

雷达对抗讲义

目 录

第 1 章 基础知识	1
1.1 符号规定	1
1.2 矩阵微积分	1
1.3 复矩阵微积分	4
1.4 常见统计量	6
1.5 拉格朗日乘数法	6
第 2 章 无源定位	9
2.1 测向定位	9
2.1.1 二维平面	9
2.1.2 三维空间	13
2.2 时差定位	16
2.3 频差定位	18
第 3 章 雷达信号处理基础	21
3.1 雷达信号的基本形式	21
3.2 目标参数测量	24
3.2.1 测距	24
3.2.2 测角	26
3.3 测速	28
第 4 章 超分辨率参数测量	29
4.1 谱估计	29
4.1.1 补零法	29
4.1.2 Capon 算法	30
4.1.3 MUSIC 算法	30
4.2 来波方位估计	30
4.2.1 Capon 算法	30
4.2.2 MUSIC 算法	31
4.2.3 ESPRIT 算法	32

4.3 时延估计	32
4.3.1 互相关法	32
4.3.2 相位拟合法	33
 第 5 章 雷达抗干扰	 35
5.1 最小二乘	35
5.2 独立成分分析	36
5.3 主偏度分析	38
5.3.1 偏度的定义	38
5.3.2 数据白化	39
5.3.3 张量基本运算	40
5.3.4 统计量映射图	42
5.3.5 任意方向的偏度	44
5.3.6 协偏度张量的计算	45
5.3.7 模型与求解	47

第 5 章 雷达抗干扰

在实际应用中，雷达系统往往受到各种干扰的影响，这些干扰可能来自环境噪声、其他信号源或系统本身的非理想特性。本章将介绍几种常用的抗干扰技术，包括最小二乘法、独立成分分析和主偏度分析。

5.1 最小二乘

设有无源相控阵雷达，其和通道和差通道的接收信号向量分别为

$$\begin{aligned}\mathbf{x} &= \mathbf{r} + \mathbf{g} + \mathbf{n}_1 \in \mathbb{C}^{L \times 1} \\ \mathbf{y} &= \alpha \mathbf{g} + \mathbf{n}_2 \in \mathbb{C}^{L \times 1}\end{aligned}$$

其中， $\mathbf{r}$ 是目标信号， $\mathbf{g}$ 是干扰信号， $\mathbf{n}_1$ 和 $\mathbf{n}_2$ 分别是和通道和差通道的噪声。我们知道，和通道往往是指向目标的，因此可以同时接收目标信号和干扰信号，而差通道在目标方向上接收干扰信号的能力较弱，因此，可以认为其接收到的信号主要是干扰信号。

由于参数 α 是未知的，我们可以通过最小二乘法来估计它。最小二乘法的目标是最小化以下损失函数：

$$J(\alpha) = \|\mathbf{x} - \alpha \mathbf{y}\|^2$$

对 α 求导并令其为零，可以得到最小二乘估计：

$$\hat{\alpha} = \frac{\mathbf{x}^H \mathbf{y}}{\mathbf{y}^H \mathbf{y}}$$

代入，可以得到去干扰后的信号：

$$\begin{aligned}\hat{\mathbf{r}} &= \mathbf{x} - \hat{\alpha} \mathbf{y} = \mathbf{r} + \mathbf{g} + \mathbf{n}_1 - \frac{\mathbf{x}^H \mathbf{y}}{\mathbf{y}^H \mathbf{y}} (\alpha \mathbf{g} + \mathbf{n}_2) \\ &= \mathbf{r} + \frac{\sigma^2}{\alpha^2 + \sigma^2} \mathbf{g} + \mathbf{n}_1 - \frac{\alpha}{\alpha^2 + \sigma^2} \mathbf{n}_2.\end{aligned}$$

不难发现，去干扰后的信号仍然包含干扰成分，但其幅度被缩小了。并且，噪声 α 越大，去干扰后的信号越接近目标信号。

实际应用中，和差通道之间的关系可能并不是简单的线性关系，如下所示：

$$\begin{aligned}\mathbf{x} &= \mathbf{r} + \mathbf{g} + \mathbf{n}_1 \in \mathbb{C}^{L \times 1} \\ \mathbf{y} &= f(\mathbf{g}) + \mathbf{n}_2 \in \mathbb{C}^{L \times 1}\end{aligned}$$

其中 $f(\cdot)$ 是一个系统。不妨假设 $f(\cdot)$ 是一个线性系统，则可以用矩阵表示为

$$f(\mathbf{g}) = \mathbf{G}\mathbf{w}$$

其中 $\mathbf{G}$ 是由 $\mathbf{g}$ 生成的矩阵:

$$\mathbf{G} = \begin{bmatrix} g_0 & g_1 & \cdots & g_{N-1} \\ g_1 & g_2 & \cdots & g_N \\ \vdots & \vdots & \ddots & \vdots \\ g_{L-N} & g_{L-N+1} & \cdots & g_{L-1} \end{bmatrix}^T$$

而 $\mathbf{w}$ 是系统的权重向量。

因此，同样利用 $\mathbf{y}$ 构造矩阵 $\mathbf{Y}$ ，我们可以得到如下的优化模型

$$\|\mathbf{x} - \mathbf{Y}\mathbf{w}\|^2$$

对 $\mathbf{w}$ 求导并令其为零，可以得到最小二乘估计：

$$\hat{\mathbf{w}} = (\mathbf{Y}^H \mathbf{Y})^{-1} \mathbf{Y}^H \mathbf{x}$$

该滤波器便是维纳滤波器（Wiener Filter），它可以有效地抑制干扰信号，并保留目标信号。

当需要进行实时处理时，维纳滤波器的计算复杂度可能较高，因此可以考虑使用自适应滤波器，如最小均方（LMS）算法。设 x_n 为和通道第 n 个采样点， $\mathbf{y}_n$ 为差通道第 n 时刻，前 N 个采样点构成的向量，则 LMS 算法的目标函数为：

$$J_n(\mathbf{w}) = |x_n - \mathbf{w}^H \mathbf{y}_n|^2$$

对 $\mathbf{w}$ 求导，可以得到梯度下降法的更新公式：

$$\mathbf{w}_{n+1} = \mathbf{w}_n + \mu \mathbf{e}_n \mathbf{y}_n$$

其中， μ 是步长因子， $\mathbf{e}_n = x_n - \mathbf{w}_n^H \mathbf{y}_n$ 是估计误差。

5.2 独立成分分析

对于一个大小为 $p \times n$ 的白化后的数据 $\mathbf{X}$ ，FastICA 算法的基本思想是找到一个投影方向 $\mathbf{u} \in \mathbb{R}^{p \times 1}$ ，使得投影后的数据 $\mathbf{u}^T \mathbf{X}$ 的某种非高斯性指标尽量大，相应的优化模型为

$$\begin{cases} \max_{\mathbf{u}} & G(\mathbf{u}^T \mathbf{X}) \mathbf{1} \\ \text{s.t.} & \mathbf{u}^T \mathbf{u} = 1 \end{cases}, \quad (5.1)$$

其中 $G(\cdot)$ 是用于衡量数据非高斯性的函数，常用的非高斯性衡量函数包括 x^3 ， x^4 ， $\log \cosh(x)$ ， $-e^{-x^2/2}$ 等。 $G(\mathbf{u}^T \mathbf{X})$ 代表对向量 $\mathbf{u}^T \mathbf{X}$ 中的每一个元素都用函数 $G(\cdot)$ 进

行非线性映射从而得到一个新的向量. 可以验证, 当 $G(x) = x^3$ 时, 优化模型 (5.1) 中的目标函数 $G(\mathbf{u}^T \mathbf{X}) \mathbf{1}$ 正好对应数据在 $\mathbf{u}$ 方向的偏度 (详见??公式??); 而当 $G(x) = x^4$ 时, 该目标函数则对应了数据在 $\mathbf{u}$ 方向的峭度.

