是它使用贝叶斯边际似然评分标准（在尖峰和平板模型下）而不是最小二乘目标。此外，它使用波束搜索的形式一次探索穿过晶格的多条路径。

13.2.3.2 随机搜索

如果我们想要近似后验概率，而不仅仅是计算模式（例如，因为我们想要计算边缘包含概率），一种选择是使用多通道蒙特卡洛方法。标准方法是使用 Metropolis Hastings，其中提案分发仅翻转单个位。这使我们能够有效地计算给定 $p(\gamma'|\mathcal{D})$ 的 $p(\gamma|\mathcal{D})$。通过计算随机游动访问该状态的次数来估计状态（位配置）的

概率。有关此类方法的综述，请参见（O'Hara 和 Sillanpaa 2009），以及（Bottolo 和 Richardson 2010），了解基于进化 MCMC 的最新方法。

然而，在离散状态空间中，多通道蒙特卡洛方法不必要地低效，因为我们可以直接使用 $p(\gamma, \mathcal{D}) = exp(-f(\gamma))$ 来计算状态的（非规范化）概率；因此，没有必要再次访问一个国家。一种更有效的替代方法是使用某种随机搜索算法，生成一组高分模型，然后进行以下近似

$$p(\gamma|\mathcal{D}) \approx \frac{e^{-f(\gamma)}}{\sum_{\gamma' \in S} e^{-f(\gamma')}} \tag{13.31}$$

请参阅（Heaton 和 Scott，2009 年），了解此类最新方法的综述。

13.2.3.3 EM 与变分推理 *

很容易将 EM 应用于尖峰和平板模型，其形式为 $\gamma_j \to w_j \to \mathbf{y}$，我们可以在 E 步中计算 $p(\gamma_j = 1|w_j)$，并在 M 步中优化 $\mathbf{w}$。然而，这是行不通的，因为当我们计算 $p(\gamma_j = 1|w_j)$ 时，我们比较了 δ 函数 $\delta_0(w_j)$ 和高斯 pdf,$\mathcal{N}(w_j|0, \sigma_w^2)$ 。我们可以用窄的高斯函数代替 δ 函数，然后 E 步相当于在两种可能的高斯模型下对 w_j 进行分类。然而，这可能会受到严重的局部极小值的影响。

另一种方法是将 EM 应用于伯努利-高斯模型，其形式为 $\gamma_j \to \mathbf{y} \leftarrow w_j$。在这种情况下，后验 $p(\gamma|\mathcal{D}, \mathbf{w})$ 难以计算，因为所有位由于解释而变得相关。然而，可以导出形式为 $\prod_j q(\gamma_j)q(w_j)$ 的平均场近似（Huang 等人，2007；Rattray 等人，2009）。

2.3 ℓ_1 正则化：基础知识

当我们有许多变量时,计算上很难找到 $p(\gamma|\mathcal{D})$ 的后验模式。虽然贪婪算法通常工作良好（参见 zhang2008 的理论分析），但它们当然会陷入局部最优。

部分问题是由于 γ_j 变量是离散的，$\gamma_j \in \{0, 1\}$。在优化领域，通常通过用连续变量替换离散变量来放松这种形式的硬约束。我们可以通过替换尖峰和板状先验来实现这一点，尖峰和板状先验将有限概率质量分配给 $w_j = 0$ 的事件，并通过在原点附近放置大量概率密度来“鼓励”$w_j = 0$ 的连续先验，例如零平均拉普拉斯分布。这在第 7.4 节稳健线性回归的背景下首次介绍。在那里，我们利用了拉普拉斯有重尾巴的事实。在这里，我们利用了它在 $\mu = 0$ 附近有一个尖峰的事实。更准确地说，考虑该形式的先验

$$p(\mathbf{w}|\lambda) = \prod_{j=1}^{D} Lap(w_j|0, 1/\lambda) \propto \prod_{j=1}^{D} e^{-\lambda|w_j|} \tag{13.32}$$

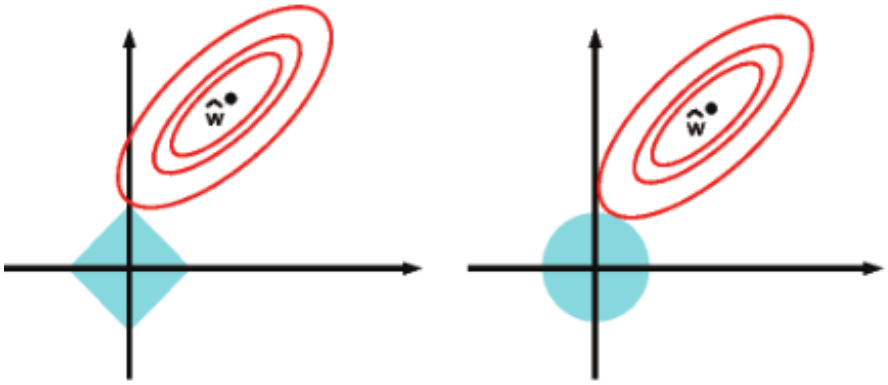


图 13.3: 演示最小二乘问题的 ℓ_1（左）和 ℓ_2（右）正则化。根据（Hastie 等人，2001 年）的图 3.12。

我们将使用一个关于偏移项的统一先验，$p(w_0) \propto 1$。让我们用这个先验来进行 MAP 估计。这个先验。惩罚性负对数似然有如下形式

$$f(\mathbf{w}) = -\log p(\mathcal{D}|\mathbf{w}) - \log p(\mathbf{w}|\lambda) = NLL(\mathbf{w}) + \lambda||\mathbf{w}||_1 \tag{13.33}$$

其中 $||\mathbf{w}||_1 = \sum_{j=1}^{D} |w_j|$ 是 $\mathbf{w}$ 的 ℓ_1 范数。对于适当大的 λ，估计 $\hat{\mathbf{w}}$ 将是稀疏的，原因我们解释如下。事实上，这可以被认为是非凸的凸近似 ℓ_0 目标

$$\arg\min_{\mathbf{w}} NLL(\mathbf{w}) + \lambda||\mathbf{w}||_0 \tag{13.34}$$

在线性回归的情况下 ℓ_1 目标变成

$$f(\mathbf{w}) = \sum_{i=1}^{N} -\frac{1}{2\sigma^2}(y_i - (w_0 + \mathbf{w}^T\mathbf{x}_i))^2 + \lambda||\mathbf{w}||_1 \tag{13.35}$$

$$= RSS(\mathbf{w}) + \lambda'||\mathbf{w}||_1 \tag{13.36}$$

其中 $\lambda' = 2\lambda\sigma^2$。这种方法称为**基追踪去噪或 BPDN**（Chen 等人，1998）。这个术语的原因将在后面变得清晰。一般来说，将零均值的拉普拉斯先验放在参数上的技术拉普拉斯先验参数并进行 MAP 估计的技术被称为正则化。它可以与任何凸的或非凸的 NLL 项相结合。许多不同的算法我们在第 13.4 节中对其中的一些算法进行了回顾。

2.3.1 为什么 ℓ_1 正则化产生稀疏解？

我们现在解释为什么 ℓ_1 正则化导致稀疏解，然而 ℓ_2 正则化不适用。我们关注线性回归的情况，尽管类似的论点适用于逻辑回归和其他 GLM。

目标是以下非光滑目标函数：

$$\min_{\mathbf{w}} RSS(\mathbf{w}) + \lambda||\mathbf{w}||_1 \tag{13.37}$$

我们可以将其改写为约束但平滑的目标（具有线性约束的二次函数）：

$$\min_{\mathbf{w}} RSS(\mathbf{w}) \quad s.t. \quad ||\mathbf{w}||_1 \leq B \tag{13.38}$$

其中，B 是权重 ℓ_1 范数的上界：小（紧）界 B 对应于大惩罚 λ，反之亦然。[6]等式 13.38 被称为 **lasso**，代表“最小绝对收缩和选择算子”（Tibshirani 1996）。我们稍后将了解它为什么有这个名称。

类似地，我们可以写成岭回归

$$\min_{\mathbf{w}} RSS(\mathbf{w}) + \lambda||\mathbf{w}||_2^2 \tag{13.39}$$

或作为绑定约束形式：

$$\min_{\mathbf{w}} RSS(\mathbf{w}) \quad s.t. \quad ||\mathbf{w}||_2^2 \leq B \tag{13.40}$$

在图 13.3 中，我们绘制了 RSS 目标函数的轮廓，以及 ℓ_2 和 ℓ_1 约束曲面。根据约束优化理论，我们知道最优解发生在目标函数的最低水平集与约束曲面相交的点（假设约束是活动的）。几何上应该很清楚，当我们放松约束 B 时，我们“成长”ℓ_1“球”，直到它达到目标；球的角比边更可能与椭圆相交，尤其是在高维情况下，因为角“突出”更多。角点对应于位于坐标轴上的稀疏解。相比之下，当我们成长的时候 ℓ_2 球，它可以在任何点与目标相交；没有“角”，因此不喜欢稀疏性。

为了避免这种情况，请注意，对于岭回归，稀疏解的先验成本（如 $\mathbf{w} = (1, 0)$ 与稠密解的成本（如 $\mathbf{w} = (1/\sqrt{2}, 1/\sqrt{2})$ 相同，只要他们有相同的 ℓ_2 规范：

$$||(1, 0)||_2 = ||(1/\sqrt{2}, 1/\sqrt{2})||_2 = 1 \tag{13.41}$$

然而，对于 lasso，设置 $\mathbf{w} = (1, 0)$ 比设置 $\mathbf{w} = (1/\sqrt{2}, 1/\sqrt{2})$ 便宜，因此

$$||(1, 0)||_1 = 1 < ||(1/\sqrt{2}, 1/\sqrt{2})||_1 = \sqrt{2} \tag{13.42}$$

查看 ℓ_1 正则化导致稀疏解的最严格方法是检查保持在最佳状态的条件。我们在第 13.3.2 节中这样做。

[6]等式 13.38 是**二次规划或 QP** 的示例，因为我们有一个受线性不等式约束的二次目标。其拉格朗日由等式 13.37 给出。

2.3.2 lasso 的最优性条件

lasso 目标的形式是

$$f(\boldsymbol{\theta}) = RSS(\boldsymbol{\theta}) + \lambda||\mathbf{w}||_1 \tag{13.43}$$

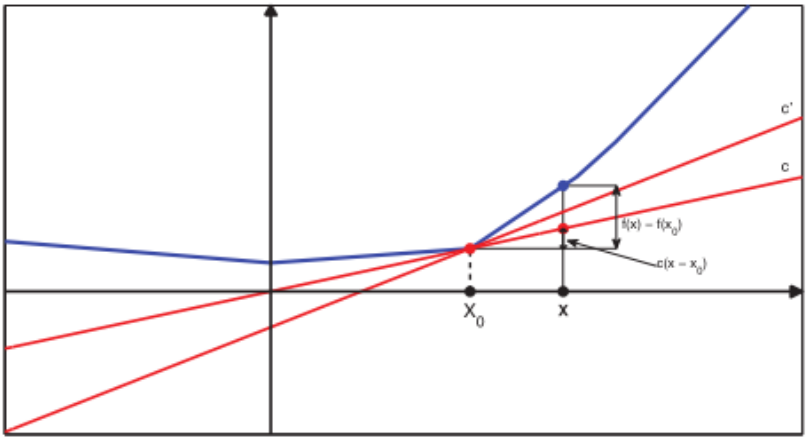


图 13.4: 基于 http://en.wikipedia.org/wiki/Subderivative 上的图，对点 x0 处函数的一次导数进行了说明。由 subgradientPlot 生成的图形。

不幸的是，当 $w_j = 0$ 时，$||\mathbf{w}||_1$ 项不可微。这是一个**非光滑**优化问题的示例。

为了处理非光滑函数，我们需要扩展导数的概念。我们定义（凸）函数 $f : I \to \mathbb{R}$ 在 θ_0 点的**次导数或次梯度**为标量 g，使得

$$f(\theta) - f(\theta_0) \geq g(\theta - \theta_0) \; \forall \theta \in I \tag{13.44}$$

其中 $\mathbf{I}$ 是包含 θ_0 的某个区间。见图 13.4。[7]我们将子导数集定义为区间 [a，b]，其中 a 和 b 是单侧极限

$$a = \lim_{\theta \to \theta_0^-} \frac{f(\theta) - f(\theta_0)}{\theta - \theta_0}, b = \lim_{\theta \to \theta_0^+} \frac{f(\theta) - f(\theta_0)}{\theta - \theta_0} \tag{13.46}$$

所有子导数的集合 [a，b] 称为函数 f 在 θ_0 处的**次微分**，并表示为 $\partial f(\theta)|_{\theta_0}$。例如，在绝对值函数 $f(\theta) = |\theta|$ 的情况下，次导数由以下公式得出：

$$\partial f(\theta) = \begin{cases} \{-1\} & if \; \theta < 0 \\ [-1, 1] & if \; \theta = 0 \\ \{+1\} & if \; \theta > 0 \end{cases} \tag{13.47}$$

如果函数处处可微，那么 $\partial f(\theta) = \{\frac{df(\theta)}{d\theta}\}$。通过类比标准微积分结果，可以证明点 $\hat{\theta}$ 是 f iff $0 \in \partial f(\theta)|\hat{\theta}$ 的局部极小值。

让我们将这些概念应用于 lasso 问题。让我们先忽略非光滑惩罚项。可以证明（练习 13.1）

$$\frac{\partial}{\partial w_j} RSS(\mathbf{w}) = a_j w_j - c_j \tag{13.48}$$

$$a_j = 2 \sum_{i=1}^{n} x_{ij}^2 \tag{13.49}$$

$$c_j = 2 \sum_{i=1}^{n} x_{ij}(y_i - \mathbf{w}_{-j}^T \mathbf{x}_{i,-j}) \tag{13.50}$$

[7]一般来说，对于一个矢量函数，如果对于所有矢量 $\boldsymbol{\theta}$，我们说 g 是 f 在 $\boldsymbol{\theta}_0$ 处的子梯度。

$$f(\boldsymbol{\theta}) - f(\boldsymbol{\theta}_0) \geq (\boldsymbol{\theta} - \boldsymbol{\theta}_0)^T \mathbf{g} \tag{13.45}$$

因此 g 是 $\boldsymbol{\theta}_0$ 处函数的线性下界。

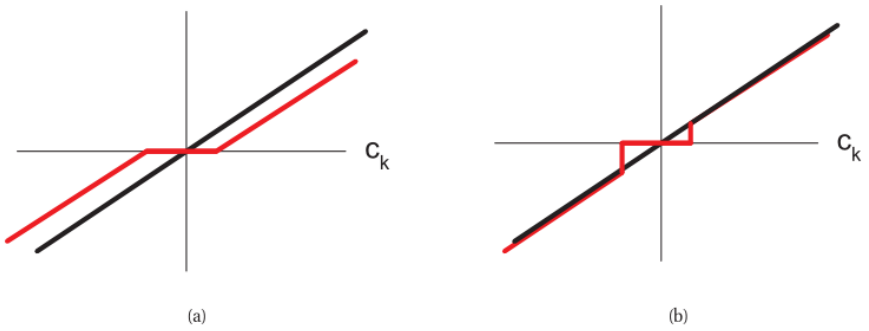


图 13.5: 左图：软阈值处理。平坦区域是区间 $[-\lambda, +\lambda]$。右图：硬阈值处理。

其中 $\mathbf{w}_{-j}$ 是不含分量 j 的 $\mathbf{w}$，类似地，对于 $\mathbf{x}_{i,-j}$，我们看到，c_j 与第 j 个特征 $\mathbf{x}$，j 和其他特征 $\mathbf{r}_{-j} = \mathbf{y} - \mathbf{X}_{:,-j}\mathbf{w}_{-j}$ 的残差之间的相关性（成比例），因此，c_j 的幅值指示了特征 j 预测 $\mathbf{y}$ 的相关程度（相对于其他特征和当前参数）。

加上惩罚项，我们发现次导数由以下公式得出：

$$\partial_{w_j} f(\mathbf{w}) = (a_j w_j - c_j) + \lambda \partial_{w_j} ||\mathbf{w}||_1 \tag{13.51}$$

$$= \begin{cases} \{a_j w_j - c_j - \lambda\} & \textbf{if } w_j < 0 \\ [-c_j - \lambda, -c_j + \lambda] & \textbf{if } w_j = 0 \\ \{a_j w_j - c_j - \lambda\} & \textbf{if } w_j > 0 \end{cases} \tag{13.52}$$

我们可以用一种更紧凑的方式来写，如下所示：

$$\mathbf{X}^T(\mathbf{X}\mathbf{w} - \mathbf{y})_j \in \begin{cases} \{-\lambda\} & \textbf{if } w_j < 0 \\ [-\lambda, \lambda] & \textbf{if } w_j = 0 \\ \{\lambda\} & \textbf{if } w_j > 0 \end{cases} \tag{13.53}$$

根据 c_j 的值，$\partial_{w_j} f(\mathbf{w}) = 0$ 的解可能出现在 3 个不同的值上 w_j 的不同值，如下所示:

1. 如果 $c_j < -\lambda$, 因此，特征与残差呈强负相关，则次梯度在 $\hat{w}_j = \frac{c_j + \lambda}{a_j} < 0$ 时为零。
2. 如果 $c_j \in [-\lambda, \lambda]$，因此特征仅与残差弱相关，则次梯度在 $\hat{w}_j = 0$ 时为零。
3. 如果 $c_j > \lambda$，则特征与残差呈强正相关，则次梯度在 $\hat{w}_j = \frac{c_j - \lambda}{a_j} > 0$ 处为零。

总之，我们有

$$\hat{w}_j(c_j) = \begin{cases} (c_j + \lambda)/a_j & if\ c_j < -\lambda \\ 0 & if\ c_j \in [-\lambda, \lambda] \\ (c_j - \lambda)/a_j & if\ c_j > \lambda \end{cases} \tag{13.54}$$

我们可以这样写：

$$\hat{w}_j = soft(\frac{c_j}{a_j}; \frac{\lambda}{a_j}) \tag{13.55}$$

其中

$$soft(a; \delta) \triangleq sign(a)(|a| - \delta)_+ \tag{13.56}$$

$x_+ = max(x, 0)$ 是 x 的正部分。这称为**软阈值**。这如图 13.5（a）所示，其中我们绘制了 $\hat{w}_j$ 与 c_j。虚线是对应于最小二乘拟合的线 $w_j = c_j/a_j$。实线表示正则化估计 $\hat{w}_j(c_j)$，将虚线向下（或向上）移动 λ，除非 $-\lambda \leq c_j \leq \lambda$, 在这种情况下，它将设置为 $w_j = 0$。

相比之下，在图 13.5（b）中，我们说明了**硬阈值**。这将 w_j 的值设置为 $-\lambda \leq c_j \leq \lambda$, 但它不会将 w_j 的值收缩到该区间之外。软阈值线的斜率与对角线不重合，这意味着即使较大的系数也会向零收缩；因此，

lasso 是一种有偏估计。这是不可取的，因为如果可能性（通过 c_j）表明系数 w_j 应该很大，我们不想缩小它。我们将在第 13.6.2 节中更详细地讨论这个问题。

现在我们终于可以理解为什么 Tibshirani 在（Tibshirani1996）中发明了术语“lasso”：它代表“最小绝对选择和收缩算子”，因为它选择变量的子集，并通过惩罚绝对值来收缩所有系数。如果 $\lambda = 0$，我们得到 OLS 解（最小 ℓ_1 范数）。如果 $\lambda \geq \lambda_{max}$，我们得到 $\hat{\mathbf{w}} = 0$，其中

$$\lambda_{max} = ||\mathbf{X}^T\mathbf{y}||_\infty = \max_j |\mathbf{y}^T\mathbf{x}_{:,j}| \tag{13.57}$$

这个值是利用这样一个事实来计算的：如果 $(\mathbf{X}^T\mathbf{y})_j \in [-\lambda, \lambda]$, 对所有 j 来说，$\mathbf{0}$ 是最优的。ℓ_1 正则化目标的最大惩罚是

$$\lambda_{max} = \max_j |\nabla_j NLL(\mathbf{0})| \tag{13.58}$$

2.3.3 最小二乘法、lasso、山脊法和子集选择法的比较

通过将 ℓ_1 正则化与最小二乘法进行比较，我们可以进一步了解 ℓ_1 正则化，以及 ℓ_2 和 ℓ_0 正则化最小二乘。为了简单起见，假设 $\mathbf{X}$ 的所有特征都是正交的，因此 $\mathbf{X}^T\mathbf{X} = \mathbf{I}$。在这种情况下，RSS 由以下公式得出：

$$RSS(\mathbf{w}) = ||\mathbf{y} - \mathbf{X}\mathbf{w}||^2 = \mathbf{y}^T\mathbf{y} + \mathbf{w}^T\mathbf{X}^T\mathbf{X}\mathbf{w} - 2\mathbf{w}^T\mathbf{X}^T\mathbf{y} \tag{13.59}$$

$$= const + \sum_k w_k^2 - 2\sum_k \sum_i w_k x_{ik} y_i \tag{13.60}$$

我们看到它分解成一个项的总和，每个维度一个。因此，我们可以分析地写下 MAP 和 ML 估计，如下所示：

- **极大似然估计** OLS 解由以下公式得出：

$$\hat{w}_k^{OLS} = \mathbf{x}_{:k}^T\mathbf{y} \tag{13.61}$$

  其中，$\mathbf{x}_{:k}$ 是 $\mathbf{X}$ 的第 k 列。这与等式 13.60 非常相似。我们可以看到 $\hat{w}_k^{OLS}$ 只是特征 k 在响应向量上的正交投影（见第 7.3.2 节）。

