19 Canonical Correlation Analysis 典型相关分析

找到两组数据的整体相关性的最大线性组合

> 人类生而好奇，这正是科学的火种。
>
> ***Men love to wonder, and that is the seed of science.***
>
> —— 拉尔夫·爱默生 (Ralph Waldo Emerson) | 美国思想家、文学家 | 1803 ~ 1882

- ◀ `numpy.linalg.eig()` 特征值分解
- ◀ `numpy.linalg.inv()` 矩阵求逆
- ◀ `seaborn.heatmap()` 绘制热图
- ◀ `seaborn.jointplot()` 绘制散点图，含边缘分布
- ◀ `seaborn.pairplot()` 成对散点图
- ◀ `seaborn.scatterplot()` 绘制散点图
- ◀ `sklearn.cross_decomposition.CCA()` 典型相关分析


典型相关分析
原理
优化问题
线性组合视角
随机变量视角
特征值分解

19.1 典型相关分析原理

典型相关分析 (Canonical Correlation Analysis, CCA) 是一种用于探究两组变量之间关系的多元统计分析方法。其核心思想是将两组变量分别投影到新的低维空间中，使得这两组变量在新空间中的投影尽可能相关。

CCA 常用于处理两组多元变量之间的关系。通过 CCA 可以发现这两组变量中的某些维度之间存在相关性，这种相关性可以帮助研究者更好地理解两组变量之间的关系。

使用 CCA 时，一般需要先对两组变量进行标准化处理，然后计算它们的相关系数矩阵。接着，CCA 会生成一组线性组合，使得两组变量在新的低维空间中的投影尽可能相关。这些线性组合称为典型变量，相关系数则称为典型相关系数。最终的结果是一组典型变量和对应的典型相关系数。

原理

下面以 $\boldsymbol{X}$ 和 $\boldsymbol{Y}$ 为例介绍典型相关分析原理。

$n \times p$ 数据矩阵 $\boldsymbol{X}$ 可以写成：

$$\boldsymbol{X}_{n\times p} = \begin{bmatrix} \boldsymbol{x}_1 & \boldsymbol{x}_2 & \cdots & \boldsymbol{x}_p \end{bmatrix} \tag{1}$$

$n \times q$ 数据矩阵 $\boldsymbol{Y}$ 可以写成：

$$\boldsymbol{Y}_{n\times q} = \begin{bmatrix} \boldsymbol{y}_1 & \boldsymbol{y}_2 & \cdots & \boldsymbol{y}_q \end{bmatrix} \tag{2}$$

⚠ 注意，$\boldsymbol{X}$ 和 $\boldsymbol{Y}$ 的行数一致。

$\boldsymbol{X}$ 朝向量 $\boldsymbol{u}_1$ 投影结果为 $\boldsymbol{s}_1$：

$$\boldsymbol{s}_1 = \boldsymbol{X}_{n\times p}\boldsymbol{u}_1 \tag{3}$$

其中，$\boldsymbol{u}_1$ 的形状为 $p \times 1$，$\boldsymbol{s}_1$ 的形状为 $n \times 1$。

⚠ 注意，很多参考文献中，向量一般记做 $\boldsymbol{a}$ 和 $\boldsymbol{b}$，投影结果一般记做 $\boldsymbol{u}$ 和 $\boldsymbol{v}$；但是本书 $\boldsymbol{u}$ 和 $\boldsymbol{v}$ 特指代表投影方向的向量，所以本章依然沿用这种记法。

展开 (3) 得到如下线性组合形式：

$$\boldsymbol{s}_1 = \begin{bmatrix} \boldsymbol{x}_1 & \boldsymbol{x}_2 & \cdots & \boldsymbol{x}_p \end{bmatrix} \begin{bmatrix} u_{1,1} \\ u_{2,1} \\ \vdots \\ u_{p,1} \end{bmatrix} = u_{1,1}\boldsymbol{x}_1 + u_{2,1}\boldsymbol{x}_2 + \cdots u_{p,1}\boldsymbol{x}_p \tag{4}$$

$\boldsymbol{Y}$ 朝向量 $\boldsymbol{v}_1$ 投影结果为 $\boldsymbol{t}_1$：

$$\boldsymbol{t}_1 = \boldsymbol{Y}_{n\times q}\boldsymbol{v}_1 \tag{5}$$

其中，$\boldsymbol{v}_1$ 的形状为 $q \times 1$，$\boldsymbol{t}_1$ 的形状为 $n \times 1$。p 和 q 可以不相等，也就是说 $\boldsymbol{u}_1$、$\boldsymbol{v}_1$ 形状可能不同。但是 $\boldsymbol{s}_1$、$\boldsymbol{t}_1$ 形状相同。

展开 (5) 得到如下线性组合形式：

$$t_1 = \begin{bmatrix} y_1 & y_2 & \cdots & y_q \end{bmatrix} \begin{bmatrix} v_{1,1} \\ v_{2,1} \\ \vdots \\ v_{q,1} \end{bmatrix} = v_{1,1} y_1 + v_{2,1} y_2 + \cdots v_{q,1} y_q \tag{6}$$