FastICA 采用牛顿迭代法求解模型 (5.1), 首先构建如下的拉格朗日函数

$$\mathcal{L}(\mathbf{u}, \lambda) = G(\mathbf{u}^T \mathbf{X}) \mathbf{1} + \frac{1}{2} \lambda (1 - \mathbf{u}^T \mathbf{u}), \quad (5.2)$$

$\mathcal{L}(\mathbf{u}, \lambda)$ 关于自变量的导数为

$$\frac{\partial \mathcal{L}(\mathbf{u}, \lambda)}{\partial \mathbf{u}} = \mathbf{X} G'(\mathbf{u}^T \mathbf{X})^T - \lambda \mathbf{u}, \quad (5.3)$$

$\mathcal{L}(\mathbf{u}, \lambda)$ 关于自变量的黑塞 (Hessian) 矩阵[HYVARINEN2000411]为

$$\begin{aligned} \frac{\partial^2 \mathcal{L}(\mathbf{u}, \lambda)}{\partial \mathbf{u} \partial \mathbf{u}^T} &= \mathbf{X} \text{diag}(G''(\mathbf{u}^T \mathbf{X})) \mathbf{X}^T - \lambda \mathbf{I}_p \\ &\approx \frac{1}{n} \mathbf{X} \mathbf{X}^T G''(\mathbf{u}^T \mathbf{X}) \mathbf{1} - \lambda \mathbf{I}_p \\ &= \left(\frac{1}{n} G''(\mathbf{u}^T \mathbf{X}) \mathbf{1} - \lambda \right) \mathbf{I}_p. \end{aligned} \quad (5.4)$$

其中 $G'(\cdot)$ 和 $G''(\cdot)$ 分别表示 $G(\cdot)$ 的一阶和二阶导数, 它们的大小均与自变量的大小相同¹.

根据 (5.3) 和 (5.4), 利用牛顿迭代法, 我们可以得到如下的迭代公式

$$\begin{aligned} \mathbf{u} &\leftarrow \mathbf{u} - \left(\frac{\partial^2 \mathcal{L}(\mathbf{u}, \lambda)}{\partial \mathbf{u} \partial \mathbf{u}^T} \right)^{-1} \frac{\partial \mathcal{L}(\mathbf{u}, \lambda)}{\partial \mathbf{u}} \\ &= \mathbf{u} - \frac{\mathbf{X} G'(\mathbf{u}^T \mathbf{X})^T - \lambda \mathbf{u}}{\frac{1}{n} G''(\mathbf{u}^T \mathbf{X}) \mathbf{1} - \lambda}. \end{aligned} \quad (5.5)$$

由于接下来我们还需要对 $\mathbf{u}$ 进行归一化, 因此在 (5.5) 右边乘以 $(\frac{1}{n} G''(\mathbf{u}^T \mathbf{X}) \mathbf{1} - \lambda)$ 并不会影响最终结果, 所以可以将 (5.5) 改写为

$$\mathbf{u} \leftarrow \frac{1}{n} G''(\mathbf{u}^T \mathbf{X}) \mathbf{1} \mathbf{u} - \mathbf{X} G'(\mathbf{u}^T \mathbf{X})^T. \quad (5.6)$$

根据 (5.6) 给出的迭代公式, 我们可以得到 FastICA 求解第一个独立成分的具体步骤 (算法 5.1) .

算法 5.1 FastICA 求解第一个独立成分的流程

1. 随机初始化向量 $\mathbf{u}$
 2. 令 $\mathbf{u} \leftarrow \frac{1}{n} G''(\mathbf{u}^T \mathbf{X}) \mathbf{1} \mathbf{u} - \mathbf{X} G'(\mathbf{u}^T \mathbf{X})^T$
 3. 向量归一化: $\mathbf{u} \leftarrow \mathbf{u} / \|\mathbf{u}\|$
 4. 重复步骤 2 和 3, 直至收敛
-

¹比如 $G(x) = x^3$ 时, 其一阶导数为 $G'(x) = 3x^2$, 二阶导数为 $G''(x) = 6x$. 当自变量为 n 维行向量 $[x_1 \ \cdots \ x_n]$ 时, 我们有 $G([x_1 \ \cdots \ x_n]) = [x_1^3 \ \cdots \ x_n^3]$, $G'([x_1 \ \cdots \ x_n]) = [3x_1^2 \ \cdots \ 3x_n^2]$, $G''([x_1 \ \cdots \ x_n]) = [6x_1 \ \cdots \ 6x_n]$.

由于 FastICA 假设所有的独立成分之间相互正交, 因此在求解第 $l+1$ 个独立成分时, 迭代过程中的每一步都需要将 $\mathbf{u}_{l+1}$ 投影到前 l 投影方向 $\mathbf{U}_l = [\mathbf{u}_1 \ \mathbf{u}_2 \ \cdots \ \mathbf{u}_l]$ 的正交补空间中. 对应的算法流程见算法 5.2.

算法 5.2 FastICA 求解第 $l+1$ 个独立成分的流程

1. 随机初始化向量 $\mathbf{u}_{l+1}$
 2. 令 $\mathbf{u}_{l+1} \leftarrow \frac{1}{n} G''(\mathbf{u}_{l+1}^T \mathbf{X}) \mathbf{1} \mathbf{u}_{l+1} - \mathbf{X} G'(\mathbf{u}_{l+1}^T \mathbf{X})^T$
 3. 将 $\mathbf{u}_{l+1}$ 投影到 $\mathbf{U}_l$ 的正交补空间中: $\mathbf{u}_{l+1} \leftarrow \mathbf{P}_{\mathbf{U}_l}^\perp \mathbf{u}_{l+1}$
 4. 向量归一化: $\mathbf{u} \leftarrow \mathbf{u} / \|\mathbf{u}\|$
 5. 重复步骤 3 和 4, 直至收敛
-

5.3 主偏度分析

为了得到数据的偏度极值方向, 我们首先给出几个基本的数学概念.

5.3.1 偏度的定义

均值和方差是两个最常用的随机变量的数字特征, 接下来我们介绍随机变量的另外一个重要的数字特征——偏度.

