Part III Multivariate Techniques

第 III 部分 多变量统计分析方法

肖 磊，2026 年 4 月 28 日

概述

第 III 部分 多变量统计分析方法

第 8 章 回归模型 (Regression Models)

第 9 章 变量选择 (Variable Selection)

第 10 章 数据矩阵的因子分解 (Decomposition of Data Matrices by Factors)

第 11 章 主成分分析 (Principal Components Analysis)

第 12 章 因子分析 (Factor Analysis)

第 13 章 聚类分析 (Cluster Analysis)

第 14 章 判别分析 (Discriminant Analysis)

第 15 章 对应分析 (Correspondence Analysis()

第 16 章 典型相关分析 (Canonical Correlation Analysis)

第 17 章 多维标度法 (Multidimensional Scaling)

第 18 章 联合测定分析 (Conjoint Measurement Analysis)

第 19 章 在金融领域的应用 (Applications in Finance)

第 20 章 计算密集型方法 (Computationally Intensive Techniques)

第 21 章 局部线性嵌入 (Locally Linear Embedding)

第 22 章 随机邻域嵌入 (Stochastic Neighborhood Embedding)

第 23 章 均匀流形近似与投影 (Uniform Manifold Approximation and Projection)

Decomposition of Data Matrices by Factors

第 10 章 数据矩阵的因子分解

Outline

数据矩阵的因子分解

The Geometric Point of View (几何的角度)

Fitting the p-Dimensional Point Cloud (拟合 p 维点集)

Fitting the n-Dimensional Point Cloud (拟合 n 维点集)

Relations Between Subspaces (子空间的关系)

Practical Computation (算法)

简介

- 讨论多变量数据集的降维问题.
  - 从描述性的角度介绍如何使用几何的方法 (最小二乘准则) 来降低数据矩阵维数的最佳办法.
  - 目的是数据矩阵的低维图形展示 (可视化).
  - 涉及数据矩阵的因子分解，这些因子会按其重要性降序排列.
  - 是各种多元统计方法的核心思想.

几何观点 (The Geometric Point of View)

- 数据矩阵 $\mathcal{X}_{n\times p}$：

$$\mathcal{X}_{n\times p} = \begin{pmatrix} X_1 & X_2 & \cdots & X_p \\ \downarrow & \downarrow & \cdots & \downarrow \\ x_{11} & x_{12} & \cdots & x_{1p} \\ x_{21} & x_{22} & \cdots & x_{2p} \\ \vdots & \vdots & \ddots & \vdots \\ x_{n1} & x_{n2} & \cdots & x_{np} \end{pmatrix} = \begin{pmatrix} \boldsymbol{x}_1^{\mathrm{T}} \\ \boldsymbol{x}_2^{\mathrm{T}} \\ \vdots \\ \boldsymbol{x}_n^{\mathrm{T}} \end{pmatrix}$$

  - 每一行 (观测值) 看作一个向量

$$\boldsymbol{x}_i = \begin{pmatrix} x_{i1} \\ x_{i2} \\ \vdots \\ x_{ip} \end{pmatrix} \in \mathbb{R}^p, \quad i = 1, 2, \ldots, n$$

Decomposition of Data Matrices by Factors

几何观点 (The Geometric Point of View)

- 数据矩阵 $\mathcal{X}_{n\times p}$：

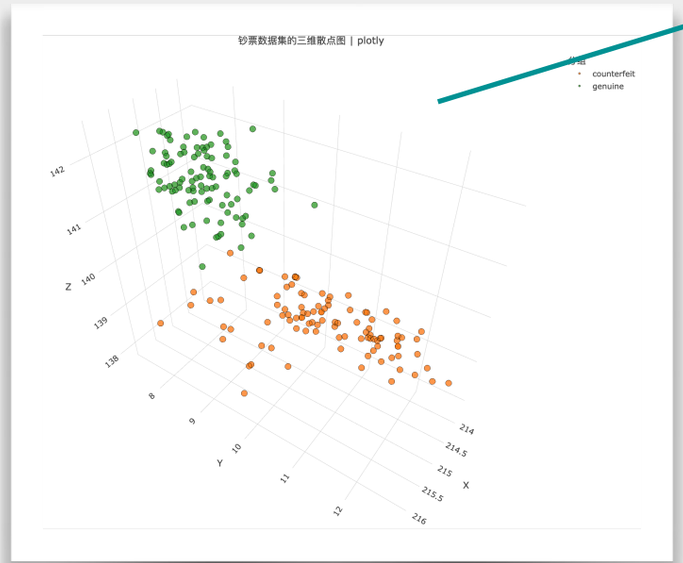
$$\mathbb{R}^p$$

$$\begin{matrix} X_1 & X_2 & \cdots & X_p \\ \downarrow & \downarrow & \cdots & \downarrow \end{matrix}$$

$$\mathcal{X}_{n\times p} = \begin{pmatrix} x_{11} & x_{12} & \cdots & x_{1p} \\ x_{21} & x_{22} & \cdots & x_{2p} \\ \vdots & \vdots & \ddots & \vdots \\ x_{n1} & x_{n2} & \cdots & x_{np} \end{pmatrix} = \begin{pmatrix} \boldsymbol{x}_1^{\mathrm{T}} \\ \boldsymbol{x}_2^{\mathrm{T}} \\ \vdots \\ \boldsymbol{x}_n^{\mathrm{T}} \end{pmatrix}$$

坐标为 $\boldsymbol{x}_i$ 的 n 个数据点

```
library(plotly)
library(mclust)
data(banknote)
str(banknote)
x <- banknote$Length
y <- banknote$Bottom
z <- banknote$Diagonal
group <- banknote$Status
df1 <- data.frame(x, y, z, group)
p1 <- plot_ly(df1,
              x = ~x,
              y = ~y,
              z = ~z,
              color = ~group,       # 分组自动配色
              colors = c("#ff7f0e","#2ca02c"),
              type = "scatter3d",   # 作三维散点图
              mode = "markers",     # 只显示点
              marker = list(
                size = 3,           # 点大小
                opacity = 0.8,      # 透明度
                line = list(width = 0.5, color = "black")
              )) %>%
  layout(
    title = list(text = "钞票数据集的三维散点图 | plotly", x =
0.5, font = list(size = 15)),
    scene = list(                   # 统一设置 3 个坐标轴名称
      xaxis = list(title = "X"),
      yaxis = list(title = "Y"),
      zaxis = list(title = "Z")
    ),
    legend = list(title = list(text = "分组"))
  ) %>%
  config(displaylogo = FALSE)
p1
```



几何观点 (The Geometric Point of View)

- 数据矩阵 $\mathcal{X}_{n\times p}$：

$$\mathcal{X}_{n\times p} = \begin{pmatrix} X_1 & X_2 & \cdots & X_p \\ \downarrow & \downarrow & \cdots & \downarrow \\ x_{11} & x_{12} & \cdots & x_{1p} \\ x_{21} & x_{22} & \cdots & x_{2p} \\ \vdots & \vdots & \ddots & \vdots \\ x_{n1} & x_{n2} & \cdots & x_{np} \end{pmatrix} = \begin{pmatrix} \boldsymbol{x}_{[1]} & \boldsymbol{x}_{[2]} & \cdots & \boldsymbol{x}_{[p]} \end{pmatrix}$$

  - 每一列 (变量) 看作一个向量

$$\boldsymbol{x}_{[j]} = \begin{pmatrix} x_{1i} \\ x_{2j} \\ \vdots \\ x_{nj} \end{pmatrix} \in \mathbb{R}^n, \quad j = 1, 2, \ \ldots, p$$

Decomposition of Data Matrices by Factors

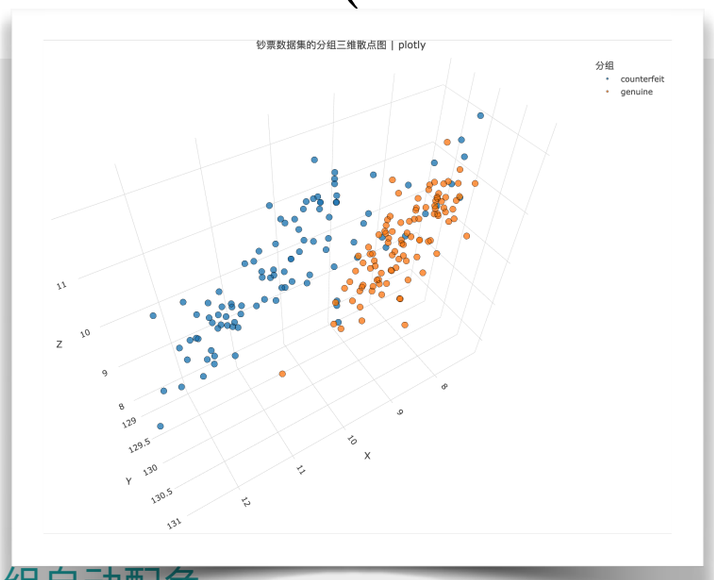
几何观点 (The Geometric Point of View)

- 数据矩阵 $\mathcal{X}_{n\times p}$：

$$\mathcal{X}_{n\times p}=\begin{array}{c} \begin{matrix} X_1 & X_2 & \cdots & X_p \end{matrix} \\ \begin{matrix} \downarrow & \downarrow & \cdots & \downarrow \end{matrix} \\ \begin{pmatrix} x_{11} & x_{12} & \cdots & x_{1p} \\ x_{21} & x_{22} & \cdots & x_{2p} \\ \vdots & \vdots & \ddots & \vdots \\ x_{n1} & x_{n2} & \cdots & x_{np} \end{pmatrix} \end{array} = \begin{pmatrix} \boldsymbol{x}_{[1]} & \boldsymbol{x}_{[2]} & \cdots & \boldsymbol{x}_{[p]} \end{pmatrix}$$

$\mathbb{R}^n$

坐标为 $\boldsymbol{x}_{[j]}$ 的 p 个变量点




```
x <- banknote$Bottom
y <- banknote$Left
z <- banknote$Top
group <- banknote$Status
df2 <- data.frame(x, y, z, group)
p2 <- plot_ly(df2,
              x = ~x,
              y = ~y,
              z = ~z,
              color = ~group,       # 分组自动配色
              colors = c("#1f77b4","#ff7f0e"),
              type = "scatter3d",   # 作三维散点图
              mode = "markers",     # 只显示点
              marker = list(
                size = 3,           # 点大小
                opacity = 0.8,      # 透明度
                line = list(width = 0.5, color = "black")
              )) %>%
  layout(
    title = list(text = "钞票数据集的分组三维散点图 | plotly",
x = 0.5, font = list(size = 15)),
    scene = list(                   # 统一设置 3 个坐标轴名称
      xaxis = list(title = "X"),
      yaxis = list(title = "Y"),
      zaxis = list(title = "Z")
    ),
    legend = list(title = list(text = "分组"))
  ) %>%
  config(displaylogo = FALSE)
p2
```

几何观点 (The Geometric Point of View)

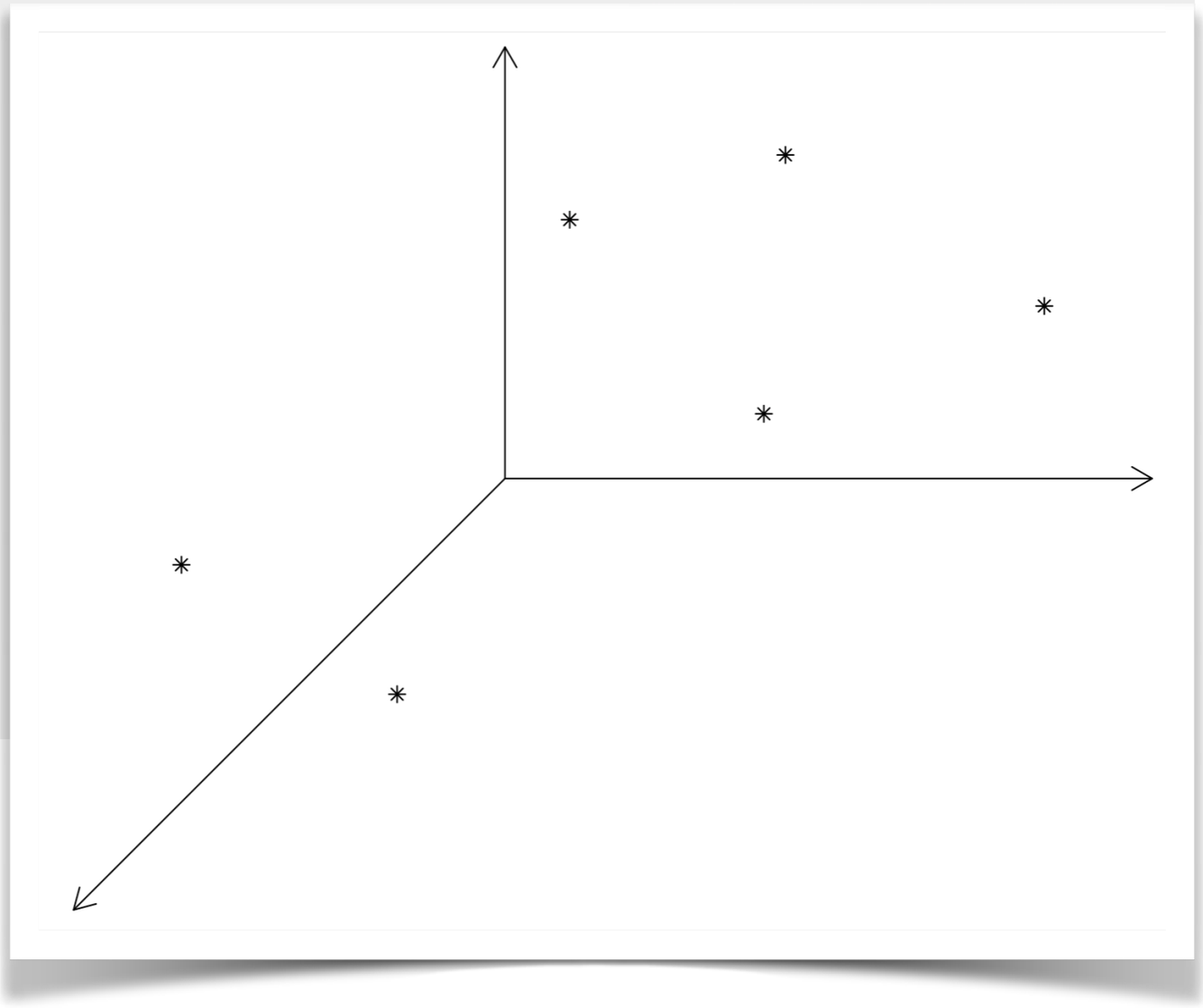
- 当 n, p 很大时，无法利用可视化对数据点或变量点进行解释.
- 我们尝试在低维子空间上同时对列空间 $C\left(\mathscr{X}\right)$ 与行空间 $C\left(\mathscr{X}^{\mathrm{T}}\right)$ 进行近似.
- 在作近似时，不能有过多的信息损失，点之间的结构不得出现显著不同.
- 有助于通过 $\mathbb{R}$, $\mathbb{R}^2$, 或 $\mathbb{R}^3$ 中的图形来深入了解 $\mathscr{X}$ 的结构.
- 核心是找到降维的因子.

