

面向时空交通数据修复及预测的低秩机器学习模型

陈新宇 (<https://xinychen.github.io>)

伍元凯 (<https://kaimaoge.github.io>)

赵熙乐 (<https://zhaoxile.github.io>)

孙立君 (<https://lijunsun.github.io>)

发布时间：2023 年 2 月

更新时间：2023 年 6 月

本文唯一下载网址：

https://xinychen.github.io/books/spatiotemporal_low_rank_models.pdf

目录

第一章 研究进展综述	9
第二章 代数结构与张量计算基础	11
2.1 基本代数结构	11
2.1.1 向量与矩阵	11
2.1.2 高阶张量	13
2.1.3 高阶张量矩阵化	15
2.1.4 高阶张量向量化	17
2.2 常见范数	17
2.2.1 向量范数	17
2.2.2 矩阵的 Frobenius 范数	18
2.3 求导数	18
2.3.1 从微积分到范数的求导数	18
2.4 延伸阅读	19
第三章 时空交通数据类型	21
第四章 张量分解基础	23
4.1 Kronecker 积定义	23
4.1.1 基本定义	23
4.1.2 Khatri-Rao 积	25
4.2 Kronecker 积基本性质	25
4.2.1 结合律与分配律	25
4.2.2 矩阵相乘	26
4.2.3 求逆矩阵	27
4.2.4 向量化	28
4.3 Kronecker 积特殊性质	29
4.3.1 矩阵的迹	29
4.3.2 矩阵的 Frobenius 范数	30
4.3.3 矩阵的行列式	31
4.3.4 矩阵的秩	31
4.4 向量外积	32
4.4.1 定义	32
4.4.2 性质	33
4.5 CP 张量分解	34
4.5.1 CP 分解形式	34

4.5.2	低秩逼近问题	35
第五章	时序矩阵分解	37
5.1	矩阵分解	37
5.1.1	模型表达式	37
5.1.2	梯度下降法	38
5.1.3	最速梯度下降法	38
5.1.4	交替优化算法	40
5.2	平滑矩阵分解	42
5.2.1	模型表达式	42
5.2.2	求解过程	43
5.3	时序矩阵分解	48
5.3.1	模型表达式	48
5.3.2	求解过程	49
5.3.3	时间序列预测	50
第六章	贝叶斯张量分解	53
6.1	基于高斯假设的 CP 分解表达式	53
6.2	贝叶斯网络	54
6.2.1	共轭分布	54
6.2.2	因子矩阵的先验分布	55
6.2.3	精度参数的先验分布	56
6.2.4	张量分解的贝叶斯网络	56
6.3	Gibbs 采样与后验推断	57
6.3.1	Gibbs 采样	57
6.3.2	因子矩阵的后验分布	57
6.3.3	超参数的后验分布	60
6.3.4	精度参数的后验分布	62
6.4	算法实现	63
6.5	Python 实现代码	64
6.5.1	贝叶斯高斯张量分解算法的函数	64
6.5.2	实验: 时空交通数据修复	66
第七章	低秩张量填充	67
7.1	基于核范数最小化的矩阵填充	67
7.1.1	核范数最小化	67
7.1.2	截断核范数最小化	67
7.1.3	加权核范数最小化	67
7.2	基于多重核范数的张量填充	67
7.3	基于多重截断核范数的张量填充	67
7.4	基于张量核范数的张量填充	67
第八章	低秩拉普拉斯卷积模型	69
8.1	循环卷积与离散傅立叶变换	69
8.1.1	循环卷积定义	69
8.1.2	一维卷积定理	70

8.1.3	二维卷积定理	71
8.1.4	Parseval 定理	72
8.2	离散傅立叶变换与循环矩阵核范数	73
8.2.1	循环矩阵定义	73
8.2.2	循环矩阵核范数	74
8.2.3	ℓ_1 范数最小化问题	75
8.2.4	循环矩阵核范数最小化问题	75
8.3	低秩拉普拉斯卷积模型	78
8.3.1	拉普拉斯卷积核	78
8.3.2	拉普拉斯时序正则	78
8.3.3	一维低秩拉普拉斯卷积模型	79
8.3.4	二维低秩拉普拉斯卷积模型	81
8.4	Python 实现代码	81
第九章 基于 Hankel 结构的张量分解		83
第十章 低秩深度学习时空预测模型		85
附录 A 公开交通数据集		87
A.1	波特兰高速公路交通流量数据集	87
A.2	西雅图高速公路交通速度数据集	87
附录 B 算法详解		89
B.1	共轭梯度法	89
B.1.1	梯度下降法	89
B.1.2	最速梯度下降法	90
B.1.3	共轭梯度下降法	92

插图

2.1	矩阵 $\mathbf{X} \in \mathbb{R}^{m \times n}$ 及其第 (i, j) 个元素 $x_{i,j}$	12
2.2	三阶张量 $\mathcal{X} \in \mathbb{R}^{m \times n \times t}$ 及其第 (i, j, k) 个元素 $x_{i,j,k}$	13
2.3	三阶张量 $\mathcal{X} \in \mathbb{R}^{m \times n \times t}$ 自三个维度的纤维	14
2.4	三阶张量 $\mathcal{X} \in \mathbb{R}^{m \times n \times t}$ 自三个维度的切片	15
4.1	向量外积得到的三阶张量 $\mathcal{Y} \in \mathbb{R}^{n_1 \times n_2 \times n_3}$	32
4.2	三阶张量 $\mathcal{Y} \in \mathbb{R}^{n_1 \times n_2 \times n_3}$ 的 CP 分解	35
5.1	矩阵分解示意图	38
5.2	NGSIM 数据集中的速度场	39
5.3	使用梯度下降法求解矩阵分解所得到的重构速度场	39
5.4	使用最速梯度下降法求解矩阵分解所得到的重构速度场	40
5.5	使用交替最小二乘法求解矩阵分解所得到的重构速度场	41
5.6	在速度场数据集上, 三种迭代法所对应的矩阵分解目标函数变化曲线	42
5.7	在西雅图高速公路车速数据集上, 三种迭代法所对应的矩阵分解目标函数变化 曲线	42
5.8	平滑矩阵分解示意图	43
5.9	使用共轭梯度法求解二元一次方程组	45
5.10	平滑矩阵分解所得到的重构速度场	48
6.1	三阶张量 $\mathcal{Y} \in \mathbb{R}^{M \times N \times T}$ 的 CP 分解	53
6.2	贝叶斯高斯张量分解的贝叶斯网络	56
8.1	循环矩阵示意图	73
8.2	基于关系型数据 $\{x_1, x_2, \dots, x_5\}$ 的无向循环图	78
8.3	低秩拉普拉斯卷积模型的示意图	80
8.4	某高速公路断面交通流的车速时间序列, 其中, 蓝色曲线表示车速时间序列; 蓝 色圆圈表示抽取的部分观测值。	81
8.5	基于一维低秩拉普拉斯卷积模型的 90% 缺失率的车速时间序列重构, 其中, 红 色曲线表示重构出来的车速时间序列。	82

第一章 研究进展综述

时空交通数据建模极具现实意义，可支撑数据驱动的智能交通系统建设。

在智能交通系统中，随着数据获取途径的多元化，可按以下几个层面进行系统建设：

- 数据层。采集数据后，需进行数据质量分析、数据隐私分析以及数据价值化分析。
- 场景层。针对特定数据或融合数据，构建现实需求导向的应用场景以及定义具有现实意义的科学问题。例如，针对短时交通流预测，可建立数据驱动模型；针对长期出行需求预测，可建立精细化出行目的为导向的宏观模型，发挥多源数据融合的价值。
- 算法层。提出具体数据建模问题的解决方案，形成有效的学习机制。
- 决策层。构建数据驱动的决策分析。
- 生态层。针对关键问题，搭建可持续的技术环境，并形成良好的技术开源生态，例如 PeMS 系统与 Uber Movement 项目。