定义 5.1 给定一个随机变量 X , 若 μ 为 X 的均值, σ 为 X 的标准差, 则称

$$\text{skew}(X) = \frac{E(X - \mu)^3}{\sigma^3} \quad (5.7)$$

为随机变量 X 的偏度.

在实际应用中, 往往只能得到随机变量的若干观测. 此时, 我们只能得到随机变量偏度的估计值, 也就是数据的三阶统计量——样本偏度. 接下来, 本书仍采用 $\text{skew}(\mathbf{x})$ 来表示观测数据 $\mathbf{x} = [x_1 \ x_2 \ \cdots \ x_n]^T$ 的样本偏度.

定义 5.2 给定随机变量 X 的 n 个观测值 $x_1, x_2, \dots, x_n$, 令 μ 为这些观测值的均值, 则称

$$\text{skew}(\mathbf{x}) = \frac{\frac{1}{n} \sum_{i=1}^n (x_i - \mu)^3}{\left(\frac{1}{n} \sum_{i=1}^n (x_i - \mu)^2\right)^{\frac{3}{2}}}, \quad (5.8)$$

为随机变量 X 的样本 $x_1, x_2, \dots, x_n$ 的样本偏度 (简称为偏度), 其中 $\mathbf{x}$ 是由 n 个观测构成的样本列向量.

对于一个矩阵 $\mathbf{X} \in \mathbb{R}^{p \times n}$, 可以将其视作 p 维随机向量的 n 个观测构成的数据. 这个数据在任意方向 $\mathbf{u} \in \mathbb{R}^{p \times 1}$ 上的偏度也能够根据 (5.8) 进行计算: 首先得到投影数据 $\mathbf{Y} = \mathbf{u}^T \mathbf{X}$, 然后根据 (5.8) 计算 $\mathbf{Y}$ 或 $\mathbf{Y}^T$ 的偏度即可.

相较于方差, 偏度是数据分布偏斜方向和程度的一个度量, 其值可以为任意实数.

如图 5.1 所示, 对称分布的数据, 其偏度值为零, 正偏态通常意味着数据的分布向左偏斜, 负偏态则反之. 因此, 偏度经常被用来描述具有非对称分布的数据. 此外, 鉴于

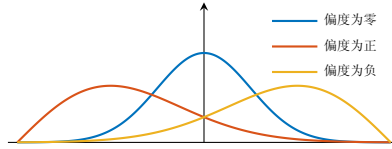


图 5.1 概率密度函数的分布与偏度的正负

远离中心的值对偏度的贡献较大, 偏度还经常被用来捕捉数据中的异常点. 图 5.2 给出了直观的示例, 分别展示了偏度为零、偏度为正和偏度为负这三种不同分布的简单模拟数据.

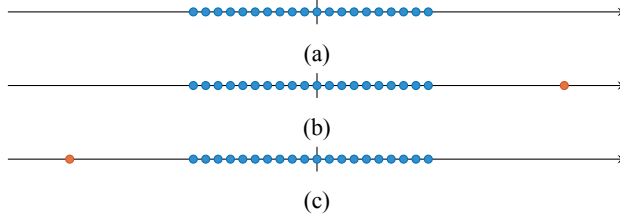


图 5.2 偏度与异常点的关系 (a) 偏度为零 (b) 偏度为正 (c) 偏度为负

5.3.2 数据白化

从公式 (5.8) 可以看出, 对于给定的数据, 其偏度的求取同时涉及分子和分母两项的计算. 为了简化这一过程, 接下来我们引入数据的白化算子, 其定义如下.

定义 5.3 给定数据 $\mathbf{X} = [\mathbf{x}_1 \ \mathbf{x}_2 \ \cdots \ \mathbf{x}_n] \in \mathbb{R}^{p \times n}$, 若其均值向量和协方差矩阵为

$$\boldsymbol{\mu} = \frac{1}{n} \mathbf{X} \mathbf{1}, \quad \boldsymbol{\Sigma} = \frac{1}{n} (\mathbf{X} - \boldsymbol{\mu} \mathbf{1}^T)(\mathbf{X} - \boldsymbol{\mu} \mathbf{1}^T)^T.$$

那么, 任意一个满足如下条件的矩阵 $\mathbf{W}$ 都可以称作数据 $\mathbf{X}$ 的白化算子

$$\mathbf{W}^T \mathbf{W} = \boldsymbol{\Sigma}^{-1}. \quad (5.9)$$

而白化后的数据有如下表达式

$$\hat{\mathbf{X}} = \mathbf{W}(\mathbf{X} - \boldsymbol{\mu} \mathbf{1}^T). \quad (5.10)$$

在本书中, 我们通常选择 $\mathbf{W} = \boldsymbol{\Sigma}^{-\frac{1}{2}}$ 作为白化算子. 可以验证, 白化后数据 $\hat{\mathbf{X}}$ 的均值向量为零向量, 协方差矩阵为单位矩阵, 即

$$\hat{\boldsymbol{\mu}} = \frac{1}{n} \hat{\mathbf{X}} \mathbf{1} = \mathbf{0}, \quad \hat{\boldsymbol{\Sigma}} = \frac{1}{n} \hat{\mathbf{X}} \hat{\mathbf{X}}^T = \mathbf{I}_p, \quad (5.11)$$

这意味着白化后的数据在任意方向上的方差均为 1.

图 5.3a 中给出了一组二维数据, 它的均值并不位于坐标系原点 $(0,0)$ 处, 沿着不同方向的方差也明显不一样. 对这个数据进行白化, 可以得到图 5.3b 中展示的结果. 特别地, 对于二维平面上任意三个不共线的点, 白化之后将变为正三角形的三个顶点;

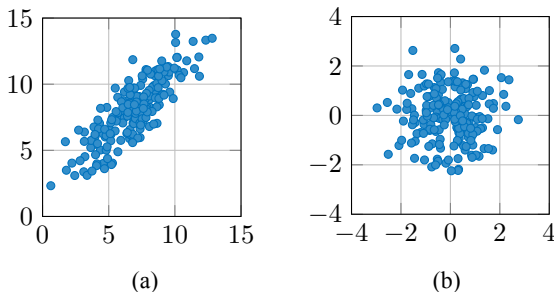


图 5.3 数据白化示例 (a) 原数据 (b) 白化后的数据

而三维空间上任意 4 个不共面的点, 白化之后将变为正四面体的 4 个顶点. 对此, 读者可自行验证.

对于白化后的数据 $\hat{\mathbf{X}}$, 其在任意方向 $\mathbf{u}$ 上的偏度计算公式为

$$\text{skew}(\mathbf{u}^T \hat{\mathbf{X}}) = \frac{1}{n} \sum_{i=1}^n \left(\mathbf{u}^T \hat{\mathbf{X}} \right)_i^3, \quad (5.12)$$

其中 $\left(\mathbf{u}^T \hat{\mathbf{X}} \right)_i$ 表示向量 $\mathbf{u}^T \hat{\mathbf{X}}$ 的第 i 个元素.