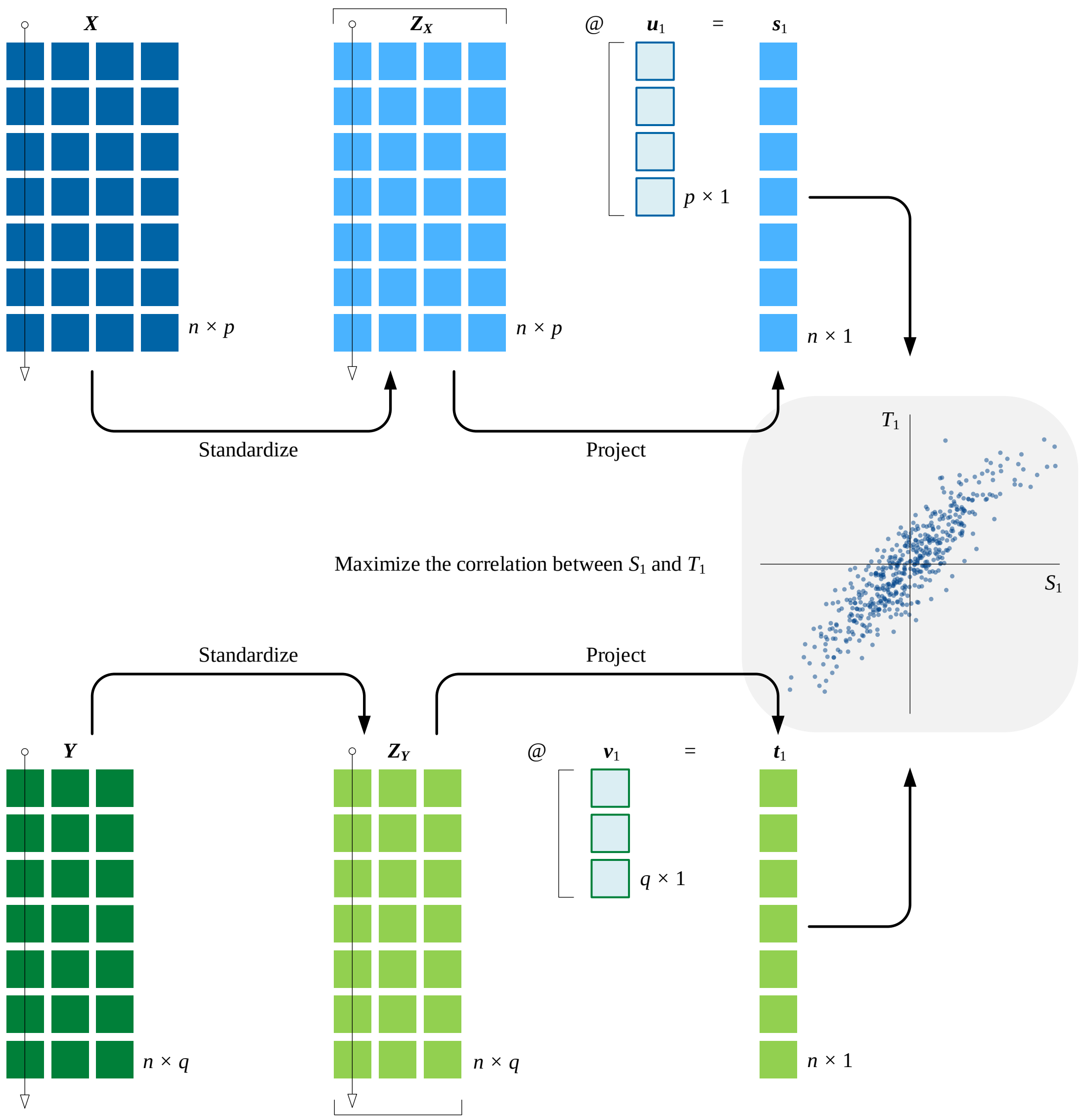

图 1. 典型相关分析原理

优化问题

如图 1 所示，典型相关分析 CCA 的问题便是找到 $\boldsymbol{u}_1$ 和 $\boldsymbol{v}_1$，使得 $\boldsymbol{s}_1$ 和 $\boldsymbol{t}_1$ 相关性最大。

⚠ 注意，如图 1 所示，从数据角度来看，一般情况 $\boldsymbol{X}$ 和 $\boldsymbol{Y}$ 都先经过标准化处理。

随机变量

用随机变量来写的话，S_1对应 $\boldsymbol{s}_1$，T_1对应 $\boldsymbol{t}_1$。随机变量 S_1可以写成如下线性变换：

$$S_1 = \boldsymbol{u}_1^{\mathrm{T}}\chi = \begin{bmatrix} u_{1,1} & u_{2,1} & \cdots & u_{p,1} \end{bmatrix} \begin{bmatrix} X_1 \\ X_2 \\ \vdots \\ X_p \end{bmatrix} = u_{1,1}X_1 + u_{2,1}X_2 + \cdots + u_{p,1}X_p \tag{7}$$

同理，随机变量 T_1可以写成：

$$T_1 = \boldsymbol{v}_1^{\mathrm{T}}\gamma = \begin{bmatrix} v_{1,1} & v_{2,1} & \cdots & v_{q,1} \end{bmatrix} \begin{bmatrix} Y_1 \\ Y_2 \\ \vdots \\ Y_q \end{bmatrix} = v_{1,1}Y_1 + v_{2,1}Y_2 + \cdots + v_{q,1}Y_q \tag{8}$$

S_1和 T_1是**第一对典型变量** (first pair of canonical variables)。

S_1和 T_1的相关性系数为：

$$\operatorname{corr}(S_1, T_1) = \frac{\operatorname{cov}(S_1, T_1)}{\sqrt{\operatorname{var}(S_1, S_1)}\sqrt{\operatorname{var}(T_1, T_1)}} \tag{9}$$