- **Ridge** 人们可以证明，Ridge 的估计值是由以下公式给出的

$$\hat{w}_k^{ridge} = \frac{\hat{w}_k^{OLS}}{1+\lambda} \tag{13.62}$$

- **Lasso** 来自等式 13.55，利用 $a_k = 2$ 和 $\hat{w}_k^{OLS} = c_k/2$ 的事实，我们得到

$$\hat{w}_k^{lasso} = sign(\hat{w}_k^{OLS})(|\hat{w}_k^{OLS}| - \frac{\lambda}{2})_+ \tag{13.63}$$

  这对应于软阈值，如图 13.5（a）所示。

- **子集选择**如果我们使用子集选择选择最佳 K 个特征，参数估计如下

$$\hat{w}_k^{SS} = \begin{cases} \hat{w}_k^{OLS} & if\ rank(|\hat{w}_k^{OLS}|) \leq K \\ 0 & othersize \end{cases} \tag{13.64}$$

  其中秩是指权重大小排序列表中的位置。这对应于硬阈值，如图 13.5（b）所示。

图 13.6（a）绘制了 14 次多项式 lasso 的均方误差与 λ 的关系，图 13.6（b）绘制了均方误差与多项式阶数的关系。我们看到，lasso 给出了与子集选择方法类似的结果。

再举一个例子，考虑一个关于前列腺癌的数据集。我们有 $D = 8$ 个特征和 $N = 67$ 个训练案例；目的是预测对数前列腺特异性抗原水平（更多生物学细节见（Hastie 等人，2009 年，p4））。表 13.1 表明，lasso 比最小二乘法、岭回归和最佳子集回归具有更好的预测精度（至少在这个特定数据集上）。（在每种情况下，正则化器的强度都是通过交叉验证来选择的。）Lasso 还产生了稀疏解。当然，对于其他问题，岭可能提供更好的预测准确性。实际上，lasso 和脊的组合，即弹性网，通常表现最好，因为它提供了稀疏性和正则化的良好组合（见第 13.5.3 节）。

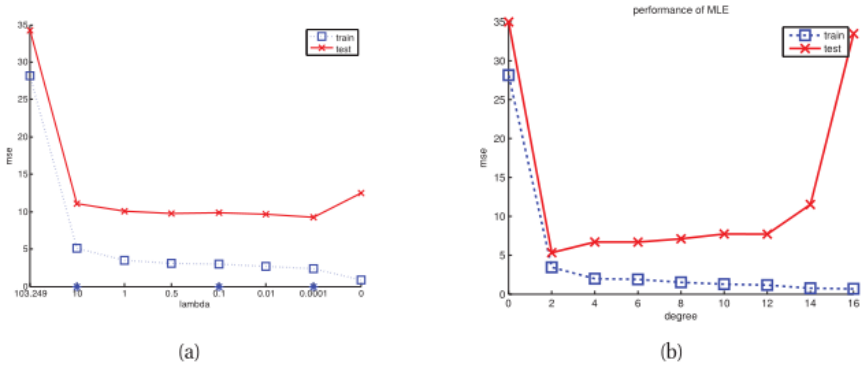


图 13.6:（a）14 次多项式的 lasso 的 MSE 与 λ。注意，当我们向右移动时，λ 减小。由 linregPolyLassoDemo 生成的图。（b）MSE 与多项式次数。请注意，模型阶数随着向右移动而增加。一些多项式回归模型的曲线图见图 1.18。由 linregPolyVsDegree 生成的图。

Term	LS	Best Subset	Ridge	Lasso
Intercept	2.452	2.481	2.479	2.480
lcavol	0.716	0.651	0.656	0.653
lweight	0.293	0.380	0.300	0.297
age	-0.143	-0.000	-0.129	-0.119
lbph	0.212	-0.000	0.208	0.200
svi	0.310	-0.000	0.301	0.289
lcp	-0.289	-0.000	-0.260	-0.236
gleason	-0.021	-0.000	-0.019	0.000
pgg45	0.277	0.178	0.256	0.226
Test Error	0.586	0.572	0.580	0.564

表 13.1: 不同方法对前列腺癌数据的结果，其中有 8 个特征和 67 个训练案例。方法有：最小二乘法，子集 = 最佳子集回归，岭，套索。行表示系数；我们看到子集回归和套索给出了稀疏解。底行是测试集的均方误差（30 例）。根据（Hastie 等人，2009 年）的表 3.3。通过前列腺比较生成的图。

2.3.4 正则化路径

随着 λ 的增加，解向量 $\hat{\mathbf{w}}(\lambda)$ 将趋于稀疏，尽管不一定单调。我们可以绘制每个特征 j 的值 $\hat{w}_j(\lambda)$ 与 λ；这被称为正则化路径。

图 13.7（a）中对 ridge 回归进行了说明，其中我们绘制了 $\hat{w}_j(\lambda)$ 随正则化子 λ 减小的曲线。我们看到当 $\lambda = \infty$, 所有系数均为零。但对于 λ 的任何有限值，所有系数都是非零的；此外，随着 λ 的减小，它们的幅值增大。

在图 13.7（b）中，我们绘制了 lasso 的类似结果。当我们向右移动时，ℓ_1 惩罚的上限 B 增加。当 $B = 0$ 时，所有系数均为零。随着 B 的增加，系数逐渐“开启”。但对于 0 和 $B_{max} = ||\hat{\mathbf{w}}_{OLS}||_1$ 之间的任何值，解是稀疏的。[8]

值得注意的是，可以证明解路径是 B 的分段线性函数（Efron 等人，2004）。也就是说，存在一组 B 的临界值，其中非零系数的活动集发生变化。对于这些临界值之间的 B 值，每个非零系数以线性方式增加或减少。这如图 13.8（a）所示。此外，可以解析求解这些临界值。这是 **LARS** 算法的基础（Efron 等人，

[8] 通常绘制与**收缩因子**（定义为 $s(B) = B/B_{max}$ 的解，而不是与 B 的解。这仅影响水平轴的比例，而不影响曲线的形状。

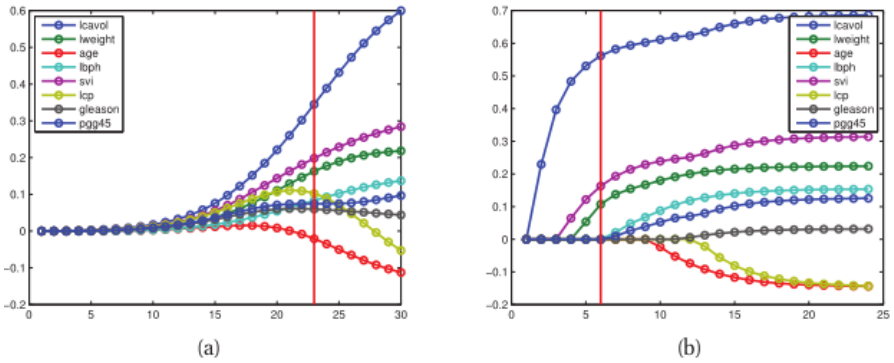


图 13.7（a）前列腺癌示例与约束条件下的 ridge 系数曲线 w 的 ℓ_2 范数，因此小 t（大 λ）在左侧。垂直线是使用 1SE 规则由 5 倍 CV 选择的值。根据（Hastie 等人，2009 年）的图 3.8。由 RidgePathProstation 生成的图。（b）前列腺癌的套索系数曲线示例与绑定 w 的 ℓ_1 范数，因此小 t（大 λ）在左侧。根据（Hastie 等人，2009 年）的图 3.10。图由 LassopathProstation 生成。

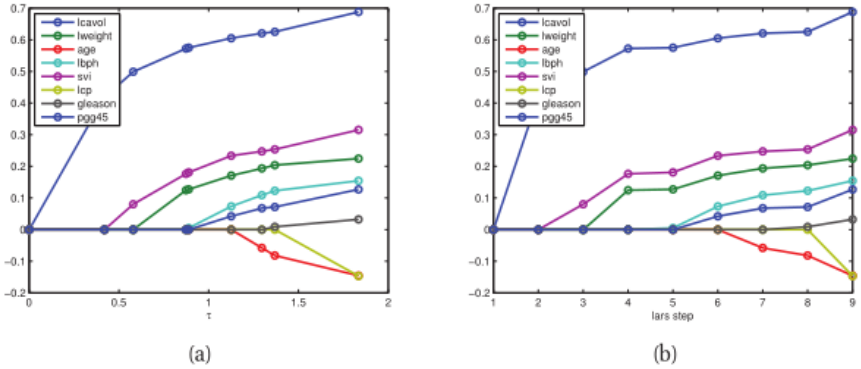


图 13.8: 前列腺癌示例上 lasso 正则化路径的分段线性图示。（a）对于 B 的临界值，我们绘制了 $\hat{w}_j(B)$。（b）我们绘制了 LARS 算法的 vs 步长。图由 LassopathProstation 生成。

2004），该算法代表“最小角度回归和收缩”（详见第 13.4.2 节）。值得注意的是，LARS 可以以与单个最小二乘拟合（即 $O(min(ND^2, DN^2))$ 大致相同的计算成本计算整个正则化路径。

在图 13.8（b）中，我们绘制了在 B 的每个临界值处计算的系数。现在分段线性更为明显。下面我们显示了沿正则化路径每一步的实际系数值（最后一行是最小二乘解）：

通过将 B 从 0 更改为 B_{max}，我们可以从所有权重均为零的解转换为所有权重均为非零的解。不幸的是，并非所有子集大小都可以使用 lasso 实现。可以证明，如果 $D > N$，最优解在达到对应于最小的 OLS 解的完整集之前，最多可以包含 N 个变量 ℓ_1 标准。在第 13.5.3 节中，我们将通过使用 ℓ_2 正则化子以及 ℓ_1 正则化（一种称为弹性网的方法），我们可以获得比训练情况包含更多变量的稀疏解。这使我们能够探索 N 和 D 之间的模型尺寸。

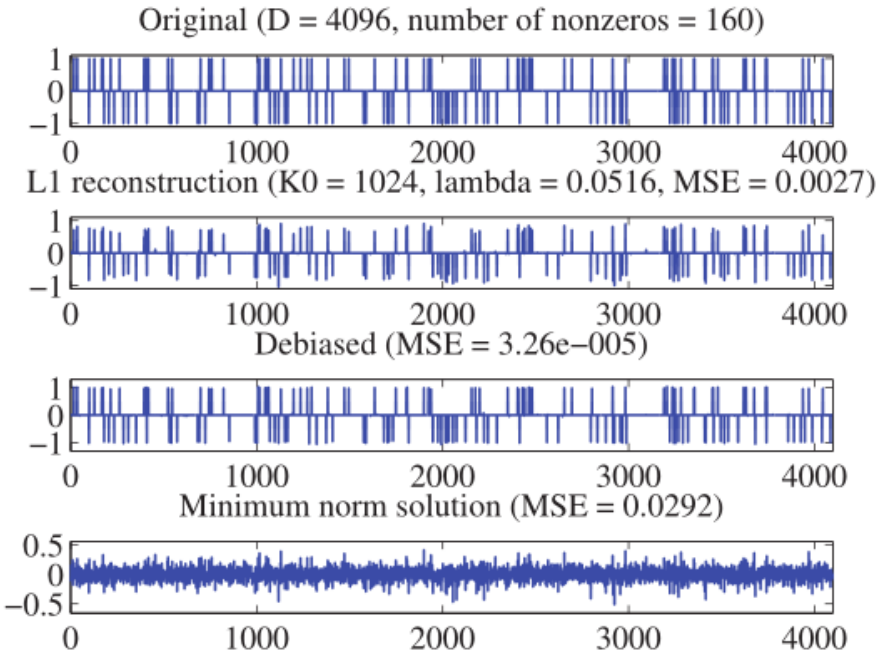


图 13.9: 使用 lasso 恢复稀疏信号的示例。有关详细信息，请参阅文本。根据图 1（Figueiredo 等人，2007）。图由马里奥·菲盖雷多创作的 sparseSensingDemo 生成。

清单 13.1: LassopathProstation 的输出

```
     0         0         0         0         0         0         0         0
0.4279         0         0         0         0         0         0         0
0.5015    0.0735         0         0         0         0         0         0
0.5610    0.1878         0         0    0.0930         0         0         0
0.5622    0.1890         0    0.0036    0.0963         0         0         0
0.5797    0.2456         0    0.1435    0.2003         0         0    0.0901
0.5864    0.2572   -0.0321    0.1639    0.2082         0         0    0.1066
0.6994    0.2910   -0.1337    0.2062    0.3003   -0.2565         0    0.2452
0.7164    0.2926   -0.1425    0.2120    0.3096   -0.2890   -0.0209    0.2773
```

2.3.5 模型选择

很容易使用 ℓ_1 正则化来估计相关变量集。在某些情况下，我们可以恢复 $\mathbf{w}^*$ 的真实稀疏模式, 生成数据的参数向量。一种可以在 $N \to \infty$ 中恢复真实模型的方法称为**模型选择一致**。关于哪些方法以及何时享受该属性的细节超出了本书的范围；详见（Buhlmann 和 van de Geer 2011）。

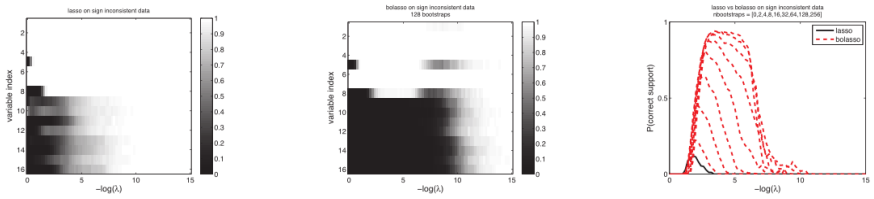
我们将只展示一个小例子，而不是进行理论讨论。我们首先生成一个稀疏信号 $\mathbf{w}^*$ 尺寸 D=4096，由 160 个随机放置的 ± 1 个尖峰组成。接下来，我们生成一个大小为 $N \times D$ 的随机设计矩阵 $\mathbf{X}$，其中 N=1024。最后，我们生成一个噪声观测 $\mathbf{y} = \mathbf{X}\mathbf{w}^* + \boldsymbol{\epsilon}$，其中 $\epsilon_i \sim \mathcal{N}(0, 0.01^2)$。然后我们根据 $\mathbf{y}$ 和 $\mathbf{X}$ 估计 $\mathbf{w}$。

原始 $\mathbf{w}^*$ 如图 13.9 的第一行所示。第二行是 ℓ_1 使用 $\lambda = 0.1\lambda_{max}$ 估计 $\hat{\mathbf{w}}_{L1}$。我们看到在正确的地方有“尖峰”，但它们太小了。第三行是基于 $\text{supp}(\hat{\mathbf{w}}_{L1})$ 估计为非零的系数的最小二乘估计。这被称为 **debiasing**，并且是必要的，因为 lasso 收缩了相关系数以及不相关的系数。最后一行是所有系数的联合最小二乘估计，忽略稀疏性。我们看到，（基于 debiased 的）稀疏估计是对原始信号的一种很好的估计。相比之下，没有稀疏性假设的最小二乘法性能很差。

当然，要进行模型选择，我们必须选择 λ。通常使用交叉验证。然而，需要注意的是，交叉验证是选择 λ 值，从而获得良好的预测精度。这通常与可能恢复“真实”模型的值不同。想知道为什么吗，记得 ℓ_1 正则化执行选择和收缩，也就是说，所选系数更接近 0。为了防止相关系数以这种方式收缩，交叉验证将倾向于选择一个不太大的 λ 值。当然，这将导致包含不相关变量（误报）的较稀疏模型。事实上，在（Meinshausen 和 Buhlmann 2006）中证明，λ 的预测最优值不会导致模型选择的一致性。在第 13.6.2 节中，我们将讨论在每维基础上自动调整 λ 的一些自适应机制，这确实会导致模型选择的一致性。

使用 ℓ_1 正则化选择变量的一个缺点是，如果数据受到轻微扰动，则会得到完全不同的结果。贝叶斯方法估计后验边缘包含概率 $p(\gamma_j = 1|\mathcal{D})$，更稳健。一种常见的解决方案是使用自举重采样（见第 6.2.1 节），并在不同版本的数据上重新运行估计器。通过计算在不同试验中选择每个变量的频率，我们可以近似后验包含概率。这种方法称为**稳定性选择**（Meinshausen 和 BÃijhlmann 2010）。

我们可以在某种程度上设定稳定性选择（bootstrap）包含概率的阈值，例如 90%，并由此导出稀疏估计器。这被称为 **bootstrap 或 bolasso**（Bach2008）。如果它发生在 lasso 返回的至少 90% 的集合中（对于固定 λ），则它将包括一个变量。这种相交集合的过程是消除 vanilla lasso 产生的误报的一种方法。（Bach 2008）中的理论结果证明，bolasso 在比 vanilla lasso 更广泛的条件下是模型选择一致的。



(a) (b) (c)

图 13.10:（a）每个变量的选择概率（白色 = 大概率，黑色 = 小概率）与 lasso 正则化参数的关系。当我们从左向右移动时，我们减少正则化的量，因此选择更多的变量。（b）与（a）相同，但适用于 bolasso。（c）符号正确估计的概率与正则化参数。Bolasso（红色，虚线）和 Lasso（黑色，普通）：引导复制的数量在 {2，4，8，16，32，64，128，256}。基于（Bach 2008）的图 1-3。图由 bolassoDemo 生成。

作为说明，我们复制了（Bach 2008）中的实验。特别是，我们创建了 256 个大小为 N=1000 的数据集，其中 D=16 个变量，其中 8 个是相关的。关于实验设置的更多细节，见 (Bach 2008)。对于数据集 n，变量 j 和稀疏程度 k，定义 $S(j,k,n) = \mathbb{I}(\hat{w}_j(\lambda_k, \mathcal{D}_n) \neq 0)$。现在定义 $P(j,k)$ 为 256 个数据集上 $S(j,k,n)$ 的平均值。在图 13.10（a-b）中，我们绘制了 lasso 和 bolasso 的 P 与 $-\log(\lambda)$ 的关系。我们看到，对于 bolasso 来说，有一个很大的 λ 范围，在这个范围内，真正的变量被选中，但对于 lasso 来说，情况并非如此。这一点在图 13.10(c) 中得到了强调，我们在图中绘制了拉索和博拉索的正确变量集被恢复的经验概率，随着引导样本数量的增加。当然，使用更多的样本需要更长的时间。在实践中，32 个自举样本似乎是速度

和准确性之间的一个很好的折中。

对于 bolasso，通常存在选择 λ 的问题。显然，我们可以使用交叉验证，但图 13.10（b）等图提出了另一种启发式方法：洗牌行以创建一个大的黑色块，然后选择 λ 位于该区域的中间。当然，操作这种直觉可能很棘手，需要各种特殊阈值（这让人想起第 11.5.2 节讨论如何为混合模型选择 K 时讨论的“在曲线中找到拐点”启发式）。贝叶斯方法为选择 λ 提供了更具原则性的方法。

2.3.6 具有拉普拉斯先验的线性模型的贝叶斯推理

我们一直专注于稀疏线性模型中的 MAP 估计。也可以进行贝叶斯推理（例如，参见（Park 和 Casella 2008；Seeger 2008））。然而，后验均值和中位数以及后验样本并不稀疏；只有模式是稀疏的。这是第 5.2.1 节中讨论的现象的另一个例子，在该节中，我们说 MAP 估计通常不典型于大部分后验值。

支持使用后验平均值的另一个论点来自等式 5.108，该等式表明，如果我们想要最小化平方预测误差，插入后验平均值而不是后验模式是最佳做法。（Schniter 等人，2008）在实验上表明，（Elad 和 Yavnch，2009）在理论上表明，使用具有尖峰和平板先验的后验平均值比使用具有拉普拉斯先验的后验模式具有更好的预测精度，尽管计算成本略高。

2.4 ℓ_1 正则化：算法

在本节中，我们简要回顾了一些可用于求解的算法 ℓ_1 正则化估计问题。我们关注的是 lasso 情况，在这里我们有一个二次损失。然而，大多数算法可以扩展到更一般的设置，如逻辑回归（参见（Yaun 等人，2010 年），以全面审查 ℓ_1 正则化逻辑回归）。请注意，机器学习的这一领域进展非常迅速，因此在阅读本章时，以下方法可能还不是最先进的。（见（Schmidt 等人，2009；Yaun 等人，2010；Yang 等人 2010）最近的一些调查。）

2.4.1 坐标下降

有时很难同时优化所有变量，但很容易逐个优化。特别是，我们可以在所有其他系数保持不变的情况下求解第 j 系数：

$$w_j^* = \arg\min_z \; f(\mathbf{w} + z\mathbf{e}_j) - f(\mathbf{w}) \tag{13.65}$$

其中 $\mathbf{e}_j$ 是第 j 个单位向量。我们可以以确定的方式在坐标中循环，也可以随机采样，或者可以选择更新梯度最陡的坐标。

如果每个一维优化问题都可以解析求解，则坐标下降法尤其有吸引力。例如，lasso 的**射击算法**（Fu 1998；Wu 和 Lange 2008）使用等式 13.54 计算给定所有其他系数的 w_j 的最优值。伪代码见算法 7（一些 Matlab 代码见 LassoShooting）。

参见（Yaun 等人，2010），了解该方法对逻辑回归案例的一些扩展。由此产生的算法是他们实验比较中最快的方法，它涉及到具有大量稀疏特征向量（代表一袋单词）的文档分类。其他类型的数据（例如密集特征和/或回归问题）可能需要不同的算法。

Algorithm 13.1 lasso 坐标下降（又名射击算法）

初始化 $\mathbf{w} = (\mathbf{X}^T\mathbf{X} + \lambda\mathbf{I})^{-1}\mathbf{X}^T\mathbf{y}$
重复
  for $j = 1,...,D$ **do**
    $a_j = 2\sum_{i=1}^n x_{ij}^2$;
    $c_j = 2\sum_{i=1}^n x_{ij}(y_i - \mathbf{w}^T\mathbf{x}_i + w_j x_{ij})$;
    w_j=soft$(\frac{c_j}{a_j}, \frac{\lambda}{a_j})$;
直到收敛

2.4.2 LARS 和其他同构方法

坐标下降的问题是，它一次只更新一个变量，因此收敛速度较慢。**活动集**方法一次更新许多变量。不幸的是，它们更复杂，因为需要识别哪些变量被约束为零，哪些变量可以自由更新。

活动集方法通常一次只添加或删除几个变量，因此如果开始时远离解决方案，则可能需要很长时间。但它们非常适合从空集开始，为 λ 的不同值生成一组解，即生成正则化路径。这些算法利用了一个事实，即可以从 $\hat{\mathbf{w}}(\lambda_k)$ 快速计算 $\hat{\mathbf{w}}(\lambda_{k-1})$ 如果 $\lambda_k \approx \lambda_{k-1}$. 这被称为**温启动**。事实上，即使我们只需要 λ 的单个值的解，也可以称之为 λ_*, 有时计算一组解（从 λ_{max} 到 λ_*）在计算上更有效, 使用温启动；这被称为延拓方法或同伦方法。这通常比在 λ_* 处直接“冷启动”快得多; 尤其如果 λ_* 是小的。

机器学习中同构方法最著名的例子可能是 **LARS** 算法，它代表“最小角度回归和收缩”（Efron 等人，2004）（在（Osborne 等人，2000b，a）中独立发明了一种类似的算法）。这可以有效地计算 λ 的所有可能值的 $\hat{\mathbf{w}}(\lambda)$.