拟合 p 维数据点 (Fitting the p-Dimensional Point Cloud)

- 1 维子空间
  - $\mathcal{X}$ 表示为 $\mathbb{R}^p$ 中的 n 个点.

$$\mathcal{X}_{n\times p} = \begin{pmatrix} x_{11} & x_{12} & \cdots & x_{1p} \\ x_{21} & x_{22} & \cdots & x_{2p} \\ \vdots & \vdots & \ddots & \vdots \\ x_{n1} & x_{n2} & \cdots & x_{np} \end{pmatrix} \begin{matrix} \uparrow \\ X_1 \end{matrix} \begin{matrix} \uparrow \\ X_2 \end{matrix} \begin{matrix} \cdots \\ \cdots \end{matrix} \begin{matrix} \uparrow \\ X_p \end{matrix}$$

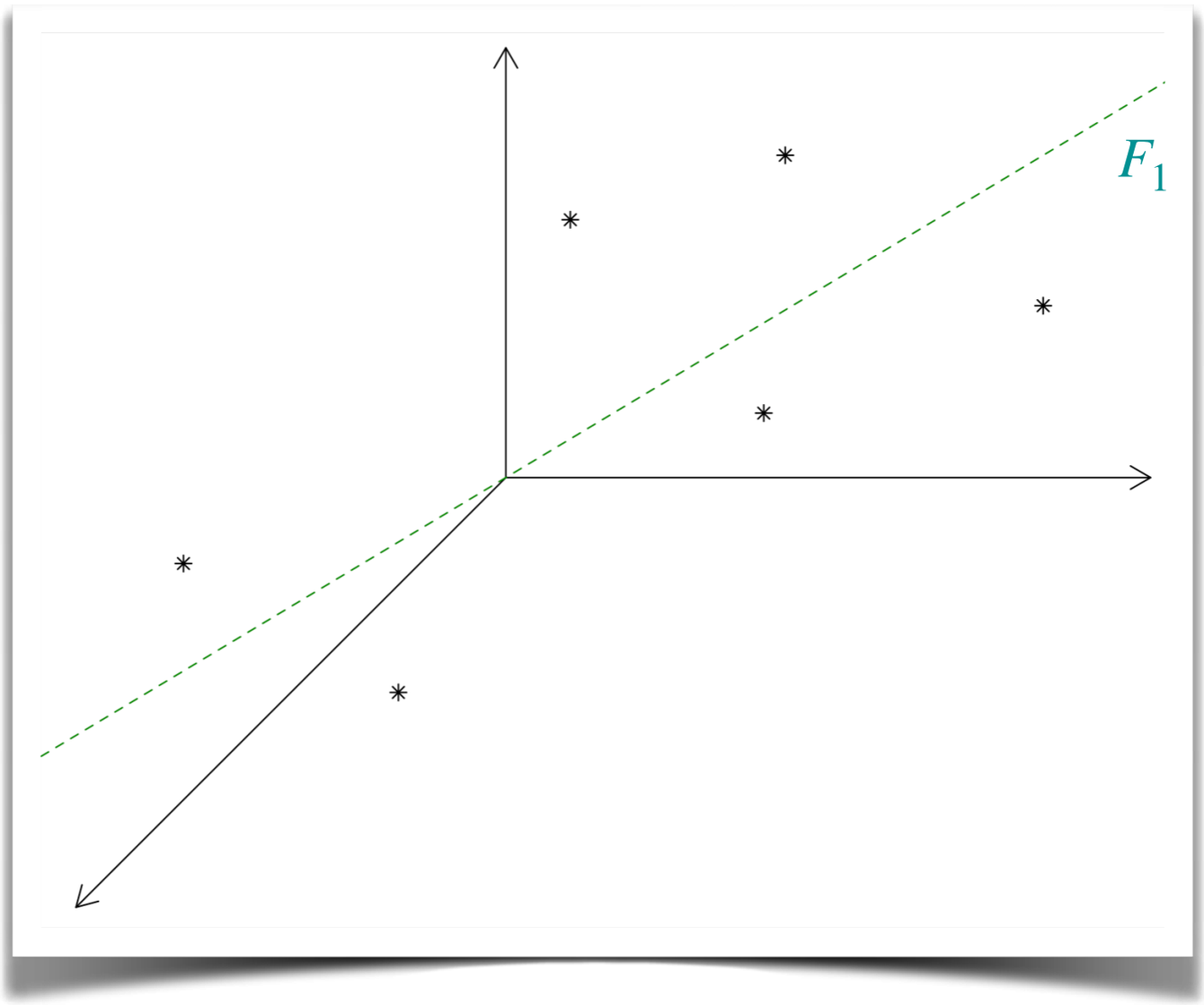
```
x <- c(-1.5, -0.5, 0.3, 1.2, 1.3, 2.5)
y <- c(-0.4, -1.0, 1.2, 0.3, 1.5, 0.8)
plot(x, y, xlim = c(-2.5, 3.0), ylim = c(-2, 2), axes = FALSE, pch = 8, cex = 1, xlab = '', ylab = '', asp = 1)
arrows(0, 0, 3.0, 0)
arrows(0, 0, 0, 2)
arrows(0, 0, -2, -2)
arrows(0, 0, 0.5, 0.3, col = 'red', lwd = 2, length = 0.15)
arrows(0, 0, 1.3, 1.5, length = 0.15)
abline(coef = c(0, 0.6), col = 'green4', lty = 2)
a <- array(0, dim = length(x))
b <- array(0, dim = length(x))
for (i in 1:length(x)) {
  a[i] <- (10 * x[i] + 6 * y[i]) / 13.6
  b[i] <- 0.6 * a[i]
  lines(c(x[i], a[i]), c(y[i], b[i]), lty=3)
}
arrows(0, 0, a[5], b[5], length = 0.15)
```



拟合 p 维数据点 (Fitting the p-Dimensional Point Cloud)

- 1 维子空间
  - $\mathscr{X}$ 表示为 $\mathbb{R}^p$ 中的 n 个点.
  - 问题：确定一条过原点的直线 F_1 .

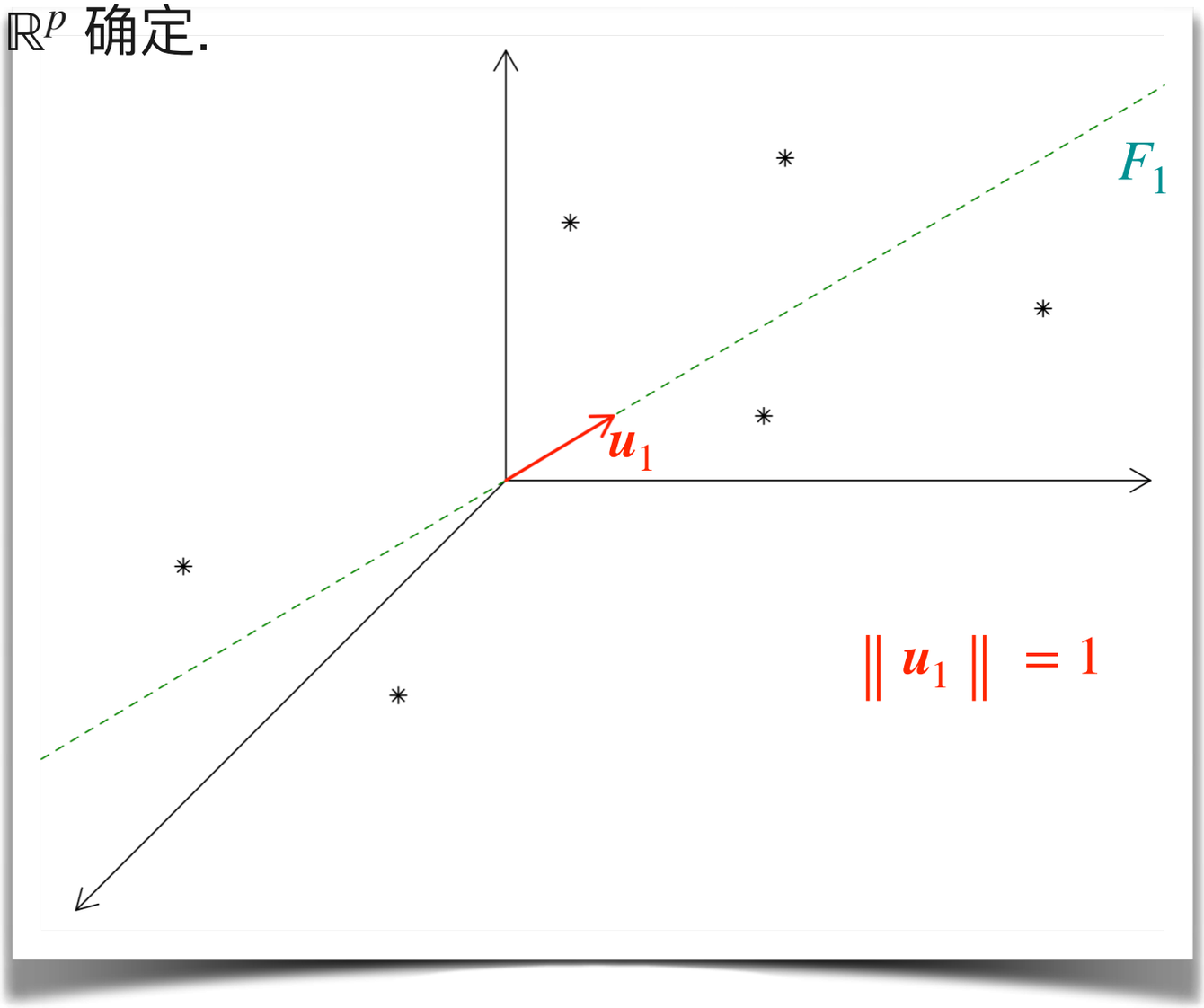
$$\mathscr{X}_{n\times p}=\begin{pmatrix} x_{11} & x_{12} & \cdots & x_{1p} \\ x_{21} & x_{22} & \cdots & x_{2p} \\ \vdots & \vdots & \ddots & \vdots \\ x_{n1} & x_{n2} & \cdots & x_{np} \\ \uparrow & \uparrow & \cdots & \uparrow \\ X_1 & X_2 & \cdots & X_p \end{pmatrix}$$



拟合 p 维数据点 (Fitting the p-Dimensional Point Cloud)

- 1 维子空间
  - $\mathcal{X}$ 表示为 $\mathbb{R}^p$ 中的 n 个点.
  - 问题：确定一条过原点的直线 F_1 .
  - 线的方向可由一个单位向量 $\boldsymbol{u}_1 \in \mathbb{R}^p$ 确定.

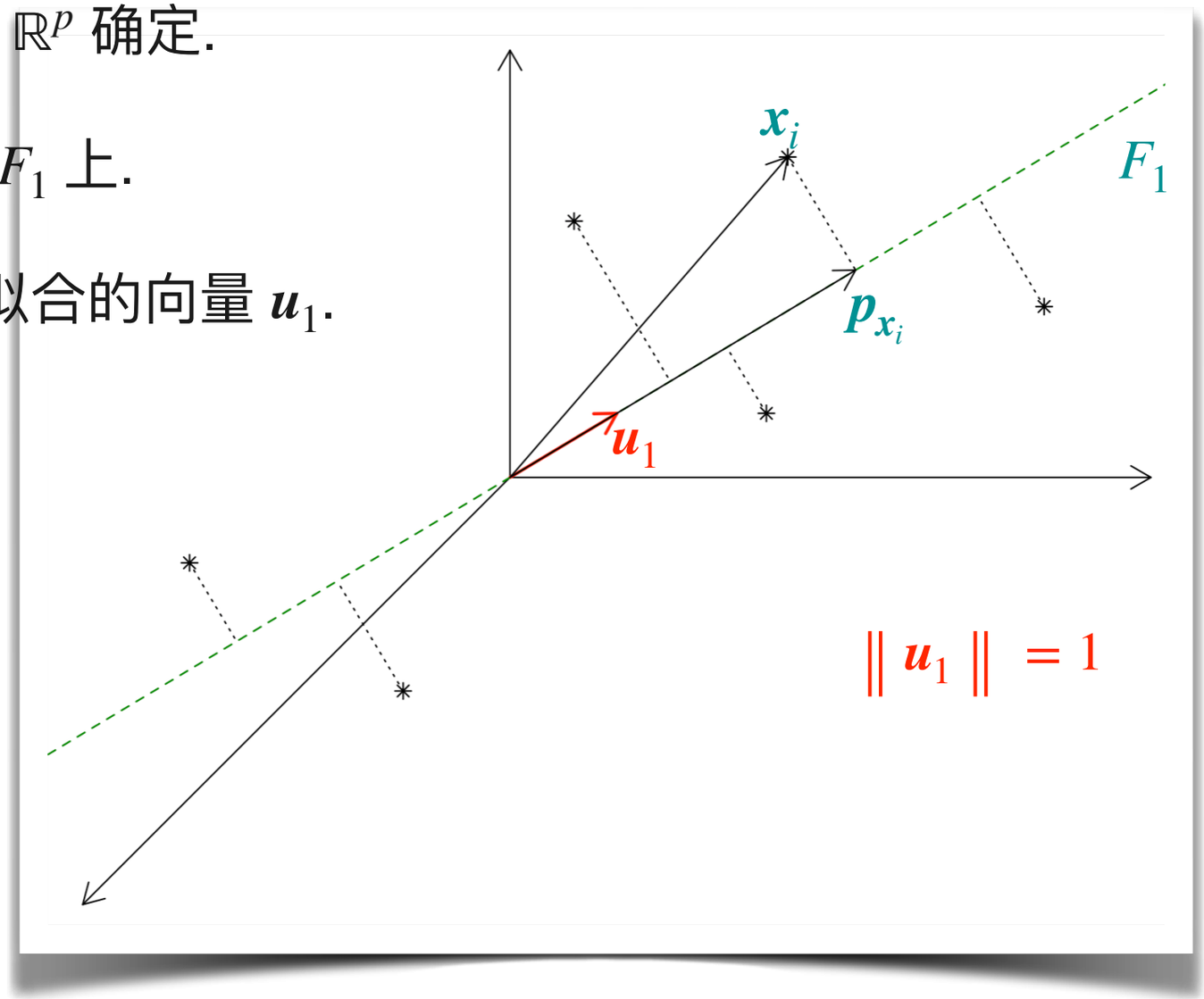
$$\mathcal{X}_{n\times p} = \begin{pmatrix} x_{11} & x_{12} & \cdots & x_{1p} \\ x_{21} & x_{22} & \cdots & x_{2p} \\ \vdots & \vdots & \ddots & \vdots \\ x_{n1} & x_{n2} & \cdots & x_{np} \\ \uparrow & \uparrow & \cdots & \uparrow \\ X_1 & X_2 & \cdots & X_p \end{pmatrix}$$



拟合 p 维数据点 (Fitting the p-Dimensional Point Cloud)

- 1 维子空间
  - $\mathcal{X}$ 表示为 $\mathbb{R}^p$ 中的 n 个点.
  - 问题：确定一条过原点的直线 F_1 .
  - 线的方向可由一个单位向量 $\boldsymbol{u}_1 \in \mathbb{R}^p$ 确定.
  - 将 $\mathbb{R}^p$ 中的所有点 $\boldsymbol{x}_i$ 投影到直线 F_1 上.
  - 我们寻找能对 n 个点进行“最佳”拟合的向量 $\boldsymbol{u}_1$.