这样寻找第一对典型变量的优化问题可以写成：

$$\underset{\boldsymbol{u}_1, \boldsymbol{v}_1}{\arg\max} \operatorname{corr}(S_1, T_1) \tag{10}$$

有关随机变量的线性变换，请大家回顾《统计至简》第 14 章。

寻找更多典型变量

如图 2 所示，再找到第一对典型变量之后，依然最大化相关性系数可以找到**第二对典型变量** (second pair of canonical variables)。约束条件是第一、第二对典型变量不相关。

用向量来写，$\boldsymbol{s}_2$也是 $\begin{bmatrix} \boldsymbol{x}_1 & \boldsymbol{x}_2 & \cdots & \boldsymbol{x}_p \end{bmatrix}$ 的线性组合：

$$\boldsymbol{s}_2 = \boldsymbol{X}\boldsymbol{u}_2 = \begin{bmatrix} \boldsymbol{x}_1 & \boldsymbol{x}_2 & \cdots & \boldsymbol{x}_p \end{bmatrix} \begin{bmatrix} u_{1,2} \\ u_{2,2} \\ \vdots \\ u_{p,2} \end{bmatrix} = u_{1,1}\boldsymbol{x}_1 + u_{2,1}\boldsymbol{x}_2 + \cdots u_{p,1}\boldsymbol{x}_p \tag{11}$$

上式相当于 $\boldsymbol{X}$朝 $\boldsymbol{u}_2$投影。

$\boldsymbol{t}_2$为 $\begin{bmatrix} \boldsymbol{y}_1 & \boldsymbol{y}_2 & \cdots & \boldsymbol{y}_q \end{bmatrix}$ 的线性组合：

$$\boldsymbol{t}_2 = \begin{bmatrix} \boldsymbol{y}_1 & \boldsymbol{y}_2 & \cdots & \boldsymbol{y}_q \end{bmatrix} \begin{bmatrix} v_{1,2} \\ v_{2,2} \\ \vdots \\ v_{q,2} \end{bmatrix} = v_{1,2}\boldsymbol{y}_1 + v_{2,2}\boldsymbol{y}_2 + \cdots v_{q,2}\boldsymbol{y}_q \tag{12}$$

上式相当于 $\boldsymbol{Y}$朝 $\boldsymbol{v}_2$投影。

通过最大化的 $\boldsymbol{s}_2$和 $\boldsymbol{t}_2$相关性系数，可以找到第二对典型变量。这步优化问题的约束条件为：

$$\begin{aligned}\boldsymbol{u}_1^{\mathrm{T}}\boldsymbol{u}_2 = 0\\ \boldsymbol{v}_1^{\mathrm{T}}\boldsymbol{v}_2 = 0\\ \boldsymbol{u}_1^{\mathrm{T}}\boldsymbol{v}_2 = 0\\ \boldsymbol{v}_1^{\mathrm{T}}\boldsymbol{u}_2 = 0\end{aligned} \tag{13}$$

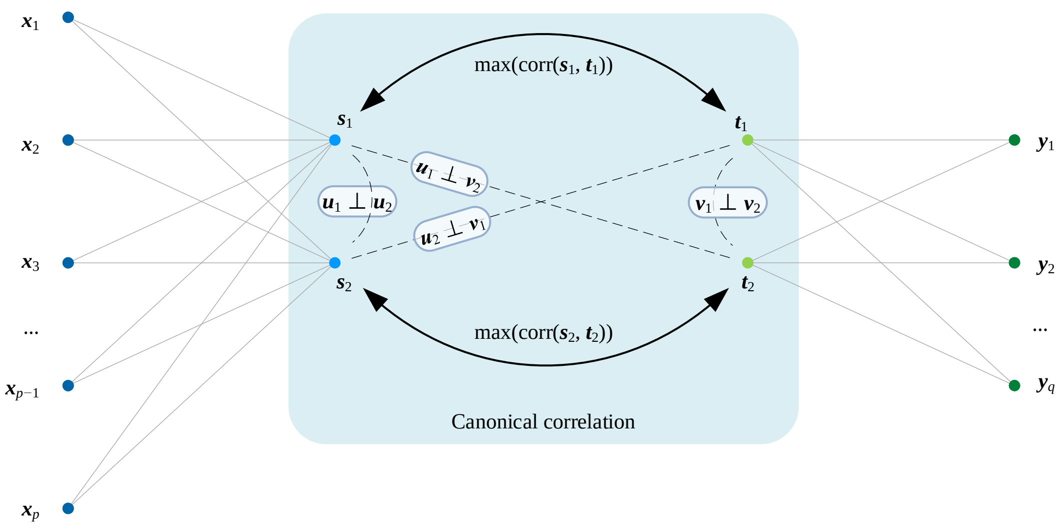

图 2. 线性组合角度看 CCA

随机变量 S_2 可以写成：

$$S_2 = \boldsymbol{u}_2^{\mathrm{T}}\boldsymbol{\chi} = \begin{bmatrix} u_{1,2} & u_{2,2} & \cdots & u_{p,2} \end{bmatrix}\begin{bmatrix} X_1 \\ X_2 \\ \vdots \\ X_p \end{bmatrix} = u_{1,2}X_1 + u_{2,2}X_2 + \cdots + u_{p,2}X_p \tag{14}$$

随机变量 T_2 可以写成：

$$T_2 = \boldsymbol{v}_2^{\mathrm{T}}\boldsymbol{\gamma} = \begin{bmatrix} v_{1,2} & v_{2,2} & \cdots & v_{q,2} \end{bmatrix}\begin{bmatrix} Y_1 \\ Y_2 \\ \vdots \\ Y_q \end{bmatrix} = v_{1,2}Y_1 + v_{2,2}Y_2 + \cdots + v_{p,2}Y_q \tag{15}$$