LARS 的工作如下。它从较大的 λ 值开始，因此仅选择与响应向量 $\mathbf{y}$ 最相关的变量。然后减小 λ，直到找到第二个变量，该变量与第一个变量的电流残差具有相同的相关性（在幅值方面），其中步骤 k 处的残差定义为 $\mathbf{r}_k = \mathbf{y} - \mathbf{X}_{:,F_k}\mathbf{w}_k$, 其中 F_k 是当前**活动集**（c.f.，等式 13.50）。值得注意的是，可以通过使用几何参数（因此称为“最小角度”）解析求解 λ 的这个新值。这使得算法能够快速“跳转”到正则化路径上的下一个点，在该点上活动集发生变化。重复该操作，直到添加所有变量。

如果我们希望解序列对应于 lasso 的正则化路径，则有必要允许从活动集移除变量。如果我们不允许删除变量，我们会得到一个稍微不同的算法，称为 **LAR**，它往往更快。特别是，LAR 的成本与单个普通最小二乘拟合相同，即 $O(NDmin(N,D))$,，如果 $N > D$，则为 $O(ND^2)$，如果 $D > N$，则为 $O(N^2D)$。LAR 非常类似于贪婪正向选择，以及一种称为最小二乘增强的方法（见第 16.4.6 节）。

有许多人试图扩展 LARS 算法来计算全正则化路径 ℓ_1 正则化 GLMs，如逻辑回归。通常，无法解析求解 λ 的临界值。相反，标准方法是从 λ_{max} 开始，然后缓慢减小 λ，在我们前进的过程中跟踪解决方案；这被称为**延拓方法或同构方法**。这些方法利用了这样一个事实，即我们可以从 $\hat{\mathbf{w}}(\lambda_k)$ 快速计算 $\hat{\mathbf{w}}(\lambda_{k-1})$ 如果 $\lambda_k \approx \lambda_{k-1}$. 这被称为**温启动**。即使我们不想要完整的路径，这种方法通常比在所需的 λ 值下直接“冷启动”快得多（如果 λ 很小，尤其如此）。

（Friedman 等人，2010）中描述的方法将坐标下降与该预热启动策略相结合，并计算任何 ℓ_1 正则 GLM。这已在 glmnet 包中实现，该包与 PMTK 捆绑在一起。

2.4.3 近端和梯度投影方法

在本节中，我们考虑了一些适用于超大规模问题的方法，其中同伦方法的速度太慢。除了 ℓ_1，我们稍后将看到。我们在本节中的介绍基于（Vandenberghe 2011；Yang 等人，2010）。

考虑形式的凸目标

$$f(\boldsymbol{\theta}) = L(\boldsymbol{\theta}) + R(\boldsymbol{\theta}) \tag{13.66}$$

其中，$L(\boldsymbol{\theta})$(表示损失) 是凸的且可微的，而 $R(\boldsymbol{\theta})$（表示正则化子）是凸但不一定可微的。例如，$L(\boldsymbol{\theta}) = RSS(\boldsymbol{\theta})$ 和 $R(\boldsymbol{\theta}) = \lambda||\boldsymbol{\theta}||_1$; 对应于 BPDN 问题。作为另一个例子，lasso 问题可以表示为：$L(\boldsymbol{\theta}) = RSS(\boldsymbol{\theta})$ 和 $R(\boldsymbol{\theta}) = \mathbf{I}_C(\boldsymbol{\theta})$, 其中 $C = \{\boldsymbol{\theta} : ||\boldsymbol{\theta}||_1 \leq B\}$，$\mathbf{I}_C(\boldsymbol{\theta})$ 是凸集 C 的指示函数，定义为

$$I_C(\boldsymbol{\theta}) \triangleq \begin{cases} 0 & \boldsymbol{\theta} \in C \\ +\infty & otherwise \end{cases} \tag{13.67}$$

在某些情况下，很容易优化方程 13.66 中形式的函数。例如，假设 $L(\boldsymbol{\theta}) = RSS(\boldsymbol{\theta})$，设计矩阵简单地为 $\mathbf{X} = \mathbf{I}$。然后目标变为 $f(\boldsymbol{\theta}) = R(\boldsymbol{\theta}) + \frac{1}{2}||\boldsymbol{\theta} - \mathbf{y}||_2^2$。其最小值由 $prox_R(\mathbf{y})$ 出，它是凸函数 R 的**近似算子**，定义如下：

$$prox_R(\mathbf{y}) = \arg\min_{\mathbf{z}}(R(\mathbf{z}) + \frac{1}{2}||\mathbf{z} - \mathbf{y}||_2^2) \tag{13.68}$$

直观地说，我们返回的是一个使 R 最小化但也接近（接近）$\mathbf{y}$ 的点。一般来说，我们将在迭代优化器中使用该操作符，在这种情况下，我们希望与前一个迭代保持接近。在这种情况下，我们使用

$$prox_R(\boldsymbol{\theta}_k) = \arg\min_{\mathbf{z}}(R(\mathbf{z}) + \frac{1}{2}||\mathbf{z} - \boldsymbol{\theta}_k||_2^2) \tag{13.69}$$

关键问题是：如何有效地计算不同正则化子 R 的近似算子，以及如何将该技术扩展到更一般的损失函数 L？我们将在下面讨论这些问题。

13.4.3.1 近端操作器

如果 $R(\boldsymbol{\theta}) = \lambda||\boldsymbol{\theta}||_1$，则通过分量软阈值给出近端算子：

$$prox_R(\boldsymbol{\theta}) = soft(\boldsymbol{\theta}, \lambda) \tag{13.70}$$

如我们在第 13.3.2 节中所示，如果 $R(\boldsymbol{\theta}) = \lambda||\boldsymbol{\theta}||_0$; 则通过分量硬阈值给出近似算子：

$$prox_R(\boldsymbol{\theta}) = hard(\boldsymbol{\theta}, \sqrt{2\lambda}) \tag{13.71}$$

其中 $hard(u, a) \triangleq u\mathbb{I}(|u| > a)$

如果 $R(\boldsymbol{\theta}) = \mathbf{I}_C(\boldsymbol{\theta})$，则通过投影到集合 C 上给出近端算子：

$$prox_R(\boldsymbol{\theta}) = \arg\min_{\mathbf{z} \in C} ||\mathbf{z} - \boldsymbol{\theta}||_2^2 = proj_C(\boldsymbol{\theta}) \tag{13.72}$$

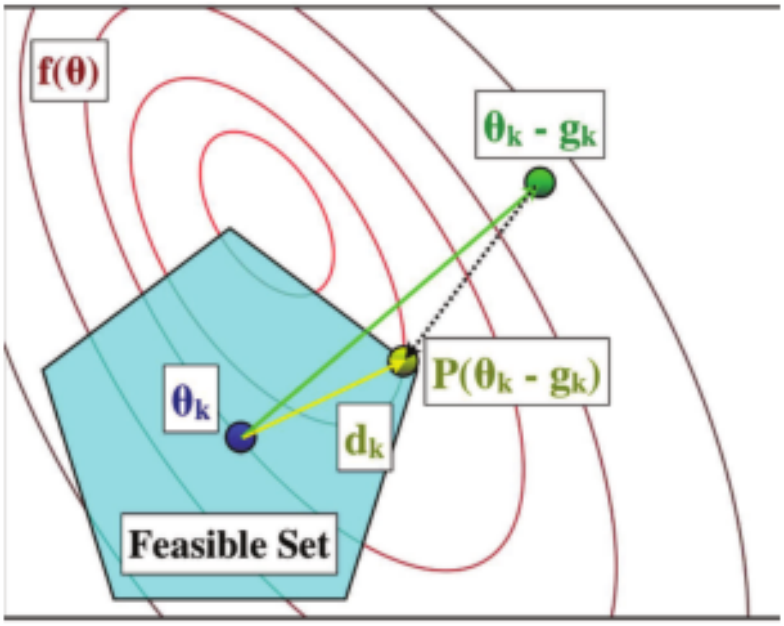


图 13.11: 投影梯度下降示意图。沿负梯度的步长，到 $\boldsymbol{\theta}_k - \mathbf{g}_k$ 将我们带到可行集之外。如果我们将该点投影到集合中最近的点上，我们得到 $\boldsymbol{\theta}_{k+1} = proj_\Theta(\boldsymbol{\theta}_k - \mathbf{g}_k)$。然后，我们可以使用 $\mathbf{d}_k = \boldsymbol{\theta}_{k+1} - \boldsymbol{\theta}_k$ 推导隐式更新方向。经马克 • 施密特许可使用。

对于某些凸集，很容易计算投影算子。例如，投影到长方体约束定义的矩形集 $C = \{\boldsymbol{\theta} : \ell_j \leq \theta_j \leq u_j\}$ 我们可以使用

$$proj_C(\boldsymbol{\theta})_j = \begin{cases} \ell_j & \theta_j \leq \ell_j \\ \theta_j & \ell_j \leq \theta_j \leq u_j \\ u_j & \theta_j \geq u_j \end{cases} \tag{13.73}$$

投影到欧几里得球 $C = \{\boldsymbol{\theta} : ||\boldsymbol{\theta}||_2 \leq 1\}$ 我们可以使用

$$proj_C(\boldsymbol{\theta}) = \begin{cases} \frac{\boldsymbol{\theta}}{||\boldsymbol{\theta}||_2} & ||\boldsymbol{\theta}||_2 > 1 \\ \boldsymbol{\theta} & ||\boldsymbol{\theta}||_2 \leq 1 \end{cases} \tag{13.74}$$

投影到 1-范数球 $C = \{\boldsymbol{\theta} : ||\boldsymbol{\theta}||_1 \leq 1\}$ 我们可以使用

$$proj_C(\boldsymbol{\theta}) = soft(\boldsymbol{\theta}, \lambda) \tag{13.75}$$

其中 $\lambda = 0$，如果 $||\boldsymbol{\theta}||_1 \leq 1$，否则 λ 是方程的解

$$\sum_{j=1}^{D} max(|\theta_j| - \lambda, 0) = 1 \tag{13.76}$$

如（Duchi 等人，2008）所述，我们可以在 $O(D)$ 时间内实现整个过程。

我们将在第 13.5.1.2 节中看到这些不同投影方法的应用。

13.4.3.2 近端梯度法

我们现在讨论如何在梯度下降例程中使用近似算子。基本思想是最小化损失函数的简单二次近似，$\boldsymbol{\theta}_k$ 为中心

$$\boldsymbol{\theta}_{k+1} = \arg\min_{\mathbf{z}} R(\mathbf{z}) + L(\boldsymbol{\theta}_k) + \mathbf{g}_k^T(\mathbf{z} - \boldsymbol{\theta}_k) + \frac{1}{2t_k}||\mathbf{z} - \boldsymbol{\theta}_k||_2^2 \tag{13.77}$$

其中 $\mathbf{g}_k = \nabla L(\boldsymbol{\theta}_k)$ 是损耗的梯度，t_k 是下面讨论的常数，最后一项来自于对形式损耗的海森近似 $\nabla^2 L(\boldsymbol{\theta}_k) \approx \frac{1}{t_k}\mathbf{I}$.

去掉与 $\mathbf{z}$ 无关的项，然后乘以 t_k，我们可以根据近似算子重写上述表达式，如下所示：

$$\boldsymbol{\theta}_{k+1} = \arg\min_{\mathbf{z}} \left[t_k R(\mathbf{z}) + \frac{1}{2}||\mathbf{z} - \mathbf{u}_k||_2^2 \right] = prox_{t_k R}(\mathbf{u}_k) \tag{13.78}$$

$$\mathbf{u}_k = \boldsymbol{\theta}_k - t_k \mathbf{g}_k \tag{13.79}$$

$$\mathbf{g}_k = \nabla L(\boldsymbol{\theta}_k) \tag{13.80}$$

如果 $R(\boldsymbol{\theta}) = 0$，这相当于**梯度下降**。如果 $R(\boldsymbol{\theta}) = \mathbf{I}_C(\boldsymbol{\theta})$，则该方法等效于投影梯度下降，如图 13.11 所示。如果 $R(\boldsymbol{\theta}) = \lambda||\boldsymbol{\theta}||_1$，则该方法称为**迭代软阈值**。

有几种方法可以选择 t_k，或者等效地，$\alpha_k = 1/t_k$。假设 $\alpha_k \mathbf{I}$ 是 Hessian 方程的近似值 $\nabla^2 L$，我们需要

$$\alpha_k(\boldsymbol{\theta}_k - \boldsymbol{\theta}_{k-1}) \approx \mathbf{g}_k - \mathbf{g}_{k-1} \tag{13.81}$$

在最小二乘意义上。因此

$$\alpha_k = \arg\min_{\alpha} ||\alpha(\boldsymbol{\theta}_k - \boldsymbol{\theta}_{k-1}) - (\mathbf{g}_k - \mathbf{g}_{k-1})||_2^2 = \frac{(\boldsymbol{\theta}_k - \boldsymbol{\theta}_{k-1})^T(\mathbf{g}_k - \mathbf{g}_{k-1})}{(\boldsymbol{\theta}_k - \boldsymbol{\theta}_{k-1})^T(\boldsymbol{\theta}_k - \boldsymbol{\theta}_{k-1})} \tag{13.82}$$

这被称为 **Barzilai-Borwein**（BB）或**光谱步长**（Barzilay 和 Borwein1988；Fletcher 2005；Raydan 1997）。该步长可用于任何梯度方法，无论是否接近。它不会导致目标的单调减少，但比标准的线搜索技术快得多。（为了确保收敛，我们要求目标“平均”减少，其中平均值是在大小为 M+1 的滑动窗口上计算的。）

当我们将 BB 步长与迭代软阈值技术（对于 $R(\boldsymbol{\theta}) = \lambda||\boldsymbol{\theta}||_1$ 结合起来，再加上逐渐减小 λ 的连续方法时，我们得到了一种用于 BPDN 问题的快速方法，称为 SpaRSA 算法，它代表“通过可分离近似进行稀疏重建”（Wright 等人，2009）。然而，我们将其称为迭代收缩和阈值算法。一些伪代码见算法 12，一些 Matlab 代码见 SpaRSA。另请参见练习 13.11，了解基于投影梯度下降的相关方法。

13.4.3.3 Nesterov 方法

通过围绕非最新参数值的点进行二次近似，可以获得更快的近似梯度下降。特别是，考虑更新表单

$$\boldsymbol{\theta}_{k+1} = prox_{t_k R}(\phi_k - t_k \mathbf{g}_k) \tag{13.83}$$

$$\mathbf{g}_k = \nabla L(\phi_k) \tag{13.84}$$

$$\phi_k = \boldsymbol{\theta}_k + \frac{k-1}{k+2}(\boldsymbol{\theta}_k - \boldsymbol{\theta}_{k-1}) \tag{13.85}$$

Algorithm 13.2 迭代收缩阈值算法（ISTA）

输入：$\mathbf{X} \in \mathbb{R}^{N \times D}, \mathbf{y} \in \mathbb{R}^N$ 参数 $\lambda \geq 0, M \geq 1, 0 < s < 1$
初始化 $\boldsymbol{\theta}_0 = \mathbf{0}, \alpha = 1, \mathbf{r} = \mathbf{y}, \lambda_0 = \infty$
重复
    $\lambda_t = max(s||\mathbf{X}^T\mathbf{r}||_\infty, \lambda)$ //调整正则化器
    重复
        $\mathbf{g} = \nabla L(\boldsymbol{\theta})$;
        $\mathbf{u} = \boldsymbol{\theta} - \frac{1}{\alpha}\mathbf{g}$;
        $\boldsymbol{\theta} = soft(\mathbf{u}, \frac{\lambda_t}{\alpha})$
        使用等式 13.82 中的 BB 步长更新 α;
    直到 $f(\boldsymbol{\theta})$ 在过去的 M 步内增加过多
    $\mathbf{r} = \mathbf{y}\mathbf{X}\boldsymbol{\theta}$//更新残差
直到 $\lambda_t = \lambda$

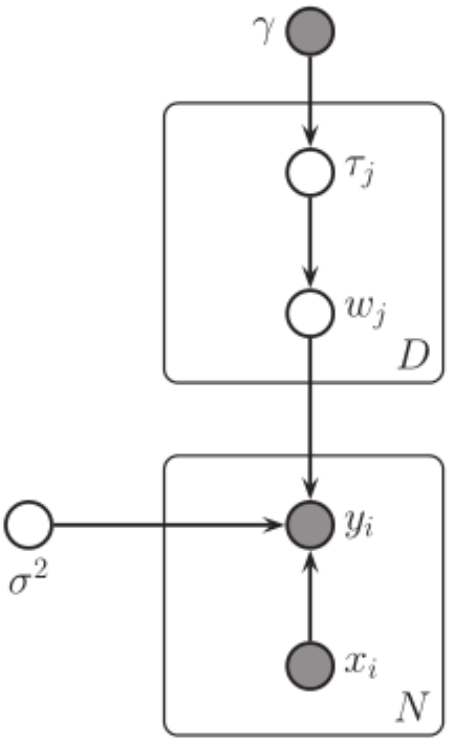


图 13.12: 使用高斯尺度混合表示 lasso。

这就是 **Nesterov 的方法**（Nesterov 2004；Tseng 2008）。和以前一样，有多种设置 t_k 的方法；通常使用线搜索。

当该方法与迭代软阈值技术（对于 $R(\boldsymbol{\theta}) = \lambda||\boldsymbol{\theta}||_1$ 结合，再加上逐渐减小 λ 的连续方法时，我们得到了一种求解 BPDN 问题的快速方法，称为**快速迭代收缩保持算法或 FISTA**（Beck 和 T eboulle 2009）。

2.4.4 EM 代表 lasso

在本节中，我们将展示如何使用 lasso 解决 lasso 问题。乍一看，这似乎有些奇怪，因为没有隐藏的变量。关键的见解是，我们可以将拉普拉斯分布表示为**高斯尺度混合**（GSM）（Andrews and Mallows 1974；West 1987），如下所示：

$$Lap(w_j|0, 1/\gamma) = \frac{\gamma}{2}e^{-\gamma|w_j|} = \int \mathcal{N}(w_j|0, \tau_j^2)Ga(\tau_j^2|1, \frac{\gamma^2}{2})d\tau_j^2 \tag{13.86}$$

因此，拉普拉斯是一个 GSM，其中方差上的混合分布是指数分布，$\text{Expon}(\tau_j^2|\frac{\gamma^2}{2} = Ga(\tau_j^2|1, \frac{\gamma^2}{2}))$。使用这种分解，我们可以表示如图 13.12 所示的套索模型。相应的联合分布具有以下形式

$$\begin{aligned} p(\mathbf{y}, \mathbf{w}, \boldsymbol{\tau}, \sigma^2|\mathbf{X}) &= \mathcal{N}(\mathbf{y}|\mathbf{X}\mathbf{w}, \sigma^2\mathbf{I}_N)\mathcal{N}(\mathbf{w}|\mathbf{0}, \mathbf{D}_r) \\ &\quad \mathbf{IG}(\sigma^2|a_\sigma, b_\sigma)\left[\prod_j Ga(\tau_j^2|1, \gamma^2/2)\right] \end{aligned} \tag{13.87}$$

其中 $\mathbf{D}_\tau = diag(\tau_j^2)$，并且为了简化符号，我们假设 $\mathbf{X}$ 是标准化的，$\mathbf{y}$ 是中心的（因此我们可以忽略偏移项 μ）。通过扩展，我们得到了

$$\begin{aligned} p(\mathbf{y}, \mathbf{w}, \boldsymbol{\tau}, \sigma^2|\mathbf{X}) \propto &(\sigma^2)^{-N/2}exp\left(-\frac{1}{2\sigma^2}||\mathbf{y} - \mathbf{X}\mathbf{w}||_2^2\right)|\mathbf{D}_\tau|^{-\frac{1}{2}} \\ &exp\left(-\frac{1}{2}\mathbf{w}^T\mathbf{D}_\tau\mathbf{w}\right)(\sigma^2)^{-(a_\sigma+1)} \\ &exp(-b_\sigma/\sigma^2)\prod_j exp(-\frac{\gamma^2}{2}\tau_j^2) \end{aligned} \tag{13.88}$$

下面，我们简要描述了如何将 EM 算法应用于图 13.12.[9]中的模型。在 E 步骤中，我们推断 τ_j^2 和 σ^2，在 M 步骤中我们估计 $\mathbf{w}$。所得估计 $\hat{\mathbf{w}}$ 与 lasso 估计相同。这种方法最早是在（Figueiredo，2003 年）提出的（一些扩展请参见（Griffin 和 Brown，2007 年；Caron 和 Doucet，2008 年；Ding 和 Harrison，2010 年）。

13.4.4.1 为什么是 EM?