5.3.3 张量基本运算

在后文的推导中涉及到张量相关的一些运算, 因此有必要对其进行简单的介绍. 其中, k 模积是最基本的张量运算之一, 它可以看作是矩阵乘法的推广, 其定义如下.

定义 5.4 给定张量 $\mathcal{A} \in \mathbb{R}^{I_1 \times I_2 \times \cdots \times I_N}$ 与矩阵 $\mathbf{U} \in \mathbb{R}^{J \times I_k}$, 两者的 k 模积操作可以表示为

$$\mathcal{A} \times_k \mathbf{U} \in \mathbb{R}^{I_1 \times \cdots \times I_{k-1} \times J \times I_{k+1} \times \cdots \times I_N}, \quad (5.13)$$

k 模积结果中的元素具有如下表达式 ($a_{i_1 i_2 \cdots i_n}, u_{j i_k}$ 分别为张量 $\mathcal{A}$ 和矩阵 $\mathbf{U}$ 对应位置的元素)

$$(\mathcal{A} \times_k \mathbf{U})_{i_1 \cdots i_{k-1} j i_{k+1} \cdots i_N} = \sum_{i_k=1}^{I_k} a_{i_1 \cdots i_{k-1} i_k i_{k+1} \cdots i_N} u_{j i_k}. \quad (5.14)$$

我们知道, 矩阵乘以一个向量可以看作是用该向量中的元素对矩阵的各个列向量进行线性组合. 同样地, 张量与向量的 k 模积也可以看作作用向量中的元素对张量沿着

第 k 个维度的切片进行线性组合. 例如, 对于一个三阶张量 $\mathcal{A} \in \mathbb{R}^{3 \times 3 \times 4}$, 该张量 3 模积一个 1×4 大小的向量将会得到一个 3×3 的矩阵, 具体操作如图 5.4 所示. 在这个例子中, 线性组合的对象不再是向量, 而是张量沿着第三个维度的切片, 也就是矩阵.

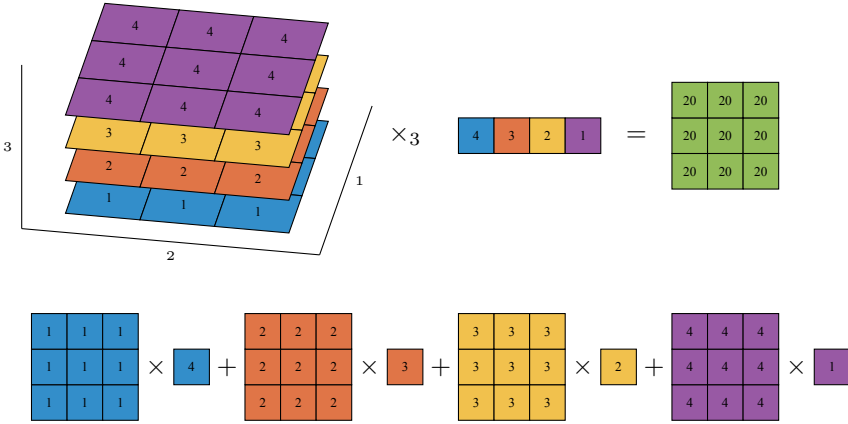


图 5.4 张量与向量 k 模积示意图

类似地, 张量与一个矩阵的 k 模积则可以看作是: 使用该矩阵的每一行对张量的第 k 个维度的切片进行线性组合得到一个新的切片, 然后将这些切片组合成一个新的张量. 例如, 对于一个三阶张量 $\mathcal{A} \in \mathbb{R}^{3 \times 3 \times 4}$, 其 3 模积一个 2×4 大小的矩阵会得到一个 $3 \times 3 \times 2$ 的张量, 具体操作如图 5.5 所示.

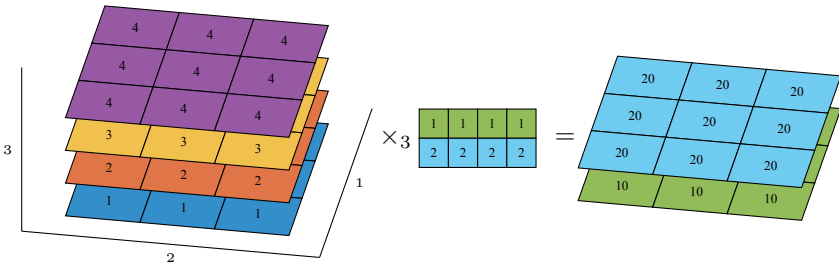


图 5.5 张量与矩阵 k 模积示意图

另一个常用的张量运算称作外积, 其定义如下.

定义 5.5 给定 N 个向量, $\mathbf{a}_i \in \mathbb{R}^{I_i \times 1}, i = 1, 2, \dots, N$, 它们的外积为一个秩为 1 的张量 $\mathcal{A} \in \mathbb{R}^{I_1 \times I_2 \times \dots \times I_N}$, 记作

$$\mathcal{A} = \mathbf{a}_1 \circ \mathbf{a}_2 \circ \dots \circ \mathbf{a}_N. \quad (5.15)$$

对于外积得到的张量，它的元素有如下表达式

$$a_{i_1 i_2 \dots i_N} = a_1^{(i_1)} a_2^{(i_2)} \dots a_N^{(i_N)}, \quad (5.16)$$

其中， $a_p^{(q)}$ 表示向量 $\mathbf{a}_p$ 的第 q 个元素。

可以发现，外积是一种特殊的 k 模积运算，即 (5.15) 可以表示为

$$\mathcal{A} = \mathbf{1} \times_1 \mathbf{a}_1 \times_2 \mathbf{a}_2 \dots \times_N \mathbf{a}_N. \quad (5.17)$$

特别地，当 $\mathbf{a}_1 = \mathbf{a}_2 = \dots = \mathbf{a}_N = \mathbf{a}$ （此时， $I_1 = I_2 = \dots = I_N = I$ ）时，(5.15) 可以简化为

$$\mathcal{A} = \mathbf{a}^{\circ N}. \quad (5.18)$$

此时， $\mathcal{A}$ 为一个 N 阶 I 维张量，其元素有如下表达式

$$a_{i_1 i_2 \dots i_N} = a^{(i_1)} a^{(i_2)} \dots a^{(i_N)}. \quad (5.19)$$

显然，任意交换 $i_1, i_2, i_3, \dots, i_N$ 的顺序，都不会改变 $a_{i_1 i_2 i_3 \dots i_N}$ 的取值，比如

$$a_{i_1 i_2 i_3 \dots i_N} = a_{i_2 i_1 i_3 \dots i_N},$$

因此 (5.18) 中的 $\mathcal{A}$ 为一个对称张量。

5.3.4 统计量映射图

对于一个高维数据 $\mathbf{X} \in \mathbb{R}^{p \times n}$ ，不妨将其在任意投影方向 $\mathbf{u}$ 上的 k 阶统计量记作 $s^{(k)}(\mathbf{u}^T \mathbf{X})$ 。如果将每个单位向量 $\mathbf{u}$ 都赋予一个长度，且该长度等于统计量 $s^{(k)}(\mathbf{u}^T \mathbf{X})$ 的模（即绝对值），则在 p 维空间可以得到一个新的向量 $|s^{(k)}(\mathbf{u}^T \mathbf{X})| \mathbf{u}$ 。所有这些新的向量（或新的点）构成的几何结构我们称之为数据 $\mathbf{X}$ 的统计量映射图。显然，统计量映射图能够清晰地反映数据在各个方向的高阶统计分布情况。具体来说，统计量映射图有如下的定义。