第二章 代数结构与张量计算基础

在线性代数中，我们所熟知的代数结构是向量与矩阵，作为基本代数结构，向量与矩阵构建了严谨的代数分析体系，并对机器学习领域产生了深远的影响。本章将首先回顾向量与矩阵这两种代数结构，以此为基础，引申出高阶张量这种更为复杂的代数结构。为了衔接代数结构与实际机器学习建模任务，我们会介绍一些面向向量、矩阵与张量的范数，同时将微积分中的求导数概念引入矩阵计算与张量计算，详解一些求导数的基本规则。

2.1 基本代数结构

长期以来，线性代数一直作为机器学习中最为核心的数学工具之一，被人们广泛用于开发各类机器学习算法。线性代数本质上是以向量与矩阵为基本代数结构，本文要讨论的张量分解等模型则主要以张量为基本代数结构。在过去的数十年间，借助线性代数这一基本数学工具，机器学习中涌现出了很多经典的代数模型，这其中不乏矩阵分解、主成分分析，而张量分解在某种程度上可看作是矩阵分解在高维的一种衍生物。

近年来，张量分解在机器学习的众多问题中得到了很好的应用 [Kolda and Bader, 2009, Sidiropoulos et al., 2017]，但关于张量的一些计算与我们所熟悉的线性代数却大相径庭，同时，张量计算相比以矩阵计算为主导的线性代数更为抽象，这使得很多与张量分解相关的内容看起来晦涩难懂。实际上，向量与矩阵都是张量的特例，可以被定义为低阶张量。一般而言，向量是一阶张量，英文表述为 first-order tensor；矩阵是二阶张量，英文表述为 second-order tensor；三阶或者更高阶数的张量被称为高阶张量，英文表述为 higher-order tensor。在各类文献中，通常提到的张量都是特指高阶张量，当然，这在本文的叙述中也不例外。需要注意的是，在各类程序语言中，人们更愿意将张量称为多维数组。

在一个矩阵中，某一元素的位置可以说是“第 i 行、第 j 列”，即要描述某一元素的位置需用到行和列索引构成的组合 (i, j) 。类似地，在一个三阶张量中，描述某一元素的位置需用到三个索引构成的组合，例如 (i, j, k) 。在处理稀疏矩阵或稀疏张量时，用索引来标记元素的位置会节省下一些不必要的存储开支。

2.1.1 向量与矩阵

向量

向量包括行向量与列向量。在写法上，为避免混淆，向量在没有特别声明的情况下是指列向量，给定任意向量 $\mathbf{x} \in \mathbb{R}^n$ 表示大小为 n 的向量，写作

$$\mathbf{x} = (x_1, x_2, \dots, x_n)^\top \quad (2.1)$$

或

$$\mathbf{x} = \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix} \quad (2.2)$$

其中, 符号 \cdot^\top 表示转置 (transpose)。

矩阵

一般而言, 给定任意矩阵 $\mathbf{X} \in \mathbb{R}^{m \times n}$, 矩阵的行数为 m 、列数为 n , 即

$$\mathbf{X} = \begin{bmatrix} x_{1,1} & x_{1,2} & \cdots & x_{1,n} \\ x_{2,1} & x_{2,2} & \cdots & x_{2,n} \\ \vdots & \vdots & \ddots & \vdots \\ x_{m,1} & x_{m,2} & \cdots & x_{m,n} \end{bmatrix} \quad (2.3)$$

其第 (i, j) 个元素 (即矩阵的第 i 行、第 j 列元素) 为

$$x_{i,j} = \mathbf{X}_{i,j} \quad (2.4)$$

其中, $i = 1, 2, \dots, m$ 与 $j = 1, 2, \dots, n$ 。

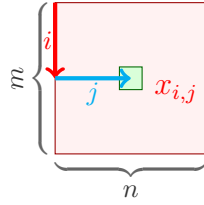


图 2.1: 矩阵 $\mathbf{X} \in \mathbb{R}^{m \times n}$ 及其第 (i, j) 个元素 $x_{i,j}$

单位矩阵一般记作 \mathbf{I}_n , 大小为 $n \times n$, 其对角线上的元素均为 1、其他位置上的元素均为 0。

矩阵向量化

给定任意矩阵 $\mathbf{X} \in \mathbb{R}^{m \times n}$, 若矩阵的列向量为 $\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_n \in \mathbb{R}^m$, 即

$$\mathbf{X} = \begin{bmatrix} | & | & \cdots & | \\ \mathbf{x}_1 & \mathbf{x}_2 & \cdots & \mathbf{x}_n \\ | & | & \cdots & | \end{bmatrix} \quad (2.5)$$

则可对矩阵按列进行向量化, 得到的向量为

$$\text{vec}(\mathbf{X}) = \begin{bmatrix} \mathbf{x}_1 \\ \mathbf{x}_2 \\ \vdots \\ \mathbf{x}_n \end{bmatrix} \in \mathbb{R}^{mn} \quad (2.6)$$

其中, 符号 $\text{vec}(\cdot)$ 表示向量化操作。与矩阵向量化相反, 也可定义向量的矩阵化规则。

例 1. 给定矩阵 $\mathbf{X} = \begin{bmatrix} 1 & 2 \\ 3 & 4 \end{bmatrix}$ 与 $\mathbf{Y} = \begin{bmatrix} 5 & 6 & 7 \\ 8 & 9 & 10 \end{bmatrix}$, 试分别写出两个矩阵的向量化形式。

解. 根据矩阵向量化操作的定义, 有

$$\text{vec}(\mathbf{X}) = \begin{bmatrix} \mathbf{x}_1 \\ \mathbf{x}_2 \end{bmatrix} = \begin{bmatrix} 1 \\ 3 \\ 2 \\ 4 \end{bmatrix} = (1, 3, 2, 4)^\top \quad (2.7)$$

其中, 矩阵 \mathbf{X} 的列向量为

$$\mathbf{x}_1 = \begin{bmatrix} 1 \\ 3 \end{bmatrix} \quad \mathbf{x}_2 = \begin{bmatrix} 2 \\ 4 \end{bmatrix} \quad (2.8)$$

矩阵 \mathbf{Y} 的向量化为

$$\text{vec}(\mathbf{Y}) = \begin{bmatrix} \mathbf{y}_1 \\ \mathbf{y}_2 \\ \mathbf{y}_3 \end{bmatrix} = \begin{bmatrix} 5 \\ 8 \\ 6 \\ 9 \\ 7 \\ 10 \end{bmatrix} = (5, 8, 6, 9, 7, 10)^\top \quad (2.9)$$

其中, 矩阵 \mathbf{Y} 的列向量为

$$\mathbf{y}_1 = \begin{bmatrix} 5 \\ 8 \end{bmatrix} \quad \mathbf{y}_2 = \begin{bmatrix} 6 \\ 9 \end{bmatrix} \quad \mathbf{y}_3 = \begin{bmatrix} 7 \\ 10 \end{bmatrix} \quad (2.10)$$

2.1.2 高阶张量

一般而言, 高阶张量可写成 $\mathcal{X} \in \mathbb{R}^{m_1 \times m_2 \times \cdots \times m_d}$, 张量的阶数为 d , 大小为 $m_1 \times m_2 \times \cdots \times m_d$ 。

三阶张量中的元素

这里以三阶张量为例, 给定任意三阶张量 $\mathcal{X} \in \mathbb{R}^{m \times n \times t}$, 其第 (i, j, k) 个元素可写作如下形式:

$$x_{i,j,k} = \mathcal{X}_{i,j,k} \quad (2.11)$$

其中, $i = 1, 2, \dots, m$ 、 $j = 1, 2, \dots, n$ 与 $k = 1, 2, \dots, t$ 。

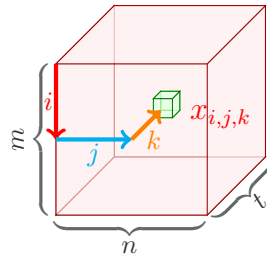


图 2.2: 三阶张量 $\mathcal{X} \in \mathbb{R}^{m \times n \times t}$ 及其第 (i, j, k) 个元素 $x_{i,j,k}$