在深入研究 EM 的细节之前，值得一问的是，既然有多种其他（通常更快）算法可以直接解决问题，我们为什么要提出这种方法 ℓ_1MAP 估计问题（经验比较见 Linregfitl1 检验）。原因是潜在变量视角带来了几个优势，例如：

- 它提供了一种简单的方法来推导一个算法来查找各种其他模型的 ℓ_1- 正则化参数估计，如稳健线性回归（练习 11.12）或概率回归（练习 13.9）。
- 它建议尝试除 $Ga(\tau_j^2|1, \gamma^2/2)$ 之外的其他方差先验。我们将在下面考虑各种扩展。
- 它清楚地说明了我们如何计算全后验概率 $p(\mathbf{w}|D)$，而不仅仅是一个 MAP 估计。这种技术被称为**贝叶斯 lasso**（Park 和 Casella 2008；Hans 2009）。

[9]为了确保后验是单峰的，可以遵循（Park 和 Casella，2008），并通过使权重的先验方差取决于观测噪声来略微修改模型：$p(w_j|\tau_j^2, \sigma^2) = \mathcal{N}(w_j|0, \sigma^2\tau_j^2)$。EM 算法易于修改。

13.4.4.2 目标函数

从等式 13.88 中，完整数据惩罚对数似然如下（不依赖于 $\mathbf{w}$ 的下降项）

$$\ell_c(\mathbf{w}) = -\frac{1}{2\sigma^2}||\mathbf{y} - \mathbf{X}\mathbf{w}||_2^2 - \frac{1}{2}\mathbf{w}^T\Lambda\mathbf{w} + const \tag{13.89}$$

其中 $\Lambda = diag(\frac{1}{\tau_j^2})$ 是 $\mathbf{w}$ 的精度矩阵。

13.4.4.3 E 步骤

关键是计算 $\mathbb{E}\left[\frac{1}{\tau_j^2}|w_j\right]$，我们可以直接推导出（见练习 13.8）。或者，我们可以推导出完整的后验概率，如下所示（Park 和 Casella，2008）：

$$p(1/\tau_j^2|\mathbf{w}, D) = InverseGaussian\left(\sqrt{\frac{\gamma^2}{w_j^2}}, \gamma^2\right) \tag{13.90}$$

（注意，**逆高斯分布**也称为 Wald 分布。）因此

$$\mathbb{E}\left[\frac{1}{\tau_j^2}|w_j\right] = \frac{\gamma}{|w_j|} \tag{13.91}$$

设 $\overline{\Lambda} = diag(\mathbb{E}[1/\tau_1^2], ..., \mathbb{E}[1/\tau_D^2])$ 表示该 E 步的结果.

我们还需要推断 σ^2。很容易证明后验概率是

$$p(\sigma^2|D, \mathbf{w}) = IG(a_\sigma + (N)/2, b_\sigma + \frac{1}{2}(\mathbf{y} - \mathbf{X}\hat{\mathbf{w}})^T(\mathbf{y} - \mathbf{X}\hat{\mathbf{w}})) = IG(a_N, b_N) \tag{13.92}$$

因此

$$\mathbb{E}[1/\sigma^2] = \frac{a_N}{b_N} \triangleq \overline{\omega} \tag{13.93}$$

13.4.4.4 M 步骤

M 步骤包括计算

$$\hat{\mathbf{w}} = \arg\max_{\mathbf{w}} -\frac{1}{2}\overline{\omega}||\mathbf{y} - \mathbf{X}\mathbf{w}||_2^2 - \frac{1}{2}\mathbf{w}^T\Lambda\mathbf{w} \tag{13.94}$$

这只是高斯先验下的 MAP 估计：

$$\hat{\mathbf{w}} = (\sigma^2\overline{\Lambda} + \mathbf{X}^T\mathbf{X})^{-1}\mathbf{X}^T\mathbf{y} \tag{13.95}$$

然而，由于我们预期许多 $w_j = 0$，因此对于许多 j，我们将得到 $\tau_j^2 = 0$。这使得反演 $\overline{\Lambda}$ 在数值上不稳定。幸运的是，我们可以使用 $\mathbf{X}$ 的奇异值分解，由 $\mathbf{X} = \mathbf{U}\mathbf{D}\mathbf{V}^T$ 给出，如下所示：

$$\hat{\mathbf{w}} = \Psi\mathbf{V}(\mathbf{V}^T\Psi\mathbf{V} + \frac{1}{\overline{\omega}}\mathbf{D}^{-2})^{-1}\mathbf{D}^{-1}\mathbf{U}^T\mathbf{y} \tag{13.96}$$

其中

$$\Psi = \overline{\Lambda}^{-1} = diag(\frac{1}{\mathbb{E}[1/\tau_j^2]}) = diag[\frac{|w_j|}{\pi'(w_j)}] \tag{13.97}$$

13.4.4.5 警告

由于 lasso 目标是凸的，该方法应始终找到全局最优值。不幸的是，由于数字原因，这种情况有时不会发生。特别地，假设在真解中 $w_j^* \neq 0$, 进一步，假设我们在 M 步中设置 $\hat{\mathbf{w}}_j = 0$。在接下来的 E 步中，我们推断 $\tau_j^2 = 0$，因此我们再次设置 $\hat{\mathbf{w}}_j = 0$；因此，我们永远无法“撤消”我们的错误。幸运的是，在实践中，这种情况似乎很少见。进一步讨论请参见（Hunter 和 Li，2005）。

2.5 ℓ_1 正则化：扩展

在本节中，我们将讨论“vanilla”的各种扩展 ℓ_1 正则化。

2.5.1 lasso 组

在标准中 ℓ_1 正则化，我们假设参数和变量之间存在 1:1 的对应关系，因此如果 $\hat{w}_j = 0$，我们将其解释为变量 j 被排除在外。但在更复杂的模型中，可能有许多参数与给定变量相关。特别是，我们可能有每个输入的权重向量 $\mathbf{w}_j$。以下是一些例子：

- **多项式逻辑回归**：每个特征与 C 个不同的权重相关，每个类别一个。
- **带分类输入的线性回归**：每个标量输入是一个热编码到长度为 C 的向量中。
- **多任务学习**在多任务学习中，我们有多个相关的预测问题。例如，我们可能有 C 独立回归或二元分类问题。因此，每个特征与 C 个不同的权重相关联。我们可能希望对所有任务使用一个功能，或者不使用任何任务，从而在组级别选择权重（Obozinski 等人。2007).

如果我们使用形式为 $||\mathbf{w}|| = \sum_j \sum_c |w_{jc}|$ 的 ℓ_1 正则化子，我们可能最终得到 $\mathbf{w}_{j,:}$ 的某些元素为零，而有些元素不为零。为了防止这种情况，我们将参数向量划分为 G 组。我们现在最小化以下目标：

$$J(\mathbf{w}) = NLL(\mathbf{w}) + \sum_{g=1}^{G} \lambda_g ||\mathbf{w}_g||_2 \tag{13.98}$$

其中

$$||\mathbf{w}_g||_2 = \sqrt{\sum_{j \in g} w_j^2} \tag{13.99}$$

是组权重向量的 2-范数。如果 NLL 是最小二乘法，这种方法称为 **lasso 组**（Yuan 和 Lin，2006）。

我们通常通过设置 $\lambda_g = \lambda\sqrt{d_g}$，其中 d_g 是群 g 中元素的数目。例如，如果我们有群 $\{1, 2\}$ 和 $\{3, 4, 5\}$，目标就变成了

$$J(\mathbf{w}) = NLL(\mathbf{w}) + \lambda \left[\sqrt{2}\sqrt{(w_1^2 + w_2^2)} + \sqrt{3}\sqrt{(w_3^2 + w_4^2 + w_5^2)} \right] \tag{13.100}$$

请注意，如果我们使用了 2-范数的平方，该模型将等价于岭回归，因为

$$\sum_{g=1}^{G} ||\mathbf{w}_g||_2^2 = \sum_g \sum_{j \in g} w_j^2 = ||\mathbf{w}||_2^2 \tag{13.101}$$

通过使用平方根，我们惩罚了包含群权重向量的球的半径：半径变小的唯一方法是所有元素都变小。因此，平方根导致群稀疏。

该技术的一种变体将 2-范数替换为无穷范数（Turlach 等人 2005；Zhao 等人 2005）：

$$||\mathbf{w}_g||_\infty = \max_{j \in g} |w_j| \tag{13.102}$$

很明显，这也会导致群体稀疏。

图 13.13 和 13.14 显示了差异的说明。在这两种情况下，我们都有一个大小为 $D = 2^{12} = 4096$ 的真实信号 $\mathbf{w}$，分为 64 组，每组大小为 64。我们随机选择 8 组 $\mathbf{w}$，并将其分配为非零值。在第一个示例中，这些值是从 $\mathcal{N}(0, 1)$ 中提取的。在第二个示例中，所有值都设置为 1。然后，我们选择一个大小为 $N \times D$ 的随机设计矩阵 $\mathbf{X}$，其中 $N = 2^{10} = 1024$。最后，我们生成 $\mathbf{y} = \mathbf{X}\mathbf{w} + \epsilon$，其中 $\epsilon \sim \mathcal{N}(\mathbf{0}, 10^{-4}\mathbf{I}_N)$。鉴于这些数据，我们使用 ℓ_1 或组 ℓ_1 然后使用最小二乘法估计非零值。我们看到群套索比香草套索做得更好，因为它尊重已知的群结构。[10]我们也看到了 ℓ_∞ 范数倾向于使块内的所有元素具有相似的大小。这适用于第二个示例，但不适用于第一个示例。（λ 的值在所有示例中都相同，并且是手动选择的。）

[10] 系统中轻微的非零“噪音”ℓ_∞ 组 lasso 结果可能是由于数值误差。

13.5.1.1 群 lasso 的 GSM 解释

组 lasso 等价于使用以下先验知识的 MAP 估计：

$$p(\mathbf{w}|\gamma,\sigma^2) \propto exp\left(-\frac{\gamma}{\sigma}\sum_{g=1}^{G}||\mathbf{w}_g||_2\right) \tag{13.103}$$

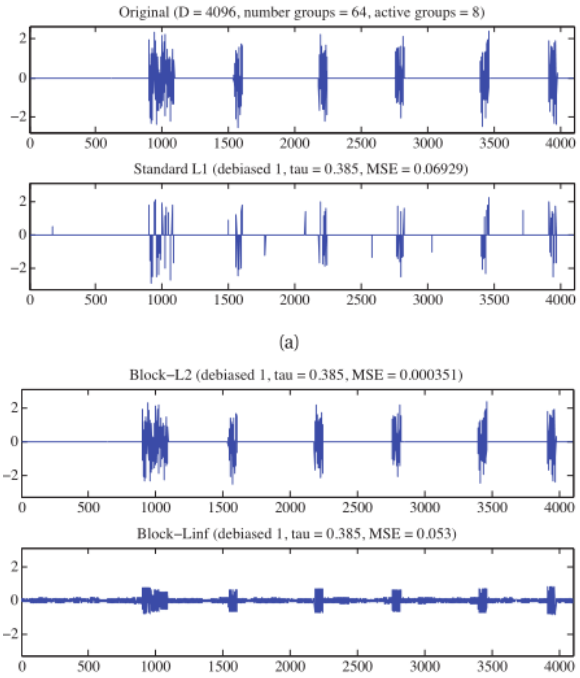


图 13.13: 原始信号为分段高斯的群套索示意图。左上角：原始信号。左下：vanilla lasso 估计。左上：使用组 lasso 估计块上的 ℓ_2 范数。右下角：使用 ℓ_∞ 范数。根据（Wright 等人。2009). 图由 groupLassoDemo 根据 Mario Figueiredo 的代码生成。

现在可以证明（练习 13.10），该先验可以写成 GSM，如下所示：

$$\mathbf{w}_g|\sigma^2,\tau_g^2 \sim \mathcal{N}(\mathbf{0},\sigma^2\tau_g^2\mathbf{I}_{d_g}) \tag{13.104}$$

$$\tau_g^2|\gamma \sim Ga(\frac{d_g+1}{2},\frac{\gamma}{2}) \tag{13.105}$$

其中 d_g 是群 g 的大小。因此我们看到每组有一个方差项，每个方差项来自伽马先验，其形状参数取决于群大小，其速率参数由 γ 控制。图 13.15 给出了一个示例，其中我们有两个组，一个大小为 2，一个为 3。

这幅图也清楚地说明了为什么应该有分组效应。假设 $w_{1,1}$,1 很小；则 τ_1^2 将被估计为小，这将迫使 $w_{1,2}$ 变小。反之，假设 $w_{1,1}$ 较大；则 τ_1^2 将被估计为大，这将允许 $w_{1,2}$ 也变大。

13.5.1.2 群 lasso 算法

群 lasso 有多种算法。这里我们简要地提到两个。第一种方法基于第 13.4.3 节中讨论的近端梯度下降。由于正则化子是可分离的，$R(\mathbf{w}) = \sum_g ||\mathbf{w}_g||_p$，近端算子分解为 G 个形式的独立算子

$$prox_R(\mathbf{b}) \underset{\mathbf{z}\in\mathbb{R}^{D_g}}{\arg\min} ||\mathbf{z}-\mathbf{b}||_2^2 + \lambda||\mathbf{z}||_p \tag{13.106}$$

其中 $\mathbf{b} = \boldsymbol{\theta}_{kg} - t_k\mathbf{g}_{kg}$。如果 p=2，可以证明（Combettes 和 Wajs 2005），这可以实现如下

$$prox_R(\mathbf{b}) = \mathbf{b} - proj_{\lambda C}(\mathbf{b}) \tag{13.107}$$

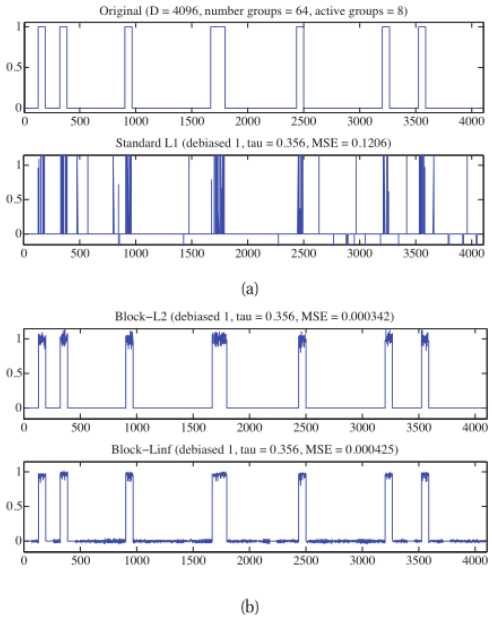


图 13.14 与图 13.13 相同，只是原始信号是分段恒定的。

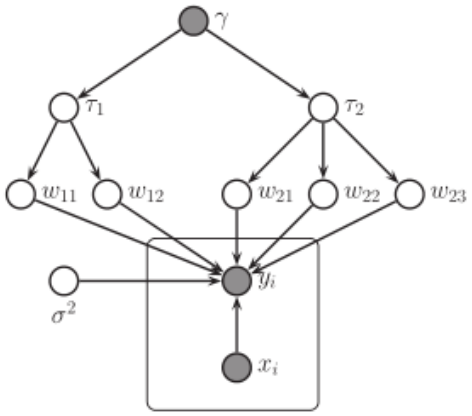


图 13.15 两组 lasso 的图形模型，第一组的尺寸 G_1=2，第二组的尺寸 G_2=3。

其中 $C = \{\mathbf{z} : ||\mathbf{z}||_2 \leq 1\}$ 是 ℓ_2 球。使用等式 13.74，如果 $||\mathbf{b}||_2 < \lambda$，则我们得到

$$prox_R(\mathbf{b}) = \mathbf{b} - \mathbf{b} = \mathbf{0} \tag{13.108}$$

否则，我们有

$$prox_R(\mathbf{b}) = \mathbf{b} - \lambda \frac{\mathbf{b}}{||\mathbf{b}||_2} = \mathbf{b} \frac{||\mathbf{b}||_2 - \lambda}{||\mathbf{b}||_2} \tag{13.109}$$

我们可以将这些组合成向量软阈值函数，如下所示（Wright 等人，2009）：

$$prox_R(\mathbf{b}) = \mathbf{b} \frac{max(||\mathbf{b}||_2 - \lambda, 0)}{max(||\mathbf{b}||_2 - \lambda, 0) + \lambda} \tag{13.110}$$

如果 $p = \infty$, 我们使用 $C = \{\mathbf{z} : ||\mathbf{z}||_1 \leq 1\}$ 是 ℓ_1 球。我们可以使用（Duchi et al.2008）中描述的算法在 $O(d_g)$ 时间内对此进行投影。

另一种方法是修改 EM 算法。该方法几乎与 vanilla lasso 相同。如果我们定义 $\tau_j^2 = \tau_{g(j)}^2$，则 $g(j)$ 是维数 j 所属的群，我们可以像以前一样对 σ^2 和 $\mathbf{w}$ 使用相同的全条件。唯一的变化如下：

- 我们必须修改权重精度的完整条件，该条件基于一组共享权重进行估计：

$$\frac{1}{\tau_g^2}|\gamma, \mathbf{w}, \sigma^2, \mathbf{y}, \mathbf{X} \sim InverseGaussian(\sqrt{\frac{\gamma^2\sigma^2}{||\mathbf{w}_g||_2^2}}, \gamma^2) \tag{13.111}$$

其中 $||\mathbf{w}_g||_2^2 = \sum_{j \in g} w_{jg}^2$。对于 E 步骤，我们可以使用

$$\mathbb{E}\left[\frac{1}{\tau_g^2}\right] = \frac{\gamma^\sigma}{||\mathbf{w}_g||_2} \tag{13.112}$$

- 我们必须修改调谐参数的完整条件，现在仅根据 τ_g^2 的 G 值进行估计：

$$p(\gamma^2|\tau) = Ga(a_\gamma + G/2, b_\gamma + \frac{1}{2}\sum_g^G \tau_g^2) \tag{13.113}$$

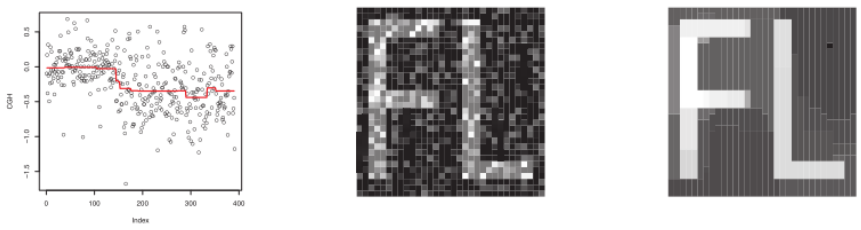


图 13.16:（a）融合 lasso 的示例。纵轴表示阵列 CGH（染色体基因组杂交）强度，横轴表示沿基因组的位置。资料来源：图 1（Hoefling 2010）。（b）噪声图像。（c）使用 2d 晶格先验的融合 lasso 估计。资料来源：图 2（Hoefling 2010）。经霍尔格 • 霍夫林许可使用。

2.5.2 融合 lasso

在某些问题设置中（例如，函数数据分析），除了稀疏之外，我们还希望相邻系数彼此相似。图 13.16（a）中给出了一个示例，其中我们希望拟合一个主要为“关闭”的信号，但另外还有一个特性，即相邻位置的值通常相似。我们可以使用该形式的先验知识对此进行建模

$$p(\mathbf{w}|\sigma^2) \propto exp\left(-\frac{\lambda_1}{\sigma}\sum_{j=1}^{D}|w_j| - \frac{\lambda_2}{\sigma}\sum_{j=1}^{D-1}|w_{j+1} - w_j|\right) \tag{13.114}$$

这就是所谓的**融合 lasso** 惩罚。在功能数据分析中，我们通常使用 $\mathbf{X} = \mathbf{I}$，因此信号中的每个位置都有一个系数（见第 4.4.2.3 节）。在这种情况下，总体目标具有以下形式：

$$J(\mathbf{w}, \lambda_1, \lambda_2) = \sum_{i=1}^{N}(y_i - w_i)^2 + \lambda_1\sum_{i=1}^{N}|w_i| + \lambda_2\sum_{i=1}^{N-1}|w_{i+1} - w_i| \tag{13.115}$$