定义 5.6 对于一个 p 维 n 个观测的数据 $\mathbf{X} \in \mathbb{R}^{p \times n}$ ，令 $s^{(k)}(\mathbf{u}^T \mathbf{X})$ 为观测向量 $\mathbf{u}^T \mathbf{X}$ 的 k 阶统计量，则如下点的集合被称作数据的 k 阶统计量映射图。

$$\left\{ \mathbf{r}^{(k)} = |s^{(k)}(\mathbf{u}^T \mathbf{X})| \mathbf{u} \mid \mathbf{u}^T \mathbf{u} = 1, \mathbf{u} \in \mathbb{R}^{p \times 1} \right\} \quad (5.20)$$

对于一个二维标准高斯分布的数据，它在任意方向的方差均为常数 $\text{var}(\mathbf{u}^T \mathbf{X}) = s^{(2)}(\mathbf{u}^T \mathbf{X}) = 1$ ，这意味着该数据的二阶统计量（方差）映射图为一个半径为 1 的圆

$$\left\{ \mathbf{r}^{(2)} = \mathbf{u} \mid \mathbf{u}^T \mathbf{u} = 1, \mathbf{u} \in \mathbb{R}^{2 \times 1} \right\} \quad (5.21)$$

统计量映射图的维度与数据的维度有关，当数据分布在二维平面上时，对应的统计量映射图是二维空间中的曲线。图 5.6 通过一个简单的示例展示了统计量映射图的

构造过程. 其中所用的数据为

$$\mathbf{X} = \begin{bmatrix} -0.1722 & 0.2090 & 0.1925 & -0.2293 \\ -0.0003 & -0.4738 & 0.2318 & 0.2423 \end{bmatrix}.$$

观察可知, $\mathbf{X}$ 的均值向量为零向量. 如果选择投影方向 $\mathbf{u} = \begin{bmatrix} 0 & 1 \end{bmatrix}^T$, 则数据在该方向的方差为

$$\text{var}(\mathbf{u}^T \mathbf{X}) = s^{(2)}(\mathbf{u}^T \mathbf{X}) = \frac{1}{4}(0.0003^2 + 0.4738^2 + 0.2318^2 + 0.2423^2) \approx 0.0842,$$

这意味着 $\mathbf{u} = \begin{bmatrix} 0 & 1 \end{bmatrix}^T$ 对应该数据的二阶统计量(方差)映射图上的点 $(0, 0.0842)$. 同样地, 根据 (5.8), 可以计算得到数据在 $\mathbf{u}$ 方向的偏度为

$$\text{skew}(\mathbf{u}^T \mathbf{X}) = s^{(3)}(\mathbf{u}^T \mathbf{X}) = \frac{\frac{1}{4}(-0.0003^3 - 0.4738^3 + 0.2318^3 + 0.2423^3)}{0.0842^{\frac{3}{2}}} \approx -0.8154,$$

这意味着 $\mathbf{u} = \begin{bmatrix} 0 & 1 \end{bmatrix}^T$ 对应该数据的三阶统计量(偏度)映射图上的点 $(0, |-0.8154|) = (0, 0.8154)$.

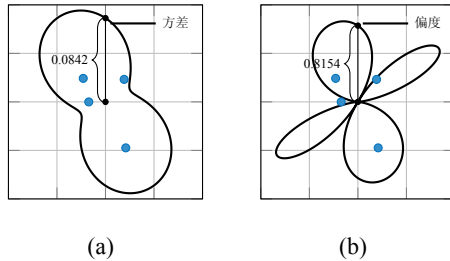


图 5.6 二维数据统计量映射图 (为了方便展示, 统计量映射图均经过了适当的缩放)
(a) 二阶统计量 (方差) (b) 三阶统计量 (偏度)

当数据分布在三维空间内时, 比如

$$\mathbf{X} = \begin{bmatrix} 2.3459 & 0.0893 & 2.2103 & 0.7440 & 0.6762 \\ -0.4959 & 1.0007 & -1.8874 & -1.2499 & -0.2327 \\ 0.1599 & -1.0078 & 0.9440 & 1.6672 & -0.9105 \end{bmatrix},$$

其所对应的统计量映射图则是三维空间中的曲面, 如图 5.7 所示. 可以发现, 从统计量映射图中, 我们能够直观地观察到数据在不同方向上投影结果的统计量大小.

主偏度分析[6782646]是主成分分析从二阶统计到三阶统计的自然拓展, 本节我们将首先介绍数据在任意方向偏度的解析表达, 然后给出主偏度分析的优化模型与基本求解算法.

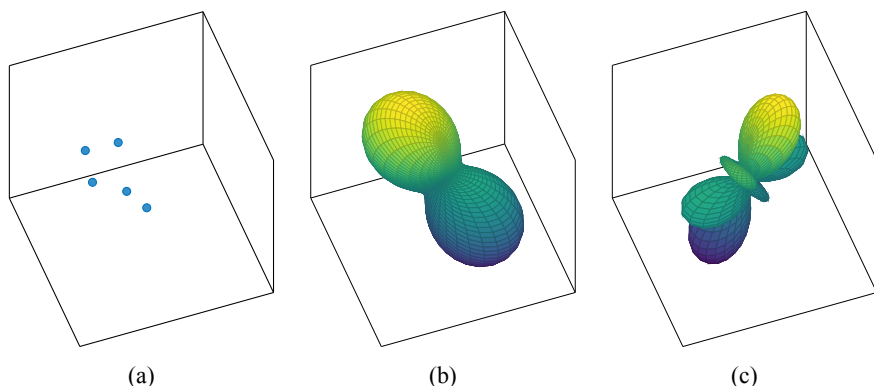


图 5.7 三维数据统计量映射图 (a) 数据 (b) 二阶统计量 (方差) (c) 三阶统计量 (偏度)