图2.2直观地展现了三阶张量元素的示意图, 可以看出: 描述三阶张量中的某一元素需用到三个索引构成的组合, 例如 (i, j, k) 。

三阶张量中的纤维

给定任意三阶张量 $\mathcal{X} \in \mathbb{R}^{m \times n \times t}$, 其各个方向的纤维 (fiber) 都是向量, 如图2.3所示, 这些纤维分别为向量 $\mathcal{X}_{:,j,k} \in \mathbb{R}^m$ 、 $\mathcal{X}_{i,:,k} \in \mathbb{R}^n$ 与 $\mathcal{X}_{i,j,:} \in \mathbb{R}^t$, 其中, $i = 1, 2, \dots, m$ 、 $j = 1, 2, \dots, n$ 与 $k = 1, 2, \dots, t$. 与矩阵中的行向量、列向量类似, 纤维是张量的基本组成部分。

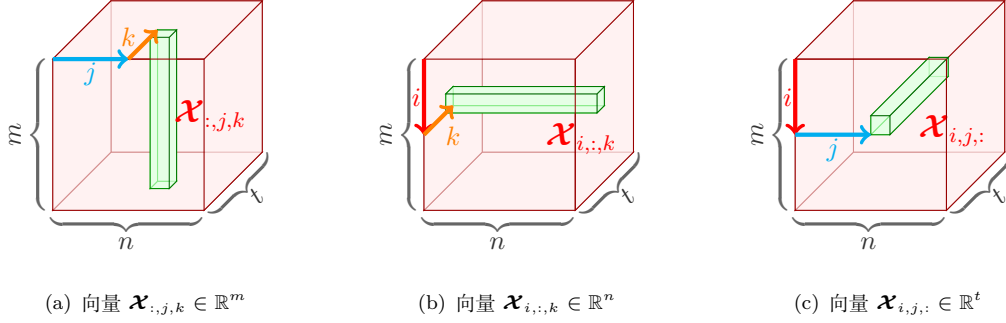


图 2.3: 三阶张量 $\mathcal{X} \in \mathbb{R}^{m \times n \times t}$ 自三个维度的纤维

具体而言, 这些纤维可写作如下形式:

$$\mathcal{X}_{:,j,k} = \begin{bmatrix} x_{1,j,k} \\ x_{2,j,k} \\ \vdots \\ x_{m,j,k} \end{bmatrix} \in \mathbb{R}^m \quad \mathcal{X}_{i,:,k} = \begin{bmatrix} x_{i,1,k} \\ x_{i,2,k} \\ \vdots \\ x_{i,n,k} \end{bmatrix} \in \mathbb{R}^n \quad \mathcal{X}_{i,j,:} = \begin{bmatrix} x_{i,j,1} \\ x_{i,j,2} \\ \vdots \\ x_{i,j,t} \end{bmatrix} \in \mathbb{R}^t \quad (2.12)$$

三阶张量中的切片

对于任意三阶张量 $\mathcal{X} \in \mathbb{R}^{m \times n \times t}$, 可用三个维度的切片 (slice) 书写该张量, 其中, horizontal 切片共有 m 个, 分别为

$$\mathcal{X}_{1,:,:), \mathcal{X}_{2,:,:), \dots, \mathcal{X}_{m,:,:) \in \mathbb{R}^{n \times t} \quad (2.13)$$

具体而言, 任意第 i 个 horizontal 切片可写作如下形式:

$$\mathcal{X}_{i,:,:) = \begin{bmatrix} x_{i,1,1} & x_{i,1,2} & \cdots & x_{i,1,t} \\ x_{i,2,1} & x_{i,2,2} & \cdots & x_{i,2,t} \\ \vdots & \vdots & \ddots & \vdots \\ x_{i,n,1} & x_{i,n,2} & \cdots & x_{i,n,t} \end{bmatrix}, \forall i \in \{1, 2, \dots, m\} \quad (2.14)$$

lateral 切片共有 n 个, 分别为

$$\mathcal{X}_{:,1,:), \mathcal{X}_{:,2,:), \dots, \mathcal{X}_{:,n,:) \in \mathbb{R}^{m \times t} \quad (2.15)$$

具体而言, 任意第 j 个 lateral 切片可写作如下形式:

$$\mathcal{X}_{:,j,:) = \begin{bmatrix} x_{1,j,1} & x_{1,j,2} & \cdots & x_{1,j,t} \\ x_{2,j,1} & x_{2,j,2} & \cdots & x_{2,j,t} \\ \vdots & \vdots & \ddots & \vdots \\ x_{m,j,1} & x_{m,j,2} & \cdots & x_{m,j,t} \end{bmatrix}, \forall j \in \{1, 2, \dots, n\} \quad (2.16)$$

frontal 切片共有 t 个, 分别为

$$\mathcal{X}_{:,:,1}, \mathcal{X}_{:,:,2}, \dots, \mathcal{X}_{:,:,t} \in \mathbb{R}^{m \times n} \quad (2.17)$$

具体而言, 任意第 k 个 frontal 切片可写作如下形式:

$$\mathcal{X}_{:, :, k} = \begin{bmatrix} x_{1,1,k} & x_{1,2,k} & \cdots & x_{1,n,k} \\ x_{2,1,k} & x_{2,2,k} & \cdots & x_{2,n,k} \\ \vdots & \vdots & \ddots & \vdots \\ x_{m,1,k} & x_{m,2,k} & \cdots & x_{m,n,k} \end{bmatrix}, \forall k \in \{1, 2, \dots, t\} \quad (2.18)$$

如图2.4所示, 这些矩阵结构的切片是张量的基本组成部分。

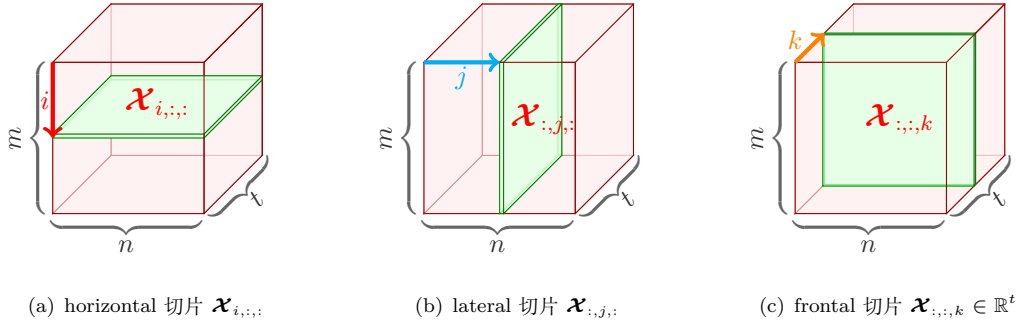


图 2.4: 三阶张量 $\mathcal{X} \in \mathbb{R}^{m \times n \times t}$ 自三个维度的切片

例 2. 给定张量 $\mathcal{X} \in \mathbb{R}^{2 \times 2 \times 2}$, 若其 *frontal* 切片为

$$\mathcal{X}_{:, :, 1} = \begin{bmatrix} x_{111} & x_{121} \\ x_{211} & x_{221} \end{bmatrix} = \begin{bmatrix} 1 & 2 \\ 3 & 4 \end{bmatrix} \quad \mathcal{X}_{:, :, 2} = \begin{bmatrix} x_{112} & x_{122} \\ x_{212} & x_{222} \end{bmatrix} = \begin{bmatrix} 5 & 6 \\ 7 & 8 \end{bmatrix} \quad (2.19)$$