这是等式 4.148 的稀疏版本。

可以将这一思想推广到链之外，并考虑其他图结构，使用形式惩罚

$$J(\mathbf{w}, \lambda_1, \lambda_2) = \sum_{s \in V}(y_s - w_s)^2 + \lambda_1\sum_{s \in V}|w_s| + \lambda_2\sum_{(s,t) \in E}|w_s - w_t| \tag{13.116}$$

这被称为**图引导融合 lasso**（参见例如（Chen 等人 2010））。该图可能来自一些先验知识，例如来自已知生物途径的数据库。另一个示例如图 13.16（b-c）所示，其中图形结构为二维晶格。

13.5.2.1 融合 lasso 的 GSM 解释

可以证明（Kyung 等人，2010 年），融合 lasso 模型等同于以下分层模型：

$$\mathbf{w}|\sigma^2, \boldsymbol{\tau}, \boldsymbol{\omega} \sim \mathcal{N}(\mathbf{0}, \sigma^2 \sum(\boldsymbol{\tau}, \boldsymbol{\omega})) \tag{13.117}$$

$$\tau_j^2|\gamma_1 \sim Expon(\frac{\gamma_1^2}{2}), j = 1:D \tag{13.118}$$

$$\omega_j^2|\gamma_2 \sim Expon(\frac{\gamma_2^2}{2}), j = 1:D-1 \tag{13.119}$$

其中 $\sum = \Omega^{-1}$ 并且 Ω 是三对角精度矩阵

$$main \ \ diagonal = \{\frac{1}{\tau_j^2} + \frac{1}{\omega_{j-1}^2} + \frac{1}{\omega_j^2}\} \tag{13.120}$$

$$off \ \ diagonal = \{-\frac{1}{\omega_j^2}\} \tag{13.121}$$

这里我们定义了 $\omega_0^{-2} = \omega_D^{-2} = 0$。这与第 4.4.2.3 节中的模型非常相似，其中我们使用链结构高斯马尔可夫随机场作为先验，具有固定方差。这里我们只让方差是随机的。在图引导 lasso 的情况下，图的结构反映在高斯精度矩阵的零模式中（见第 19.4.4 节）。

13.5.2.2 融合 lasso 算法

通过利用高斯先验的马尔可夫结构来提高效率，可以推广 EM 算法来拟合融合的 lasso 模型。还可以导出直接解算器（不使用潜变量技巧）（例如，参见（Hoefling，2010））。然而，不可否认，与我们考虑的其他模型相比，该模型的安装成本更高。

2.5.3 弹性网（ridge 和 lasso 组合）

尽管 lasso 已被证明是一种有效的变量选择技术，但它存在以下几个问题（Zou 和 Hastie 2005）：

- 如果存在一组高度相关的变量（例如，在同一路径中的基因），那么 lasso 倾向于只选择其中一个，选择相当随意。（从 LARS 算法中可以明显看出：一旦选择了组中的一个成员，组中的其余成员将不会与新残差非常相关，因此不会被选择。）通常最好选择组中的所有相关变量。如果我们知道分组结构，我们可以使用组 lasso，但通常我们不知道分组结构。
- 在 $D > N$ 的情况下，lasso 在饱和之前最多可以选择 N 个变量。
- 如果 $N > D$，但变量是相关的，根据经验观察，ridge 的预测性能优于 lasso。

Zou 和 Hastie（Zou 和 Hastie，2005）提出了一种称为**弹性网**的方法，它是 lasso 回归和 ridge 回归的混合，解决了所有这些问题。它显然被称为“弹性网”，因为它“像一个可伸缩的渔网，保留了' 所有的大鱼'”（Zou 和 Hastie，2005 年）。

13.5.3.1 vanilla 版

模型的普通版本定义了以下目标函数：

$$J(\mathbf{w}, \lambda_1, \lambda_2) = ||\mathbf{y} - \mathbf{X}\mathbf{w}||^2 + \lambda_2||\mathbf{w}||_2^2 + \lambda_1||\mathbf{w}||_1 \tag{13.122}$$

请注意，该惩罚函数是严格凸的（假设 $\lambda_2 > 0$），因此存在唯一的全局最小值，即使 $\mathbf{X}$ 不是满秩。

可以证明（Zou 和 Hastie 2005），$\mathbf{w}$ 上的任何严格凸惩罚都会表现出**分组效应**，这意味着高度相关变量的回归系数趋于相等（如果它们是负相关的，则符号会发生变化）。例如，如果两个特征相等，那么 $\mathbf{X}_{:j} = \mathbf{X}_{:k}$，可以证明它们的估计也相等，$\hat{w}_j = \hat{w}_k$。相比之下，对于 lasso，我们可以得到 $\hat{w}_j = 0$ 和 $\hat{w}_k \neq 0$ 或反之亦然。

13.5.3.2 vanilla 弹性网的算法

很容易证明（练习 13.5），弹性网问题可以简化为修正数据上的 lasso 问题。特别是

$$\widetilde{X} = c\begin{pmatrix}\mathbf{X}\\ \sqrt{\lambda_2}\mathbf{I}_D\end{pmatrix}, \widetilde{y} = \begin{pmatrix}\mathbf{y}\\ \mathbf{0}_{D\times 1}\end{pmatrix} \tag{13.123}$$

其中 $c = (1+\lambda_2)^{\frac{1}{2}}$, 然后我们解决

$$\widetilde{w} = arg\ \min_{\widetilde{w}} ||\widetilde{y} - \widetilde{X}\widetilde{w}||^2 + c\lambda_1||\widetilde{w}||_1 \tag{13.124}$$

并设置 $\mathbf{w} = c\widetilde{w}$。

我们可以使用 LARS 来解决这个子问题；这被称为 LARS-EN 算法。如果我们在包含 m 个变量后停止算法，则成本为 $O(m^3 + Dm^2)$。请注意，如果我们愿意，我们可以使用 $m = D$，因为 $\hat{\mathbf{X}}$ 具有秩 D。这与 lasso 不同，lasso 在 $N < D$ 时不能选择超过 N 个变量（在跳到 OLS 解之前）。

当使用 LARS-EN（或其他 ℓ_1 解算器）时，通常使用交叉验证来选择 λ_1 和 λ_2。

13.5.3.3 改进版

不幸的是，“vanilla”弹性网不能产生非常精确的预测函数，除非它非常接近纯 ridge 或纯 lasso。从直觉上看，原因是它会收缩两次：一次是由于 ℓ_2. 处罚和再次由于以下原因 ℓ_1 惩罚。解决方案很简单：撤销 ℓ_2 通过扩大普通版本的估计值来缩小。换言之 $\mathbf{w}^*$ 为方程 13.124 的解，则为更好的估计

$$\hat{\mathbf{w}} = \sqrt{1+\lambda_2}\widetilde{w} \tag{13.125}$$

我们称之为修正估计。

可以证明，修正后的估计由下式给出：

$$\hat{\mathbf{w}} = arg\ \min_{\mathbf{w}} \mathbf{w}^T\left(\frac{\mathbf{X}^T\mathbf{X} + \lambda_2\mathbf{I}}{1+\lambda_2}\right)\mathbf{w} - 2\mathbf{y}^T\mathbf{X}\mathbf{w} + \lambda_1||\mathbf{w}||_1 \tag{13.126}$$

现在

$$\frac{\mathbf{X}^T\mathbf{X} + \lambda_2\mathbf{I}}{1+\lambda_2} = (1-\rho)\hat{\sum} + \rho\mathbf{I} \tag{13.127}$$

其中 $\rho = \lambda_2/(1+\lambda_2)$。因此，弹性网络就像 lasso，但我们使用的 $\hat{\sum}$ 版本向 $\mathbf{I}$ 收缩（有关协方差矩阵正则化估计的更多讨论，请参见第 4.2.6 节）

13.5.3.4 弹性网的 GSM 解释

弹性网络使用的隐式先验显然具有以下形式：

$$p(\mathbf{w}|\sigma^2) \propto exp\left(-\frac{\gamma_1}{\sigma}\sum_{j=1}^{D}|w_j| - \frac{\gamma_2}{2\sigma^2}\sum_{j=1}^{D}w_j^2\right) \tag{13.128}$$

它只是高斯分布和拉普拉斯分布的乘积。

这可以写成如下的层次先验（Kyung 等人 2010；Chen 等人 2011）：

$$w_j|\sigma^2, \tau_j^2 \sim \mathcal{N}(0, \sigma^2(\tau_j^{-2} + \gamma_2)^{-1}) \tag{13.129}$$

$$\tau_j^2|\gamma_1 \sim Expon(\frac{\gamma_1^2}{2}) \tag{13.130}$$

显然，如果 $\gamma_2 = 0$，这将减少到常规 lasso。

可以使用 EM 在该模型中进行 MAP 估计，或者使用 MCMC（Kyung 等人 2010）或变分贝叶斯（Chen 等人 2011）进行贝叶斯推断。

2.6 非凸正则化子

虽然拉普拉斯先验导致凸优化问题，但从统计角度来看，该先验并不理想。它有两个主要问题。首先，它没有在 0 附近放置足够的概率质量，因此它没有充分抑制噪声。其次，它没有将足够的概率质量放在大的值上，因此它会导致相应于“信号”的相关系数收缩。（如图 13.5（a）所示）：我们看到 ℓ_1 系数的估计值明显小于其 ML 估计值，这种现象称为偏差。）

这两个问题都可以通过使用更灵活的先验知识来解决，这些先验知识在 0 处具有更大的尖峰和更重的尾部。即使我们再也找不到全局最优解，这些非凸方法的性能往往优于 ℓ_1 正则化，无论是在预测精度方面还是在检测相关变量方面（Fan 和 Li，2001；Schniter 等人，2008）。下面我们给出一些例子。

2.6.1 桥回归

ℓ_1 正则化的自然推广，称为**桥回归**（Frank 和 Friedman，1993），具有以下形式：

$$\hat{\mathbf{w}} = NLL(\mathbf{w}) + \lambda \sum_j |w_j|^b \tag{13.131}$$

对于 $b \geq 0$ 这对应于使用由下式给出的**指数幂分布**的 MAP 估计：

$$ExpPower(w|\mu, a, b) \triangleq \frac{b}{2a\Gamma(1+1/b)} exp\left(-\frac{|x-\mu|^b}{a}\right) \tag{13.132}$$

如果 $b = 2$，我们得到高斯分布 $(a = \sigma\sqrt{2})$，对应于岭回归；如果我们设 $b = 1$，我们得到拉普拉斯分布，对应于 lasso；如果我们设 $b = 0$，我们得到 ℓ_0 回归，相当于最佳子集选择。不幸的是，对于 $b < 1$，目标不是凸的，对于 $b > 1$，目标也不是稀疏促进的 ℓ_1 范数是 ℓ_0 范数。

图 13.17 显示了改变 b 的效果，其中我们绘制了 $b = 2$、$b = 1$ 和 $b = 0.4$ 的先验值；我们假设 $p(\mathbf{w}) = p(w_1)p(w_2)$。我们还绘制了在看到单个观测后的后验概率（$\mathbf{x}$，y），这施加了形式的单一线性约束，$y = \mathbf{w}^T\mathbf{x}$，具有由观测噪声控制的一定公差（与图 7.11 相比）。我们看到拉普拉斯的模式在垂直轴上，对应于 $w_1 = 0$。相反，当使用 $b = 0.4$ 时，有两种模式，对应于两种不同的稀疏解。当使用高斯时，MAP 估计不稀疏（模式不位于任一坐标轴上）。

2.6.2 分层自适应 lasso

回想一下，lasso 的主要问题之一是它会导致有偏差的估计。这是因为它需要使用较大的 λ 值来“挤压”不相关的参数，但这会过度惩罚相关参数。如果我们可以将不同的惩罚参数与每个参数相关联，那会更好。当然，通过交叉验证来调整 D 参数是完全不可行的，但这对贝叶斯没有问题：我们只是让每个 τ_j^2 有自己的私人调整参数 γ_j，现在将其视为来自共轭先验 $\gamma_j \sim \mathbf{IG}(a, b)$ 的随机变量。完整模型如下：

$$\gamma_j \sim IG(a, b) \tag{13.133}$$

$$\tau_j^2|\gamma_j \sim Ga(1, \gamma_j^2/2) \tag{13.134}$$

$$w_j|\tau_j^2 \sim \mathcal{N}(0, \tau_j^2) \tag{13.135}$$

见图 13.18（a）。这被称为**分层自适应 lasso**（HAL）（Lee 等人 2010）（另见（Lee 等人 2011；Cevher 2009；Armagan 等人 2011））。我们可以积分出 τ_j^2，这会像以前一样在 w_j 上产生重叠 $(w_j|0, 1/\gamma_j)$ 分布。结果是 $p(w_j)$ 现在是拉普拉斯算子的比例混合。事实证明，我们可以使用 EM 拟合该模型（即计算局部后验模），如下所述。结果估计值,$\hat{\mathbf{w}}_{HAL}$ 通常比 lasso 返回的估计值 $\hat{\mathbf{w}}_{L1}$ 工作得更好，因为它更可能在正确的位置包含零（模型选择一致性），更可能产生良好的预测（预测一致性）（Lee 等人，2010）。我们在第 13.6.2.2 节中对此行为进行了解释。

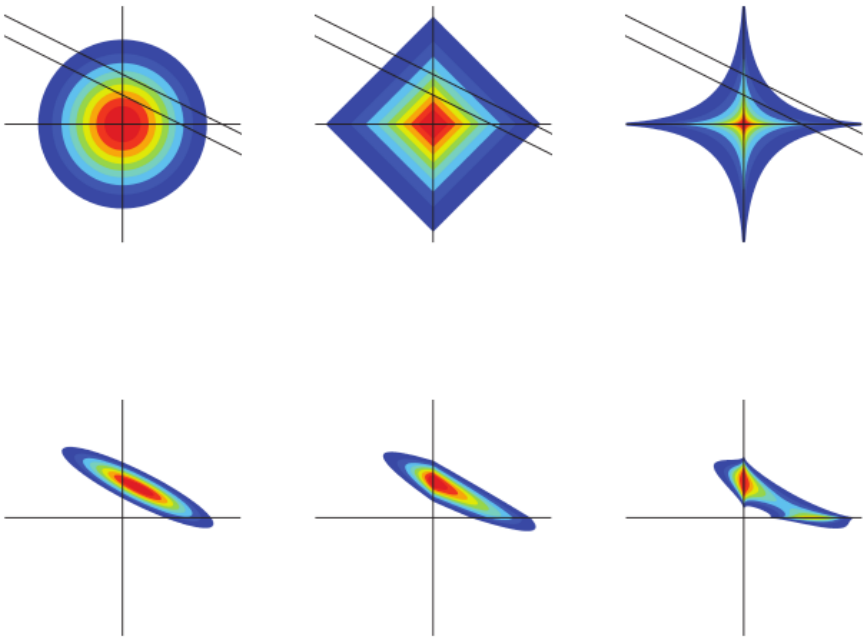


图 13.17: 顶部：具有单位方差的三种不同分布的对数先验图：高斯、拉普拉斯和指数幂。底部：观察单个观察后的对数后验曲线图，对应于单个线性约束。该观测的精度由上图中的对角线表示。在高斯先验的情况下，后验是单峰对称的。在拉普拉斯先验的情况下，后验是单峰和不对称的（倾斜的）。在指数先验的情况下，后验是双峰的。基于（Seeger 2008）的图 1。图由 sparsePostPlot 生成，由 Florian Steinke 编写。

13.6.2.1 用于 HAL 的 EM

由于逆伽马与拉普拉斯共轭，我们发现 γ_j 的 E 步由下式给出：

$$p(\gamma_j|w_j) = IG(a+1, b+|w_j|) \tag{13.136}$$

σ^2 的 E 步与 vanilla lasso 的 E 步相同。

$\mathbf{w}$ 的优先权具有以下形式：

$$p(\mathbf{w}|\gamma) = \prod_j \frac{1}{2\gamma_j} exp(-|w_j|/\gamma_j) \tag{13.137}$$

因此，M 步必须优化

$$\hat{\mathbf{w}}^{(t+1)} = \arg\max_{\mathbf{w}} \log \mathcal{N}(\mathbf{y}|\mathbf{X}\mathbf{w}, \sigma^2) - \sum_j |w_j| \mathbb{E}[1/\gamma_j] \tag{13.138}$$

期望值由下式给出：

$$\mathbb{E}[1/\gamma_j] = \frac{a+1}{b+|w_j^{(t)}|} \triangleq s_j^{(t)} \tag{13.139}$$

因此，M 步成为加权 lasso 问题：

$$\hat{\mathbf{w}}^{(t+1)} = \arg\min_{\mathbf{w}} ||\mathbf{y} - \mathbf{X}\mathbf{w}||_2^2 + \sum_j s_j^{(t)} |w_j| \tag{13.140}$$

使用标准方法（如 LARS）很容易解决这一问题。请注意，如果在上一次迭代中估计系数较大（因此 $w_j^{(t)}$ 较大），则缩放因子 $s_j^{(t)}$ 较小，因此较大的系数不会受到严重惩罚。相反，小系数确实会受到严重惩罚。这是算法调整每个系数惩罚强度的方式。结果是一个估计值，通常比 lasso 返回的估计值要稀疏得多，但偏差也较小。

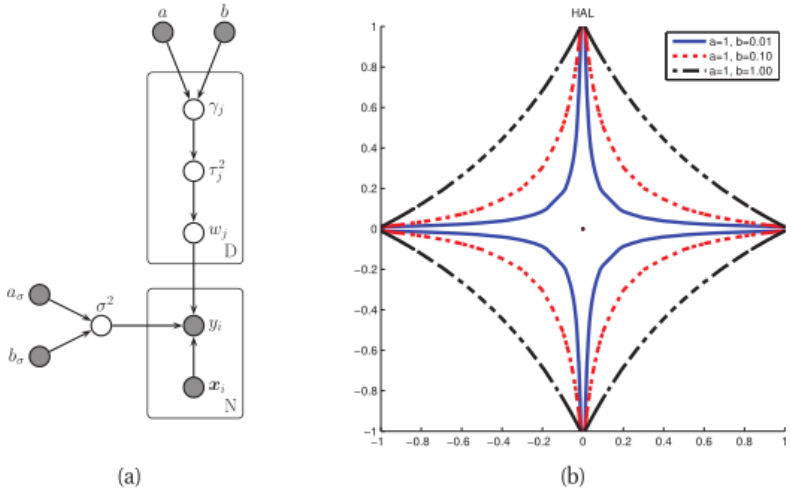


图 13.18:（a）分层自适应 lasso 的 DGM。（b）层次调整拉普拉斯的轮廓。基于图 1（Lee 等人，2010）。图由 normalGammaPenaltyPlotDemo 生成。

请注意，如果我们设置 $a = b = 0$，并且只执行一次 EM 迭代，我们得到的方法与（Zou2006；Zou 和 Li2008）的**自适应 lasso** 密切相关。该 EM 算法也与某些迭代重加权算法密切相关 ℓ_1 信号处理界提出的方法（Chartrand 和 Yin，2008；Candes 等人，2008）。

13.6.2.2 理解 HAL 的行为

通过积分 γ_j，我们可以更好地理解 HAL，从而得到以下边缘分布，

$$p(w_j|a,b) = \frac{a}{2b}\left(\frac{|w_j|}{b}+1\right)^{-(a+1)} \tag{13.141}$$

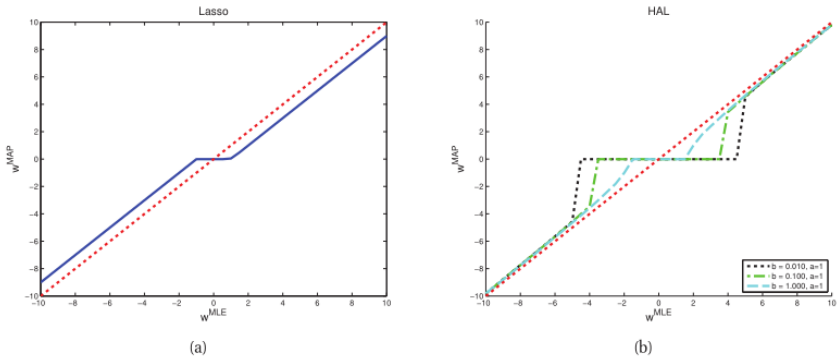


图 13.19: 两个惩罚函数的阈值行为（负对数先验）。（a）拉普拉斯。（b）分层自适应拉普拉斯。基于图 2（Lee 等人，2010）。图由 normalGammaThresholdPlotDemo 生成。

这是**广义 t 分布**的一个例子（McDonald 和 Newey 1988）（在（Cevher 2009；Armagan 等人 2011）中，这被称为双帕累托分布），定义为：

$$GT(w|\mu,a,c,q) \triangleq \frac{q}{2ca^{1/q}B(1/q,a)}\left(1+\frac{|w-\mu|^q}{ac^q}\right)^{-(a+1/q)} \tag{13.142}$$

其中 c 是比例参数（控制稀疏度），a 与自由度相关。当 $q = 2$ 且 $c = \sqrt{2}$. 恢复标准 t 分布；当 $a \to \infty$,

我们恢复指数功率分布；当 $q = 1$ 且 $a = \infty$ 我们得到拉普拉斯分布。在当前模型的上下文中，我们看到 $p(w_j|a, b) = GT(w_j|0, a, b/a, 1)$。

由此产生的惩罚条款具有以下形式：

$$\pi_{\boldsymbol{\lambda}}(w_j) \triangleq -\log p(w_j) = (a+1)\log(1+\frac{|w_j|}{b}) + const \tag{13.143}$$