5.3.5 任意方向的偏度

在主成分分析中, 对于任意 p 个特征 n 个观测的数据 $\mathbf{X}$, 其在任意方向 $\mathbf{u}$ 的方差可以通过数据的协方差矩阵解析表达, 即 $\text{var}(\mathbf{X}) = \mathbf{u}^T \boldsymbol{\Sigma} \mathbf{u}$. 那么, 既然偏度是方差的三阶推广, 我们不禁要问, 数据在任意方向的偏度是否也有类似的简洁表达呢? 接下来, 为了方便起见, 不妨假设数据 $\mathbf{X}$ 为白化后的数据, 即其均值向量为零向量, 协方差矩阵为单位矩阵. 于是, 根据公式 (5.12), 数据 $\mathbf{X}$ 在任意方向 $\mathbf{u}$ 的偏度有

$$\begin{aligned}
 \text{skew}(\mathbf{u}^T \mathbf{X}) &= \text{skew} \left(\begin{bmatrix} \mathbf{u}^T \mathbf{x}_1 & \mathbf{u}^T \mathbf{x}_2 & \cdots & \mathbf{u}^T \mathbf{x}_n \end{bmatrix} \right) \\
 &= \frac{1}{n} \sum_{i=1}^n (\mathbf{u}^T \mathbf{x}_i)^3 = \frac{1}{n} \sum_{i=1}^n ((\mathbf{x}_i \circ \mathbf{x}_i \circ \mathbf{x}_i) \times_1 \mathbf{u}^T \times_2 \mathbf{u}^T \times_3 \mathbf{u}^T) \\
 &= \left(\left(\frac{1}{n} \sum_{i=1}^n \mathbf{x}_i \circ \mathbf{x}_i \circ \mathbf{x}_i \right) \times_1 \mathbf{u}^T \times_2 \mathbf{u}^T \times_3 \mathbf{u}^T \right).
 \end{aligned} \tag{5.22}$$

不难发现, $\frac{1}{n} \sum_{i=1}^n \mathbf{x}_i \circ \mathbf{x}_i \circ \mathbf{x}_i$ 不再是一个矩阵, 而是一个大小为 $p \times p \times p$ 的三阶对称张量 (如图 5.8 所示), 将其记作 $\mathcal{S}$, 则 (5.22) 可以简化为

$$\text{skew}(\mathbf{u}^T \mathbf{X}) = \mathcal{S} \times_1 \mathbf{u}^T \times_2 \mathbf{u}^T \times_3 \mathbf{u}^T. \tag{5.23}$$

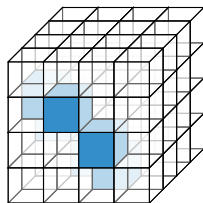


图 5.8 三阶对称张量 $\mathcal{S}$ 的示意图 ($p = 4$), 蓝色方块具有相同的数值

出于传统习惯以及表达上的简洁, 我们通常将 (5.23) 中 $\mathbf{u}^T$ 的转置符号省略, 即用下式表示数据在 $\mathbf{u}$ 方向的偏度

$$\text{skew}(\mathbf{u}^T \mathbf{X}) = \mathcal{S} \times_1 \mathbf{u} \times_2 \mathbf{u} \times_3 \mathbf{u}. \quad (5.24)$$

事实上, 数据在任意方向的方差公式??也可以类似于 (5.23) 重新表示为

$$\text{var}(\mathbf{u}^T \mathbf{X}) = \mathbf{u}^T \boldsymbol{\Sigma} \mathbf{u} = \boldsymbol{\Sigma} \times_1 \mathbf{u}^T \times_2 \mathbf{u}^T. \quad (5.25)$$

类似于 (5.24), 我们也可以将其简化为

$$\text{var}(\mathbf{u}^T \mathbf{X}) = \boldsymbol{\Sigma} \times_1 \mathbf{u} \times_2 \mathbf{u}. \quad (5.26)$$

类比主成分分析中协方差矩阵的概念, 我们将 (5.23) 或 (5.24) 中的三阶张量命名为数据的协偏度张量 (Coskewness Tensor). 与协方差矩阵包含数据所有的二阶统计信息类似, 协偏度张量包含了数据所有的三阶统计信息, 这也正是数据在任意方向的偏度能有 (5.23) 或 (5.24) 这样解析表达式的原因所在. 对于一个 N 阶张量 $\mathcal{S} \in \mathbb{R}^{I \times I \times \cdots \times I}$, 记

$$\mathcal{S} \mathbf{u}^m = \mathcal{S} \times_{N-m+1} \mathbf{u} \times_{N-m+2} \mathbf{u} \cdots \times_N \mathbf{u}, \quad (5.27)$$

其中 $m \leq N$. 容易验证, 对于协偏度张量 $\mathcal{S} \in \mathbb{R}^{p \times p \times p}$, 有

$$\mathcal{S} \mathbf{u}^3 = \mathcal{S} \times_1 \mathbf{u} \times_2 \mathbf{u} \times_3 \mathbf{u}, \quad \mathcal{S} \mathbf{u}^2 = \mathcal{S} \times_2 \mathbf{u} \times_3 \mathbf{u},$$

那么, 数据在 $\mathbf{u}$ 方向的偏度可以表示为

$$\text{skew}(\mathbf{u}^T \mathbf{X}) = \mathcal{S} \mathbf{u}^3. \quad (5.28)$$

与任意方向方差的解析表达一样, 公式 (5.23) 和 (5.28) 简洁优雅, 为后续数据偏度极值方向的求解提供了极大便利.

5.3.6 协偏度张量的计算

在??, 根据数据观测矩阵分块方式的不同, 我们给出了两种协方差矩阵的计算方式. 在本章, 我们仍按照观测矩阵的不同分块方式, 给出两种不同的协偏度张量的计算方法. 仍假设 $\mathbf{X}$ 为白化后的数据, 且其两种分块方式分别为

$$\mathbf{X} = \begin{bmatrix} \mathbf{x}_1 & \mathbf{x}_2 & \cdots & \mathbf{x}_n \end{bmatrix} = \begin{bmatrix} X_1 \\ X_2 \\ \vdots \\ X_p \end{bmatrix} \in \mathbb{R}^{p \times n}.$$

基于列向量的外积, 可以给出协偏度张量的第一种计算公式为

$$\mathcal{S} = \frac{1}{n} \sum_{i=1}^n \mathbf{x}_i \circ \mathbf{x}_i \circ \mathbf{x}_i, \quad (5.29)$$

即, 协偏度张量 $\mathcal{S}$ 为 n 个秩为 1 的张量 $\mathbf{x}_i \circ \mathbf{x}_i \circ \mathbf{x}_i$ 的平均, 如图 5.9b 所示.

基于行向量的“内积”, 可以给出协偏度张量中每一个元素的计算公式为

$$s_{ijk} = \frac{1}{n} \sum_{l=1}^n X_i(l) X_j(l) X_k(l). \quad (5.30)$$

其中 s_{ijk} 表示行向量 X_i, X_j, X_k 之间的协偏度, 如图 5.9c 所示.