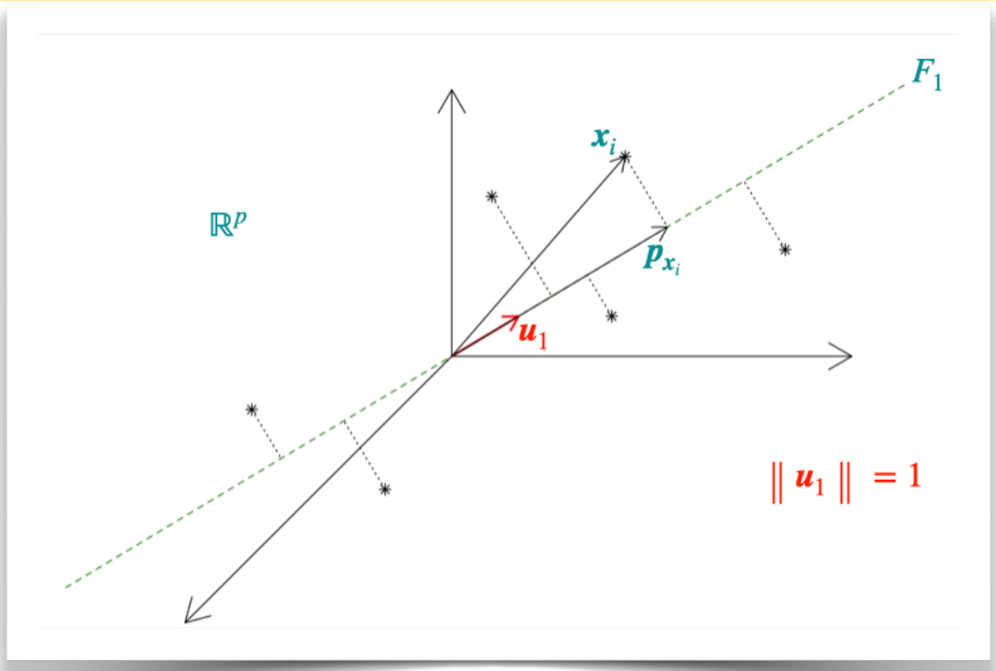
$$\mathcal{X}_{n\times p} = \begin{pmatrix} x_{11} & x_{12} & \cdots & x_{1p} \\ x_{21} & x_{22} & \cdots & x_{2p} \\ \vdots & \vdots & \ddots & \vdots \\ x_{n1} & x_{n2} & \cdots & x_{np} \\ \uparrow & \uparrow & \cdots & \uparrow \\ X_1 & X_2 & \cdots & X_p \end{pmatrix}$$



拟合 p 维数据点 (Fitting the p-Dimensional Point Cloud)

- 1 维子空间
  - 点 $\boldsymbol{x}_i \in \mathbb{R}^p$ 在直线 F_1 上表示为 $\boldsymbol{x}_i$ 在 $\boldsymbol{u}_1$ 上的投影

$$\boldsymbol{p}_{\boldsymbol{x}_i} = \boldsymbol{x}_i^{\mathrm{T}} \frac{\boldsymbol{u}_1}{\| \boldsymbol{u}_1 \|} = \boldsymbol{x}_i^{\mathrm{T}} \boldsymbol{u}_1$$



  - 定义“最小二乘”意义下的“最佳直线” F_1：使得下式达到最小的 $\boldsymbol{u}_1 \in \mathbb{R}^p$

$$\sum_{i=1}^{n} \| \boldsymbol{x}_i - \boldsymbol{p}_{\boldsymbol{x}_i} \|^2$$

  - 勾股定理:

$$\| \boldsymbol{x}_i - \boldsymbol{p}_{\boldsymbol{x}_i} \|^2 = \| \boldsymbol{x}_i \|^2 - \| \boldsymbol{p}_{\boldsymbol{x}_i} \|^2$$

$$\Longrightarrow \sum_{i=1}^{n} \| \boldsymbol{x}_i - \boldsymbol{p}_{\boldsymbol{x}_i} \|^2 = \sum_{i=1}^{n} \| \boldsymbol{x}_i \|^2 - \sum_{i=1}^{n} \| \boldsymbol{p}_{\boldsymbol{x}_i} \|^2$$

最小　　最大

拟合 p 维数据点 (Fitting the p-Dimensional Point Cloud)

- 1 维子空间
  - 问题：在约束条件 $\| \boldsymbol{u}_1 \| = 1$ 下，求使得 $\sum_{i=1}^{n} \| \boldsymbol{p}_{\boldsymbol{x}_i} \|^2$ 最大的 $\boldsymbol{u}_1 \in \mathbb{R}^p$.

$$\begin{pmatrix} p_{\boldsymbol{x}_1} \\ p_{\boldsymbol{x}_2} \\ \vdots \\ p_{\boldsymbol{x}_n} \end{pmatrix} = \begin{pmatrix} \boldsymbol{x}_1^{\mathrm{T}}\boldsymbol{u}_1 \\ \boldsymbol{x}_2^{\mathrm{T}}\boldsymbol{u}_1 \\ \vdots \\ \boldsymbol{x}_n^{\mathrm{T}}\boldsymbol{u}_1 \end{pmatrix} = \begin{pmatrix} \boldsymbol{x}_1^{\mathrm{T}} \\ \boldsymbol{x}_2^{\mathrm{T}} \\ \vdots \\ \boldsymbol{x}_n^{\mathrm{T}} \end{pmatrix} \boldsymbol{u}_1 = \mathcal{X}\boldsymbol{u}_1$$

  - 问题最终可表示为：$\max_{\boldsymbol{u}_1^{\mathrm{T}}\boldsymbol{u}_1=1} \boldsymbol{u}_1^{\mathrm{T}} \left(\mathcal{X}^{\mathrm{T}}\mathcal{X} \right) \boldsymbol{u}_1$.

Theorem 2.5 *If $\mathcal{A}$ and $\mathcal{B}$ are symmetric and $\mathcal{B} > 0$, then the maximum of $\frac{\boldsymbol{x}^{\mathrm{T}}\mathcal{A}\,\boldsymbol{x}}{\boldsymbol{x}^{\mathrm{T}}\mathcal{B}\,\boldsymbol{x}}$ is given by the largest eigenvalues of $\mathcal{B}^{-1}\mathcal{A}$. More generally,*

$$\max_{x} \frac{\boldsymbol{x}^{\mathrm{T}}\mathcal{A}\,\boldsymbol{x}}{\boldsymbol{x}^{\mathrm{T}}\mathcal{B}\,\boldsymbol{x}} = \lambda_1 \geq \lambda_2 \geq \cdots \geq \lambda_p = \min_{x} \frac{\boldsymbol{x}^{\mathrm{T}}\mathcal{A}\,\boldsymbol{x}}{\boldsymbol{x}^{\mathrm{T}}\mathcal{B}\,\boldsymbol{x}}$$

where $\lambda_1, \lambda_2, \ldots, \lambda_p$ denote the eigenvalues of $\mathcal{B}^{-1}\mathcal{A}$. The vector which maximizes (minimizes) $\frac{\boldsymbol{x}^{\mathrm{T}}\mathcal{A}\,\boldsymbol{x}}{\boldsymbol{x}^{\mathrm{T}}\mathcal{B}\,\boldsymbol{x}}$ is the eigenvector of $\mathcal{B}^{-1}\mathcal{A}$ which corresponds to the largest (smallest) eigenvalue of $\mathcal{B}^{-1}\mathcal{A}$. If $\boldsymbol{x}^{\mathrm{T}}\mathcal{B}\,\boldsymbol{x} = 1$, we get

$$\max_{x} \boldsymbol{x}^{\mathrm{T}}\mathcal{A}\,\boldsymbol{x} = \lambda_1 \geq \lambda_2 \geq \cdots \geq \lambda_p = \min_{x} \boldsymbol{x}^{\mathrm{T}}\mathcal{A}\,\boldsymbol{x}$$

$\mathcal{A}$

$\mathcal{B} = \mathcal{I}_p$

$\boldsymbol{x} = \boldsymbol{u}_1$

拟合 p 维数据点 (Fitting the p-Dimensional Point Cloud)

- 1 维子空间

定理 10.1 使得 $\sum_{i=1}^{n} \| \boldsymbol{x}_i - \boldsymbol{p}_{\boldsymbol{x}_i} \|^2$ 最小的向量 $\boldsymbol{u}_1$ 是 $\mathcal{X}^{\mathrm{T}}\mathcal{X}$ 的最大特征值对应的特征向量.

- 若数据已中心化，即 $\bar{\boldsymbol{x}} = \boldsymbol{0}$，则 $\mathcal{X} = \mathcal{X}_c$，其中 $\mathcal{X}_c$ 表示中心化的数据矩阵，则此时的 $\frac{1}{n}\mathcal{X}^{\mathrm{T}}\mathcal{X}$ 就是协方差矩阵.

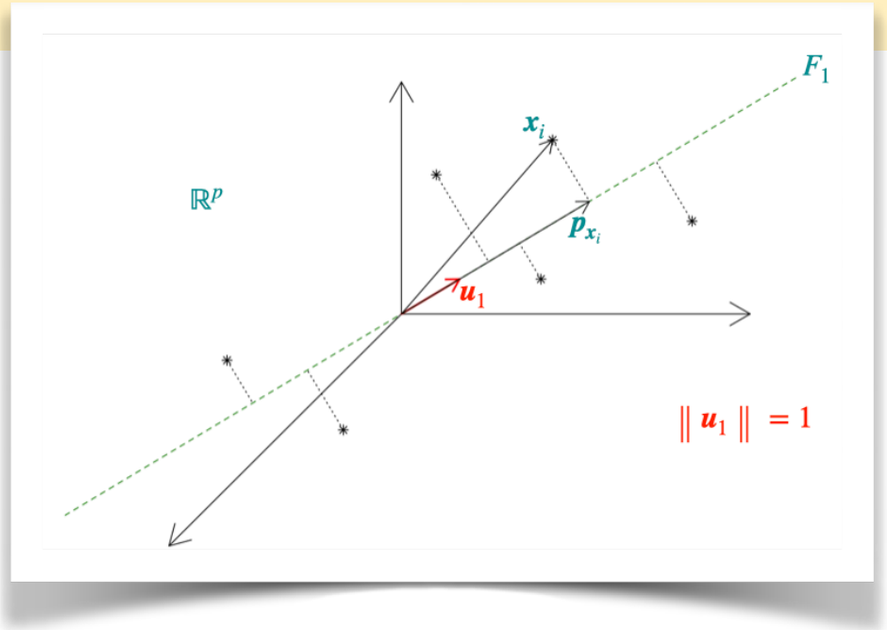
- 定理 10.1 表明，我们要确定的是二次型 $\max\limits_{\boldsymbol{u}_1^{\mathrm{T}}\boldsymbol{u}_1=1} \boldsymbol{u}_1^{\mathrm{T}} \left(\mathcal{X}^{\mathrm{T}}\mathcal{X}\right) \boldsymbol{u}_1$ 关于协方差矩阵 $\mathcal{S}_{\mathcal{X}} = \frac{1}{n}\mathcal{X}^{\mathrm{T}}\mathcal{X}$ 的最大值.

拟合 p 维数据点 (Fitting the p-Dimensional Point Cloud)

- 数据点在 F_1 上的表示
  - n 个观测点在 F_1 上的坐标为

$\mathcal{X}\boldsymbol{u}_1$

首个因子变量或首因子　　$\boldsymbol{u}_1$ 是首个因子轴



  - n 个观测点 $\boldsymbol{x}_i$，现在由一个新的因子变量 $z_1 = \mathcal{X}\boldsymbol{u}_1$ 来表示.

$$\boldsymbol{u}_1 = \begin{pmatrix} u_{11} \\ u_{21} \\ \vdots \\ u_{p1} \end{pmatrix} \qquad \mathcal{X}_{n\times p} = \begin{pmatrix} x_{11} & x_{12} & \cdots & x_{1p} \\ x_{21} & x_{22} & \cdots & x_{2p} \\ \vdots & \vdots & \ddots & \vdots \\ x_{n1} & x_{n2} & \cdots & x_{np} \end{pmatrix} \begin{matrix} \uparrow & \uparrow & \cdots & \uparrow \\ X_{[1]} & X_{[2]} & \cdots & X_{[p]} \end{matrix}$$

  - 首个因子变量是初始变量的一个线性组合

$$Z_1 = u_{11}X_{[1]} + u_{21}X_{[2]} + \cdots + u_{p1}X_{[p]}$$

拟合 p 维数据点 (Fitting the p-Dimensional Point Cloud)

- 2 维子空间
  - 如果用一个平面 (2 维) 去近似 n 个观测点，利用定理 2.5 可以证明该空间一定含有 $\boldsymbol{u}_1$.
  - 该平面由最佳现性拟合 $\left(\boldsymbol{u}_1\right)$ 以及与 $\boldsymbol{u}_1$ 正交的一个单位向量 $\boldsymbol{u}_2$ 所确定，并且 $\boldsymbol{u}_2$ 应使得二次型 $\boldsymbol{u}_2^{\mathrm{T}}\left(\mathscr{X}^{\mathrm{T}}\mathscr{X}\right)\boldsymbol{u}_2$ 在约束条件

$$\| \boldsymbol{u}_2 \| = 1, \qquad \boldsymbol{u}_1^{\mathrm{T}}\boldsymbol{u}_2 = 0$$

下达到最大.

定理 10.2 第二个因子轴 $\boldsymbol{u}_2$ 是 $\mathscr{X}^{\mathrm{T}}\mathscr{X}$ 的第二大特征值 λ_2 对应的单位特征向量.

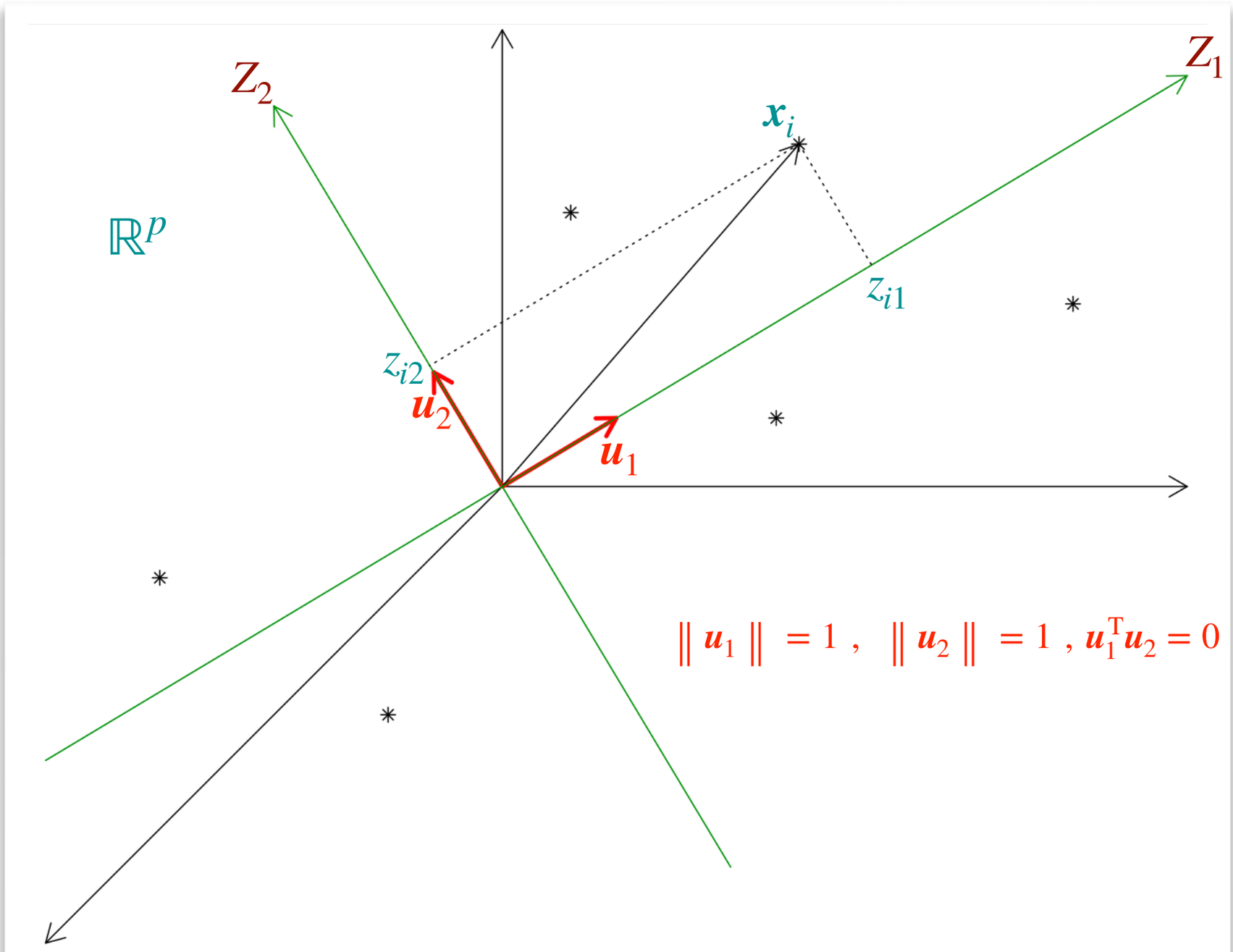
拟合 p 维数据点 (Fitting the p-Dimensional Point Cloud)

- 2 维子空间

```
x <- c(-1.5, -0.5, 0.3, 1.2, 1.3, 2.5)
y <- c(-0.4, -1.0, 1.2, 0.3, 1.5, 0.8)
plot(x, y, xlim = c(-2.5, 3.0), ylim = c(-2, 2), axes = FALSE, pch = 8, cex = 1, xlab = '', ylab = '', asp = 1)
arrows(0, 0, 3.0, 0, length = 0.15)
arrows(0, 0, 0, 2, length = 0.15)
arrows(0, 0, -2, -2, length = 0.15)
arrows(0, 0, 0.5, 0.3, col = 'red', lwd = 3, length = 0.15)
arrows(0, 0, -0.3, 0.5, col = 'red', lwd = 3, length = 0.15)
arrows(0, 0, 1.3, 1.5, length = 0.15)
arrows(-2, -1.2, 3, 1.8, col = 'green4', length = 0.15)
arrows(1, -10/6, -1, 10/6, col = 'green4', length = 0.15)
# abline(coef = c(0, -10/6), col = 'green4', lty = 2)
a <- array(0, dim = length(x)); b <- array(0, dim = length(x))
c <- array(0, dim = length(x)); d <- array(0, dim = length(x))
for (i in 1:length(x)) {
  a[i] <- (10 * x[i] + 6 * y[i]) / 13.6; b[i] <- 0.6 * a[i]
  c[i] <- ((6/10) * x[i] - y[i]) / (10/6 + 6/10); d[i] <- -(10/6) * c[i]
}
lines(c(x[5], a[5]), c(y[5], b[5]), lty=3)
lines(c(x[5], c[5]), c(y[5], d[5]), lty=3)
```

拟合 p 维数据点 (Fitting the p-Dimensional Point Cloud)

- 2 维子空间



拟合 p 维数据点 (Fitting the p-Dimensional Point Cloud)

- 2 维子空间
  - 单位向量 $\boldsymbol{u}_2$ 确定了数据点要投影到的第二条直线 F_2.
  - n 个数据点在 F_2 上的坐标为 $\boldsymbol{z}_2 = \mathcal{X}\boldsymbol{u}_2$.
  - 称变量 Z_2 为第二因子变量 (*second factorial variable*) 或者第二因子 (*second factor*).