其中 $\boldsymbol{\lambda} = (a, b)$ 是调谐参数。我们在图 13.18（b）中绘制了各种 b 值的 2d 惩罚图（即，我们绘制了 $\pi_{\boldsymbol{\lambda}}(w_1) + \pi_{\boldsymbol{\lambda}}(w_2)$。与图 13.3（a）中所示的菱形拉普拉斯惩罚相比，我们看到 HAL 惩罚更像是一条“星鱼”：它沿“棘”放置了更多密度，从而更积极地强制稀疏性。请注意，此惩罚显然不是凸的。

通过考虑将该罚函数应用于具有正交设计矩阵的线性回归问题，我们可以进一步了解该罚函数的行为。在这种情况下，可以证明目标变为

$$J(\mathbf{w}) = \frac{1}{2}||\mathbf{y} - \mathbf{X}\mathbf{w}||_2^2 + \sum_{j=1}^{D} \pi_{\boldsymbol{\lambda}}(|w_j|) \tag{13.144}$$

$$= \frac{1}{2}||\mathbf{y} - \hat{\mathbf{y}}||^2 + \frac{1}{2}\sum_{j=1}^{D}(\hat{w}_j^{mle} - w_j)^2 + \sum_{j=1}^{D} \pi_{\boldsymbol{\lambda}}(|w_j|) \tag{13.145}$$

$p(\tau_j^2)$	$p(\gamma_j)$	$p(w_j)$	Ref
$\mathrm{Ga}(1, \frac{\gamma^2}{2})$	Fixed	$\mathrm{Lap}(0, 1/\gamma)$	(Andrews and Mallows 1974; West 1987)
$\mathrm{Ga}(1, \frac{\gamma^2}{2})$	$\mathrm{IG}(a, b)$	$\mathrm{GT}(0, a, b/a, 1)$	(Lee et al. 2010, 2011; Cevher 2009; Armagan et al. 2011)
$\mathrm{Ga}(1, \frac{\gamma^2}{2})$	$\mathrm{Ga}(a, b)$	$\mathrm{NEG}(a, b)$	(Griffin and Brown 2007, 2010; Chen et al. 2011)
$\mathrm{Ga}(\delta, \frac{\gamma^2}{2})$	Fixed	$\mathrm{NG}(\delta, \gamma)$	(Griffin and Brown 2007, 2010)
$\mathrm{Ga}(\tau_j^2\|0, 0)$	-	$\mathrm{NJ}(w_j)$	(Figueiredo 2003)
$\mathrm{IG}(\frac{\delta}{2}, \frac{\delta\gamma^2}{2})$	Fixed	$\mathcal{T}(0, \delta, \gamma)$	(Andrews and Mallows 1974; West 1987)
$C^+(0, \gamma)$	$C^+(0, b)$	horseshoe(b)	(Carvahlo et al. 2010)

表 13.2: 高斯的一些比例混合。缩写：C^+ = 半-校正柯西；Ga= 伽马（形状和速率参数化）；GT 广义 t；IG= 逆伽马；NEG= 正态指数 Algamma；NG= 正常伽马；NJ= 正常的杰弗里斯。马蹄形分布是我们对先前描述的 w_j 上诱导的分布的名称（Carvahlo 等人，2010）；这没有简单的解析形式。负密度和负密度的定义有点复杂，但可以在参考文献中找到。文本中定义了其他分布。

其中 $\hat{\mathbf{w}}^{mel} = \mathbf{X}^T\mathbf{y}$ 是最大似然估计，且 $\hat{\mathbf{y}} = \mathbf{X}\hat{\mathbf{w}}^{mel}$。因此，我们可以通过解决以下 1d 优化问题，一次计算一维 MAP 估计：

$$\hat{w}_j = \arg\min_{w_j} \frac{1}{2}(\hat{w}_j^{mle} - w_j)^2 + \pi_{\boldsymbol{\lambda}}(w_j) \tag{13.146}$$

在图 13.19（a）中，我们绘制了 lasso 估计值 $\hat{w}^{L1}$ 与 ML 估计值,$\hat{w}^{mle}$。我们看到 ℓ_1 估计器具有如图 13.5（a）所示的通常软阈值行为。然而，这种行为是不可取的，因为大震级系数也向 0 收缩，而我们希望它们等于其 unshrunken ML 估计。

在图 13.19（b）中，我们绘制了 HAL 估计值 $\hat{w}^{HAL}$ 与 ML 估计值 $\hat{w}^{mle}$。我们发现，这更接近图 13.5（b）中所示的更理想的硬阈值行为。

2.6.3 其他等级先验

已经提出了许多其他等级稀疏促进先验；简要总结见表 13.2。在某些情况下，我们可以解析地导出 w_j 的边际先验形式。一般来说，该先验不是凹的。

一个特别有趣的先验是不适当的正常杰弗里斯先验，它已在（Figueiredo 2003）中使用。这将非信息性 Jeffreys 先验置于方差 $Ga(\tau_j^2|0, 0) \propto 1/\tau_j^2$, 所得边际具有形式 $p(w_j) = NJ(w_j) \propto 1/|w_j|$。这导致了阈值规则，看起来非常类似于图 13.19（b）中的 HAL，而这又非常类似到硬阈值。然而，这个先验没有自由参数，这是一件好事（无需调整）和一件坏事（没有能力适应稀疏程度）。

2.7 自动相关性确定（ARD）/稀疏贝叶斯学习（SBL）

到目前为止，我们考虑的所有方法（第 13.2.1 节中的尖峰法和平板法除外）都使用了形式为 $p(\mathbf{w}) = \prod_j p(w_j)$ 的**阶乘先验**。我们已经看到了这些先验是如何用高斯尺度 $w_j \sim \mathcal{N}(0, \tau_j^2)$ 形式的混合来表示的，其中 τ_j^2 具有表 13.2 中列出的先验之一。使用这些潜在方差，我们可以以 $\tau_j^2 \to w_j \to \mathbf{y} \leftarrow \mathbf{X}$ 的形式表示模型, 然后，我们可以使用 EM 进行 MAP 估计，在 E 步骤中，我们推断 $p(\tau_j^2|w_j)$，在 M 步骤中，我们根据 $\mathbf{y}$、$\mathbf{X}$ 和 $\boldsymbol{\tau}$ 估计 $\mathbf{w}$。这个 M 步要么涉及闭式加权 ℓ_1 优化（在高斯尺度混合的情况下），或加权 ℓ_1 优化（在拉普拉斯尺度混合物的情况下）。我们还讨论了如何在此类模型中执行贝叶斯推理，而不仅仅是计算 MAP 估计。

在本节中，我们讨论了一种基于 $\mathbb{I}$ 型 ML 估计（经验贝叶斯）的替代方法，据此我们将 $\mathbf{w}$ 积分并最大化边际似然 wrt $\boldsymbol{\tau}$。该 EB 程序可以通过 EM 或通过重新加权 ℓ_1 方案，如下所述。估计方差后，我们将其插入计算权重的后验平均值 $\mathbb{E}[\mathbf{w}|\hat{\boldsymbol{\tau}}, D]$；令人惊讶的是（考虑到高斯先验），结果是（近似）稀疏估计，原因如下所述。

在神经网络的背景下，这种方法被称为**自动相关性确定**或 **ARD**（MacKay 1995b；Neal 1996）：参见第 16.5.7.5 节。在本章中我们考虑的线性模型的背景中，这种方法称为**稀疏贝叶斯学习**或 **SBL**（Tipping 2001）。将 ARD/SBL 与线性模型中的基函数展开相结合，产生了一种称为相关向量机（RVM）的技术，我们将在第 14.3.2 节中讨论。

2.7.1 线性回归的 ARD

我们将在线性回归的背景下解释该过程；GLMs 的 ARD 需要使用拉普拉斯（或其他）近似。在讨论 ARD/SBL 时，通常用 $\alpha_j = 1/\tau_j^2$ 表示权重精度，用 $\beta = 1/\sigma^2$ 表示测量精度（不要将此与统计中使用 β 表示回归系数混淆！）。特别是，我们将假设以下模型：

$$p(y|\mathbf{x}, \mathbf{w}, \beta) = \mathcal{N}(y|w^T\mathbf{x}, 1/\beta) \tag{13.147}$$

$$p(\mathbf{w}) = \mathcal{N}(\mathbf{w}|\mathbf{0}, \mathbf{A}^{-1}) \tag{13.148}$$

其中 $\mathbf{A} = diag(\boldsymbol{\alpha})$。边际似然可通过以下方式进行分析计算：

$$p(\mathbf{y}|\mathbf{X}, \boldsymbol{\alpha}, \beta) = \int \mathcal{N}(\mathbf{y}|\mathbf{X}\mathbf{w}, \beta\mathbf{I}_N)\mathcal{N}(\mathbf{w}|\mathbf{0}, \mathbf{A})d\mathbf{w} \tag{13.149}$$

$$= \mathcal{N}(\mathbf{y}|\mathbf{0}, \beta\mathbf{I}_N + \mathbf{X}\mathbf{A}^{-1}\mathbf{X}^T) \tag{13.150}$$

$$= (2\pi)^{-N/2}|\mathbf{C}_\alpha|^{-\frac{1}{2}}exp(-\frac{1}{2}\mathbf{y}^T\mathbf{C}_\alpha^{-1}\mathbf{y}) \tag{13.151}$$

其中

$$\mathbf{C}_{\boldsymbol{\alpha}} \triangleq \beta^{-1}\mathbf{I}_N + \mathbf{X}\mathbf{A}^{-1}\mathbf{X}^T \tag{13.152}$$

将其与尖峰和平板模型中等式 13.13 中的边际似然进行比较；取第二项中缺失的 $\beta = 1/\sigma^2$ 因子的模，方程是相同的，只是我们替换了二元 $\gamma_j \in \{0, 1\}$ 具有连续的 $\alpha_j \in \mathbb{R}^+$。在对数形式下，目标变为：

$$\ell(\boldsymbol{\alpha}, \beta) \triangleq -\frac{1}{2}\log p(\mathbf{y}|\mathbf{X}, \boldsymbol{\alpha}, \beta) = \log|\mathbf{C}_\alpha| + \mathbf{y}^T\mathbf{C}_{\boldsymbol{\alpha}}^{-1}\mathbf{y} \tag{13.153}$$

为了使问题正则化，我们可以对每个精度设置共轭先验 $\alpha_j \sim Ga(a, b)$ 和 $\beta \sim Ga(c, d)$。修改后的目标变为

$$\ell(\boldsymbol{\alpha}, \beta) \triangleq -\frac{1}{2}\log p(\mathbf{y}|\mathbf{X}, \boldsymbol{\alpha}, \beta) + \sum_j \log Ga(\alpha_j|a, b) + \log Ga(\beta|c, d) \tag{13.154}$$

$$= \log|\mathbf{C}_\alpha| + \mathbf{y}^T\mathbf{C}_{\boldsymbol{\alpha}}^{-1}\mathbf{y} + \sum_j (a\log\alpha_j - b\alpha_j) + c\log\beta - d\beta \tag{13.155}$$

这在对 $\boldsymbol{\alpha}$ 和 β 进行贝叶斯推断时非常有用（Bishop 和 Tipping 2000）。然而，当执行（II 型）点估计时，我们将使用不适当的先验 $a = b = c = d = 0$，这将导致最大稀疏性。

下面我们描述如何优化 $\ell(\boldsymbol{\alpha}, \beta)$wrt 精度项 $\boldsymbol{\alpha}$ 和 β。[11]这是在尖峰和平板模型中寻找 $\boldsymbol{\gamma}$ 的最可能模型设置的代理，这反过来又与 ℓ_0 正则化。特别是，可以证明（Wipf 等人，2010 年），等式 13.153 中的目标比 ℓ_0 目标具有更少的局部最优值，因此更容易优化。

一旦我们估计了 $\boldsymbol{\alpha}$ 和 β，我们就可以使用

$$p(\mathbf{w}|D, \hat{\boldsymbol{\alpha}}, \hat{\beta}) = \mathcal{N}(\boldsymbol{\mu}, \sum) \tag{13.156}$$

$$\sum^{-1} = \hat{\beta}\mathbf{X}^T\mathbf{X} + \mathbf{A} \tag{13.157}$$

$$\boldsymbol{\mu} = \hat{\beta} \sum \mathbf{X}^T\mathbf{y} \tag{13.158}$$

我们计算 $\mathbf{w}$ 上的后验概率，同时鼓励稀疏性，这就是为什么这种方法被称为“稀疏贝叶斯学习”。然而，由于有许多方法可以实现稀疏和贝叶斯，因此我们将使用“ARD”术语，即使在线性模型上下文中也是如此。（此外，SBL 只是关于系数值的“贝叶斯”，而不是反映相关变量集的不确定性，这通常更令人感兴趣。）

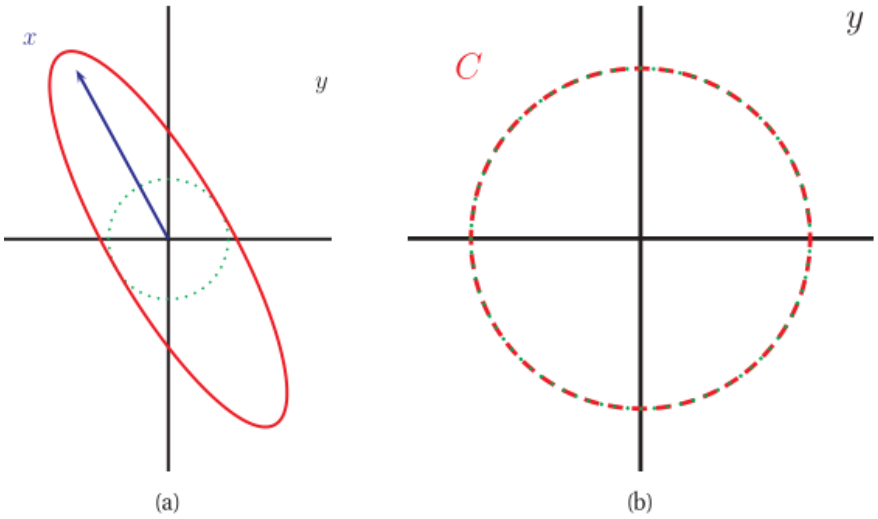


图 13.20: 说明 ARD 导致稀疏性的原因。输入 $\mathbf{x}$ 的矢量不指向输出 $\mathbf{y}$ 的矢量，因此应删除该特征。（a）对于有限 α，当 $\alpha = \infty$, $\mathbf{y}$ 处的概率密度最大化。基于（2001 年）的图 8。

2.7.2 稀疏性从何而来？

如果 $\hat{\alpha}_j \approx 0$，我们找到 $\hat{w}_j \approx \hat{w}_j^{mle}$，因为高斯先验收缩 w_j 朝向 0 具有零精度。然而，如果我们发现 $\hat{\alpha}_j \approx 0$, 那么先验知识非常确信 $w_j = 0$，因此特征 j 是“不相关的”。因此，后验平均值将具有 $\hat{w}_j \approx 0$ 因此，不相关特征的权重会自动“关闭”或“删除”。

我们现在根据（Tipping 2001）给出了一个直观的论点，关于 ML-II 为什么应该鼓励 $\alpha_j \to \infty$ 对于不相关的特征。考虑一个具有 2 个训练示例的一维线性回归，因此 $\mathbf{X} = \mathbf{x} = (x_1, x_2)$ 和 $\mathbf{y} = (y_1, y_2)$。我们可以将 $\mathbf{x}$ 和 $\mathbf{y}$ 绘制为平面中的向量，如图 13.20 所示。假设该特征与预测响应无关，因此 $\mathbf{x}$ 指向与 $\mathbf{y}$ 几乎正交的方向。让我们看看当我们改变 α 时，边际似然会发生什么。边际似然由 $p(\mathbf{y}|\mathbf{x}, \alpha, \beta) = \mathcal{N}(\mathbf{y}|\mathbf{0}, \mathbf{C})$，其中

$$\mathbf{C} = \frac{1}{\beta}\mathbf{I} + \frac{1}{\alpha}\mathbf{x}\mathbf{x}^T \tag{13.159}$$

如果 α 是有限的，则后部将沿 x 方向拉长，如图 13.20（a）所示。然而，如果 $\alpha = \infty$, 我们发现 $\mathbf{C} = \frac{1}{\beta}\mathbf{I}$, $\mathbf{C}$ 是球形的，如图 13.20（b）所示。如果 $|\mathbf{C}|$ 保持不变，则后者为观测响应向量 y 分配更高的概率密度，因

[11]优化 β 的另一种方法是在 β 上加一个伽马先验，并将其积分，得到 $\mathbf{w}$ 的学生后验值（Buntine 和 Weigend 1991）。然而，结果表明，这导致 $\boldsymbol{\alpha}$ 的估计不太准确（MacKay，1999）。此外，使用高斯分布比使用学生分布更容易，并且高斯情况更容易推广到其他情况，如逻辑回归。

此这是首选解决方案。换句话说，边际似然“惩罚”解，其中 α_j 很小，但 $\mathbf{X}_{:,j}$ 不相关，因为这些浪费了概率质量。消除冗余维度更为节省（从贝叶斯-奥卡姆剃刀的角度来看）。

2.7.3 连接到 MAP 估计

ARD 似乎与我们在本章前面考虑的 MAP 估计方法有很大不同。特别是，在 ARD 中，我们没有积分出 $\boldsymbol{\alpha}$ 并优化 $\mathbf{w}$，反之亦然。因为参数 w_j 在后验中变得相关（由于解释的原因），当我们估计 α_j 时，我们借用了所有特征的信息，而不仅仅是特征 j。因此，有效先验 $p(\mathbf{w}|\hat{\boldsymbol{\alpha}})$ 是**非阶乘**的，而且它取决于数据 $\mathbf{D}$（和 σ^2）。然而，在（Wipf 和 Nagarajan，2007 年）中，ARD 可被视为以下 MAP 估计问题：

$$\hat{\mathbf{w}}^{ARD} = \arg\min_{\mathbf{w}} \beta||\mathbf{y} - \mathbf{X}\mathbf{w}||_2^2 + g_{ARD}(\mathbf{w}) \tag{13.160}$$

$$g_{ARD}(\mathbf{w}) \triangleq \min_{\boldsymbol{\alpha}\geq 0} \sum_j \alpha_j w_j^2 + \log|\mathbf{C}_{\boldsymbol{\alpha}}| \tag{13.161}$$

基于凸分析的证明有点复杂，因此省略。

此外，（Wipf 和 Nagarajan，2007 年；Wipf 等人，2010 年）证明了具有非阶乘先验的 MAP 估计在以下意义上严格优于具有任何可能阶乘先验值的 MAP 估计：非阶乘目标总是比阶乘目标具有更少的局部极小值，同时仍然满足非阶乘目标的全局最优对应于 ℓ_0 目标-一个属性 ℓ_1 正则化没有局部极小值，因此不受欢迎。

2.7.4 ARD 算法 *

在本节中，我们将回顾几种用于实现 ARD 的不同算法。

13.7.4.1 EM 算法

实现 SBL/ARD 的最简单方法是使用 EM。预期的完整数据日志可能性由下式给出：

$$\begin{aligned} Q(\boldsymbol{\alpha}, \beta) &= \mathbb{E}[\log \mathcal{N}(\mathbf{y}|\mathbf{X}\mathbf{w}, \sigma^2\mathbf{I}) + \log \mathcal{N}(\mathbf{w}|\mathbf{0}, \mathbf{A}^{-1})] && (13.162)\\ &= \frac{1}{2}\mathbb{E}\left[N\log\beta - \beta||\mathbf{y} - \mathbf{X}\mathbf{w}||^2 + \sum_j \log\alpha_j - tr(\mathbf{A}\mathbf{w}\mathbf{w}^T)\right] + const && (13.163)\\ &= \frac{1}{2}N\log\beta - \frac{\beta}{2}(||\mathbf{y} - \mathbf{X}\boldsymbol{\mu}||^2 + tr(\mathbf{X}^T\mathbf{X}\textstyle\sum)) \\ &\quad + \frac{1}{2}\sum_j \log\alpha_j - \frac{1}{2}tr[\mathbf{A}(\boldsymbol{\mu}\boldsymbol{\mu}^T + \textstyle\sum)] + const && (13.164) \end{aligned}$$

其中 $\boldsymbol{\mu}$ 和 $\sum$ 在 E 步中使用等式 13.158 计算。

假设我们把 $Ga(a, b)$ 放在 α_j 上，把 $Ga(c, d)$ 放在 β 上。被处罚的目标变为：

$$Q'(\boldsymbol{\alpha}, \beta) = Q(\boldsymbol{\alpha}, \beta) + \sum_j (a\log\alpha_j - b\alpha_j) + c\log\beta - d\beta \tag{13.165}$$

设置 $\frac{dQ'}{d\alpha_j} = 0$ 我们得到以下 M 步：

$$\alpha_j = \frac{1 + 2a}{\mathbb{E}[w_j^2] + 2b} = \frac{1 + 2a}{m_j^2 + \textstyle\sum_{\boldsymbol{jj}} + 2b} \tag{13.166}$$

如果 $\alpha_j = \alpha$，$a = b = 0$，则更新变为

$$\alpha = \frac{D}{\mathbb{E}[\mathbf{w}^T\mathbf{w}]} = \frac{D}{\boldsymbol{\mu}^T\boldsymbol{\mu} + tr(\textstyle\sum)} \tag{13.167}$$