考虑到一般情况下 $\boldsymbol{X}$ 和 $\boldsymbol{Y}$ 已经标准化，$\mathrm{E}(\boldsymbol{X}) = \boldsymbol{0}$ 且 $\mathrm{E}(\boldsymbol{Y}) = \boldsymbol{0}$。这样 $\mathrm{E}(U_1) = 0$，$\mathrm{E}(V_1) = 0$。

这个步骤最多重复 $\min(p, q)$ 次，可以最多找到 $\min(p, q)$ 对典型变量。$\min(p, q)$ 对应 $\boldsymbol{X}$ 和 $\boldsymbol{Y}$ 的列数最小值。

19.2 从一个协方差矩阵考虑

$[\boldsymbol{X}, \boldsymbol{Y}]$ 的协方差矩阵可以按图 3 所示形式分成四个子块。$\boldsymbol{\Sigma}_{XX}$ 为 $\boldsymbol{X}$ 的协方差矩阵，$\boldsymbol{\Sigma}_{YY}$ 为 $\boldsymbol{Y}$ 的协方差矩阵，它俩都是方阵。$\boldsymbol{\Sigma}_{XY}$、$\boldsymbol{\Sigma}_{YX}$ 都是 $\boldsymbol{X}$、$\boldsymbol{Y}$ 的**互协方差矩阵** (cross-covariance matrix)，它俩互为转置。

《统计至简》第 13 章特别介绍过协方差矩阵分块，请大家回顾。

S_1 和 T_1 各自的方差、协方差为：

$$\begin{aligned}\operatorname{var}(S_1) &= \boldsymbol{u}_1^{\mathrm{T}} \boldsymbol{\Sigma}_{XX} \boldsymbol{u}_1 \\ \operatorname{var}(T_1) &= \boldsymbol{v}_1^{\mathrm{T}} \boldsymbol{\Sigma}_{YY} \boldsymbol{v}_1 \\ \operatorname{cov}(S_1, T_1) &= \boldsymbol{u}_1^{\mathrm{T}} \boldsymbol{\Sigma}_{XY} \boldsymbol{v}_1\end{aligned} \tag{16}$$

如果大家对上式概念模糊的话，请回顾《统计至简》第 14 章。

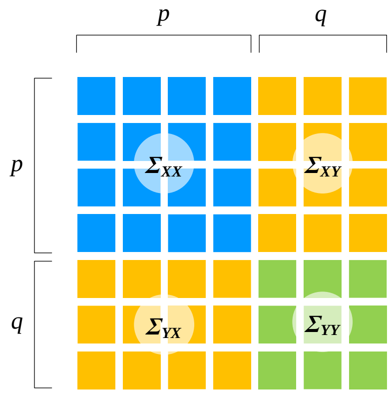

图 3. [$\boldsymbol{X}$, $\boldsymbol{Y}$] 的协方差矩阵分块

这样，(9) 的相关性系数可以写成：

$$\operatorname{corr}(S_1, T_1) = \frac{\boldsymbol{u}_1^{\mathrm{T}} \boldsymbol{\Sigma}_{XY} \boldsymbol{v}_1}{\sqrt{\boldsymbol{u}_1^{\mathrm{T}} \boldsymbol{\Sigma}_{XX} \boldsymbol{u}_1} \sqrt{\boldsymbol{v}_1^{\mathrm{T}} \boldsymbol{\Sigma}_{YY} \boldsymbol{v}_1}} \tag{17}$$

观察上式，大家是否发现它实际上是个**瑞利商** (Rayleigh quotient)。

我们在《矩阵力量》第 14 章了解过瑞利商。

优化结果

利用拉格朗日乘子法，我们可以求得优化问题的解。此处，省略推导过程，直接给出结果。

向量 $\boldsymbol{u}$ 是 $\boldsymbol{P} = \boldsymbol{\Sigma}_{XX}^{-1} \boldsymbol{\Sigma}_{XY} \boldsymbol{\Sigma}_{YY}^{-1} \boldsymbol{\Sigma}_{YX}$ 的特征向量。如图 4 所示，$\boldsymbol{P}$ 为 $p \times p$ 方阵。

向量 $\boldsymbol{v}$ 是 $\boldsymbol{Q} = \boldsymbol{\Sigma}_{YY}^{-1} \boldsymbol{\Sigma}_{YX} \boldsymbol{\Sigma}_{XX}^{-1} \boldsymbol{\Sigma}_{XY}$ 的特征向量。如图 5 所示，$\boldsymbol{Q}$ 为 $q \times q$ 方阵。