拟合 p 维数据点 (Fitting the p-Dimensional Point Cloud)

- q $(q \le p)$ 维子空间
  - 最佳子空间由 $\mathcal{X}^{\mathrm{T}}\mathcal{X}$ 的特征值 $\lambda_1 \ge \lambda_2 \ge \cdots \ge \lambda_q$ 对应的特征向量 $\boldsymbol{u}_1, \boldsymbol{u}_2, \ldots, \boldsymbol{u}_q$ 生成.
  - n 个数据点在第 k 个因子轴 $\boldsymbol{u}_k$ 上的坐标由第 k 个因子变量 $\boldsymbol{z}_k = \mathcal{X}\boldsymbol{u}_k$ 给出 $(k = 1, 2, \ldots, q)$.
  - 每一个因子变量 $\boldsymbol{z}_k = (z_{1k}, z_{2k}, \ldots, z_{nk})^{\mathrm{T}}$ 是初始变量 $X_{[1]}, X_{[2]}, \ldots, X_{[p]}$ 的一个线性组合，线性组合的系数由第 k 个向量 $\boldsymbol{u}_k$ 的元素给出：

$$z_{ik} = \sum_{m=1}^{p} x_{im} u_{mk} .$$

拟合 n 维数据点 (Fitting the n-Dimensional Point Cloud)

- 1 维子空间
  - 将 $\mathcal{X}$ 看作是由 $\mathbb{R}^n$ 当中的 p 个点构成.
  - 问题：确定一条过原点的直线 G_1.
  - 直线的方向可由 $\mathbb{R}^n$ 当中的一个单位向量 $\boldsymbol{v}_1 \in \mathbb{R}^n$ 确定.
  - 我们寻找向量 $\boldsymbol{v}_1$ 以使它能对最初的 p 个点集给出 “最佳” 拟合.

$$\mathcal{X}_{n\times p} = \begin{pmatrix} x_{11} & x_{12} & \cdots & x_{1p} \\ x_{21} & x_{22} & \cdots & x_{2p} \\ \vdots & \vdots & \ddots & \vdots \\ x_{n1} & x_{n2} & \cdots & x_{np} \end{pmatrix} = \begin{pmatrix} \uparrow & \uparrow & \cdots & \uparrow \\ \boldsymbol{x}_{[1]} & \boldsymbol{x}_{[2]} & \cdots & \boldsymbol{x}_{[p]} \end{pmatrix}$$

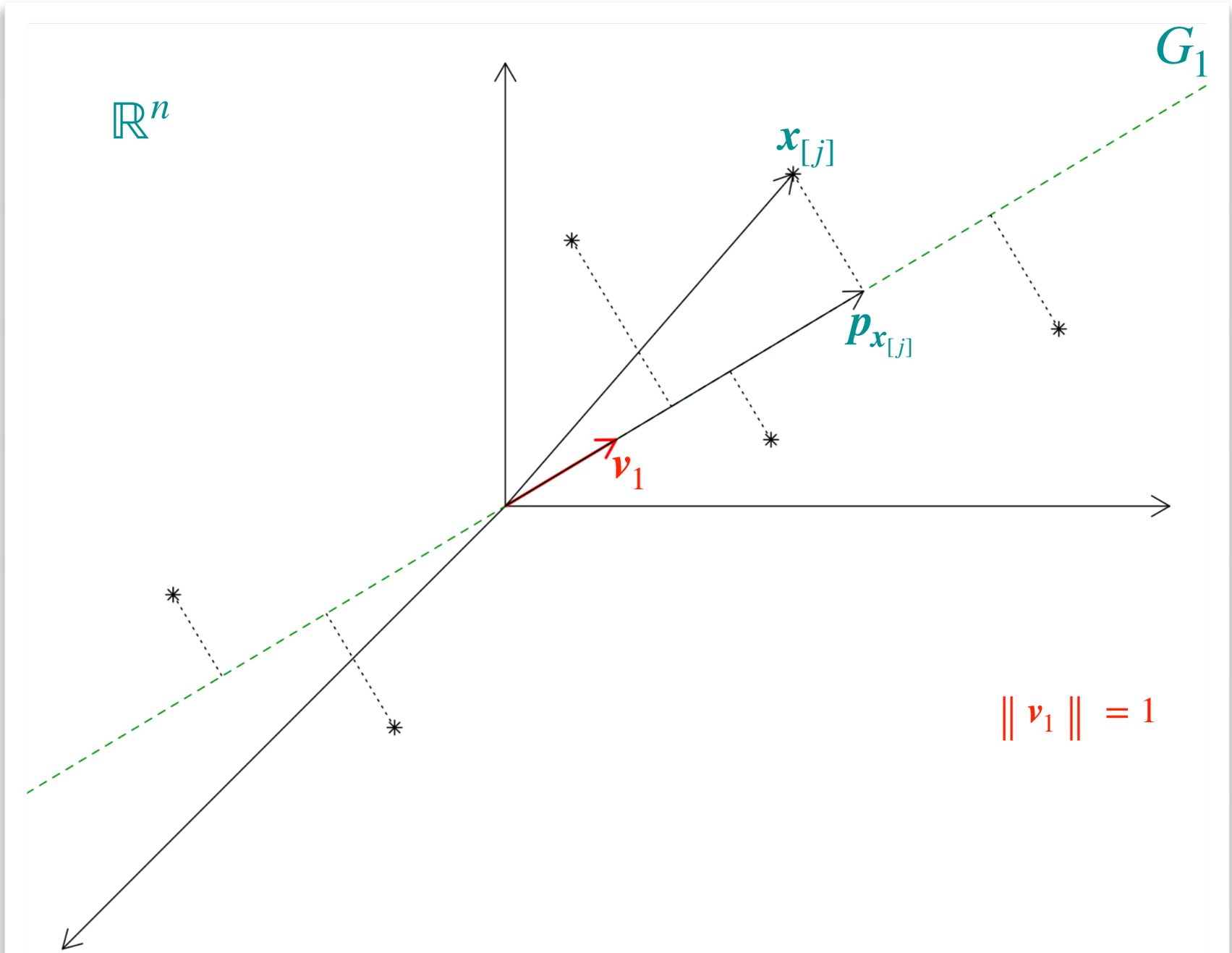
拟合 n 维数据点 (Fitting the n-Dimensional Point Cloud)

- 1 维子空间

```
x <- c(-1.5, -0.5, 0.3, 1.2, 1.3, 2.5)
y <- c(-0.4, -1.0, 1.2, 0.3, 1.5, 0.8)
plot(x, y, xlim = c(-2.5, 3.0), ylim = c(-2, 2), axes = FALSE, pch = 8, cex = 1,
    xlab = '', ylab = '', asp = 1)
arrows(0, 0, 3.0, 0, length = 0.15)
arrows(0, 0, 0, 2, length = 0.15)
arrows(0, 0, -2, -2, length = 0.15)
arrows(0, 0, 0.5, 0.3, col = 'red', lwd = 2, length = 0.15)
arrows(0, 0, 1.3, 1.5, length = 0.15)
abline(coef = c(0, 0.6), col = 'green4', lty = 2)
a <- array(0, dim = length(x))
b <- array(0, dim = length(x))
for (i in 1:length(x)) {
  a[i] <- (10 * x[i] + 6 * y[i]) / 13.6
  b[i] <- 0.6 * a[i]
  lines(c(x[i], a[i]), c(y[i], b[i]), lty=3)
}
arrows(0, 0, a[5], b[5], length = 0.15)
```

拟合 n 维数据点 (Fitting the n-Dimensional Point Cloud)

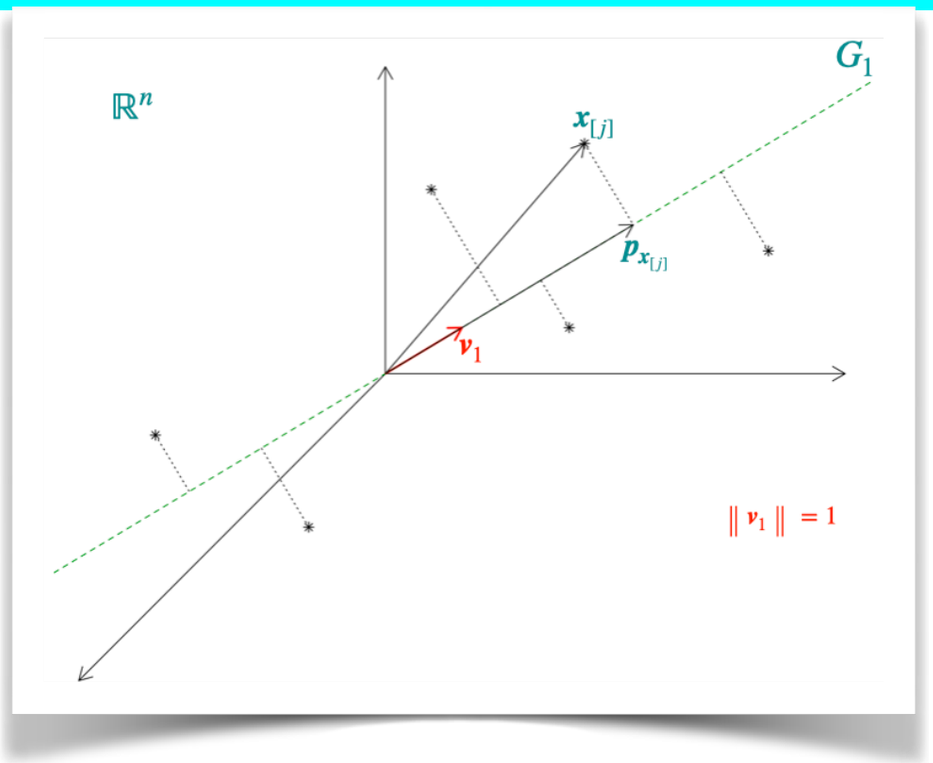
- 1 维子空间



拟合 n 维数据点 (Fitting the n-Dimensional Point Cloud)

- 1 维子空间

  - $\boldsymbol{x}_{[j]} \in \mathbb{R}^n$ 在直线 G_1 上的表示即 $\boldsymbol{x}_{[j]}$ 在 $\boldsymbol{v}_1$ 上的投影



$$\boldsymbol{p}_{\boldsymbol{x}_{[j]}} = \boldsymbol{x}_{[j]}^{\mathrm{T}} \frac{\boldsymbol{v}_1}{\| \boldsymbol{v}_1 \|} = \boldsymbol{x}_{[j]}^{\mathrm{T}} \boldsymbol{v}_1$$

  - 与 p 维情形类似，我们求使得

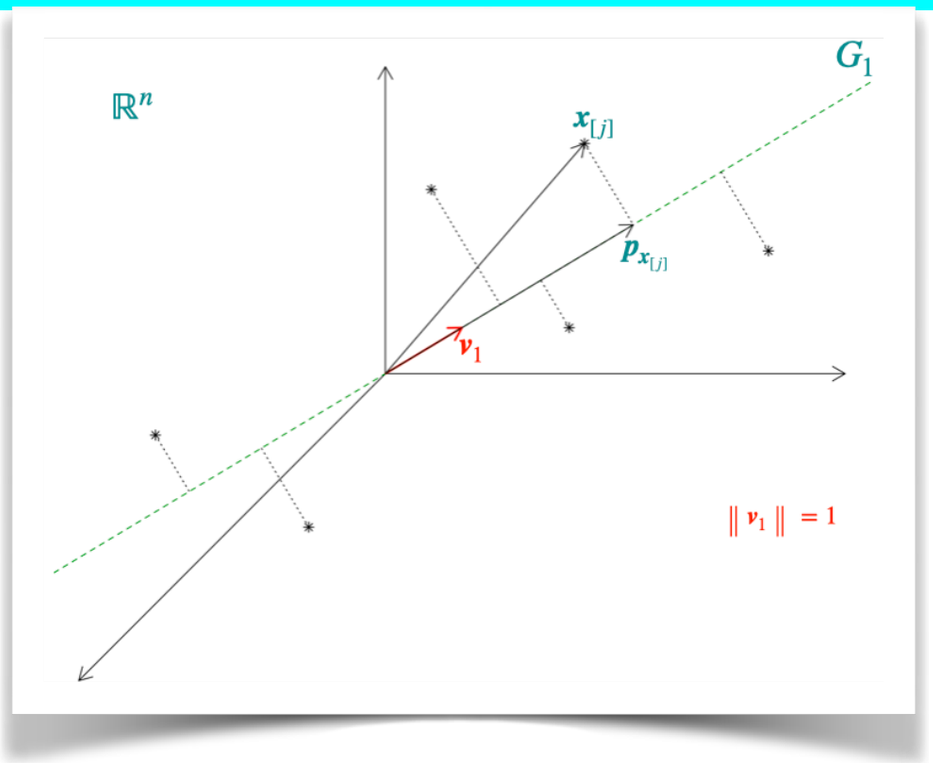
$$\sum_{j=1}^{p} \| \boldsymbol{x}_{[j]} - \boldsymbol{p}_{\boldsymbol{x}_{[j]}} \|^2 = \sum_{j=1}^{p} \| \boldsymbol{x}_{[j]} \|^2 - \sum_{j=1}^{p} \| \boldsymbol{p}_{\boldsymbol{x}_{[j]}} \|^2$$

    最小的一个单位向量 $\boldsymbol{v}_1$.

  - 等价于求使得 $\sum_{j=1}^{p} \| \boldsymbol{p}_{\boldsymbol{x}_{[j]}} \|^2$ 最大的一个单位向量 $\boldsymbol{v}_1$.