β 的更新由下式给出：

$$\beta_{new}^{-1} = \frac{||\mathbf{y} - \mathbf{X}\boldsymbol{\mu}||^2 + \beta^{-1}\sum_j(1 - \alpha_j \sum_{jj}) + 2d}{N + 2c} \tag{13.168}$$

（这是练习 13.2 的推导。）

13.7.4.2 不动点算法

一种更快更直接的方法是直接优化等式 13.155 中的目标。可以证明（练习 13.3），等式 $\frac{d\ell}{d\alpha_j} = 0$ 和 $\frac{d\ell}{d\beta} = 0$ 导致以下定点更新：

$$\alpha_j \leftarrow \frac{\gamma_j + 2a}{m_j^2 + 2b} \tag{13.169}$$

$$\beta^{-1} \leftarrow \frac{||\mathbf{y} - \mathbf{X}\boldsymbol{\mu}||^2 + 2d}{N - \sum_j \gamma_j + 2c} \tag{13.170}$$

$$\gamma_j \triangleq 1 - \alpha_j \sum_{jj} \tag{13.171}$$

量 γ_j 是数据确定 w_j 的良好程度的量度（MacKay 1992）。因此 $\gamma = \sum_j \gamma_j$ 是模型的有效自由度。进一步讨论见第 7.5.3 节。

由于 $\boldsymbol{\alpha}$ 和 β 都依赖于 $\boldsymbol{\mu}$ 和 $\sum$（可使用方程 13.158 或拉普拉斯近似计算），我们需要重新估计这些方程，直到收敛。（该算法的收敛性已在（Wipf 和 Nagarajan，2007）中进行了研究。）在收敛时，结果在形式上与 EM 获得的结果相同，但由于目标是非凸的，结果可能取决于初始值。

13.7.4.3 迭代重加权 ℓ_1 算法

另一种解决 ARD 问题的方法是基于这是一个 MAP 估计问题的观点。虽然对数先验 $g(\mathbf{w})$ 的形式相当复杂，但它可以被证明是 $|w_j|$ 的非递减凹函数。这意味着它可以通过迭代重加权 ℓ_1 表格问题

$$\mathbf{w}^{t+1} = arg\min_{\mathbf{w}} NLL(\mathbf{w}) + \sum_j \lambda_j^{(t)}|w_j| \tag{13.172}$$

在（Wipf 和 Nagarajan 2010）中，建议采用以下程序设置惩罚项（基于惩罚函数的凸边界）。我们用 $\lambda_j^{(0)} = 1$ 初始化，然后在迭代 $t+1$ 时，通过迭代以下方程几次来计算 $\lambda_j^{(t+1)}$：[12]

$$\lambda_j \leftarrow \left[\mathbf{X}_{:,j}\left(\sigma^2\mathbf{I} + \mathbf{X}diag(1/\lambda_j)diag(|w_j^{(t+1)}|)\right)^{-1}\mathbf{X}^T)^{-1}\mathbf{X}_{:,j}\right]^{\frac{1}{2}} \tag{13.173}$$

我们看到新惩罚 λ_j 依赖于所有旧权重。这与第 13.6.2 节的自适应 lasso 方法大不相同。

为了理解这种差异，考虑 $\sigma^2 = 0$ 的无噪声情况，并假设 $D \gg N$ 在这种情况下，存在完全重构数据的 $\binom{D}{N}$ 解，$\mathbf{X}\mathbf{w} = \mathbf{y}$，并且具有稀疏性 $||\mathbf{w}||_0 = N$；这些被称为基本可行解或 BFS。我们想要的是 $\mathbf{X}\mathbf{w} = \mathbf{y}$ 的解，但比这个解要稀疏得多。假设该方法已找到一个 BFS。我们不希望仅仅因为重量小（如自适应 lasso）就增加重量的惩罚，因为这只会加强我们当前的局部最优值。相反，如果权重很小并且 $||\mathbf{w}^{(t+1)}|| < N||$，我们希望增加对权重的惩罚。协方差项（$\mathbf{X}diag(1/\lambda_j)diag(|w_j^{(t+1)}|)^{-1}$ 具有这样的效果：如果 $\mathbf{w}$ 是 BFS，则该矩阵将是满秩的，因此惩罚不会增加太多，但如果 $\mathbf{w}$ 比 N 稀疏，则矩阵将不会为满秩，因此与零值系数相关的惩罚将增加，从而加强了该解决方案（Wipf 和 Nagarajan，2010）。

2.7.5 逻辑回归的 ARD

现在考虑二元逻辑回归，$p(y|\mathbf{x}, \mathbf{w}) = Ber(y|sigm(\mathbf{w}^T\mathbf{x}))$，使用相同的高斯先验，$p(\mathbf{w}) = \mathcal{N}(\mathbf{w}|\mathbf{0}, \mathbf{A}^{-1})$. 我们不能再使用 EM 来估计 $\boldsymbol{\alpha}$，因为高斯先验不与 logistic 似然共轭，因此 E 步无法精确完成。一种方法

[12]（Wipf 和 Nagarajan，2007）中的算法相当于方程 13.173 的一次迭代。然而，由于方程计算成本低（只有 $O(ND||\mathbf{w}^{(t+1)}||_0)$ 次），因此在解决更昂贵的 ℓ_1 问题之前，值得迭代几次。

是使用 E 步的变分近似，如第 21.8.1.1 节所述。更简单的方法是在 E 步骤中使用拉普拉斯近似（见第 8.4.1 节）。然后，我们可以在与之前相同的 EM 过程中使用此近似，但我们不再需要更新 β。但是，请注意，这并不能保证收敛。

另一种方法是使用第 13.7.4.3 节中的技术。在这种情况下，我们可以使用精确的方法来计算内部加权 ℓ_1 正则逻辑回归问题，无需近似。

2.8 稀疏编码 *

到目前为止，我们一直专注于监督学习的稀疏先验。在本节中，我们将讨论如何将它们用于无监督学习。

在第 12.6 节中，我们讨论了独立分量分析，它与主成分分析类似，只是对潜在因子 $\mathbf{z}_i$ 使用了非高斯先验。如果我们使非高斯先验具有稀疏性，例如拉普拉斯分布，我们将每个观察到的向量 $\mathbf{x}_i$ 近似为基向量（$\mathbf{W}$ 列）的稀疏组合；请注意，稀疏模式（由 zi 控制）随数据情况而变化。如果我们放松 $\mathbf{W}$ 是正交的约束，我们就会得到一种称为**稀疏编码**的方法。在这种情况下，我们称因子加载矩阵 $\mathbf{W}$ 为**字典**；每列称为一个**原子**。[13]鉴于稀疏表示，$L > D$ 是常见的，在这种情况下，我们称其为**过完备**表示。

在稀疏编码中，字典可以固定或学习。如果它是固定的，则通常使用小波或 DCT 基，因为许多自然信号可以由少量此类基函数很好地近似。然而，通过最大化可能性，也可以学习字典

$$\log p(D|\mathbf{W}) = \sum_{i=1}^{N} \log \int_{\mathbf{z}_i} \mathcal{N}(\mathbf{x}_i|\mathbf{W}\mathbf{z}_i, \sigma^2\mathbf{I}) p(\mathbf{z}_i) d\mathbf{z}_i \tag{13.174}$$

我们将在下面讨论优化方法，然后介绍几个有趣的应用程序。

不要将稀疏编码与**稀疏 PCA** 混淆（参见例如（Witten 等人 2009；Journee 等人。2010））：这将稀疏性提升先验放在回归权重 $\mathbf{W}$ 上，而在稀疏编码中，我们将稀疏性提高先验放在潜在因子 $\mathbf{z}_i$ 上。当然，这两种技术可以结合使用；我们称之为**稀疏矩阵分解**的结果，尽管这个术语是非标准的。有关我们的术语总结，请参见表 13.3。

2.8.1 学习稀疏编码字典

由于等式 13.174 是一个难以最大化的目标，因此通常采用以下近似值：

$$\log p(D|\mathbf{W}) \approx \sum_{i=1}^{N} \max_{\mathbf{z}_i} [\log \mathcal{N}(\mathbf{x}_i|\mathbf{W}\mathbf{z}_i, \sigma^2\mathbf{I}) + \log p(\mathbf{z}_i)] \tag{13.175}$$

如果 $p(\mathbf{z}_i)$ 是拉普拉斯，我们可以将 NLL 重写为

$$NLL(\mathbf{W}, \mathbf{Z}) = \sum_{i=1}^{N} \frac{1}{2}||\mathbf{x}_i - \mathbf{W}\mathbf{z}_i||_2^2 + \lambda||\mathbf{z}_i||_1 \tag{13.176}$$

为了防止 $\mathbf{W}$ 变得任意大，通常约束 ℓ_2 其列的范数小于或等于 1。让我们用

$$\mathbf{C} = \{\mathbf{W} \in \mathbb{R}^{D \times L} \;\; s.t. \;\; \mathbf{w}_j^T \mathbf{w}_j \leq 1\} \tag{13.177}$$

然后我们要解决 $min_{\mathbf{w} \in c, \mathbf{z} \in \mathbb{R}^{N \times L}} NLL(\mathbf{W}, \mathbf{Z})$。对于固定的 $\mathbf{z}_i$，$\mathbf{W}$ 上的优化是一个简单的最小二乘问题。对于固定字典 $\mathbf{W}$，$\mathbf{Z}$ 上的优化问题与 lasso 问题相同，存在许多快速算法。这表明了一个明显的迭代优化方案，我们在优化 $\mathbf{W}$ 和 $\mathbf{Z}$ 之间交替进行。（Mumford 1994）将这种方法称为**分析合成循环**，其中估计基 $\mathbf{W}$ 是分析阶段，估计系数 $\mathbf{Z}$ 是合成阶段。在速度太慢的情况下，可以使用更复杂的算法，例如（Mairal 等人，2010）。

多种其他模型导致类似于等式 13.176 的优化问题。例如，非负矩阵分解或 NMF（Paatero 和 Tapper 1994；Lee 和 Seung 2001）需要求解形式的目标

[13]通常用 $\mathbf{D}$ 表示字典，用 $\boldsymbol{\alpha}_i$ 表示潜在因素。然而，我们将坚持使用 $\mathbf{W}$ 和 $\mathbf{z}_i$ 符号。

$$\min_{\mathbf{w}\in c,\mathbf{z}\in\mathbb{R}^{N\times L}} \frac{1}{2}\sum_{i=1}^{N}||\mathbf{x}_i-\mathbf{W}\mathbf{z}_i||_2^2 \;\; \mathbf{s}.t. \;\; \mathbf{W}\geq 0, \mathbf{z}_i\geq 0 \tag{13.178}$$

（请注意，这没有可调整的超参数。）该约束背后的直觉是，如果学习字典是正“部分”的正和，而不是正或负的原子稀疏和，则它可能更易于解释。当然，我们可以将 NMF 与潜在因素的稀疏性提升先验结合起来。这称为**非负稀疏编码**（Hoyer 2004）。

或者，我们可以放弃正性约束，但对因子 $\mathbf{z}_i$ 和字典 $\mathbf{W}$ 施加稀疏性约束。我们称之为**稀疏矩阵分解**。为了确保严格的凸性，我们可以对权重使用弹性网类型惩罚（Mairal et al.2010），从而产生：

$$\min_{\mathbf{W},\mathbf{Z}} \frac{1}{2}\sum_{i=1}^{N}||\mathbf{x}_i-\mathbf{W}\mathbf{z}_i||_2^2+\lambda||\mathbf{z}_i||_1 \;\; s.t. \;\; ||\mathbf{w}_j||_2^2+\gamma||\mathbf{w}_j||_1\leq 1 \tag{13.179}$$

有几个相关的目标可以写下来。例如，我们可以用组 lasso 或融合 lasso 代替 lasso NLL（Witten 等人，2009）。除了拉普拉斯，我们还可以使用其他稀疏性促进先验。例如（Zhou 等人。2009）提出一种模型，其中使用第 13.2.2 节的二进制掩码模型使潜在因子 $\mathbf{z}_i$ 稀疏。掩码的每一位都可以从具有参数 π 的伯努利分布中生成，该分布可以从贝塔分布中提取。或者，我们可以使用非参数先验，如 beta 过程。这允许模型使用大小无限的字典，而不必事先指定 L。可以使用例如吉布斯采样或变分贝叶斯在该模型中执行贝叶斯推断。人们发现，由于贝叶斯奥卡姆剃刀，字典的有效大小随着噪声水平的升高而减小。这可以防止过度装配。详情见（Zhou 等人，2009 年）。

2.8.2 从图像块进行字典学习的结果

稀疏编码最近引起如此多兴趣的一个原因是它解释了神经科学中一个有趣的现象。特别是，通过对自然图像块应用稀疏编码学习的字典由基向量组成，这些基向量看起来像哺乳动物大脑初级视觉皮层中简单细胞中的滤波器（Olshausen 和 Field 1996）。特别是，滤波器看起来像条形和边缘检测器，如图 13.21（b）所示。（在本例中，选择参数 λ 的目的是使主动基函数（$\mathbf{z}_i$ 的非零分量）的数量约为 10。）有趣的是，使用 ICA 可以得到视觉上类似的结果，如图 13.21（a）所示。相比之下，对相同数据应用主成分分析会产生正弦光栅，如图 13.21（c）所示；这些看起来不像皮质细胞反应模式。[14]因此，推测部分皮层可能正在执行感觉输入的稀疏编码；由此产生的潜在表征然后由更高层次的大脑进一步处理。

图 13.21（d）显示了使用 NMF 的结果，图 13.22（e-f）显示了稀疏 PCA 的结果，因为我们增加了基向量的稀疏性。

[14]PCA 发现正弦光栅图案的原因是，它试图对数据的协方差进行建模，在图像块的情况下，协方差是平移不变的。这意味着 $cov[I(x,y),I(x',y')]=f[(x-x')^2+(y-y')^2]$. 对于某些函数 f，其中 $I(x,y)$ 是位置 (x,y) 处的图像强度。可以证明（Hyvarinen 等人，2009 年，第 125 页），此类矩阵的特征向量始终是不同相位的正弦曲线，即 PCA 发现傅里叶基。

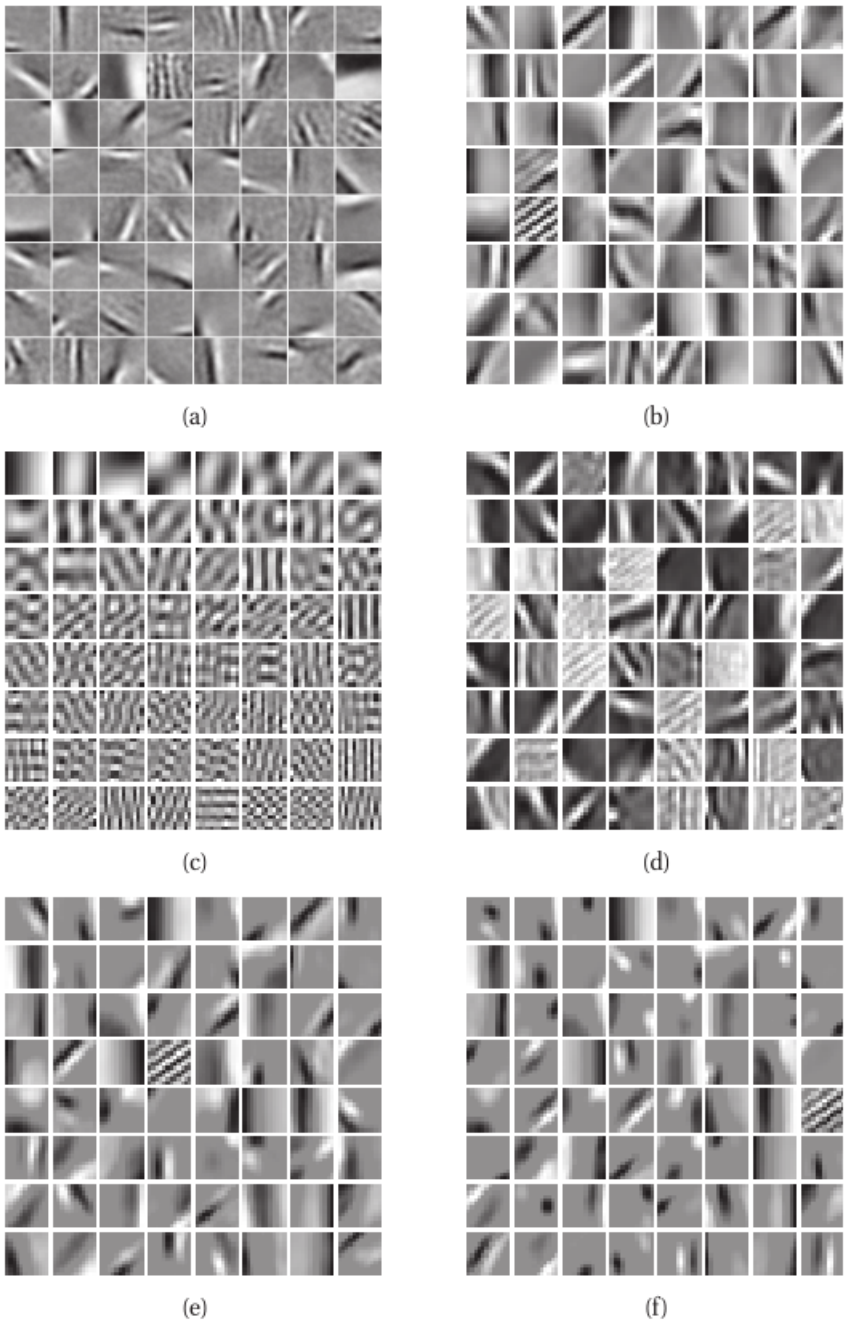


图 13.21：当应用于自然图像块时，通过各种方法学习的滤波器示意图。(每个面片首先居中并归一化为单位范数。)（a）ICA。图由 icaBasisDemo 生成，由 Aapo Hyvarinen 提供。（b）稀疏编码。（c）主成分分析。（d）非负矩阵分解。（e）权重矩阵稀疏性低的稀疏主成分分析。（f）权重矩阵上具有高稀疏性的稀疏 PCA。图由 sparseDictDemo 生成，由 Julien Mairal 编写。

2.8.3 压缩传感

虽然通过稀疏编码学习的字典很有趣，但它不一定非常有用。然而，稀疏编码有一些实际应用，我们将在下面讨论。

想象一下，与其观察数据 $x \in \mathbb{R}^D$ 我们观察到它的低维投影，$\mathbf{y} = \mathbf{R}\mathbf{x} + \boldsymbol{\epsilon}$, 其中，$\mathbf{y} \in \mathbb{R}^M$,$\mathbf{R}$ 是 $M \times D$ 矩阵，$M \ll D$、然后 ϵ 是噪声项（通常为高斯）。我们假设 $\mathbf{R}$ 是一个已知的传感矩阵，对应于 $\mathbf{x}$ 的不同线性投影。例如，考虑一个 MRI 扫描仪：每个波束方向对应一个矢量，在 $\mathbf{R}$ 中编码为一行。图 13.22 说明了建模假设。

我们的目标是推断 $p(\mathbf{x}|\mathbf{y}, \mathbf{R})$。如果我们不测量所有的 $\mathbf{x}$，我们怎么可能希望恢复所有的 $\mathbf{x}$? 答案是：我们可以使用具有适当先验的贝叶斯推理，利用自然信号可以表示为少量适当选择的基函数的加权组合这一事实。也就是说，我们假设 $\mathbf{x} = \mathbf{W}\mathbf{z}$，其中 $\mathbf{z}$ 具有稀疏先验，并且 $\mathbf{W}$ 是合适的字典。这被称为压缩传感或压缩传感（Candes 等人，2006 年；Baruniak 2007 年；Candes 和 Wakin 2008 年；Bruckstein 等人，2009 年）。

对于 CS 来说，以正确的基础表示信号很重要，否则它将不会稀疏。在传统的 CS 应用程序中，字典固定为标准形式，如小波。然而，通过使用稀疏编码学习特定领域的词典，可以获得更好的性能（Zhou 等人，2009）。至于传感矩阵 $\mathbf{R}$，由于（Candes 和 Wakin 2008）中解释的原因，它通常被选择为随机矩阵。然而，通过将投影矩阵调整到字典中，可以获得更好的性能（Seeger 和 Nickish，2008；Chang 等人，2009）。

2.8.4 图像修复与去噪

假设我们有一幅图像在某种程度上被破坏了，例如，上面稀疏地叠加了文字或划痕，如图 13.23 所示。我们可能需要估计底层的“干净”图像。这称为**图像修复**。可以使用类似的技术进行**图像去噪**。

我们可以将其建模为一种特殊的压缩感知问题。基本思想如下。我们将图像分割为重叠的面片 $\mathbf{y}_i$，并将它们连接起来形成 $\mathbf{y}$。我们定义 $\mathbf{R}$，使得第 i 行选择面片 i。现在定义 $\mathcal{V}$ 为 $\mathbf{y}$ 的可见（未损坏）分量，$\mathcal{H}$ 为隐藏分量。为了进行图像修复，我们只需计算 $p(\mathbf{y}_\mathcal{H}|\mathbf{y}_\mathcal{V}, \boldsymbol{\theta})$，其中 $\boldsymbol{\theta}$ 是模型参数，指定字典 $\mathbf{W}$ 和 $\mathbf{z}$ 的稀疏度水平 λ。我们可以从图像数据库离线学习字典，也可以基于未损坏的补丁为该图像学习字典。