试写出张量 \mathcal{X} 的 *lateral* 切片与 *horizontal* 切片。

解. 张量 \mathcal{X} 的 *lateral* 切片为

$$\mathcal{X}_{:, 1, :} = \begin{bmatrix} x_{111} & x_{112} \\ x_{211} & x_{212} \end{bmatrix} = \begin{bmatrix} 1 & 5 \\ 3 & 7 \end{bmatrix} \quad \mathcal{X}_{:, 2, :} = \begin{bmatrix} x_{121} & x_{122} \\ x_{221} & x_{222} \end{bmatrix} = \begin{bmatrix} 2 & 6 \\ 4 & 8 \end{bmatrix} \quad (2.20)$$

张量 \mathcal{X} 的 *horizontal* 切片为

$$\mathcal{X}_{1, :, :} = \begin{bmatrix} x_{111} & x_{112} \\ x_{121} & x_{122} \end{bmatrix} = \begin{bmatrix} 1 & 5 \\ 2 & 6 \end{bmatrix} \quad \mathcal{X}_{2, :, :} = \begin{bmatrix} x_{211} & x_{212} \\ x_{221} & x_{222} \end{bmatrix} = \begin{bmatrix} 3 & 7 \\ 4 & 8 \end{bmatrix} \quad (2.21)$$

2.1.3 高阶张量矩阵化

给定任意张量 $\mathcal{X} \in \mathbb{R}^{m_1 \times m_2 \times \cdots \times m_d}$, 阶数为 d , 若其自第 k 维度展开得到的矩阵为 $\mathbf{X}_{(k)}$, 大小为 $n_k \times \prod_{h \neq k} n_h$, 则张量 \mathcal{X} 的第 (i_1, i_2, \dots, i_d) 个元素对应着矩阵 $\mathbf{X}_{(k)}$ 的第 (i_k, j_k) 个元素:

$$j_k = 1 + \sum_{h=1, h \neq k}^d (i_h - 1) J_h \quad (2.22)$$

其中, $J_h = \prod_{l=1, l \neq k}^{h-1} n_l$. 张量展开的过程往往也被称为张量矩阵化 (matricization).

为更容易理解张量展开的规则, 不妨以大小为 $3 \times 4 \times 2$ 的张量 \mathcal{X} 为例 [Kolda and Bader, 2009].

例 3. 给定张量 $\mathcal{X} \in \mathbb{R}^{3 \times 4 \times 2}$, 若其 *frontal* 切片为

$$\mathcal{X}_{:, :, 1} = \begin{bmatrix} 1 & 4 & 7 & 10 \\ 2 & 5 & 8 & 11 \\ 3 & 6 & 9 & 12 \end{bmatrix} \quad \mathcal{X}_{:, :, 2} = \begin{bmatrix} 13 & 16 & 19 & 22 \\ 14 & 17 & 20 & 23 \\ 15 & 18 & 21 & 24 \end{bmatrix} \quad (2.23)$$

试写出张量矩阵化的结果 $\mathbf{X}_{(1)} \in \mathbb{R}^{3 \times 8}$ 、 $\mathbf{X}_{(2)} \in \mathbb{R}^{4 \times 6}$ 与 $\mathbf{X}_{(3)} \in \mathbb{R}^{2 \times 12}$ 。

解. 根据张量矩阵化规则, 张量 \mathcal{X} 自第 1 维度展开得到的矩阵为 $\mathbf{X}_{(1)}$ 为

$$\mathbf{X}_{(1)} = \begin{bmatrix} \mathcal{X}_{:, :, 1} & \mathcal{X}_{:, :, 2} \end{bmatrix} = \begin{bmatrix} 1 & 4 & 7 & 10 & 13 & 16 & 19 & 22 \\ 2 & 5 & 8 & 11 & 14 & 17 & 20 & 23 \\ 3 & 6 & 9 & 12 & 15 & 18 & 21 & 24 \end{bmatrix} \quad (2.24)$$

张量 \mathcal{X} 自第 2 维度展开得到的矩阵为 $\mathbf{X}_{(2)}$ 为

$$\mathbf{X}_{(2)} = \begin{bmatrix} \mathcal{X}_{:, :, 1}^\top & \mathcal{X}_{:, :, 2}^\top \end{bmatrix} = \begin{bmatrix} 1 & 2 & 3 & 13 & 14 & 15 \\ 4 & 5 & 6 & 16 & 17 & 18 \\ 7 & 8 & 9 & 19 & 20 & 21 \\ 10 & 11 & 12 & 22 & 23 & 24 \end{bmatrix} \quad (2.25)$$

张量 \mathcal{X} 自第 3 维度展开得到的矩阵为 $\mathbf{X}_{(3)}$ 为

$$\mathbf{X}_{(3)} = \begin{bmatrix} \mathcal{X}_{:, 1, :}^\top & \mathcal{X}_{:, 2, :}^\top \end{bmatrix} = \begin{bmatrix} 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 & 9 & 10 & 11 & 12 \\ 13 & 14 & 15 & 16 & 17 & 18 & 19 & 20 & 21 & 22 & 23 & 24 \end{bmatrix} \quad (2.26)$$

对于任意三阶张量 $\mathcal{X} \in \mathbb{R}^{m \times n \times t}$, 其自三个维度展开得到的矩阵分别为

$$\begin{cases} \mathbf{X}_{(1)} = \begin{bmatrix} \mathcal{X}_{:, :, 1} & \mathcal{X}_{:, :, 2} & \cdots & \mathcal{X}_{:, :, t} \end{bmatrix} \in \mathbb{R}^{m \times (nt)} \\ \mathbf{X}_{(2)} = \begin{bmatrix} \mathcal{X}_{:, :, 1}^\top & \mathcal{X}_{:, :, 2}^\top & \cdots & \mathcal{X}_{:, :, t}^\top \end{bmatrix} \in \mathbb{R}^{n \times (mt)} \\ \mathbf{X}_{(3)} = \begin{bmatrix} \mathcal{X}_{:, 1, :}^\top & \mathcal{X}_{:, 2, :}^\top & \cdots & \mathcal{X}_{:, t, :}^\top \end{bmatrix} \in \mathbb{R}^{t \times (mn)} \end{cases} \quad (2.27)$$

例 4. 给定张量 $\mathcal{X} \in \mathbb{R}^{3 \times 4 \times 2}$, 若其 *frontal* 切片为

$$\mathcal{X}_{:, :, 1} = \begin{bmatrix} 1 & 4 & 7 & 10 \\ 2 & 5 & 8 & 11 \\ 3 & 6 & 9 & 12 \end{bmatrix} \quad \mathcal{X}_{:, :, 2} = \begin{bmatrix} 13 & 16 & 19 & 22 \\ 14 & 17 & 20 & 23 \\ 15 & 18 & 21 & 24 \end{bmatrix} \quad (2.28)$$

试根据张量矩阵化定义写出张量 \mathcal{X} 的第 (2, 3, 2) 个元素 (即 20) 在 $\mathbf{X}_{(1)} \in \mathbb{R}^{3 \times 8}$ 、 $\mathbf{X}_{(2)} \in \mathbb{R}^{4 \times 6}$ 与 $\mathbf{X}_{(3)} \in \mathbb{R}^{2 \times 12}$ 的位置。

解. 在矩阵 $\mathbf{X}_{(1)}$ 中, 元素 20 的位置为 (2, 7), 即

$$\begin{aligned} i_1 &= 2 \\ j_1 &= 1 + (i_2 - 1)J_2 + (i_3 - 1)J_3 \\ &= 1 + (3 - 1) \times 1 + (2 - 1) \times 4 = 7 \end{aligned} \quad (2.29)$$