拟合 n 维数据点 (Fitting the n-Dimensional Point Cloud)

- 1 维子空间
  - $\boldsymbol{x}_{[j]} \in \mathbb{R}^n$ 在直线 G_1 上的表示即 $\boldsymbol{x}_{[j]}$ 在 $\boldsymbol{v}_1$ 上的投影



$$\boldsymbol{p}_{\boldsymbol{x}_{[j]}} = \boldsymbol{x}_{[j]}^{\mathrm{T}} \frac{\boldsymbol{v}_1}{\| \boldsymbol{v}_1 \|} = \boldsymbol{x}_{[j]}^{\mathrm{T}} \boldsymbol{v}_1$$

  - 等价地，我们求使得

$$\left(\mathcal{X}^{\mathrm{T}} \boldsymbol{v}_1\right)^{\mathrm{T}} \left(\mathcal{X}^{\mathrm{T}} \boldsymbol{v}_1\right) = \boldsymbol{v}_1^{\mathrm{T}} \left(\mathcal{X} \mathcal{X}^{\mathrm{T}}\right) \boldsymbol{v}_1$$

    最大的一个单位向量 $\boldsymbol{v}_1$.

replace $\mathcal{X}$ by $\mathcal{X}^{\mathrm{T}}$

  - 等价于求使得 $\sum_{j=1}^{p} \| \boldsymbol{p}_{\boldsymbol{x}_{[j]}} \|^2$ 最大的一个单位向量 $\boldsymbol{v}_1$.

拟合 n 维数据点 (Fitting the n-Dimensional Point Cloud)

- 1 维子空间

定理 10.3 使得 $\boldsymbol{v}_1^{\mathrm{T}}\left(\mathscr{X}\mathscr{X}^{\mathrm{T}}\right)\boldsymbol{v}_1$ 达到最大的向量 $\boldsymbol{v}_1$ 是 $\mathscr{X}\mathscr{X}^{\mathrm{T}}$ 的最大特征值 μ_1 对应的特征向量.

- 数据点在 G_1 上的表示
  - 这 p 个变量在 G_1 (第一因子轴) 上的坐标为 $\boldsymbol{w}_1 = \mathscr{X}^{\mathrm{T}}\boldsymbol{v}_1$.
  - p 个变量由初始观测值 $\boldsymbol{x}_1, \boldsymbol{x}_2, \ldots, \boldsymbol{x}_n$ 的一个线性组合来表示，其系数由向量 $\boldsymbol{v}_1$ 确定

$$w_{1j} = v_{11}x_{1j} + v_{12}x_{2j} + \cdots + v_{1n}x_{nj}$$

拟合 n 维数据点 (Fitting the n-Dimensional Point Cloud)

- q $(q \le n)$ 维子空间
  - 最佳子空间由 $\mathcal{X}\mathcal{X}^{\mathrm{T}}$ 的特征值 $\mu_1 \ge \mu_2 \ge \cdots \ge \mu_q$ 对应的正交特征向量 $\boldsymbol{v}_1, \boldsymbol{v}_2, \ldots, \boldsymbol{v}_q$ 生成.
  - p 个变量在第 k $(k = 1, 2, \ldots, q)$ 个因子轴上的坐标为因子变量 $\boldsymbol{w}_k = \mathcal{X}^{\mathrm{T}}\boldsymbol{v}_k$.
  - 每个因子变量 $\boldsymbol{w}_k = \left(w_{k1}, w_{k2}, \ldots, w_{kp}\right)^{\mathrm{T}}$ 是初始观测点 $\boldsymbol{x}_1, \boldsymbol{x}_2, \ldots, \boldsymbol{x}_n$ 的一个线性组合，线性组合的系数是第 k 个向量 $\boldsymbol{v}_k$ 的元素：

$$w_{kj} = \sum_{m=1}^{n} v_{km}\, x_{mj}$$

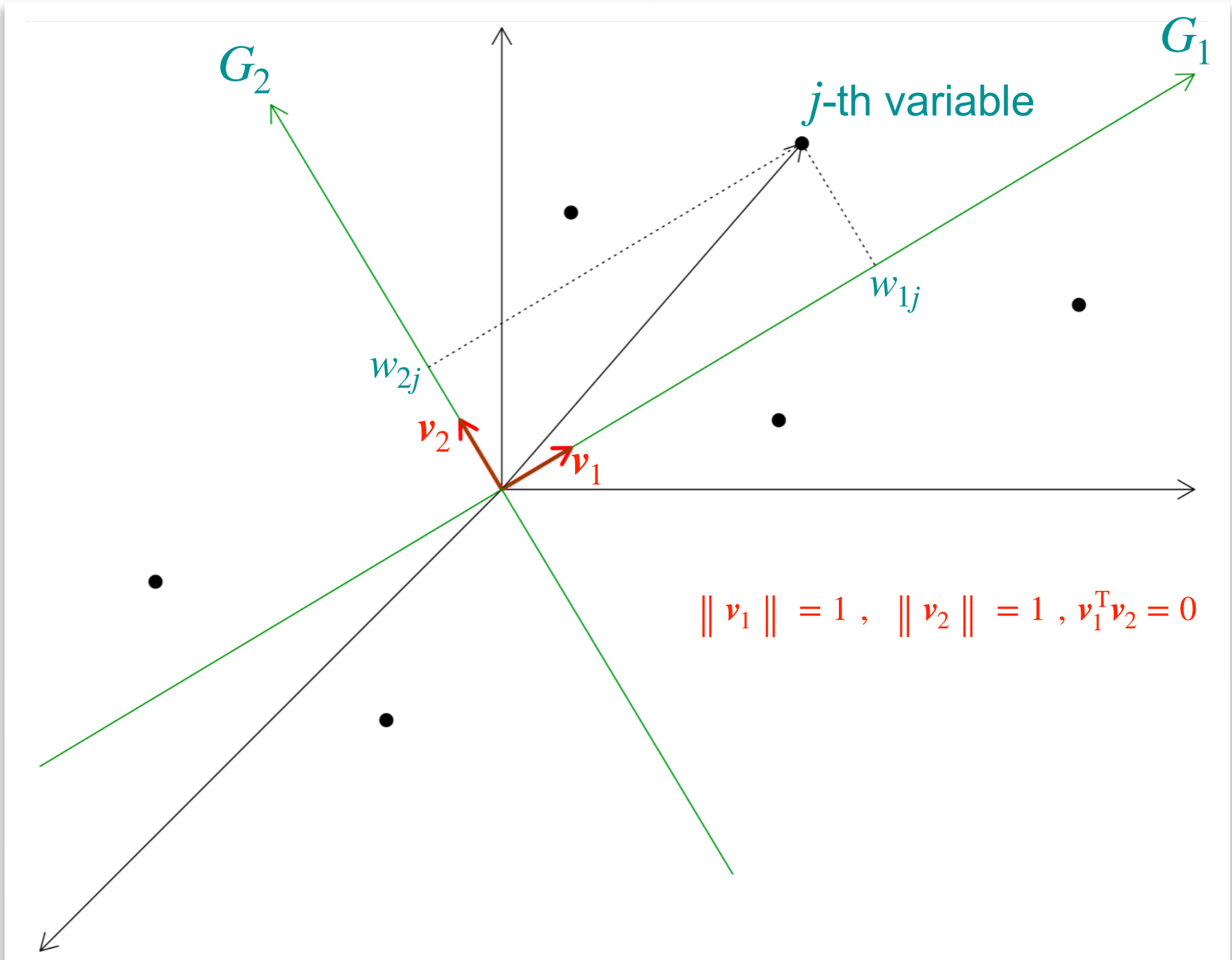
拟合 n 维数据点 (Fitting the n-Dimensional Point Cloud)

- q $(q \le n)$ 维子空间

```
x <- c(-1.5, -0.5, 0.3, 1.2, 1.3, 2.5)
y <- c(-0.4, -1.0, 1.2, 0.3, 1.5, 0.8)
plot(x, y, xlim = c(-2.5, 3.0), ylim = c(-2, 2), axes = FALSE, pch = 16, cex = 1.5, xlab = '', ylab = '', asp = 1)
arrows(0, 0, 3.0, 0, length = 0.15)
arrows(0, 0, 0, 2, length = 0.15)
arrows(0, 0, -2, -2, length = 0.15)
arrows(0, 0, 0.3, 0.18, col = 'red', lwd = 3, length = 0.15)
arrows(0, 0, -0.18, 0.3, col = 'red', lwd = 3, length = 0.15)
arrows(0, 0, 1.3, 1.5, length = 0.15)
arrows(-2, -1.2, 3, 1.8, col = 'green4', length = 0.15)
arrows(1, -10/6, -1, 10/6, col = 'green4', length = 0.15)
a <- array(0, dim = length(x))
b <- array(0, dim = length(x))
c <- array(0, dim = length(x))
d <- array(0, dim = length(x))
for (i in 1:length(x)) {
  a[i] <- (10 * x[i] + 6 * y[i]) / 13.6
  b[i] <- 0.6 * a[i]
  c[i] <- ((6/10) * x[i] - y[i]) / (10/6 + 6/10)
  d[i] <- -(10/6) * c[i]
}
lines(c(x[5], a[5]), c(y[5], b[5]), lty=3)
lines(c(x[5], c[5]), c(y[5], d[5]), lty=3)
```

拟合 n 维数据点 (Fitting the n-Dimensional Point Cloud)

- q $(q \le n)$ 维子空间



子空间之间的关系 (Relations Between Subspaces)

- 上述两种方法之间的对偶关系.
  - 考虑 $\mathbb{R}^n$ 中的特征向量方程

$$\left(\mathcal{X}\mathcal{X}^{\mathrm{T}}\right)\boldsymbol{v}_k = \mu_k \boldsymbol{v}_k, \quad k \le r = \operatorname{rank}\left(\mathcal{X}\mathcal{X}^{\mathrm{T}}\right) = \operatorname{rank}\left(\mathcal{X}\right) \le \min\left\{p\ ,\ n\right\}$$

Multiplying by $\mathcal{X}^{\mathrm{T}}$

$$\mathcal{X}^{\mathrm{T}}\left(\mathcal{X}\mathcal{X}^{\mathrm{T}}\right)\boldsymbol{v}_k = \mu_k\ \mathcal{X}^{\mathrm{T}}\ \boldsymbol{v}_k$$

$$\left(\mathcal{X}^{\mathrm{T}}\mathcal{X}\right)\left(\mathcal{X}^{\mathrm{T}}\boldsymbol{v}_k\right) = \mu_k\left(\mathcal{X}^{\mathrm{T}}\ \boldsymbol{v}_k\right)$$

  - 对同一个特征值 μ_k:

对 $\mathcal{X}\mathcal{X}^{\mathrm{T}}$ 的每一个特征向量 $\boldsymbol{v}_k$ $\Longrightarrow$ $\left(\mathcal{X}^{\mathrm{T}}\boldsymbol{v}_k\right)$ 是 $\mathcal{X}^{\mathrm{T}}\mathcal{X}$ 的一个特征向量

  - $\mathcal{X}\mathcal{X}^{\mathrm{T}}$ 的每一个非零特征值都是 $\mathcal{X}^{\mathrm{T}}\mathcal{X}$ 的一个特征值.
  - 对应的特征向量是 $\boldsymbol{u}_k = c_k\,\mathcal{X}^{\mathrm{T}}\boldsymbol{v}_k$.

c_k 是某个常数

子空间之间的关系 (Relations Between Subspaces)

- 本章所介绍的两种方法之间的对偶关系.
  - 现在，考虑 $\mathbb{R}^p$ 中的特征向量方程

$$\left(\mathcal{X}^{\mathrm{T}}\mathcal{X}\right)\boldsymbol{u}_k = \lambda_k \boldsymbol{u}_k, \quad k \le r = \operatorname{rank}\left(\mathcal{X}\mathcal{X}^{\mathrm{T}}\right) = \operatorname{rank}\left(\mathcal{X}\right) \le \min\left\{p\ ,\ n\right\}$$

Multiplying by $\mathcal{X}$

$$\mathcal{X}\left(\mathcal{X}^{\mathrm{T}}\mathcal{X}\right)\boldsymbol{u}_k = \lambda_k \mathcal{X}\boldsymbol{u}_k$$

$$\left(\mathcal{X}\mathcal{X}^{\mathrm{T}}\right)\left(\mathcal{X}\boldsymbol{u}_k\right) = \lambda_k\left(\mathcal{X}\ \boldsymbol{u}_k\right)$$

  - 对同一个特征值 λ_k :

    对 $\mathcal{X}^{\mathrm{T}}\mathcal{X}$ 的每一个特征向量 $\boldsymbol{u}_k \Longrightarrow \left(\mathcal{X}\boldsymbol{u}_k\right)$ 是 $\mathcal{X}\mathcal{X}^{\mathrm{T}}$ 的一个特征向量

  - $\mathcal{X}^{\mathrm{T}}\mathcal{X}$ 的每一个非零特征值是 $\mathcal{X}\mathcal{X}^{\mathrm{T}}$ 的一个特征值.
  - 对应的特征向量是 $\boldsymbol{v}_k = d_k \mathcal{X}\boldsymbol{u}_k$.

d_k 是某个常数

子空间之间的关系 (Relations Between Subspaces)

- 本章所介绍的两种方法之间的对偶关系.
  - 因为

$$\boldsymbol{u}_k^{\mathrm{T}}\boldsymbol{u}_k = \boldsymbol{v}_k^{\mathrm{T}}\boldsymbol{v}_k = 1$$

$$\left(\mathcal{X}^{\mathrm{T}}\mathcal{X}\right)\boldsymbol{u}_k = \lambda_k \boldsymbol{u}_k$$

$$\boldsymbol{v}_k^{\mathrm{T}}\boldsymbol{v}_k = \left(d_k\mathcal{X}\boldsymbol{u}_k\right)^{\mathrm{T}}\left(d_k\mathcal{X}\boldsymbol{u}_k\right) = d_k^2\boldsymbol{u}_k^{\mathrm{T}}\left(\mathcal{X}^{\mathrm{T}}\mathcal{X}\right)\boldsymbol{u}_k = \lambda_k d_k^2\boldsymbol{u}_k^{\mathrm{T}}\boldsymbol{u}_k = \lambda_k d_k^2 = 1$$

$$\Longrightarrow \quad d_k = \frac{1}{\sqrt{\lambda_k}}$$

$$\left(\mathcal{X}\mathcal{X}^{\mathrm{T}}\right)\boldsymbol{v}_k = \mu_k \boldsymbol{v}_k$$

$$\boldsymbol{u}_k^{\mathrm{T}}\boldsymbol{u}_k = \left(c_k\mathcal{X}^{\mathrm{T}}\boldsymbol{v}_k\right)^{\mathrm{T}}\left(c_k\mathcal{X}^{\mathrm{T}}\boldsymbol{v}_k\right) = c_k^2\boldsymbol{v}_k^{\mathrm{T}}\left(\mathcal{X}\mathcal{X}^{\mathrm{T}}\right)\boldsymbol{v}_k = \mu_k c_k^2\boldsymbol{v}_k^{\mathrm{T}}\boldsymbol{v}_k = \mu_k c_k^2 = 1$$

$$\Longrightarrow \quad c_k = \frac{1}{\sqrt{\mu_k}}$$

  - 因为 $\mathcal{X}^{\mathrm{T}}\mathcal{X}$ 与 $\mathcal{X}\mathcal{X}^{\mathrm{T}}$ 有相同的非零特征值. $\Longrightarrow$ $\lambda_k = \mu_k$

$$\Longrightarrow \quad c_k = d_k = \frac{1}{\sqrt{\lambda_k}}$$

子空间之间的关系 (Relations Between Subspaces)

- 本章所介绍的两种方法之间的对偶关系.