值得大家注意的是，如图 1 所示，一般 CCA 算法中，数据先要经过标准化处理。也就是说图 3 中真正参与运算的是相关性系数矩阵，而非协方差矩阵。

本章下面要使用的 sklearn.cross_decomposition.CCA() 函数就是先对数据标准化，再进行 CCA 分析。

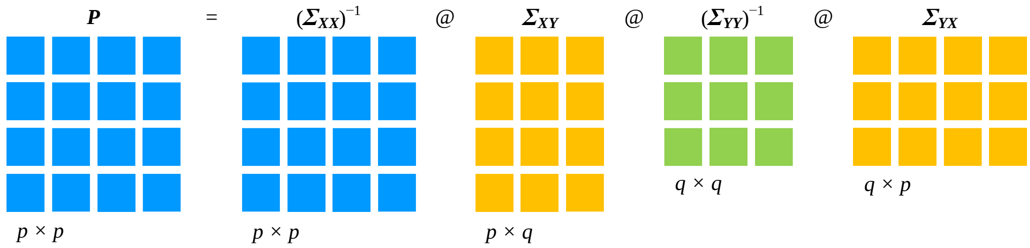

图 4. $\Sigma_{XX}^{-1}\Sigma_{XY}\Sigma_{YY}^{-1}\Sigma_{YX}$ 对应运算

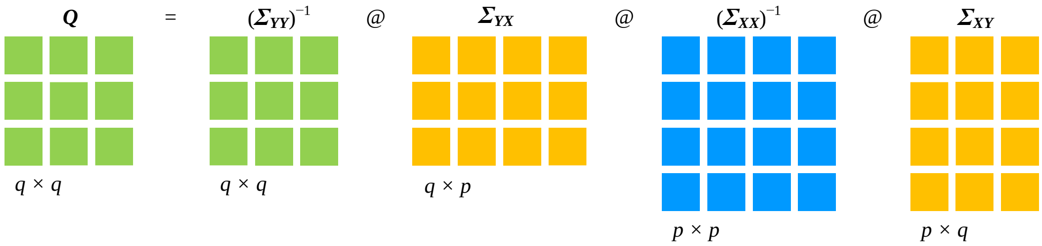

图 5. $\Sigma_{YY}^{-1}\Sigma_{YX}\Sigma_{XX}^{-1}\Sigma_{XY}$ 对应运算

19.3 以鸢尾花数据为例

本节以鸢尾花数据为例介绍如何完成典型相关分析。

如所示，我们把鸢尾花数据 4 列均分为 $\boldsymbol{X}$ 和 $\boldsymbol{Y}$ 两个矩阵。$\boldsymbol{X}$ 代表花萼 (长度、宽度)，$\boldsymbol{Y}$ 代表花瓣 (长度、宽度)。

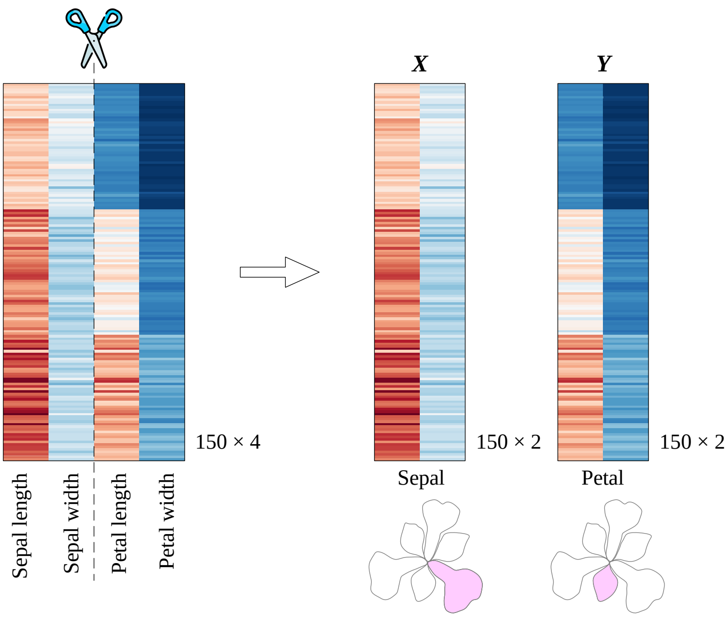

图 6. 把鸢尾花数据均分成两个子块

典型相关分析就是，将花萼数据 $\boldsymbol{X}$ 的两列合成一列 $\boldsymbol{s}_1$，将花瓣数据 $\boldsymbol{Y}$ 的两列合成一列 $\boldsymbol{t}_1$。通过合适的组合方式，让 $\boldsymbol{s}_1$ 和 $\boldsymbol{t}_1$ 的相关性最大。可以理解为找到花萼、花瓣之间“整体”关系。