在矩阵 $\mathbf{X}_{(2)}$ 中, 元素 20 的位置为 (3, 5), 即

$$\begin{aligned} i_2 &= 3 \\ j_2 &= 1 + (i_1 - 1)J_1 + (i_3 - 1)J_3 \\ &= 1 + (2 - 1) \times 1 + (2 - 1) \times 3 = 5 \end{aligned} \quad (2.30)$$

在矩阵 $\mathbf{X}_{(3)}$ 中, 元素 20 的位置为 (2, 8), 即

$$\begin{aligned} i_3 &= 2 \\ j_3 &= 1 + (i_1 - 1)J_1 + (i_2 - 1)J_2 \\ &= 1 + (2 - 1) \times 1 + (3 - 1) \times 3 = 8 \end{aligned} \quad (2.31)$$

2.1.4 高阶张量向量化

给定任意张量 $\mathcal{X} \in \mathbb{R}^{m_1 \times m_2 \times \cdots \times m_d}$, 阶数为 d , 若其以第一个维度展开得到的矩阵为 $\mathbf{X}_{(1)}$, 则张量向量化可写作如下形式:

$$\text{vec}(\mathcal{X}) = \text{vec}(\mathbf{X}_{(1)}) \quad (2.32)$$

例 5. 给定张量 $\mathcal{X} \in \mathbb{R}^{2 \times 2 \times 2}$, 若其 *frontal* 切片为

$$\mathcal{X}_{::,1} = \begin{bmatrix} x_{111} & x_{121} \\ x_{211} & x_{221} \end{bmatrix} = \begin{bmatrix} 1 & 2 \\ 3 & 4 \end{bmatrix} \quad \mathcal{X}_{::,2} = \begin{bmatrix} x_{112} & x_{122} \\ x_{212} & x_{222} \end{bmatrix} = \begin{bmatrix} 5 & 6 \\ 7 & 8 \end{bmatrix} \quad (2.33)$$

试写出张量向量化的结果 $\text{vec}(\mathcal{X})$ 。

解. 根据张量向量化规则, 有

$$\begin{aligned} \text{vec}(\mathcal{X}) &= \text{vec}(\mathbf{X}_{(1)}) \\ &= \text{vec}([\mathcal{X}_{::,1} \quad \mathcal{X}_{::,2}]) \\ &= \text{vec}\left(\begin{bmatrix} 1 & 2 & 5 & 6 \\ 3 & 4 & 7 & 8 \end{bmatrix}\right) \\ &= (1, 3, 2, 4, 5, 7, 6, 8)^\top \end{aligned} \quad (2.34)$$

2.2 常见范数

在机器学习中, 常见范数包括向量范数与矩阵范数, 这些范数通常可用于构造特定的损失函数、正则项等, 是各类机器学习模型的重要基础。

2.2.1 向量范数

ℓ_1 范数

从定义出发, 向量的 ℓ_1 范数表示向量元素的绝对值之和。给定任意向量 $\mathbf{x} \in \mathbb{R}^n$, 写作

$$\mathbf{x} = (x_1, x_2, \dots, x_n)^\top \quad (2.35)$$

其 ℓ_1 范数为

$$\|\mathbf{x}\|_1 = |x_1| + |x_2| + \cdots + |x_n| = \sum_{i=1}^n |x_i| \quad (2.36)$$

其中, $|\cdot|$ 为绝对值符号。

ℓ_2 范数

ℓ_2 范数是一种常见的向量范数¹, 表示向量元素的平方和开根号, 一般用 $\|\cdot\|_2$ 表示。给定任意向量 $\mathbf{x} \in \mathbb{R}^n$, 其 ℓ_2 范数为

$$\|\mathbf{x}\|_2 = \sqrt{x_1^2 + x_2^2 + \cdots + x_n^2} = \sqrt{\sum_{i=1}^n x_i^2} \quad (2.37)$$

¹ ℓ_2 范数又被称为 Euclidean 范数, 即欧式范数。

ℓ_p 范数

相比于向量的 ℓ_1 范数与 ℓ_2 范数, ℓ_p 范数是更为一般形式的范数。给定任意向量 $\mathbf{x} \in \mathbb{R}^n$, 其 ℓ_p 范数为

$$\|\mathbf{x}\|_p = (|x_1|^p + |x_2|^p + \cdots + |x_n|^p)^{1/p} = \left(\sum_{i=1}^n |x_i|^p \right)^{1/p} \quad (2.38)$$

其中, $p \geq 1$ 。当 $p = 1$ 时, ℓ_p 范数即为 ℓ_1 范数; 当 $p = 2$ 时, ℓ_p 范数即为 ℓ_2 范数。

2.2.2 矩阵的 Frobenius 范数

从定义出发, 矩阵的 Frobenius 范数表示矩阵元素的平方和开根号, 一般用 $\|\cdot\|_F$ 表示。对于任意矩阵 $\mathbf{X} \in \mathbb{R}^{m \times n}$, 其 Frobenius 范数为

$$\|\mathbf{X}\|_F = \sqrt{\sum_{i=1}^m \sum_{j=1}^n x_{ij}^2} \quad (2.39)$$

2.3 求导数

2.3.1 从微积分到范数的求导数

在微积分中, 给定函数 $f(x)$, 其导数可写作

$$f'(x) = \frac{df(x)}{dx} = \lim_{\Delta x \rightarrow 0} \frac{f(x + \Delta x) - f(x)}{\Delta x} = \lim_{\Delta x \rightarrow 0} \frac{\Delta f}{\Delta x} \quad (2.40)$$

若 $f(x) = x^2$, 则 $f'(x) = 2x$ 。

给定函数 $f(x, y)$, 则关于变量 x 和 y 的偏导数分别为

$$\begin{cases} \frac{\partial f(x, y)}{\partial x} = \lim_{\Delta x \rightarrow 0} \frac{f(x + \Delta x, y) - f(x, y)}{\Delta x} \\ \frac{\partial f(x, y)}{\partial y} = \lim_{\Delta y \rightarrow 0} \frac{f(x, y + \Delta y) - f(x, y)}{\Delta y} \end{cases} \quad (2.41)$$

若 $f(x, y) = 2x^2 + y^2$, 则

$$\begin{cases} \frac{\partial f(x, y)}{\partial x} = 4x \\ \frac{\partial f(x, y)}{\partial y} = 2y \end{cases} \quad (2.42)$$

微积分中的求导数遵循一定的规则, 这些规则实际上同样适用于向量、矩阵的求导数过程。给定任意向量 $\mathbf{x} \in \mathbb{R}^n$, 令其 ℓ_2 范数的平方为函数 $f(\mathbf{x}) = \|\mathbf{x}\|_2^2$, 则

$$\begin{aligned} f(\mathbf{x}) &= f(x_1, x_2, \dots, x_n) \\ &= x_1^2 + x_2^2 + \cdots + x_n^2 \\ &= \sum_{i=1}^n x_i^2 \end{aligned} \quad (2.43)$$

分别对 x_1, x_2, \dots, x_n 求偏导数, 有

$$\begin{aligned} \frac{\partial f(x_1, x_2, \dots, x_n)}{\partial x_1} &= 2x_1 \\ \frac{\partial f(x_1, x_2, \dots, x_n)}{\partial x_2} &= 2x_2 \\ &\vdots \\ \frac{\partial f(x_1, x_2, \dots, x_n)}{\partial x_n} &= 2x_n \end{aligned} \quad (2.44)$$

因此, 函数 $f(\mathbf{x})$ 的导数为

$$\frac{df(\mathbf{x})}{d\mathbf{x}} = \begin{bmatrix} \frac{\partial f(x_1, x_2, \dots, x_n)}{\partial x_1} \\ \frac{\partial f(x_1, x_2, \dots, x_n)}{\partial x_2} \\ \vdots \\ \frac{\partial f(x_1, x_2, \dots, x_n)}{\partial x_n} \end{bmatrix} = \begin{bmatrix} 2x_1 \\ 2x_2 \\ \vdots \\ 2x_n \end{bmatrix} = 2\mathbf{x} \quad (2.45)$$