定理 10.4 (对偶关系) 设 $\mathcal{X}$ 的秩为 r. 对于 $k \le r$, $\mathcal{X}^{\mathrm{T}}\mathcal{X}$ 与 $\mathcal{X}\mathcal{X}^{\mathrm{T}}$ 的第 k 个特征值 λ_k 相同, 对应的特征向量 (分别记为 $\boldsymbol{u}_k$ 与 $\boldsymbol{v}_k$) 之间的关系如下

$$\boldsymbol{u}_k = \frac{1}{\sqrt{\lambda_k}} \mathcal{X}^{\mathrm{T}} \boldsymbol{v}_k , \quad \boldsymbol{v}_k = \frac{1}{\sqrt{\lambda_k}} \mathcal{X} \boldsymbol{u}_k .$$

- p 个变量在因子轴 $\boldsymbol{v}_k$ 上的投影为

$$w_k = \mathcal{X}^{\mathrm{T}} \boldsymbol{v}_k = \frac{1}{\sqrt{\lambda_k}} \mathcal{X}^{\mathrm{T}} \mathcal{X} \boldsymbol{u}_k = \frac{1}{\sqrt{\lambda_k}} \lambda_k \boldsymbol{u}_k = \sqrt{\lambda_k}\ \boldsymbol{u}_k$$

$$\left(\mathcal{X}^{\mathrm{T}}\mathcal{X}\right) \boldsymbol{u}_k = \lambda_k \boldsymbol{u}_k$$

- 所以，为了得到 w_k，并不需要精确计算出特征向量 $\boldsymbol{v}_k$.

子空间之间的关系 (Relations Between Subspaces)

- 本章所介绍的两种方法之间的对偶关系.

定理 10.4 (对偶关系) 设 $\mathcal{X}$ 的秩为 r. 对于 $k \le r$, $\mathcal{X}^{\mathrm{T}}\mathcal{X}$ 与 $\mathcal{X}\mathcal{X}^{\mathrm{T}}$ 的第 k 个特征值 λ_k 相同, 对应的特征向量 (分别记为 $\boldsymbol{u}_k$ 与 $\boldsymbol{v}_k$) 之间的关系如下

$$\boldsymbol{u}_k = \frac{1}{\sqrt{\lambda_k}} \mathcal{X}^{\mathrm{T}} \boldsymbol{v}_k , \quad \boldsymbol{v}_k = \frac{1}{\sqrt{\lambda_k}} \mathcal{X} \boldsymbol{u}_k .$$

- 注意, $\boldsymbol{u}_k$ 和 $\boldsymbol{v}_k$ 构成了 $\mathcal{X}$ 的奇异值分解 (SVD, singular value decomposition).

- **Theorem 2.2** (Singular Value Decomposition)

  Each matrix $\mathcal{A}_{n\times p}$ with rank r can be decomposed as

  $$\mathcal{A} = \Gamma \, \Lambda \, \Delta^{\mathrm{T}}$$

  where $\Gamma_{n\times r}$ and $\Delta_{p\times r}$. Both Γ and Δ are column orthogonal, i.e.,

  $$\Gamma^{\mathrm{T}} \Gamma = \Delta^{\mathrm{T}} \Delta = \boldsymbol{I}_r$$

  and

  $$\Lambda = \mathrm{diag}\left(\lambda_1^{1/2}, \lambda_2^{1/2}, \ \ldots, \lambda_r^{1/2}\right) \ , \lambda_j > 0$$

  the values $\lambda_1, \lambda_2, \ \ldots, \lambda_r$ are the none zero eigenvalues of the matrices $\mathcal{A}\mathcal{A}^{\mathrm{T}}$ and $\mathcal{A}^{\mathrm{T}}\mathcal{A}$. Γ and Δ consist of the corresponding r eigenvectors of these matrices.

子空间之间的关系 (Relations Between Subspaces)

- 本章所介绍的两种方法之间的对偶关系.

定理 10.4 (对偶关系) 设 $\mathcal{X}$ 的秩为 r. 对于 $k \le r$, $\mathcal{X}^{\mathrm{T}}\mathcal{X}$ 与 $\mathcal{X}\mathcal{X}^{\mathrm{T}}$ 的第 k 个特征值 λ_k 相同, 对应的特征向量 (分别记为 $\boldsymbol{u}_k$ 与 $\boldsymbol{v}_k$) 之间的关系如下

$$\boldsymbol{u}_k = \frac{1}{\sqrt{\lambda_k}} \mathcal{X}^{\mathrm{T}} \boldsymbol{v}_k\,, \quad \boldsymbol{v}_k = \frac{1}{\sqrt{\lambda_k}} \mathcal{X}\, \boldsymbol{u}_k\,.$$

- 注意, $\boldsymbol{u}_k$ 和 $\boldsymbol{v}_k$ 构成了 $\mathcal{X}$ 的奇异值分解 (SVD, singular value decomposition).

- 令

$$\left.\begin{aligned} U &= \left(\boldsymbol{u}_1 \quad \boldsymbol{u}_2 \quad \cdots \quad \boldsymbol{u}_r\right) \\ V &= \left(\boldsymbol{v}_1 \quad \boldsymbol{v}_2 \quad \cdots \quad \boldsymbol{v}_r\right) \\ \Lambda &= \mathrm{diag}\left(\lambda_1\,, \lambda_2\,, \ \ldots\ , \lambda_r\right) \end{aligned}\right\} \Longrightarrow \quad \mathcal{X} = V\Lambda^{1/2}U^{\mathrm{T}}$$

$$\Longrightarrow \quad x_{ij} = \sum_{k=1}^{r} \lambda_k^{1/2}\, v_{ik}\, u_{jk}$$

- **Theorem 2.2** (Singular Value Decomposition)

Each matrix $\mathcal{A}_{n\times p}$ with rank r can be decomposed as

$$\mathcal{A} = \Gamma\, \Lambda\, \Delta^{\mathrm{T}}$$

where $\Gamma_{n\times r}$ and $\Delta_{p\times r}$. Both Γ and Δ are column orthogonal, i.e.,

$$\Gamma^{\mathrm{T}}\Gamma = \Delta^{\mathrm{T}}\Delta = \boldsymbol{I}_r$$

and

$$\Lambda = \mathrm{diag}\left(\lambda_1^{1/2}, \lambda_2^{1/2}, \ \ldots, \lambda_r^{1/2}\right)\ , \lambda_j > 0$$

the values $\lambda_1, \lambda_2, \ \ldots, \lambda_r$ are the none zero eigenvalues of the matrices $\mathcal{A}\mathcal{A}^{\mathrm{T}}$ and $\mathcal{A}^{\mathrm{T}}\mathcal{A}$. Γ and Δ consist of the corresponding r eigenvectors of these matrices.

实际计算 (Practical Computations)

- 求 $\mathcal{X}^{\mathrm{T}}\mathcal{X}$ 的特征值 $\lambda_1 \geq \lambda_2 \geq \cdots \geq \lambda_p$ 及其对应的特征向量 $\boldsymbol{u}_1, \boldsymbol{u}_2, \ldots, \boldsymbol{u}_p$.
  - 通过绘制 $\boldsymbol{z}_1 = \mathcal{X}\boldsymbol{u}_1$ 关于 $\boldsymbol{z}_2 = \mathcal{X}\boldsymbol{u}_2$ 的图形，即可得到 n 个观测点在一个平面上的表现.
  - 如果需要增加第三个维度，我们可以添加 $\boldsymbol{z}_3 = \mathcal{X}\boldsymbol{u}_3$.
  - 通过绘制 $\boldsymbol{w}_1 = \sqrt{\lambda_1}\boldsymbol{u}_1$ 关于 $\boldsymbol{w}_2 = \sqrt{\lambda_2}\boldsymbol{u}_2$ 的图形，即可得到 p 个变量在一个平面上的表现.
  - 如果需要增加第三个维度，我们可以添加 $\boldsymbol{w}_3 = \sqrt{\lambda_3}\boldsymbol{u}_3$.
  - 更高维的因子解可以通过计算 $k > 3$ 时的 $\boldsymbol{z}_k$ 与 $\boldsymbol{w}_k$ 得到，当然，此时无法进行图形表示.

实际计算 (Practical Computations)

- 评估维度为 q 的子空间中因子表示质量的一种标准方法为

$$\tau_q = \frac{\lambda_1 + \lambda_2 + \cdots + \lambda_q}{\left(\lambda_1 + \lambda_2 + \cdots + \lambda_q\right) + \lambda_{q+1} + \cdots + \lambda_p} \quad \Longrightarrow \quad 0 \le \tau_q \le 1$$

  - 一般来说，我们称标量积 $\boldsymbol{y}^{\mathrm{T}}\boldsymbol{y}$ 为 $\boldsymbol{y} \in \mathbb{R}^n$ 关于原点的惯量 (inertia).
  - 因为 $\left(\mathscr{X}^{\mathrm{T}}\mathscr{X}\right)\boldsymbol{u}_j = \lambda_j \boldsymbol{u}_j \quad \Longrightarrow \quad \boldsymbol{u}_j^{\mathrm{T}}\left(\mathscr{X}^{\mathrm{T}}\mathscr{X}\right)\boldsymbol{u}_j = \lambda_j \boldsymbol{u}_j^{\mathrm{T}}\boldsymbol{u}_j = \lambda_j = \left(\mathscr{X}\boldsymbol{u}_j\right)^{\mathrm{T}}\left(\mathscr{X}\boldsymbol{u}_j\right) = \boldsymbol{z}_j^{\mathrm{T}}\boldsymbol{z}_j$
  - 因此, λ_j 是第 j 个因子变量关于原点的惯量.
  - 比值 τ_q 常被看作是惯量可以由前 q 个因子解释的比例.
  - $\lambda_1 + \lambda_2 + \cdots + \lambda_p$ 是 p 个变量的总惯量的一个度量：

$$\sum_{j=1}^{p} \lambda_j = \mathrm{tr}\left(\mathscr{X}^{\mathrm{T}}\mathscr{X}\right) = \sum_{j=1}^{p}\sum_{i=1}^{n} x_{ij}^2 = \sum_{j=1}^{p} \boldsymbol{x}_{[j]}^{\mathrm{T}}\boldsymbol{x}_{[j]}$$

实际计算 (Practical Computations)

- 例: 研究不同子女数量 (2 个、3 个、4 个或 5 个孩子) 的法国家庭 (体力劳动者 = MA，雇员 = EM，经理 = CA) 的食品支出情况.
  - 我们感兴趣的是研究某些家庭类型是否偏爱特定的食物. 利用这里推导的因子近似法，我们能够回答这个问题.
  - 数据集:

```
setwd("~/Desktop/2023_Applied Multivariate Statistical Analysis/R Codes with data/Data")
library(data.table)
x <- read.csv("food.csv", header = FALSE)
x
```

```
> x
    V2   V3  V4   V5   V6  V7  V8
1  332  428 354 1437  526 247 427
2  293  559 388 1527  567 239 258
3  372  767 562 1948  927 235 433
4  406  563 341 1507  544 324 407
5  386  608 396 1501  558 319 363
6  438  843 689 2345 1148 243 341
7  534  660 367 1620  638 414 407
8  460  699 484 1856  762 400 416
9  385  789 621 2366 1149 304 282
10 655  776 423 1848  759 495 486
11 584  995 548 2056  893 518 319
12 515 1097 887 2630 1167 561 284
```

实际计算 (Practical Computations)

- 例: 研究不同子女数量 (2 个、3 个、4 个或 5 个孩子) 的法国家庭 (体力劳动者 = MA，雇员 = EM，经理 = CA) 的食品支出情况.
  - 请注意，在这个特定问题中，原点没有特定含义 (它代表一个“零消费”的消费者).
  - 因此，将任何一个家庭的消费与 “普通家庭” 的消费进行比较，而不是与原点 (零消费) 进行比较，这样做更有意义.
  - 因此，首先对数据进行中心化处理 (将原点平移至重心点 $\bar{x}$)

```
x = x[, 2:ncol(x)]
n = nrow(x)
p = ncol(x)
a = x - matrix(apply(x, 2, mean), n, p, byrow = T)  # substracts mean
round(a, digits = 3)
```

```
> round(a, digits = 3)
      bread veget fruit    meat    poult    milk     wine
1  -114.667  -304  -151 -449.75 -277.167 -111.25   58.417
2  -153.667  -173  -117 -359.75 -236.167 -119.25 -110.583
3   -74.667    35    57   61.25  123.833 -123.25   64.417
4   -40.667  -169  -164 -379.75 -259.167  -34.25   38.417
5   -60.667  -124  -109 -385.75 -245.167  -39.25   -5.583
6    -8.667   111   184  458.25  344.833 -115.25  -27.583
7    87.333   -72  -138 -266.75 -165.167   55.75   38.417
8    13.333   -33   -21  -30.75  -41.167   41.75   47.417
9   -61.667    57   116  479.25  345.833  -54.25  -86.583
10 208.333    44   -82  -38.75  -44.167  136.75  117.417
11 137.333   263    43  169.25   89.833  159.75  -49.583
12  68.333   365   382  743.25  363.833  202.75  -84.583
```

实际计算 (Practical Computations)

- 例: 研究不同子女数量 (2 个、3 个、4 个或 5 个孩子) 的法国家庭 (体力劳动者 = MA，雇员 = EM，经理 = CA) 的食品支出情况.
  - 回顾一下中心化矩阵为 $\mathscr{H} = \mathscr{I}_n - \frac{1}{n}\mathbf{1}_n\mathbf{1}_n^{\mathrm{T}}$. 因此，我们可以通过以下方式得到中心化后的数据:

```
one = matrix(1, n, n)
h = diag(1, n, n) - one/n  # centering the matrix
round(h %*% as.matrix(x), digits = 3)
```

```
> one = matrix(1, n, n)
> h = diag(1, n, n) - one/n  # centering the matrix
> round(h %*% as.matrix(x), digits = 3)
        bread veget fruit    meat    poult    milk     wine
 [1,] -114.667  -304  -151 -449.75 -277.167 -111.25   58.417
 [2,] -153.667  -173  -117 -359.75 -236.167 -119.25 -110.583
 [3,]  -74.667    35    57   61.25  123.833 -123.25   64.417
 [4,]  -40.667  -169  -164 -379.75 -259.167  -34.25   38.417
 [5,]  -60.667  -124  -109 -385.75 -245.167  -39.25   -5.583
 [6,]   -8.667   111   184  458.25  344.833 -115.25  -27.583
 [7,]   87.333   -72  -138 -266.75 -165.167   55.75   38.417
 [8,]   13.333   -33   -21  -30.75  -41.167   41.75   47.417
 [9,]  -61.667    57   116  479.25  345.833  -54.25  -86.583
[10,]  208.333    44   -82  -38.75  -44.167  136.75  117.417
[11,]  137.333   263    43  169.25   89.833  159.75  -49.583
[12,]   68.333   365   382  743.25  363.833  202.75  -84.583
```

实际计算 (Practical Computations)

- 例: 研究不同子女数量 (2 个、3 个、4 个或 5 个孩子) 的法国家庭 (体力劳动者 = MA，雇员 = EM，经理 = CA) 的食品支出情况.
  - 此外，由于这七个变量的离散程度差异很大，每个变量都要进行标准化处理，以便在分析中每个变量具有相同的权重 (均值为0，方差为1).
  - 最后，为方便起见，我们将数据矩阵中的每个元素都除以 $\sqrt{n} = \sqrt{12}$. (这只会改变图形表示中绘图的比例.)
  - 要分析的数据矩阵为 $\mathcal{X}_* = \frac{1}{\sqrt{n}}\mathscr{H}\mathcal{X}\mathscr{D}^{-1/2}$, 其中 $\mathscr{D} = \text{diag}\left(s_{X_iX_i}\right)$.

```
d = diag(1/sqrt(colSums(a^2)/n))
xs = h %*% as.matrix(x) %*% d  # standardized data
xs1 = xs/sqrt(n)
round(xs1, digits = 3)
```

```
> d = diag(1/sqrt(colSums(a^2)/n))
> xs = h %*% as.matrix(x) %*% d  # standardized data
> xs1 = xs/sqrt(n)
> round(xs1, digits = 3)
      [,1]   [,2]   [,3]   [,4]   [,5]   [,6]   [,7]
 [1,] -0.323 -0.485 -0.276 -0.343 -0.335 -0.286  0.245
 [2,] -0.432 -0.276 -0.214 -0.274 -0.285 -0.307 -0.464
 [3,] -0.210  0.056  0.104  0.047  0.150 -0.317  0.271
 [4,] -0.114 -0.269 -0.300 -0.289 -0.313 -0.088  0.161
 [5,] -0.171 -0.198 -0.199 -0.294 -0.296 -0.101 -0.023
 [6,] -0.024  0.177  0.336  0.349  0.417 -0.297 -0.116
 [7,]  0.246 -0.115 -0.252 -0.203 -0.200  0.144  0.161
 [8,]  0.038 -0.053 -0.038 -0.023 -0.050  0.107  0.199
 [9,] -0.174  0.091  0.212  0.365  0.418 -0.140 -0.364
[10,]  0.586  0.070 -0.150 -0.030 -0.053  0.352  0.493
[11,]  0.386  0.419  0.079  0.129  0.109  0.411 -0.208
[12,]  0.192  0.582  0.698  0.566  0.440  0.522 -0.355
```