图 7 所示为鸢尾花数据的相关性系数矩阵。请大家特别关注热图中黄色框高亮的两个子块，花萼和花瓣之间最大的相关性存在于花萼长度和花瓣长度 (0.87)。

比 0.87 更大的相关性系数是 0.96，这个相关性系数是花瓣长度、宽度之间的关系，而非花萼、花瓣之间的关系。

此外，CCA 分析中，图 7 的相关性系数矩阵就相当于图 3 的协方差矩阵。

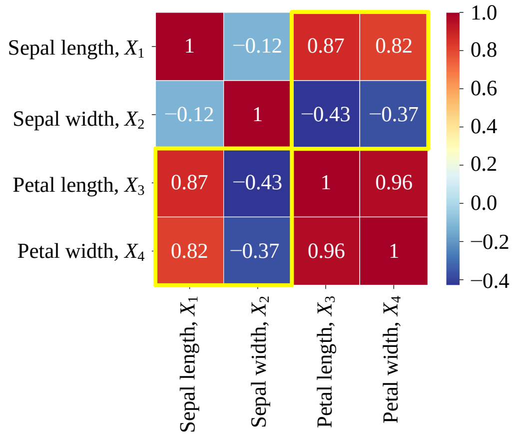

图 7. 鸢尾花数据的相关性系数矩阵

CCA 结果

通过 CCA 分析，我们得到的结果如图 8 (a) 所示。大家可以在本章代码中自行验算，可以发现图 8 (a) 中每一列均值均为 0。

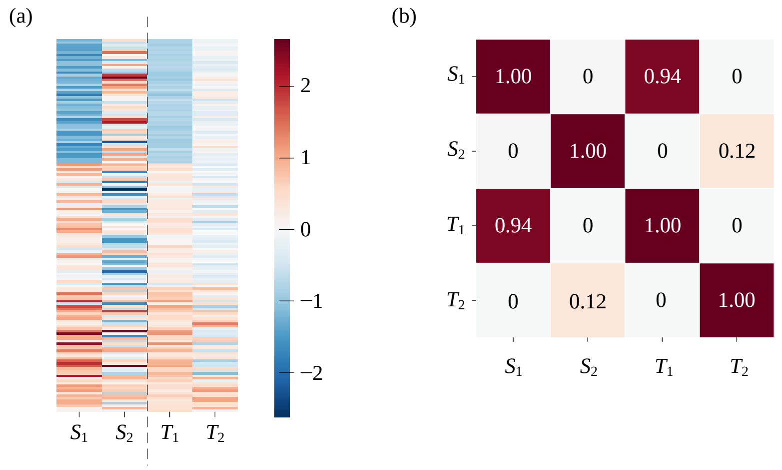

图 8. CCA 分析结果

图 8 (b) 所示为图 8 (a) 结果的相关性系数矩阵。S_1 和 T_1 的相关性系数达到 0.94。此外，大家发现图 8 (b) 中很多相关性系数为 0 的情况，这就是本章前文介绍的优化问题约束条件。

图 9 所示为用散点图可视化 S_1 和 T_1 的关系。图 9 (b) 还考虑了鸢尾花分类。观察图 9 (a)，大家可能已经发现 S_1 和 T_1 均方差明显不同。

图 10 所示为 CCA 结果成对特征散点图。

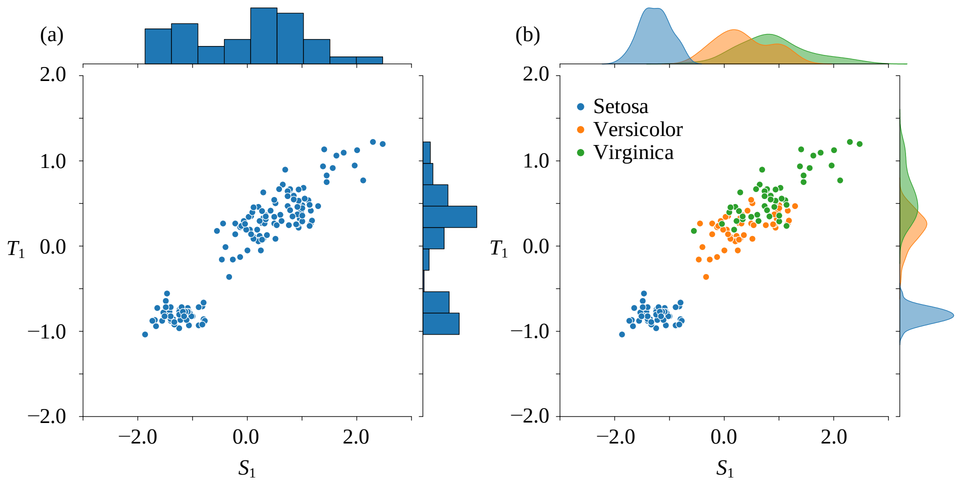

图 9. S_1 和 T_1 的散点图

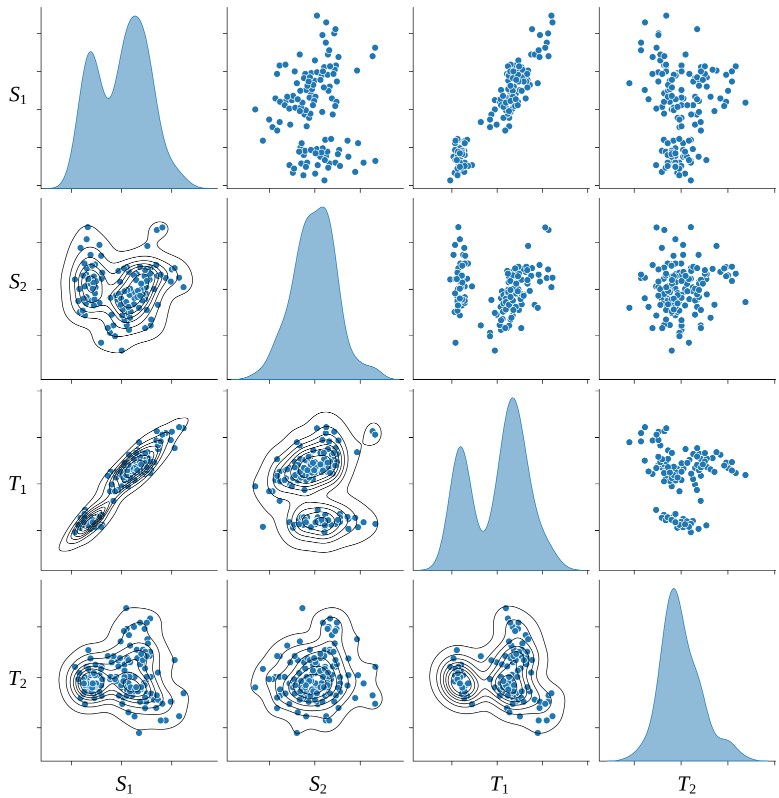

图 10. CCA 结果成对特征散点图

投影

大家可能会好奇到底怎样的 $\boldsymbol{u}_1$、$\boldsymbol{v}_1$ 让 S_1 和 T_1 的相关性系数如此之大？

sklearn.cross_decomposition.CCA() 函数同样返回 $\boldsymbol{u}_1$、$\boldsymbol{v}_1$，具体如图 11 所示。