例 6. 给定任意向量 $\mathbf{x}, \mathbf{y} \in \mathbb{R}^n$, 试写出函数 $f(\mathbf{x}, \mathbf{y}) = \mathbf{x}^\top \mathbf{y}$ 的偏导数 $\frac{\partial f(\mathbf{x}, \mathbf{y})}{\partial \mathbf{x}}$ 与 $\frac{\partial f(\mathbf{x}, \mathbf{y})}{\partial \mathbf{y}}$ 。

解. 函数 $f(\mathbf{x}, \mathbf{y})$ 的偏导数分别为

$$\frac{\partial f(\mathbf{x}, \mathbf{y})}{\partial \mathbf{x}} = \mathbf{y} \quad \frac{\partial f(\mathbf{x}, \mathbf{y})}{\partial \mathbf{y}} = \mathbf{x} \quad (2.46)$$

2.4 延伸阅读

矩阵求导数是机器学习中十分重要的一部分内容, 上述介绍的求导数仅局限于本文所涉及到的推导, 更多介绍可参考《The Matrix Cookbook》²。

²网址为<https://www.math.uwaterloo.ca/~hwolkowi/matrixcookbook.pdf>。

第三章 时空交通数据类型

时空交通数据的分类方式有很多种，从代数结构角度来看，可通过矩阵或张量来表示时空交通数据。举例来说，带有空间信息与时间信息的交通流参数（如流量与车速）可整理成矩阵；出行数据可整理成张量，该张量的维度信息包括起点、终点、时间信息。但有时候，也会根据需要对时空交通数据进行升维处理，将矩阵形式的数据转换成张量形式或是将张量形式的数据转换成更高阶的张量，从而利用维度信息进行建模。

第四章 张量分解基础

张量分解在过去的数年间被应用于众多领域的数据分析以及理论建模。本章将从矩阵计算中最为常见的 Kronecker 积出发, 介绍其基本定义与一些常见性质, 另外, 我们会介绍向量外积的定义及性质, 从而引出经典的 CP 张量分解形式。当然, 张量分解也有其他经典的分解形式, 例如 Tucker 分解, 可参考 2009 年的综述文献《Tensor Decompositions and Applications》[Kolda and Bader, 2009]。

4.1 Kronecker 积定义

4.1.1 基本定义

Kronecker 积是以德国数学家 Leopold Kronecker 的名字命名的运算规则, 已广泛应用于各类矩阵计算以及张量计算算法中。从定义出发, 给定任意矩阵 $\mathbf{X} \in \mathbb{R}^{m \times n}$ 与 $\mathbf{Y} \in \mathbb{R}^{p \times q}$, 则两者之间的 Kronecker 积为

$$\mathbf{X} \otimes \mathbf{Y} = \begin{bmatrix} x_{11}\mathbf{Y} & x_{12}\mathbf{Y} & \cdots & x_{1n}\mathbf{Y} \\ x_{21}\mathbf{Y} & x_{22}\mathbf{Y} & \cdots & x_{2n}\mathbf{Y} \\ \vdots & \vdots & \ddots & \vdots \\ x_{m1}\mathbf{Y} & x_{m2}\mathbf{Y} & \cdots & x_{mn}\mathbf{Y} \end{bmatrix} \in \mathbb{R}^{(mp) \times (nq)} \quad (4.1)$$

其中, 符号 \otimes 表示 Kronecker 积。这里的 Kronecker 积得到的矩阵大小为 $(mp) \times (nq)$, 在写法上符合线性代数中对分块矩阵 (block matrix) 的定义, 其中, 分块矩阵的子矩阵是由矩阵 \mathbf{X} 的每个元素与矩阵 \mathbf{Y} 相乘得到。

矩阵 \mathbf{X} 与 \mathbf{Y} 之间的 Kronecker 积存在前后顺序, 根据 Kronecker 积的定义, 可得到矩阵 \mathbf{Y} 与 \mathbf{X} 之间的 Kronecker 积为

$$\mathbf{Y} \otimes \mathbf{X} = \begin{bmatrix} y_{11}\mathbf{X} & y_{12}\mathbf{X} & \cdots & y_{1q}\mathbf{X} \\ y_{21}\mathbf{X} & y_{22}\mathbf{X} & \cdots & y_{2q}\mathbf{X} \\ \vdots & \vdots & \ddots & \vdots \\ y_{p1}\mathbf{X} & y_{p2}\mathbf{X} & \cdots & y_{pq}\mathbf{X} \end{bmatrix} \in \mathbb{R}^{(mp) \times (nq)} \quad (4.2)$$

尽管矩阵 $\mathbf{X} \otimes \mathbf{Y}$ 与矩阵 $\mathbf{Y} \otimes \mathbf{X}$ 大小一致, 但两者并不相等, 因此, Kronecker 积不存在交换律。

例 7. 给定矩阵 $\mathbf{X} = \begin{bmatrix} 1 & 2 \\ 3 & 4 \end{bmatrix}$ 与 $\mathbf{Y} = \begin{bmatrix} 5 & 6 & 7 \\ 8 & 9 & 10 \end{bmatrix}$, 试写出两者之间的 Kronecker 积 $\mathbf{X} \otimes \mathbf{Y}$ 与 $\mathbf{Y} \otimes \mathbf{X}$ 。

解. 根据 *Kronecker* 积定义, 有

$$\mathbf{X} \otimes \mathbf{Y} = \begin{bmatrix} 1 \times \begin{bmatrix} 5 & 6 & 7 \\ 8 & 9 & 10 \end{bmatrix} & 2 \times \begin{bmatrix} 5 & 6 & 7 \\ 8 & 9 & 10 \end{bmatrix} \\ 3 \times \begin{bmatrix} 5 & 6 & 7 \\ 8 & 9 & 10 \end{bmatrix} & 4 \times \begin{bmatrix} 5 & 6 & 7 \\ 8 & 9 & 10 \end{bmatrix} \end{bmatrix} = \begin{bmatrix} 5 & 6 & 7 & 10 & 12 & 14 \\ 8 & 9 & 10 & 16 & 18 & 20 \\ 15 & 18 & 21 & 20 & 24 & 28 \\ 24 & 27 & 30 & 32 & 36 & 40 \end{bmatrix} \quad (4.3)$$

$$\mathbf{Y} \otimes \mathbf{X} = \begin{bmatrix} 5 \times \begin{bmatrix} 1 & 2 \\ 3 & 4 \end{bmatrix} & 6 \times \begin{bmatrix} 1 & 2 \\ 3 & 4 \end{bmatrix} & 7 \times \begin{bmatrix} 1 & 2 \\ 3 & 4 \end{bmatrix} \\ 8 \times \begin{bmatrix} 1 & 2 \\ 3 & 4 \end{bmatrix} & 9 \times \begin{bmatrix} 1 & 2 \\ 3 & 4 \end{bmatrix} & 10 \times \begin{bmatrix} 1 & 2 \\ 3 & 4 \end{bmatrix} \end{bmatrix} = \begin{bmatrix} 5 & 10 & 6 & 12 & 7 & 14 \\ 15 & 20 & 18 & 24 & 21 & 28 \\ 8 & 16 & 9 & 18 & 10 & 20 \\ 24 & 32 & 27 & 36 & 30 & 40 \end{bmatrix} \quad (4.4)$$

例 8. 给定矩阵 $\mathbf{X} = \begin{bmatrix} 1 & 2 \\ 3 & 4 \end{bmatrix}$ 与 $\mathbf{Y} = \begin{bmatrix} 5 & 6 & 7 \\ 8 & 9 & 10 \end{bmatrix}$, 试问等式 $(\mathbf{X} \otimes \mathbf{Y})^\top = \mathbf{X}^\top \otimes \mathbf{Y}^\top$ 是否成立。