实际计算 (Practical Computations)

- 例: 研究不同子女数量 (2 个、3 个、4 个或 5 个孩子) 的法国家庭 (体力劳动者 = MA，雇员 = EM，经理 = CA) 的食品支出情况.
  - 相关矩阵为

```
xs2 = t(xs1) %*% xs1
round(xs2, digits = 2)
```

```
> xs2 = t(xs1) %*% xs1
> round(xs2, digits = 2)
     [,1]  [,2]  [,3]  [,4]  [,5] [,6]  [,7]
[1,] 1.00  0.59  0.20  0.32  0.25 0.86  0.30
[2,] 0.59  1.00  0.86  0.88  0.83 0.66 -0.36
[3,] 0.20  0.86  1.00  0.96  0.93 0.33 -0.49
[4,] 0.32  0.88  0.96  1.00  0.98 0.37 -0.44
[5,] 0.25  0.83  0.93  0.98  1.00 0.23 -0.40
[6,] 0.86  0.66  0.33  0.37  0.23 1.00  0.01
[7,] 0.30 -0.36 -0.49 -0.44 -0.40 0.01  1.00
```

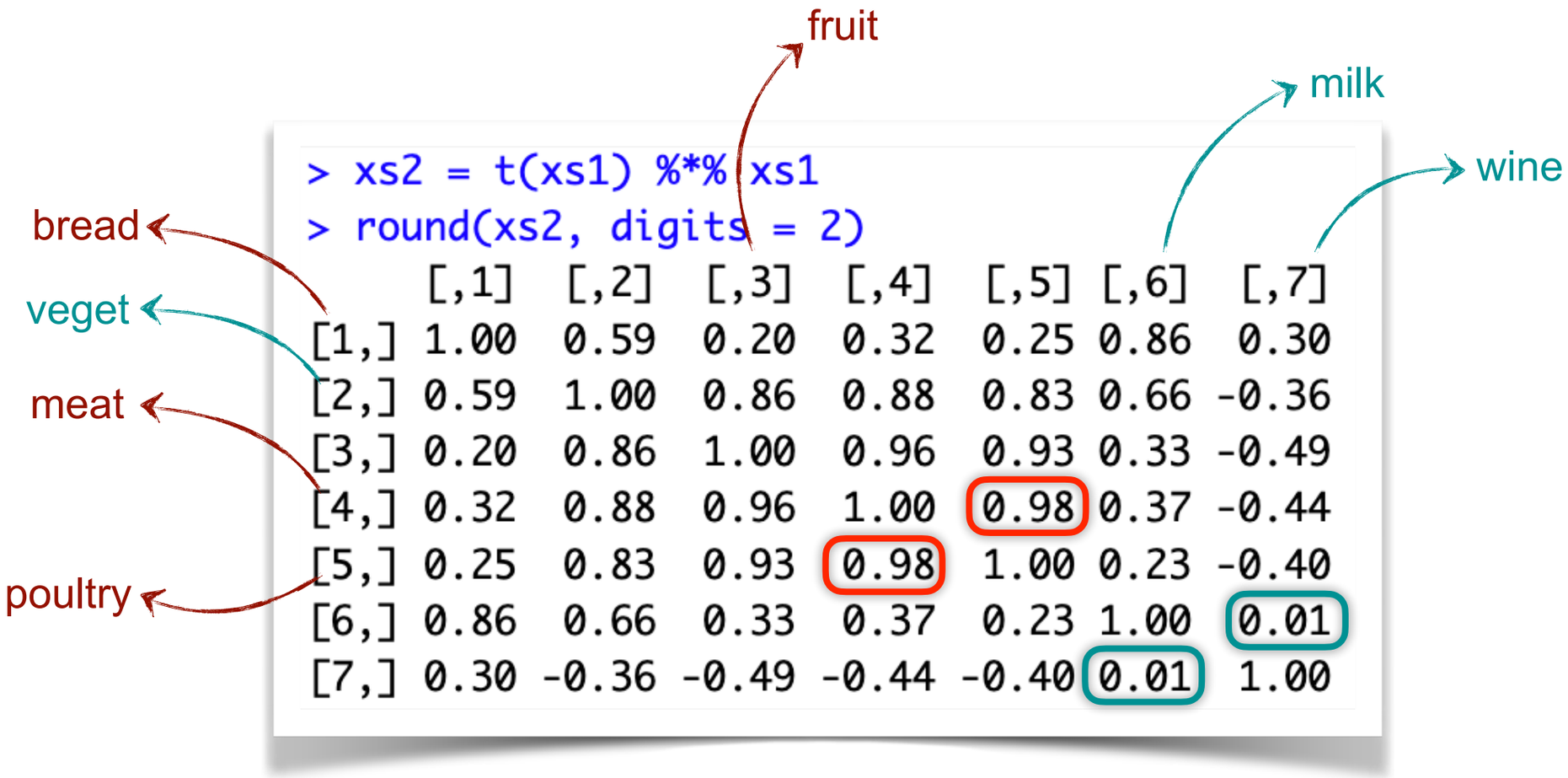
```
> round(cor(x), digits = 2)
      bread veget fruit  meat poult milk  wine
bread  1.00  0.59  0.20  0.32  0.25 0.86  0.30
veget  0.59  1.00  0.86  0.88  0.83 0.66 -0.36
fruit  0.20  0.86  1.00  0.96  0.93 0.33 -0.49
meat   0.32  0.88  0.96  1.00  0.98 0.37 -0.44
poult  0.25  0.83  0.93  0.98  1.00 0.23 -0.40
milk   0.86  0.66  0.33  0.37  0.23 1.00  0.01
wine   0.30 -0.36 -0.49 -0.44 -0.40 0.01  1.00
```

  - 我们也可以通过以下方式从原始数据中得到相关矩阵

```
round(cor(x), digits = 2)
```

实际计算 (Practical Computations)

- 例: 研究不同子女数量 (2 个、3 个、4 个或 5 个孩子) 的法国家庭 (体力劳动者 = MA，雇员 = EM，经理 = CA) 的食品支出情况.
  - 相关矩阵为



  - 是否存在某些类型的家庭，相对于面包更喜欢肉类呢?

实际计算 (Practical Computations)

- 例: 研究不同子女数量 (2 个、3 个、4 个或 5 个孩子) 的法国家庭 (体力劳动者 = MA，雇员 = EM，经理 = CA) 的食品支出情况.
  - 相关矩阵的特征值和对应的特征向量为

```
eig = eigen(xs2)  # spectral decomposition
lambda = eig$values
round(lambda, digits = 2)
gamma = eig$vectors
round(gamma, digits = 2)
```

```
> round(lambda, digits = 2)
[1] 4.33 1.83 0.63 0.13 0.06 0.02 0.00
```

```
> round(gamma, digits = 2)
      [,1]  [,2]  [,3]  [,4]  [,5]  [,6]  [,7]
[1,] -0.24  0.62  0.01 -0.54 -0.04  0.51 -0.02
[2,] -0.47  0.10  0.06 -0.02  0.81 -0.30  0.16
[3,] -0.45 -0.21 -0.15  0.55  0.07  0.63 -0.20
[4,] -0.46 -0.14 -0.21 -0.05 -0.41 -0.09  0.74
[5,] -0.44 -0.20 -0.36 -0.32 -0.22 -0.35 -0.60
[6,] -0.28  0.52  0.44  0.45 -0.34 -0.33 -0.15
[7,]  0.21  0.48 -0.78  0.31  0.07 -0.14  0.04
```

$$\boldsymbol{\lambda} = \begin{pmatrix} \lambda_1 \\ \lambda_2 \\ \lambda_3 \\ \lambda_4 \\ \lambda_5 \\ \lambda_6 \\ \lambda_7 \end{pmatrix} = \begin{pmatrix} 4.33 \\ 1.83 \\ 0.63 \\ 0.13 \\ 0.06 \\ 0.02 \\ 0.00 \end{pmatrix}$$

$$\tau_2 = \frac{\lambda_1 + \lambda_2}{\lambda_1 + \lambda_2 + \lambda_3 + \lambda_4 + \lambda_5 + \lambda_6 + \lambda_7} = 0.8805039$$

- 二维图应足以解释这个数据集.

实际计算 (Practical Computations)

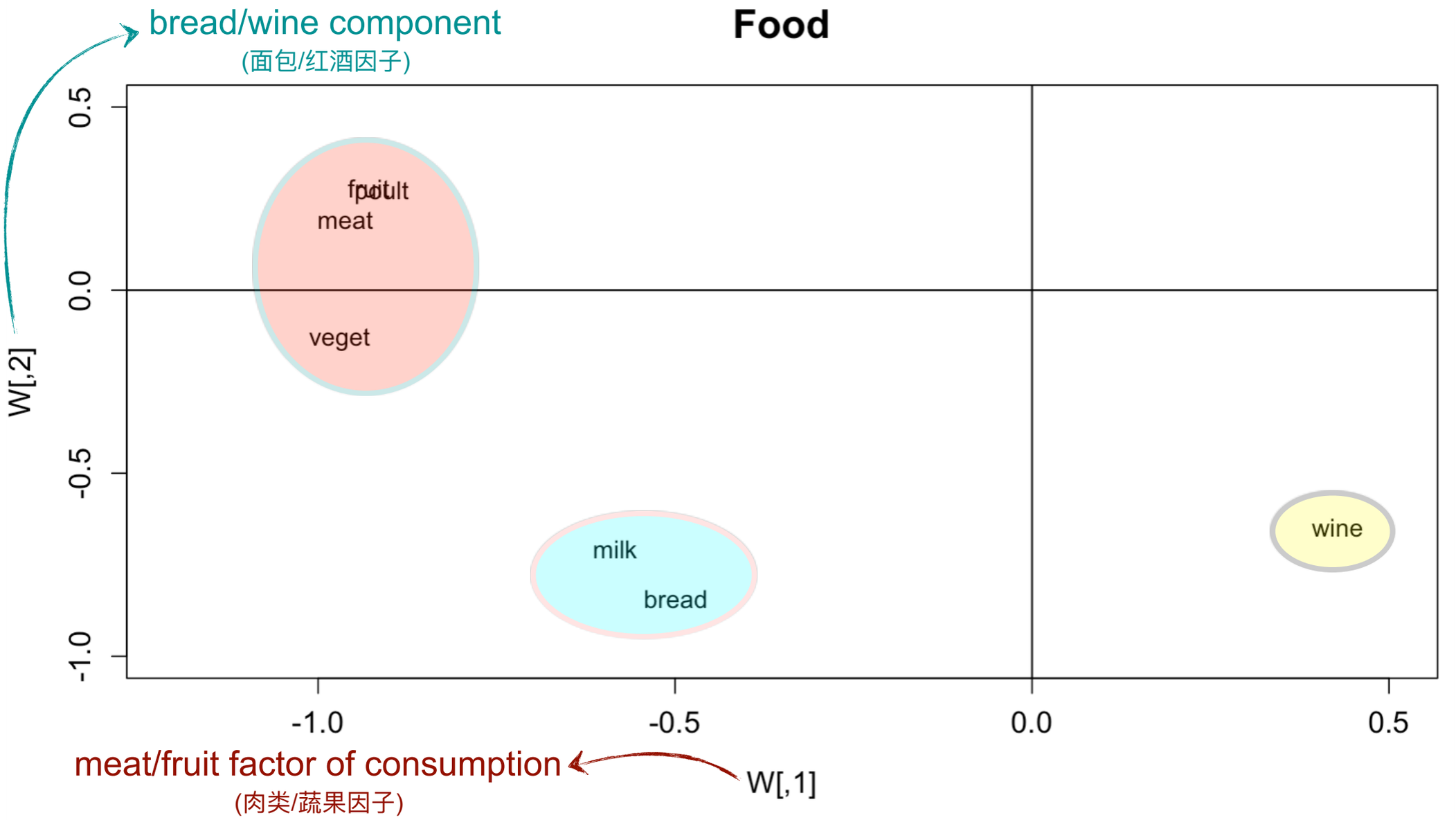
- 例: 研究不同子女数量 (2 个、3 个、4 个或 5 个孩子) 的法国家庭 (体力劳动者 = MA，雇员 = EM，经理 = CA) 的食品支出情况.
  - $p = 7$ 个变量在一个平面上的表现可以通过 $\boldsymbol{w}_1 = \sqrt{\lambda_1}\boldsymbol{u}_1$ 关于 $\boldsymbol{w}_2 = \sqrt{\lambda_2}\boldsymbol{u}_2$ 的散点图来表示.

```
w = gamma * (matrix(sqrt(lambda), nrow = nrow(gamma), ncol = ncol(gamma), byrow = T))  # coordinates of food
w = w[, 1:2]
w = round(w, 3)
namew = c("bread", "veget", "fruit", "meat", "poult", "milk", "wine")

par(mfrow = c(2, 1))
plot(w[, 1], -w[, 2], type = "n", xlab = "W[,1]", ylab = "W[,2]", main = "Food", cex.axis = 1.2,
     cex.lab = 1.2, cex.main = 1.6, xlim = c(-1.2, 0.5), ylim = c(-1, 0.5))
text(w[, 1], -w[, 2], namew, xpd = NA)
abline(h = 0, v = 0, lwd = 1.2)

for (i in 1:7) {
  mtext(namew[i], side = 1, line = 5 + i, at = -1.15)
  mtext(toString(c(sprintf("%.3f", w[i, 1]))), side = 1, line = 5 + i, at = -0.55)
  mtext(toString(c(sprintf("%.3f", w[i, 2]))), side = 1, line = 5 + i, at = 0)
}
```

实际计算 (Practical Computations)



bread	-0.499	0.842
veget	-0.970	0.133
fruit	-0.929	-0.278
meat	-0.962	-0.191
poult	-0.911	-0.266
milk	-0.584	0.707
wine	0.428	0.648

实际计算 (Practical Computations)

- 例: 研究不同子女数量 (2 个、3 个、4 个或 5 个孩子) 的法国家庭 (体力劳动者 = MA，雇员 = EM，经理 = CA) 的食品支出情况.
  - n 个数据点在一个平面上的表现可以通过绘制 $z_1 = \mathcal{X}\boldsymbol{u}_1$关于 $z_2 = \mathcal{X}\boldsymbol{u}_2$ 的散点图来实现.