图 11. CCA 投影向量结果

假设 $\boldsymbol{X} = [\boldsymbol{x}_1, \boldsymbol{x}_2]$ 已经标准化，$\boldsymbol{x}_1$ 和 $\boldsymbol{x}_2$ 按如下方式线性组合得到 $\boldsymbol{s}_1$：

$$\boldsymbol{s}_1 = \boldsymbol{X}_{150\times 2}\boldsymbol{u}_1 = \begin{bmatrix} \boldsymbol{x}_1 & \boldsymbol{x}_2 \end{bmatrix}\begin{bmatrix} 0.92 \\ -0.39 \end{bmatrix} = 0.92\boldsymbol{x}_1 - 0.39\boldsymbol{x}_2 \tag{18}$$

大家可以自己验证 $\boldsymbol{u}_1$ 为单位向量。

同样，假设 $\boldsymbol{Y} = [\boldsymbol{y}_1, \boldsymbol{y}_2]$ 已经标准化，$\boldsymbol{y}_1$ 和 $\boldsymbol{y}_2$ 按如下方式线性组合得到 $\boldsymbol{t}_1$：

$$\boldsymbol{t}_1 = \boldsymbol{Y}_{150\times 2}\boldsymbol{v}_1 = \begin{bmatrix} \boldsymbol{y}_1 & \boldsymbol{y}_2 \end{bmatrix}\begin{bmatrix} 0.94 \\ -0.33 \end{bmatrix} = 0.94\boldsymbol{x}_1 - 0.33\boldsymbol{x}_2 \tag{19}$$

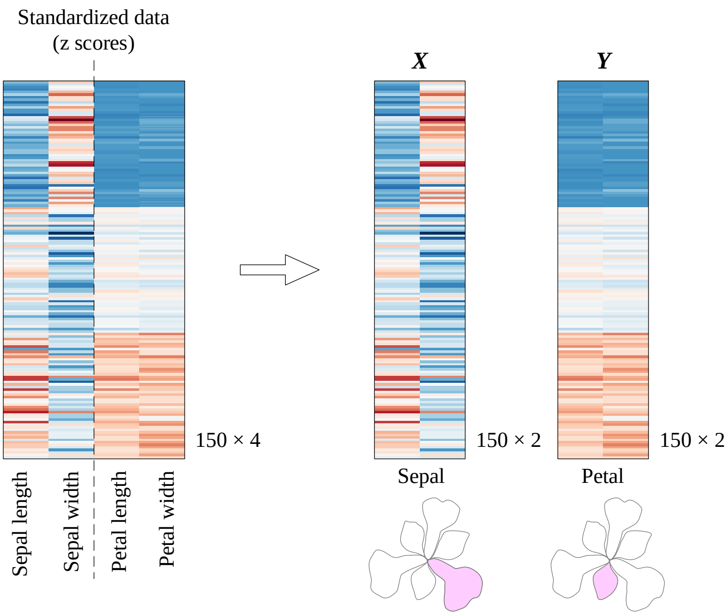

图 12. 标准化的鸢尾花数据

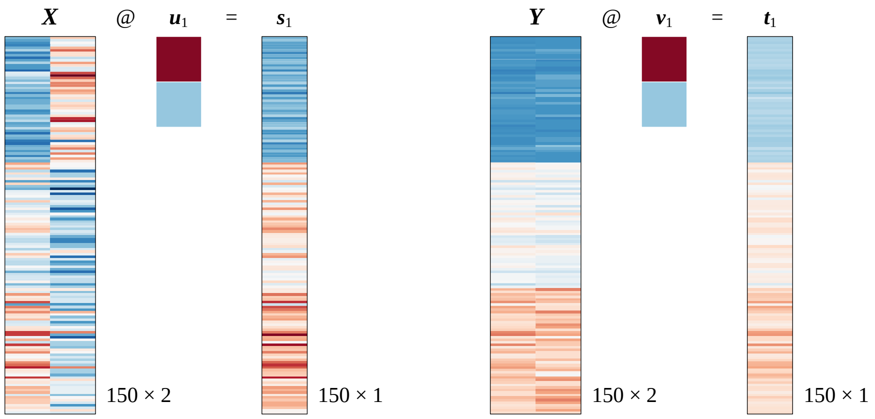

图 13. 通过投影计算 $\boldsymbol{s}_1$ 和 $\boldsymbol{t}_1$

特征值分解

下面我们利用特征值分解自行求解 $\boldsymbol{u}_1$、$\boldsymbol{v}_1$。根据图 4 和图 5，我们先需要计算 $\boldsymbol{P}$ 和 $\boldsymbol{Q}$ 两个方阵。具体过程如图 14、图 15 所示。

$$\underbrace{\begin{bmatrix} 0.73 & -0.37 \\ -0.30 & 0.17 \end{bmatrix}}_{\boldsymbol{P}} = \underbrace{\begin{bmatrix} 1.01 & 0.12 \\ 0.12 & 1.01 \end{bmatrix}}_{(\boldsymbol{\Sigma}_{XX})^{-1}} @ \underbrace{\begin{bmatrix} 0.87 & 0.82 \\ -0.43 & -0.37 \end{bmatrix}}_{\boldsymbol{\Sigma}_{XY}} @ \underbrace{\begin{bmatrix} 13.72 & -13.21 \\ -13.21 & 13.72 \end{bmatrix}}_{(\boldsymbol{\Sigma}_{YY})^{-1}} @ \underbrace{\begin{bmatrix} 0.87 & -0.43 \\ 0.82 & -0.37 \end{bmatrix}}_{\boldsymbol{\Sigma}_{YX}}$$