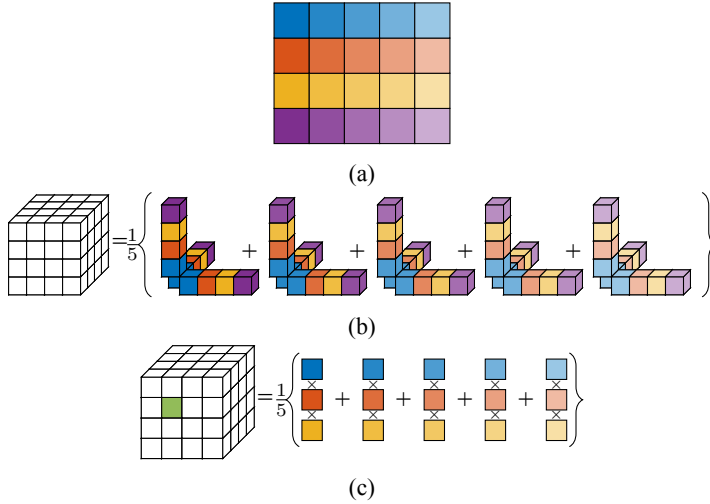


图 5.9 协偏度张量计算示意图 (a) 4×5 大小的白化数据 (b) 基于列向量外积的计算方式 (c) 基于行向量“内积”的计算方式来计算 s_{123}

尽管第一种分块对应的计算方式看上去更加简洁, 但是在实际应用中, n 往往远远大于 p , 因此在一些循环较慢的语言中 (比如 MATLAB), 这种计算方式的效率会非常低下. 而第二种分块对应的计算方式只需要进行 p^3 次计算, 且每次计算都可以利用软件内置的向量点乘以及求平均的函数来获得 s_{ijk} , 因此效率会更高. 此外, 如果能充分利用统计张量的对称性, 计算次数还可以进一步减少.

利用列克罗内克积算符 (一种特殊的 Khatri-Rao 积), 我们还可以将张量的计算写成矩阵乘法的形式. 对于两个拥有相同列数的矩阵 $\mathbf{A} = [\mathbf{a}_1 \ \mathbf{a}_2 \ \cdots \ \mathbf{a}_n]$ 和 $\mathbf{B} = [\mathbf{b}_1 \ \mathbf{b}_2 \ \cdots \ \mathbf{b}_n]$, 它们的列克罗内克积有如下表达式

$$\mathbf{A} * \mathbf{B} = [\mathbf{a}_1 \otimes \mathbf{b}_1 \ \mathbf{a}_2 \otimes \mathbf{b}_2 \ \cdots \ \mathbf{a}_n \otimes \mathbf{b}_n],$$

其中 $\otimes$ 代表克罗内克积, 其定义和相关性质请参考???. 对于数据矩阵 $\mathbf{X}$, 其自身与自身的列克罗内克积为

$$\mathbf{X} * \mathbf{X} = [\mathbf{x}_1 \otimes \mathbf{x}_1 \ \mathbf{x}_2 \otimes \mathbf{x}_2 \ \cdots \ \mathbf{x}_n \otimes \mathbf{x}_n] \in \mathbb{R}^{p^2 \times n}.$$

可以验证, $\mathbf{X} * \mathbf{X}$ 与 $\mathbf{X}$ 的互协方差矩阵

$$\mathbf{S} = \frac{1}{n}(\mathbf{X} * \mathbf{X})\mathbf{X}^T,$$

为一个大小为 $p^2 \times p$ 的矩阵, 将其转化为一个 $p \times p \times p$ 的三阶对称张量后, 其中的元素将与协偏度张量 $\mathcal{S}$ 中的元素一一对应.

事实上, $\mathbf{X} * \mathbf{X}$ 的每一列都可以看作是原数据的二次非线性项. 不妨将原数据与其二次非线性项组合起来, 得到如下的新的数据矩阵

$$\tilde{\mathbf{X}} = \begin{bmatrix} \mathbf{X} \\ \mathbf{X} * \mathbf{X} \end{bmatrix} \in \mathbb{R}^{(p^2+p) \times n}.$$

该数据矩阵的协方差矩阵可以表示为如下的分块矩阵

$$\tilde{\Sigma} = \frac{1}{n}\tilde{\mathbf{X}}\tilde{\mathbf{X}}^T = \begin{bmatrix} \Sigma & \mathbf{S}^T \\ \mathbf{S} & \mathbf{K} \end{bmatrix},$$

其中 Σ 为原数据 $\mathbf{X}$ 的协方差矩阵, 包含了数据的全部二阶统计信息. 需要注意的是, 在本小节中由于 $\mathbf{X}$ 为白化数据, 因此 Σ 为单位矩阵. 而矩阵 $\mathbf{S} = \frac{1}{n}(\mathbf{X} * \mathbf{X})\mathbf{X}^T \in \mathbb{R}^{p^2 \times p}$ 对应了数据的协偏度张量 $\mathcal{S}$, 因此其包含了数据的全部三阶统计信息. 至于矩阵 $\mathbf{K} = \frac{1}{n}(\mathbf{X} * \mathbf{X})(\mathbf{X} * \mathbf{X})^T \in \mathbb{R}^{p^2 \times p^2}$ 则包含了数据的全部四阶统计信息, 将其转换为一个 $p \times p \times p \times p$ 的四阶对称张量后, 其中的元素将与数据的四阶统计张量 $\mathcal{K} \in \mathbb{R}^{p \times p \times p \times p}$ 中的元素一一对应 (读者可自行查阅四阶统计量——峭度的相关定义, 并进行验证).

5.3.7 模型与求解

在得到数据在任意方向偏度的解析表达之后, 我们可以给出如下的主偏度分析优化模型

$$\begin{cases} \max_{\mathbf{u}} & \mathcal{S}\mathbf{u}^3 \\ \text{s.t.} & \mathbf{u}^T \mathbf{u} = 1 \end{cases}. \quad (5.31)$$

为了得到模型的最优解, 类似于主成分分析, 我们仍然采用拉格朗日乘子法. 首先构建模型的拉格朗日函数为

$$\mathcal{L}(\mathbf{u}, \lambda) = \frac{1}{3}\mathcal{S}\mathbf{u}^3 + \frac{\lambda}{2}(1 - \mathbf{u}^T \mathbf{u}). \quad (5.32)$$

(5.32) 两边同时对自变量求偏导有

$$\frac{\partial \mathcal{L}(\mathbf{u}, \lambda)}{\partial \mathbf{u}} = \mathcal{S}\mathbf{u}^2 - \lambda \mathbf{u},$$

令其等于零向量, 可得

$$\mathcal{S}\mathbf{u}^2 = \lambda \mathbf{u}. \quad (5.33)$$

这意味着, 数据的偏度极值方向必然满足 (5.33). 进一步地, 可以发现 (5.33) 与矩阵的特征值与特征向量问题非常相似, 因此我们称 (5.33) 为协偏度张量的特征值与特征向量问题. Lim 与 Qi 两位学者在 2005 年也分别从纯数学角度关注了类似的问题, 并分别独立给出了张量 $\mathbf{Z}$ -特征对的概念, 其定义如下.