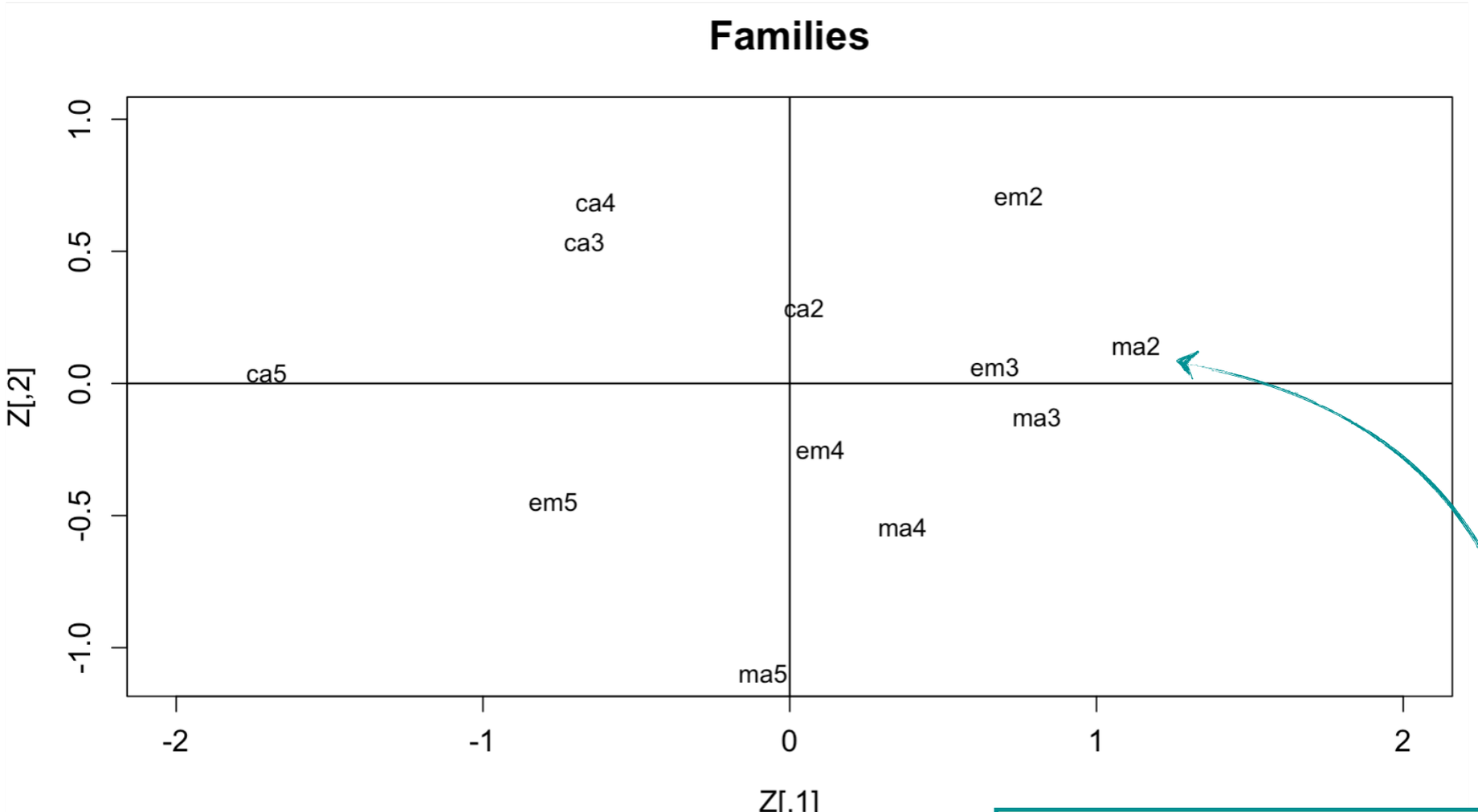
```
z1 = xs1 %*% gamma  # coordinates of families
z2 = sqrt(n/p) * z1
z = z2[, 1:2]
z = round(z, 3)
namez = c("ma2", "em2", "ca2", "ma3", "em3", "ca3", "ma4", "em4", "ca4", "ma5", "em5", "ca5")

graphics.off()
par(mfrow = c(2, 1))
plot(z[, 1], -z[, 2], type = "n", xlim = c(-2, 2), ylim = c(-1.1, 1), xlab = "Z[,1]",
     ylab = "Z[,2]", main = "Families", cex.axis = 1.2, cex.lab = 1.2, cex.main = 1.6)
text(z[, 1], -z[, 2], namez, xpd = NA)
abline(h = 0, v = 0, lwd = 1.2)

for (i in 1:12) {
  mtext(namez[i], side = 1, line = 5 + i, at = -2)
  mtext(toString(c(sprintf("%.3f", z[i, 1]))), side = 1, line = 5 + i, at = -1)
  mtext(toString(c(sprintf("%.3f", z[i, 2]))), side = 1, line = 5 + i, at = -0)
}
```

Decomposition of Data Matrices by Factors

实际计算 (Practical Computations)

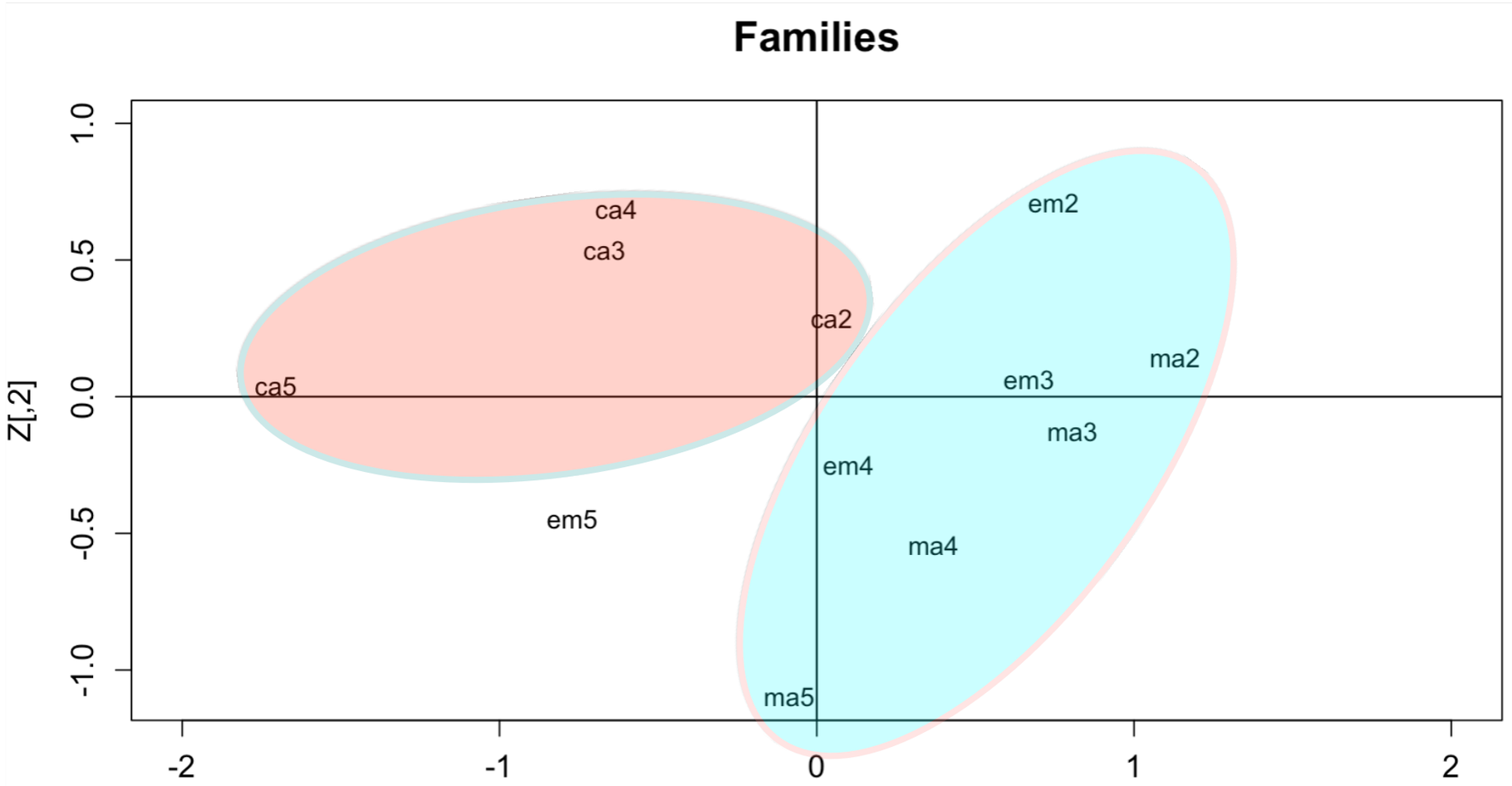


ma2	1.129	-0.144
em2	0.746	-0.708
ca2	0.047	-0.286
ma3	0.806	0.128
em3	0.669	-0.064
ca3	-0.669	-0.535
ma4	0.368	0.542
em4	0.100	0.250
ca4	-0.632	-0.685
ma5	-0.087	1.096
em5	-0.770	0.447
ca5	-1.705	-0.040

图中显示的点是相对于由原点代表的平均消费绘制的.

Decomposition of Data Matrices by Factors

实际计算 (Practical Computations)



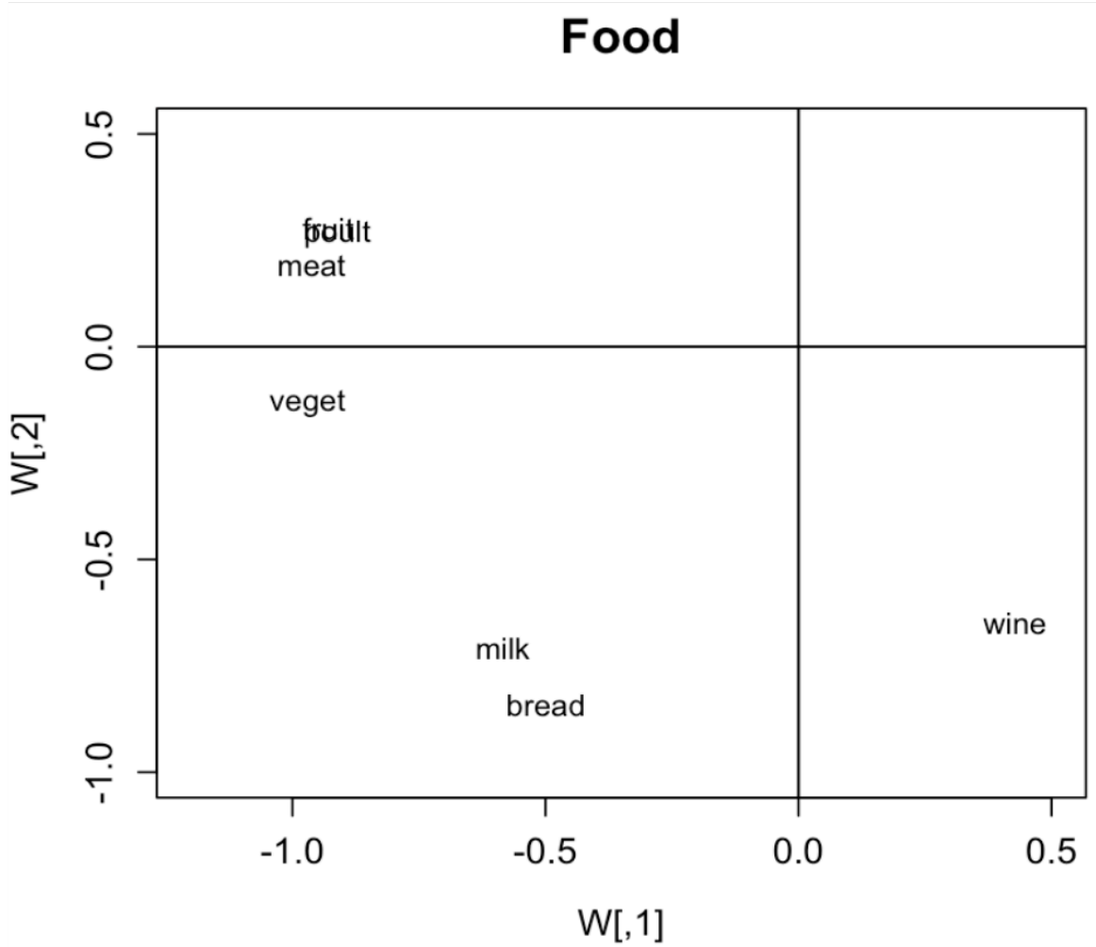
ma2	1.129	-0.144
em2	0.746	-0.708
ca2	0.047	-0.286
ma3	0.806	0.128
em3	0.669	-0.064
ca3	-0.669	-0.535
ma4	0.368	0.542
em4	0.100	0.250
ca4	-0.632	-0.685
ma5	-0.087	1.096
em5	-0.770	0.447
ca5	-1.705	-0.040

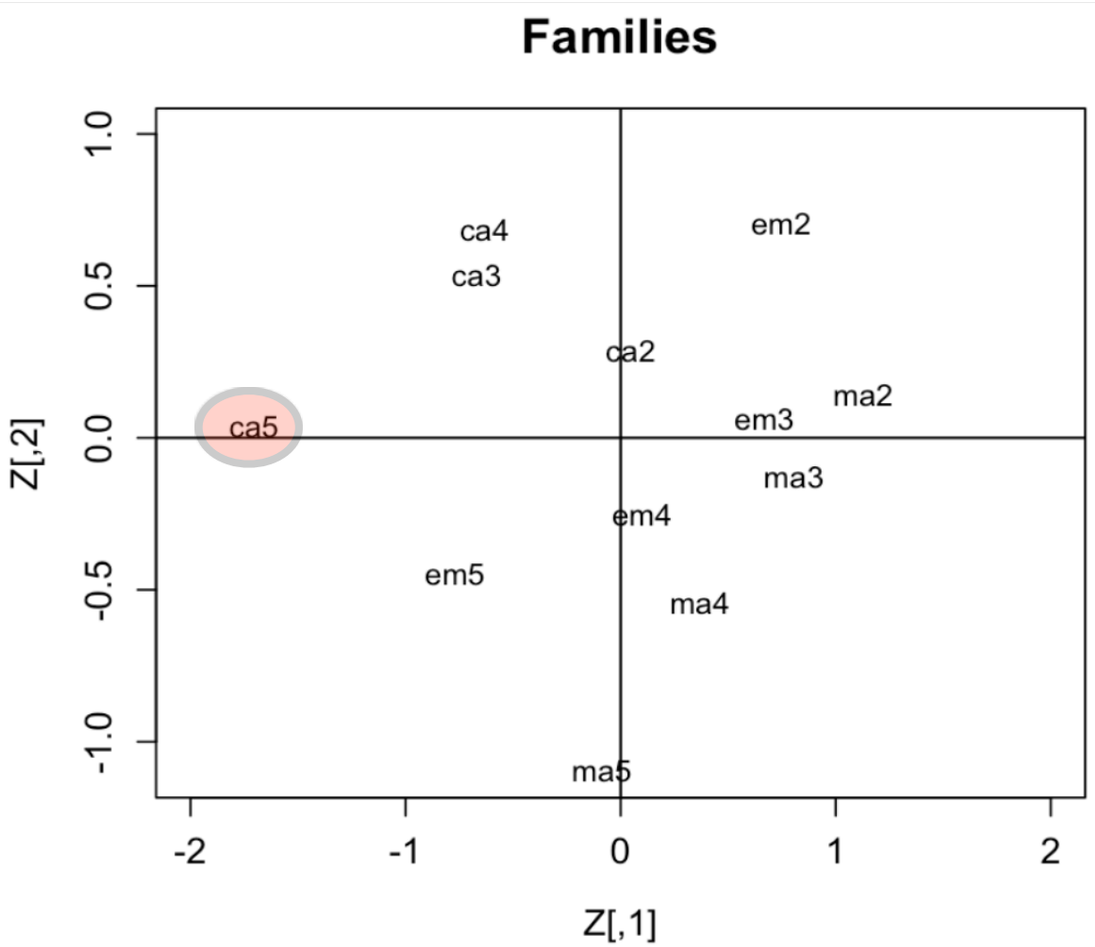
经理家庭位于图表的左上角.

体力劳动者和雇员家庭则倾向于右上角.

Decomposition of Data Matrices by Factors

实际计算 (Practical Computations)





对于 CA5 (有五个孩子的经理家庭)，其因子变量靠近肉类/水果因子. 相较于普通消费者，这类家庭是肉类/家禽以及水果/蔬菜的大量消费群体.

poult	-0.911	-0.266	em3	0.669	-0.064
milk	-0.584	0.707	ca3	-0.669	-0.535
ma5	-0.087	1.096			
em5	-0.770	0.447			
ca5	-1.705	-0.040			

在第 11 章，我们将重新审视这些图表，并进行更深入的解读.