图 13.23 显示了该技术的实际应用。字典（大小为 256 个原子）是从 1200 万像素图像中的 7×10^6 个未损坏的 12×12 色块中学习的。

另一种方法是使用图形模型（例如**专家领域**模型（S.and Black 2009）），该模型直接编码相邻图像块之间的相关性，而不是使用潜变量模型。不幸的是，这种模型在计算上往往更昂贵。

练习

练习 13.1 RSS 的偏导数

定义

$$RSS(\mathbf{w}) = ||\mathbf{X}\mathbf{w} - \mathbf{y}||_2^2 \tag{13.180}$$

a. 证明

$$\frac{\partial}{\partial_{w_k}} RSS(\mathbf{w}) = a_k w_k - c_k \tag{13.181}$$

$$a_k = 2\sum_{i=1}^{n} x_{ik}^2 = 2||x_{:,k}||^2 \tag{13.182}$$

$$c_k = 2\sum_{i=1}^{n} x_{ik}(y_i - \mathbf{w}_{-k}^T \mathbf{x}_{i,-k}) = 2\mathbf{x}_{:,k}^T \mathbf{r}_k \tag{13.183}$$

其中 $\mathbf{w}_{-k} = \mathbf{w}$，不含分量 $k, \mathbf{x}_{i,-k}$ 是不含分量 k 的 $\mathbf{x}_i$，并且 $\mathbf{r}_k = \mathbf{y} - \mathbf{w}_{-k}^T \mathbf{x}_{:,-k}$ 是由于使用除特征 k 之外的所有特征而产生的残差。提示：将权重划分为涉及 k 的权重和不涉及 k 的权值。

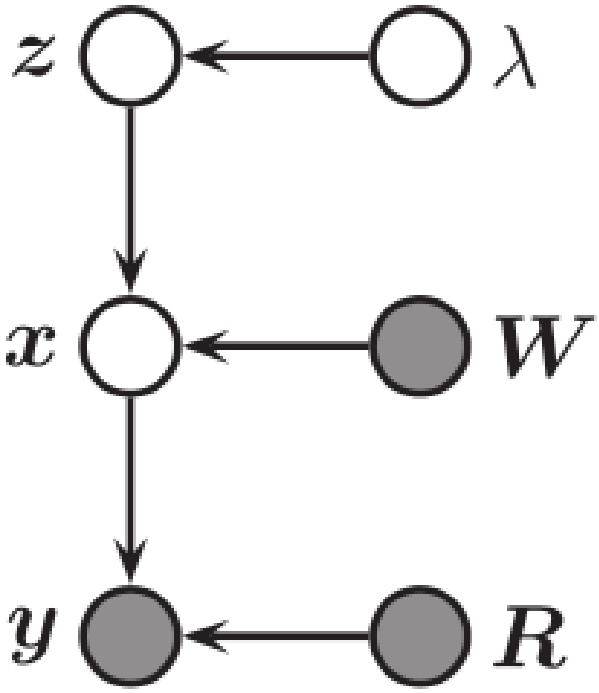


图 13.22: 压缩传感的 DGM 示意图。我们观察到一个低维测量 **y**，该测量 y 通过测量矩阵 **R** 传递 **x** 生成，并且可能受到方差 σ^2 的观测噪声。我们假设 **x** 具有字典 **W** 和潜变量 **z** 的稀疏分解。参数 λ 控制稀疏度水平。

b. 证明, 当 $\frac{\partial}{\partial_{w_k}} RSS(\mathbf{w}) = 0$,

$$\hat{w}_k = \frac{\mathbf{x}_{:,k}^T \mathbf{r}_k}{||\mathbf{x}_{:,k}||^2} \tag{13.184}$$

因此，当我们顺序添加特征时，特征 k 的最佳权重是通过计算将 $\mathbf{x}_{:,k}$ 正交投影到当前残差上来计算的。

练习 13.2 线性回归 EB M 步的推导

推导公式 13.166 和 13.168。提示：以下等式应该有用

$$\begin{aligned}
\sum \mathbf{X}^T\mathbf{X} &= \sum \mathbf{X}^T\mathbf{X} + \beta^{-1}\sum \mathbf{A} - \beta^{-1}\sum \mathbf{A} && (13.185)\\
&= \sum(\mathbf{X}^T\mathbf{X}\beta + \mathbf{A})\beta^{-1} - \beta^{-1}\sum \mathbf{A} && (13.186)\\
&= (\mathbf{A} + \beta\mathbf{X}^T\mathbf{X})^{-1}(\mathbf{X}^T\mathbf{X}\beta + \mathbf{A})\beta^{-1} - \beta^{-1}\sum \mathbf{A} && (13.187)\\
&= (\mathbf{I} - \mathbf{A}\sum)\beta^{-1} && (13.188)
\end{aligned}$$

练习 13.3 线性回归 EB 不动点更新的推导

推导方程 13.169 和 13.170。提示：推导该结果的最简单方法是重写方程 8.54 中的 $\log p(D|\boldsymbol{\alpha}, \beta)$。这完全等价，因为在高斯先验和似然的情况下，后验也是高斯的，因此拉普拉斯“近似”是精确的。在这种情况下，我们得到

$$\log p(D|\boldsymbol{\alpha}, \beta) = \frac{N}{2}\log - \frac{\beta}{2}||\mathbf{y} - \mathbf{X}\mathbf{w}||^2$$

(a) (b)

图 13.23: 使用稀疏编码的图像修复示例。左：原始图像。右图：重建。资料来源：图 13（Mairal 等人，2008 年）。经朱利安 • 迈拉尔善意许可使用。

$$+\frac{1}{2}\sum_j \log\alpha_j - \frac{1}{2}\mathbf{m}^T\mathbf{A}\mathbf{m} + \frac{1}{2}\log|\sum| - \tfrac{D}{2}\log(2\pi) \tag{13.189}$$

剩下的就是简单的代数。

练习 13.4 线性回归的边际似然

假设我们使用 $\sum_\gamma = g(\mathbf{X}_\gamma^T\mathbf{X}_\gamma)^{-1}$ 形式的 g 先验. 表明等式 13.16 简化为：

$$p(D|\gamma) \propto (1+g)^{-D_\gamma/2}(2b_\sigma + S(\gamma))^{-(2a_\sigma+N-1)/2} \tag{13.190}$$

$$S(\gamma) = \mathbf{y}^T\mathbf{y} - \frac{g}{1+g}\mathbf{y}^T\mathbf{X}_\gamma(\mathbf{X}_\gamma^T\mathbf{X}_\gamma)^{-1}\mathbf{X}_\gamma^T\mathbf{y} \tag{13.191}$$

练习 13.5 将弹性网还原为 lasso

定义

$$J_1(\mathbf{w}) = |\mathbf{y} - \mathbf{X}\mathbf{w}|^2 + \lambda_2|\mathbf{w}|^2 + \lambda_1|\mathbf{w}|_1 \tag{13.192}$$

和

$$J_2(\mathbf{w}) = |\widetilde{y} - \widetilde{X}\widetilde{w}|^2 + c\lambda_1|\mathbf{w}|_1 \tag{13.193}$$

其中 $c = (1+\lambda_2)^{-\frac{1}{2}}$,

$$\widetilde{\boldsymbol{X}} = c\begin{pmatrix}\mathbf{X}\\ \sqrt{\lambda_2}\mathbf{I}_d\end{pmatrix}, \widetilde{\boldsymbol{y}} = \begin{pmatrix}\mathbf{y}\\ \mathbf{0}_{d\times 1}\end{pmatrix} \tag{13.194}$$

证明

$$arg\ min J_1(\mathbf{w}) = c(arg\ min J_2(\mathbf{w})) \tag{13.195}$$

即

$$J_1(c\mathbf{w}) = J_2(\mathbf{w}) \tag{13.196}$$

因此，可以在修改的数据上使用 lasso 解算器来解决弹性网问题。

练习 13.6 线性回归中的收缩

（来源：Jaakkola。）考虑使用正交设计矩阵进行线性回归，因此，对于每列（特征）k，$||\mathbf{x}_{:,k}||_2^2 = 1$，$\mathbf{x}_{:,k}^T\mathbf{x}_{:,j} = 0$，因此我们可以分别估计每个参数 w_k。

图 13.24 绘制了 3 种不同估计方法的 $\hat{w}_k$ vs $c_k = 2\mathbf{y}^T\mathbf{x}_{:,k}$，特征 k 与响应的相关性：普通最小二乘法（OLS）、带参数 λ_2 的岭回归和带参数 λ_1 的 lasso。

a. 不幸的是，我们忘了给这些地块贴标签。实线（1）、虚线（2）和虚线（3）对应于哪种方法？提示：见第 13.3.3 节。

b. λ_1 的值是多少？

c. λ_2 的值是多少？

练习 13.7 之前，针对尖峰和平板模型中的伯努利速率参数

考虑第 13.2.1 节中的模型。假设我们对稀疏率 $\pi_j \sim Beta(\alpha_1, \alpha_2)$。在积分 π_j 之后，导出 $p(\boldsymbol{\gamma}|\boldsymbol{\alpha})$ 的表达式。讨论与固定 π_0 假设 $\pi_j = \pi_0$ 相比，该方法的一些优点和缺点。

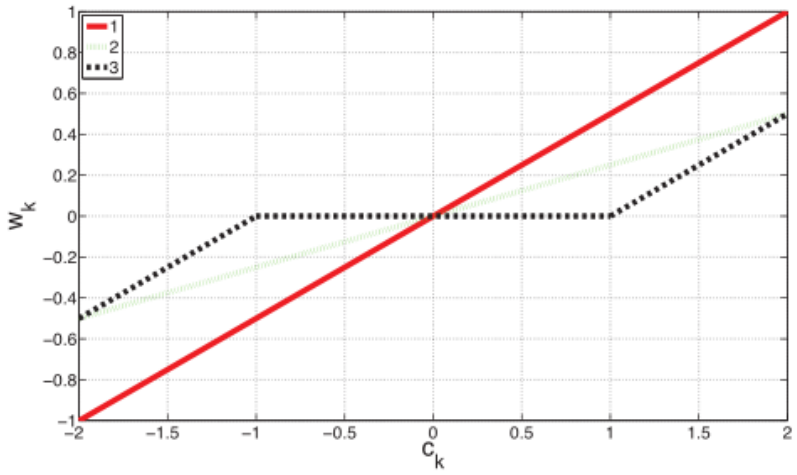


图 13.24: 三种不同估计器的 $\hat{w}_k$ 与相关量 c_k 的曲线图。

练习 13.8 推导 GSM 之前的 E 步骤

证明

$$\mathbb{E}\left[\frac{1}{\tau_j^2}|w_j\right] = \frac{\pi'(w_j)}{|w_j|} \tag{13.197}$$

其中 $\pi(w_j) = -\log p(w_j)$ 和 $p(w_j) = int\mathcal{N}(w_j|0,\tau_j^2)p(\tau_j^2)d\tau_j^2$ 提示 1：

$$\frac{1}{\tau_j^2}\mathcal{N}(w_j|0,\tau_j^2) \propto \frac{1}{\tau_j^2}exp(-\frac{w_j^2}{2\tau_j^2}) \tag{13.198}$$

$$= \frac{-1}{|w_j|}\frac{-2w_j}{2\tau_j^2}exp(-\frac{w_j^2}{2\tau_j^2}) \tag{13.199}$$

$$= \frac{-1}{|w_j|}\frac{d}{d|w_j|}\mathcal{N}(w_j|0,\tau_j^2) \tag{13.200}$$

提示 2：

$$\frac{d}{d|w_j|}p(w_j) = \frac{1}{p(w_j)}\frac{d}{d|w_j|}\log p(w_j) \tag{13.201}$$

练习 13.9 拉普拉斯先验稀疏概率回归的 EM

推导一种 EM 算法，用于使用拉普拉斯先验对权重进行二元概率分类器拟合（第 9.4 节）。（如果你陷入困境，请参见（Figueiredo 2003；丁和哈里森 2010）

练习 13.10 组 lasso 的 GSM 表示

考虑先验 $\tau_j^2 \sim Ga(\delta, \rho^2/2)$，暂时忽略分组问题。由伽马混合分布在权重上产生的边际分布称为**正态伽马分布**，由下式给出：

$$NG(w_j|\delta,\rho) = \int \mathcal{N}(w_j|0,\tau_j^2)Ga(\tau_j^2|\delta,\rho^2/2)d\tau_j^2 \tag{13.202}$$

$$= \frac{1}{Z}|w_j|^{\delta-1/2}\mathcal{K}_{\delta-\frac{1}{2}}(\rho|w_j|) \tag{13.203}$$

$$1/Z = \frac{\rho^{\delta+\frac{1}{2}}}{\sqrt{\pi}2^{\delta-1/2}\rho(\delta)} \tag{13.204}$$

其中 $\mathcal{K}_\alpha(x)$ 是第二类修正贝塞尔函数（Matlab 中的贝塞尔函数）。

现在假设我们有以下关于方差的先验知识：

$$p(\boldsymbol{\sigma}^2_{1:D}) = \prod_{g=1}^{G} p(\boldsymbol{\sigma}^2_{1:d_g}), p(\boldsymbol{\sigma}^2_{1:d_g}) = \prod_{j\in g} Ga(\tau_j^2|\delta_g, \rho^2/2) \tag{13.205}$$

每组权重的相应边缘具有以下形式：

$$p(\mathbf{w}_g) \propto |u_g|^{\delta_g - d_g/2}\mathcal{K}_{\delta_g-d_g/2}(\rho u_g) \tag{13.206}$$

其中

$$u_g \triangleq \sqrt{\sum_{j\in g} w_{g,j}^2} = ||\mathbf{w}_g||_2 \tag{13.207}$$

现在假设 $\delta_g = (d_g+1)/2$，所以 $\delta_g - d_g/2 = \frac{1}{2}$。方便地，我们有 $\mathcal{K}_{\frac{1}{2}}(z) = \sqrt{\frac{\pi}{2z}}exp(-z)$。结果表明，得到的 MAP 估计等价于群 lasso。

练习 13.11 ℓ_1 正则化最小二乘的投影梯度下降

考虑 BPDN 问题 $argmin_{\boldsymbol{\theta}} RSS(\boldsymbol{\theta}) + \lambda||\boldsymbol{\theta}||_1$, 通过使用第 7.4 节中介绍的分裂变量技巧（即，通过定义 $\boldsymbol{\theta} = [\boldsymbol{\theta}_+, \boldsymbol{\theta}_-]$, 将其重写为具有简单边界约束的二次规划。然后简述如何使用投影梯度下降来解决此问题。（如果你陷入困境，请咨询（Figueiredo 等人，2007 年）

练习 13.12 铰链损失函数的次导数

设 $f(x) = (1-x)_+$ 是铰链损失函数，其中 $(z)_+ = max(0, z)$。什么是 $\partial f(0), \partial f(1), \partial f(2)$?

练习 13.13 凸函数的下界

设 f 是凸函数。解释如何在任意点 $\mathbf{x} \in dom(f)$ 处找到 f 的全局仿射下界

3 线性回归模型

到目前为止，本书的重点一直放在无监督学习上，包括密度估计和数据聚类等主题。现在我们转向讨论监督学习，从回归开始。回归的目标是在给定 D 维**输入变量** $\mathbf{x}$ 的情况下预测一个或多个连续**目标变量** t 的值。在第一章中，我们已经遇到了回归问题的一个示例，即多项式曲线拟合。多项式是一类特定的函数，属于被称为线性回归模型的广泛函数类别。这些模型共享可调参数的线性函数属性，并且将是本章的重点。最简单形式的线性回归模型也是输入变量的线性函数。然而，通过对一组固定的非线性函数（称为**基函数**）进行线性组合，我们可以得到一类更加有用的函数。这些模型在参数方面是线性函数，因此具有简单的分析性质，但在输入变量方面可能是非线性的。

给定一个包含 N 个观测值 $\{\mathbf{x}_n\}$ 的训练数据集，其中 $n = 1, \cdots, N$，以及相应的目标值 $\{t_n\}$，目标是预测新值 $\mathbf{x}$ 的 t 值。在最简单的方法中，这可以通过直接构造一个适当的函数 $y(\mathbf{x})$ 来完成，其对新输入 $\mathbf{x}$ 的值构成了 t 相应值的预测。更一般地，从概率的角度来看，我们的目标是建模预测分布 $p(t|\mathbf{x})$，因为这表达了我们对每个 $\mathbf{x}$ 值的 t 值的不确定性。从这个条件分布中，我们可以对任何新的值 $\mathbf{x}$ 进行 t 的预测，以最小化适当选择的损失函数的期望值。如第 1.5.5 节所述，实值变量的常用损失函数是平方损失，其最优解由 t 的条件期望给出。

尽管线性模型作为模式识别的实用技术具有很大的局限性，特别是对于涉及高维输入空间的问题，但它们具有良好的分析性质，并为后面章节将要讨论的更复杂模型奠定了基础。

3.1 线性基函数模型

最简单的线性回归模型是指涉及输入变量线性组合的模型。

$$y(\mathbf{x}, \mathbf{w}) = w_0 + w_1 x_1 + \ldots + w_D x \tag{3.1}$$

其中 $\mathbf{x} = (x_1, \ldots, x_D)^{\mathrm{T}}$。这通常被称为**线性回归**。该模型的关键特性是它是参数 $w_0, \ldots, w_D$ 的线性函数。然而，它同时也是输入变量 x_i 的线性函数，这对模型施加了显著的限制。因此，我们通过考虑输入变量 x_i 的固定非线性函数的线性组合来扩展模型的类别，形式为：

$$y(\mathbf{x}, \mathbf{w}) = w_0 + \sum_{j=1}^{M-1} w_j \phi_j(\mathbf{x}) \tag{3.2}$$

其中，$\phi_j(\mathbf{x})$ 被称为**基函数**。通过用 $M-1$ 表示索引 j 的最大值，该模型中的参数总数将为 M。参数 w_0 允许数据中的存在任何固定偏移，并且有时被称为**偏置参数**（与统计意义上的“偏差”不要混淆）。通常方便定义一个额外的虚拟“基函数” $\phi_0(\mathbf{x}) = 1$，以便于以下形式的表示：

$$y(\mathbf{x}, \mathbf{w}) = \sum_{j=0}^{M-1} w_j \phi_j(\mathbf{x}) = \mathbf{w}^{\mathrm{T}} \boldsymbol{\phi}(\mathbf{x}) \tag{3.3}$$

其中，$\mathbf{w} = (w_0, \ldots, w_{M-1})^{\mathrm{T}}$ 和 $\boldsymbol{\phi} = (\phi_0, \ldots, \phi_{M-1})^{\mathrm{T}}$。在模式识别的许多实际应用中，我们会对原始数据变量应用某种形式的固定预处理或特征提取。如果原始变量包含向量 $\mathbf{x}$，则特征可以用基函数 $\{\phi_j(\mathbf{x})\}$ 来表示。

通过使用非线性基函数，我们允许函数 $y(\mathbf{x}, \mathbf{w})$ 成为输入向量 $\mathbf{x}$ 的非线性函数。式 (3.2) 中的函数被称为线性模型，因为该函数对 $\mathbf{w}$ 是线性的。正是这种对参数 $\mathbf{w}$ 的线性性质极大地简化了对这类模型的分析。然而，它也导致了一些显著的限制，正如我们在第 3.6 节中所讨论的那样。

第 1 章中考虑的多项式回归示例就是这个模型的一个特例，其中只有一个输入变量 $\mathbf{x}$，并且基函数采用 $\mathbf{x}$ 的幂次形式，即 $\phi_j(\mathbf{x}) = x^j$。多项式基函数的一个限制是它们是输入变量的全局函数，因此输入空间的一个区域的变化会影响所有其他区域。这可以通过将输入空间分割成不同的区域，并在每个区域内拟合不同的多项式来解决，从而导致样条函数（Spline functions）(Hastie et al., 2001)。

还有许多其他可能的基函数选择，例如

$$\phi_j(x) = \exp\left\{-\frac{(x-\mu_j)^2}{2s^2}\right\} \tag{3.4}$$

$$\phi_j(x) = \sigma\left(\frac{x-\mu_j}{s}\right) \tag{3.5}$$

$$\sigma(a) = \frac{1}{1+\exp(-a)} \tag{3.6}$$

$$t = y(\mathbf{x}, \mathbf{w}) + \epsilon \tag{3.7}$$

$$p(t|\mathbf{x}, \mathbf{w}, \beta) = \mathcal{N}(t|y(\mathbf{x}, \mathbf{w}), \beta^{-1}). \tag{3.8}$$

$$\mathbb{E}[t|\mathbf{x}] = \int tp(t|\mathbf{x})\mathrm{d}t = y(\mathbf{x}, \mathbf{w}) \tag{3.9}$$

$$p(\mathbf{t}|\mathbf{X}, \mathbf{w}, \beta) = \prod_{n=1}^{N} \mathcal{N}(t_n|\mathbf{w}^{\mathrm{T}}\boldsymbol{\phi}(\mathbf{x}_n), \beta^{-1}) \tag{3.10}$$

$$\begin{aligned}\ln p(\mathbf{t}|\mathbf{w}, \beta) &= \sum_{n=1}^{N} \ln \mathcal{N}(t_n|\mathbf{w}^{\mathrm{T}}\boldsymbol{\phi}(\mathbf{x}_n), \beta^{-1}) \\ &= \frac{N}{2}\ln\beta - \frac{N}{2}\ln(2\pi) - \beta E_D(\mathbf{w})\end{aligned} \tag{3.11}$$

3.1.1 最大似然和最小平方

3.2 偏置-方差分解