定义 5.7 (对称张量的 $\mathbf{Z}$ -特征对) 给定一个 m 阶 n 维的对称张量 $\mathcal{S}$, 若如下式子满足

$$\mathcal{S}\mathbf{u}^{m-1} = \lambda\mathbf{u}, \quad (5.34)$$

则称 $(\lambda, \mathbf{u})$ 为张量 $\mathcal{S}$ 的一个 $\mathbf{Z}$ -特征对. 其中, $\lambda \in \mathbb{R}$ 为 $\mathcal{S}$ 的特征值, $\mathbf{u} \in \mathbb{R}^{n \times 1}$ 为 λ 对应的特征向量, 其满足 $\mathbf{u}^T \mathbf{u} = 1$.

显然, 当 $m = 2$ 时, 对称张量的 $\mathbf{Z}$ -特征对将会退化为矩阵特征对. 而公式 (5.33) 则可认为是公式 (5.34) 在 $m = 3$, 且 $\mathcal{S}$ 为数据的协偏度张量时的特例. 可以验证, 当向量 $\mathbf{u}$ 满足 (5.33) 时, 其相应的特征值 λ 则正好为数据在这个方向的偏度值, 即

$$\text{skew}(\mathbf{u}^T \mathbf{X}) = \mathcal{S}\mathbf{u}^3 = (\mathcal{S}\mathbf{u}^2)^T \mathbf{u} = (\lambda\mathbf{u})^T \mathbf{u} = \lambda. \quad (5.35)$$

我们知道, 矩阵的特征值与特征向量问题已经得到了极为广泛和极其详尽的研究, 相关的参考资料浩如烟海. 而遗憾的是, 张量的特征对问题的研究目前尚处于初级阶段, 可供参考的文献极为匮乏. 为此, 我们采用比较常用的优化策略——固定点迭代法来得到 (5.33) 的一个特征对. 具体步骤如下.

算法 5.3 固定点迭代法求解第一个特征对

1. 随机初始化向量 $\mathbf{u}$
 2. 令 $\mathbf{u} \leftarrow \mathcal{S}\mathbf{u}^2$
 3. 向量归一化: $\mathbf{u} \leftarrow \mathbf{u}/\|\mathbf{u}\|$
 4. 重复步骤 2 和 3, 直至收敛
-

接下来的问题是如何求解 (5.33) 的第二个特征对以及余下的所有特征对. 为此, 我们这里采取正交约束的策略. 假设已获得 l 个特征向量 $\mathbf{U}_l = [\mathbf{u}_1 \ \mathbf{u}_2 \ \cdots \ \mathbf{u}_l]$, 对于第 $l+1$ 个特征向量 $\mathbf{u}_{l+1}$, 为了防止其收敛到前 l 个特征向量, 我们可以将 $\mathbf{u}_{l+1}$ 的搜索范围限制在前 l 个特征向量张成的空间的正交补空间. 相应地, 求解第 $l+1$ 个特征向量的算法为:

算法 5.4 固定点迭代法求解第 $l+1$ 个特征对

1. 随机初始化向量 $\mathbf{u}_{l+1}$
 2. 令 $\mathbf{u}_{l+1} \leftarrow \mathcal{S}\mathbf{u}_{l+1}^2$
 3. 将 $\mathbf{u}_{l+1}$ 投影到 $\mathbf{U}_l$ 的列空间的正交补空间: $\mathbf{u}_{l+1} \leftarrow \mathbf{P}_{\mathbf{U}_l}^\perp \mathbf{u}_{l+1}$
 4. 向量归一化: $\mathbf{u}_{l+1} \leftarrow \mathbf{u}_{l+1}/\|\mathbf{u}_{l+1}\|$
 5. 重复步骤 2、3 和 4, 直至收敛
-

其中, $\mathbf{P}_{\mathbf{U}_l}^\perp = \mathbf{I}_p - \mathbf{U}_l(\mathbf{U}_l^T \mathbf{U}_l)^{-1} \mathbf{U}_l^T$ 为矩阵 $\mathbf{U}_l$ 的正交补投影算子. 显然, $\mathbf{U}_l$ 为列正交矩阵, 它的正交补投影算子可以简化为 $\mathbf{P}_{\mathbf{U}_l}^\perp = \mathbf{I}_p - \mathbf{U}_l \mathbf{U}_l^T$.

从上面的步骤可以看出, 在算法 5.4 中, 每次迭代都需要额外将向量投影到已有的特征向量的正交补空间. 事实上, 这个步骤是没必要的, 即我们可以把上述步骤的 2,3 合并为

$$\begin{aligned} \mathbf{u}_{l+1} &= \mathbf{P}_{\mathbf{U}_l}^\perp (\mathcal{S}(\mathbf{P}_{\mathbf{U}_l}^\perp \mathbf{u}_{l+1}))^2 \\ &= \mathcal{S} \times_1 \mathbf{P}_{\mathbf{U}_l}^\perp \times_2 (\mathbf{P}_{\mathbf{U}_l}^\perp \mathbf{u}_{l+1}) \times_3 (\mathbf{P}_{\mathbf{U}_l}^\perp \mathbf{u}_{l+1}) \\ &= (\mathcal{S} \times_1 \mathbf{P}_{\mathbf{U}_l}^\perp \times_2 \mathbf{P}_{\mathbf{U}_l}^\perp \times_3 \mathbf{P}_{\mathbf{U}_l}^\perp) \mathbf{u}_{l+1}^2. \end{aligned} \quad (5.36)$$

记 $\mathcal{S}_l = \mathcal{S} \times_1 \mathbf{P}_{\mathbf{U}_l}^\perp \times_2 \mathbf{P}_{\mathbf{U}_l}^\perp \times_3 \mathbf{P}_{\mathbf{U}_l}^\perp$, 则调整后的固定点迭代算法的具体步骤如下.

算法 5.5 简化后的固定点迭代法求解第 $l+1$ 个特征对

1. 计算投影后的张量 $\mathcal{S}_l = \mathcal{S} \times_1 \mathbf{P}_{\mathbf{U}_l}^\perp \times_2 \mathbf{P}_{\mathbf{U}_l}^\perp \times_3 \mathbf{P}_{\mathbf{U}_l}^\perp$
 2. 随机初始化向量 $\mathbf{u}_{l+1}$
 3. 令 $\mathbf{u}_{l+1} \leftarrow \mathcal{S}_l \mathbf{u}_{l+1}^2$
 4. 向量归一化: $\mathbf{u}_{l+1} \leftarrow \mathbf{u}_{l+1} / \|\mathbf{u}_{l+1}\|$
 5. 重复步骤 3 和 4, 直至收敛
-

有趣的是, 对于任意一个位于前 l 个特征向量所在的子空间中的向量 $\mathbf{u}$, 都有 $\mathcal{S}_l \mathbf{u}^3 = 0$. 这一定程度揭示了正交约束的工作机制: 对原本的协偏度张量进行正交补投影得到新协偏度张量使得数据在前 l 个特征向量张成子空间中任意方向上偏度均为零, 从而阻止第 $l+1$ 个特征向量收敛到前 l 个特征向量.