图 14. 计算矩阵 $\boldsymbol{P}$

$$\underbrace{\begin{bmatrix} 1.31 & 1.19 \\ -0.46 & -0.41 \end{bmatrix}}_{\boldsymbol{Q}} = \underbrace{\begin{bmatrix} 13.72 & -13.21 \\ -13.21 & 13.72 \end{bmatrix}}_{(\boldsymbol{\Sigma}_{YY})^{-1}} @ \underbrace{\begin{bmatrix} 0.87 & -0.43 \\ 0.82 & -0.37 \end{bmatrix}}_{\boldsymbol{\Sigma}_{YX}} @ \underbrace{\begin{bmatrix} 1.01 & 0.12 \\ 0.12 & 1.01 \end{bmatrix}}_{(\boldsymbol{\Sigma}_{XX})^{-1}} @ \underbrace{\begin{bmatrix} 0.87 & 0.82 \\ -0.43 & -0.37 \end{bmatrix}}_{\boldsymbol{\Sigma}_{XY}}$$

图 15. 计算矩阵 $\boldsymbol{Q}$

然后对 $\boldsymbol{P}$ 和 $\boldsymbol{Q}$ 分别进行特征值分解，具体如图 16、图 17 所示。

注意，图 17 中矩阵 $\boldsymbol{V}$ 的第 2 列向量 $\boldsymbol{v}_2$ 和图 11 中不同，但是两者为倍数关系，即共线。

$$\underbrace{\begin{bmatrix} 0.73 & -0.37 \\ -0.30 & 0.17 \end{bmatrix}}_{\boldsymbol{P}} = \overbrace{\begin{array}{cc} \boldsymbol{u}_1 & \boldsymbol{u}_2 \end{array}}^{\boldsymbol{U}} \begin{bmatrix} 0.92 & 0.46 \\ -0.39 & 0.89 \end{bmatrix} @ \underbrace{\begin{bmatrix} 0.89 & 0 \\ 0 & 0.02 \end{bmatrix}}_{\boldsymbol{\Lambda}_P} @ \underbrace{\begin{bmatrix} 0.89 & -0.46 \\ 0.39 & 0.92 \end{bmatrix}}_{\boldsymbol{U}^{-1}}$$

图 16. 矩阵 P 特征值分解

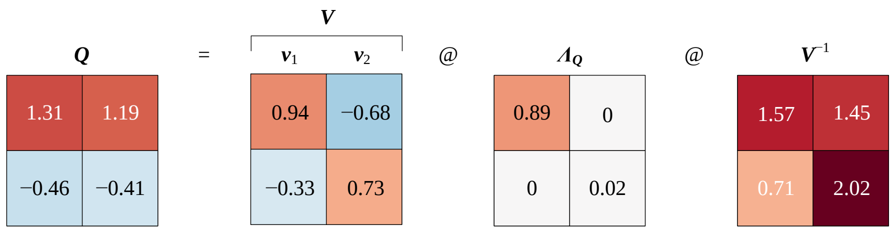

图 17. 矩阵 Q 特征值分解

Bk7_Ch19_01.ipynb 完成本章 CCA 分析及可视化。下面聊聊其中关键语句。

ⓐ导入鸢尾花数据。

ⓑ取出花萼两个特征数据。ⓒ取出花瓣两个特征数据。

ⓓ用 `sklearn.cross_decomposition.CCA()`创建一个 CCA 对象，指定要保留的主成分数为 2。

ⓔ使用 `fit()` 方法拟合模型，将花萼特征 (X) 和花瓣特征 (Y) 传递给 CCA 模型。

ⓕ使用 `transform()` 方法将原始数据投影到 CCA 空间，得到投影后的数据 S 和 T。

Bk7_Ch19_01.ipynb 这段代码还复刻了上述 CCA 运算，请大家自行学习。

```python
from sklearn.cross_decomposition import CCA
from sklearn.datasets import load_iris

# 导入鸢尾花数据
iris_sns = sns.load_dataset("iris")              # a

X_df = iris_sns[['sepal_length', 'sepal_width',
                 'petal_length', 'petal_width']]

# 花萼两个特征
X = iris_sns[['sepal_length', 'sepal_width']]    # b

# 花瓣两个特征
Y = iris_sns[['petal_length', 'petal_width']]    # c

# CCA分析
Iris_CCA = CCA(n_components=2)                   # d
Iris_CCA.fit(X, Y)                               # e
S, T = Iris_CCA.transform(X, Y)                  # f

# 整理结果
S_T_df = pd.DataFrame({"s1":S[:, 0],
                       "s2":S[:, 1],
                       "t1":T[:, 0],
                       "t2":T[:, 1]})
```

代码 1. 利用 `sklearn.cross_decomposition.CCA()`完成典型相关分析 | Bk7_Ch19_01.ipynb

至此，我们完成了本书所有有关“降维”算法的学习。请大家务必掌握六种不同主成分的异同，以及经济型 SVD 分解、截断型 SVD 分解。

另外，大家需要了解 PCA 算法的局限性。对于非线性数据降维，大家可以试着用核 PCA；核 PCA 将非线性数据投影到高维度空间，再投影。也请大家自行学习流形学习等其他降维算法。