解. 根据 *Kronecker* 积定义, 有

$$\mathbf{X}^\top \otimes \mathbf{Y}^\top = \begin{bmatrix} 1 \times \begin{bmatrix} 5 & 8 \\ 6 & 9 \\ 7 & 10 \end{bmatrix} & 3 \times \begin{bmatrix} 5 & 8 \\ 6 & 9 \\ 7 & 10 \end{bmatrix} \\ 2 \times \begin{bmatrix} 5 & 8 \\ 6 & 9 \\ 7 & 10 \end{bmatrix} & 4 \times \begin{bmatrix} 5 & 8 \\ 6 & 9 \\ 7 & 10 \end{bmatrix} \end{bmatrix} = \begin{bmatrix} 5 & 8 & 15 & 24 \\ 6 & 9 & 18 & 27 \\ 7 & 10 & 21 & 30 \\ 10 & 16 & 20 & 32 \\ 12 & 18 & 24 & 36 \\ 14 & 20 & 28 & 40 \end{bmatrix} \quad (4.5)$$

在这里, 等式 $(\mathbf{X} \otimes \mathbf{Y})^\top = \mathbf{X}^\top \otimes \mathbf{Y}^\top$ 是成立的。

例 9. 给定向量 $\mathbf{x} = (1, 2)^\top$ 与 $\mathbf{y} = (3, 4)^\top$, 试写出 $\mathbf{x} \otimes \mathbf{y}$ 与 $\mathbf{x} \otimes \mathbf{y}^\top$ 。

解. 根据 *Kronecker* 积定义, 有

$$\mathbf{x} \otimes \mathbf{y} = \begin{bmatrix} 1 \times \begin{bmatrix} 3 \\ 4 \end{bmatrix} \\ 2 \times \begin{bmatrix} 3 \\ 4 \end{bmatrix} \end{bmatrix} = \begin{bmatrix} 3 \\ 4 \\ 6 \\ 8 \end{bmatrix} \quad (4.6)$$

$$\mathbf{x} \otimes \mathbf{y}^\top = \begin{bmatrix} 1 \times \begin{bmatrix} 3 & 4 \end{bmatrix} \\ 2 \times \begin{bmatrix} 3 & 4 \end{bmatrix} \end{bmatrix} = \begin{bmatrix} 3 & 4 \\ 6 & 8 \end{bmatrix} \quad (4.7)$$

在这里, $\mathbf{x} \otimes \mathbf{y}^\top = \mathbf{x} \mathbf{y}^\top$, 即向量外积。

例 10 (向量自回归). 对于多元时间序列, 向量自回归可写作如下形式 (参见例 4):

$$\mathbf{X} \mathbf{\Psi}_0^\top = \sum_{k=1}^d \mathbf{A}_k \mathbf{X} \mathbf{\Psi}_k^\top + \mathbf{E} \quad (4.8)$$

若令

$$\begin{aligned} \mathbf{A} &= \begin{bmatrix} \mathbf{A}_1 & \mathbf{A}_2 & \cdots & \mathbf{A}_d \end{bmatrix} \in \mathbb{R}^{N \times (dN)} \\ \mathbf{\Psi} &= \begin{bmatrix} \mathbf{\Psi}_1 & \mathbf{\Psi}_2 & \cdots & \mathbf{\Psi}_d \end{bmatrix} \in \mathbb{R}^{(T-d) \times (dT)} \end{aligned} \quad (4.9)$$

则向量自回归可进一步写作如下形式:

$$\mathbf{X} \mathbf{\Psi}_0^\top = \mathbf{A}(\mathbf{I}_d \otimes \mathbf{X}) \mathbf{\Psi}^\top + \mathbf{E} \quad (4.10)$$

4.1.2 Khatri-Rao 积

以 Kronecker 积为基础, 可定义另一种十分重要的运算规则, 即 Khatri-Rao 积。给定任意矩阵

$$\mathbf{X} = \begin{bmatrix} | & | & \cdots & | \\ \mathbf{x}_1 & \mathbf{x}_2 & \cdots & \mathbf{x}_d \\ | & | & \cdots & | \end{bmatrix} \in \mathbb{R}^{m \times d} \quad \mathbf{Y} = \begin{bmatrix} | & | & \cdots & | \\ \mathbf{y}_1 & \mathbf{y}_2 & \cdots & \mathbf{y}_d \\ | & | & \cdots & | \end{bmatrix} \in \mathbb{R}^{n \times d} \quad (4.11)$$

若两个矩阵列数相同, 则两者之间的 Khatri-Rao 积为

$$\mathbf{X} \odot \mathbf{Y} = \begin{bmatrix} | & | & \cdots & | \\ \mathbf{x}_1 \otimes \mathbf{y}_1 & \mathbf{x}_2 \otimes \mathbf{y}_2 & \cdots & \mathbf{x}_d \otimes \mathbf{y}_d \\ | & | & \cdots & | \end{bmatrix} \in \mathbb{R}^{(mn) \times d} \quad (4.12)$$

其中, 列向量是由 \mathbf{X} 与 \mathbf{Y} 的列向量进行 Kronecker 积运算得到的。

例 11. 给定矩阵 $\mathbf{X} = \begin{bmatrix} 1 & 2 \\ 3 & 4 \end{bmatrix}$ 与 $\mathbf{Y} = \begin{bmatrix} 5 & 6 \\ 7 & 8 \\ 9 & 10 \end{bmatrix}$, 试写出 $\mathbf{X} \odot \mathbf{Y}$ 。

解. 根据 Khatri-Rao 积定义, 有

$$\mathbf{X} \odot \mathbf{Y} = \begin{bmatrix} \begin{bmatrix} 1 \\ 3 \end{bmatrix} \otimes \begin{bmatrix} 5 \\ 7 \\ 9 \end{bmatrix} & \begin{bmatrix} 2 \\ 4 \end{bmatrix} \otimes \begin{bmatrix} 6 \\ 8 \\ 10 \end{bmatrix} \end{bmatrix} = \begin{bmatrix} 5 & 12 \\ 7 & 16 \\ 9 & 20 \\ 15 & 24 \\ 21 & 32 \\ 27 & 40 \end{bmatrix} \quad (4.13)$$

4.2 Kronecker 积基本性质

4.2.1 结合律与分配律

在小学数学中, 我们学习了加减乘除的运算规则。以乘法为例, 不妨重温一下烙印在我们脑海中的基本概念:

- 乘法结合律: $x \times y \times z = x \times (y \times z)$
- 乘法分配律: $x \times z + y \times z = (x + y) \times z$

由于 Kronecker 积本质上也是元素间相乘, 所以同样存在结合律与分配律。对于任意矩阵 \mathbf{X} 、 \mathbf{Y} 与 \mathbf{Z} , 结合律可归纳为

$$\mathbf{X} \otimes \mathbf{Y} \otimes \mathbf{Z} = \mathbf{X} \otimes (\mathbf{Y} \otimes \mathbf{Z}) \quad (4.14)$$

分配律可归纳为

$$\mathbf{X} \otimes \mathbf{Z} + \mathbf{Y} \otimes \mathbf{Z} = (\mathbf{X} + \mathbf{Y}) \otimes \mathbf{Z} \quad (4.15)$$

例 12. 给定矩阵 $\mathbf{X} = \begin{bmatrix} 1 & 2 \\ 3 & 4 \end{bmatrix}$ 、 $\mathbf{Y} = \begin{bmatrix} 5 & 6 \\ 7 & 8 \end{bmatrix}$ 与 $\mathbf{Z} = \begin{bmatrix} 1 & 1 \\ 1 & 1 \end{bmatrix}$, 试写出 $\mathbf{X} \otimes \mathbf{Y} \otimes \mathbf{Z}$ 与 $\mathbf{X} \otimes (\mathbf{Y} \otimes \mathbf{Z})$ 。

解. 根据 *Kronecker* 积定义, 有

$$\mathbf{X} \otimes \mathbf{Y} = \begin{bmatrix} 5 & 6 & 10 & 12 \\ 7 & 8 & 14 & 16 \\ 15 & 18 & 20 & 24 \\ 21 & 24 & 28 & 32 \end{bmatrix} \quad (4.16)$$

$$\mathbf{Y} \otimes \mathbf{Z} = \begin{bmatrix} 5 & 5 & 6 & 6 \\ 5 & 5 & 6 & 6 \\ 7 & 7 & 8 & 8 \\ 7 & 7 & 8 & 8 \end{bmatrix} \quad (4.17)$$

从而, 可得到

$$\mathbf{X} \otimes \mathbf{Y} \otimes \mathbf{Z} = \begin{bmatrix} 5 & 5 & 6 & 6 & 10 & 10 & 12 & 12 \\ 5 & 5 & 6 & 6 & 10 & 10 & 12 & 12 \\ 7 & 7 & 8 & 8 & 14 & 14 & 16 & 16 \\ 7 & 7 & 8 & 8 & 14 & 14 & 16 & 16 \\ 15 & 15 & 18 & 18 & 20 & 20 & 24 & 24 \\ 15 & 15 & 18 & 18 & 20 & 20 & 24 & 24 \\ 21 & 21 & 24 & 24 & 28 & 28 & 32 & 32 \\ 21 & 21 & 24 & 24 & 28 & 28 & 32 & 32 \end{bmatrix} = \mathbf{X} \otimes (\mathbf{Y} \otimes \mathbf{Z}) \quad (4.18)$$

例 13. 给定 $\mathbf{X} = \begin{bmatrix} 1 & 2 \\ 3 & 4 \end{bmatrix}$, $\mathbf{Y} = \begin{bmatrix} 5 & 6 \\ 7 & 8 \end{bmatrix}$ 与 $\mathbf{Z} = \begin{bmatrix} 1 & 1 \\ 1 & 1 \end{bmatrix}$, 试写出 $\mathbf{X} \otimes \mathbf{Z} + \mathbf{Y} \otimes \mathbf{Z}$ 与 $(\mathbf{X} + \mathbf{Y}) \otimes \mathbf{Z}$.

解. 根据 *Kronecker* 积定义, 有

$$\mathbf{X} \otimes \mathbf{Z} + \mathbf{Y} \otimes \mathbf{Z} = \begin{bmatrix} 1 & 1 & 2 & 2 \\ 1 & 1 & 2 & 2 \\ 3 & 3 & 4 & 4 \\ 3 & 3 & 4 & 4 \end{bmatrix} + \begin{bmatrix} 5 & 5 & 6 & 6 \\ 5 & 5 & 6 & 6 \\ 7 & 7 & 8 & 8 \\ 7 & 7 & 8 & 8 \end{bmatrix} = \begin{bmatrix} 6 & 6 & 8 & 8 \\ 6 & 6 & 8 & 8 \\ 10 & 10 & 12 & 12 \\ 10 & 10 & 12 & 12 \end{bmatrix} \quad (4.19)$$

$$(\mathbf{X} + \mathbf{Y}) \otimes \mathbf{Z} = \begin{bmatrix} 6 & 8 \\ 10 & 12 \end{bmatrix} \otimes \begin{bmatrix} 1 & 1 \\ 1 & 1 \end{bmatrix} = \begin{bmatrix} 6 & 6 & 8 & 8 \\ 6 & 6 & 8 & 8 \\ 10 & 10 & 12 & 12 \\ 10 & 10 & 12 & 12 \end{bmatrix} \quad (4.20)$$

4.2.2 矩阵相乘

对于任意矩阵 $\mathbf{X} \in \mathbb{R}^{m \times n}$, $\mathbf{Y} \in \mathbb{R}^{s \times t}$, $\mathbf{U} \in \mathbb{R}^{n \times p}$ 与 $\mathbf{V} \in \mathbb{R}^{t \times q}$, 则矩阵 $\mathbf{X} \otimes \mathbf{Y} \in \mathbb{R}^{(ms) \times (nt)}$ 的列数 nt 与矩阵 $\mathbf{U} \otimes \mathbf{V} \in \mathbb{R}^{(nt) \times (pq)}$ 的行数 nt 一致, 可进行矩阵相乘, 两者相乘得到的矩

阵满足：

$$\begin{aligned}
 (\mathbf{X} \otimes \mathbf{Y})(\mathbf{U} \otimes \mathbf{V}) &= \begin{bmatrix} x_{11}\mathbf{Y} & \cdots & x_{1n}\mathbf{Y} \\ \vdots & \ddots & \vdots \\ x_{m1}\mathbf{Y} & \cdots & x_{mn}\mathbf{Y} \end{bmatrix} \begin{bmatrix} u_{11}\mathbf{V} & \cdots & u_{1p}\mathbf{V} \\ \vdots & \ddots & \vdots \\ u_{n1}\mathbf{V} & \cdots & u_{np}\mathbf{V} \end{bmatrix} \\
 &= \begin{bmatrix} \sum_{k=1}^n x_{1k}u_{k1}\mathbf{YV} & \cdots & \sum_{k=1}^n x_{1k}u_{kp}\mathbf{YV} \\ \vdots & \ddots & \vdots \\ \sum_{k=1}^n x_{mk}u_{k1}\mathbf{YV} & \cdots & \sum_{k=1}^n x_{mk}u_{kp}\mathbf{YV} \end{bmatrix} \\
 &= \begin{bmatrix} \sum_{k=1}^n x_{1k}u_{k1} & \cdots & \sum_{k=1}^n x_{1k}u_{kp} \\ \vdots & \ddots & \vdots \\ \sum_{k=1}^n x_{mk}u_{k1} & \cdots & \sum_{k=1}^n x_{mk}u_{kp} \end{bmatrix} \otimes (\mathbf{YV}) \\
 &= (\mathbf{XU}) \otimes (\mathbf{YV}) \in \mathbb{R}^{(ms) \times (pq)}
 \end{aligned} \tag{4.21}$$

例 14 (矩阵的奇异值分解). 给定任意矩阵 $\mathbf{X} \in \mathbb{R}^{m \times n}$ 与 $\mathbf{Y} \in \mathbb{R}^{p \times q}$, 若奇异值分解分别为

$$\mathbf{X} = \mathbf{W}\mathbf{S}\mathbf{Q}^\top \quad \mathbf{Y} = \mathbf{U}\mathbf{D}\mathbf{V}^\top \tag{4.22}$$

试证明矩阵 $\mathbf{X} \otimes \mathbf{Y}$ 的奇异值分解可由矩阵 \mathbf{X} 与 \mathbf{Y} 的奇异值分解计算得到, 即

$$\mathbf{X} \otimes \mathbf{Y} = (\mathbf{W} \otimes \mathbf{U})(\mathbf{S} \otimes \mathbf{D})(\mathbf{Q} \otimes \mathbf{V})^\top \tag{4.23}$$

解. 根据 Kronecker 积性质, 有

$$\begin{aligned}
 \mathbf{X} \otimes \mathbf{Y} &= (\mathbf{W}\mathbf{S}\mathbf{Q}^\top) \otimes (\mathbf{U}\mathbf{D}\mathbf{V}^\top) \\
 &= (\mathbf{W} \otimes \mathbf{U})((\mathbf{S}\mathbf{Q}^\top) \otimes (\mathbf{D}\mathbf{V}^\top)) \\
 &= (\mathbf{W} \otimes \mathbf{U})(\mathbf{S} \otimes \mathbf{D})(\mathbf{Q}^\top \otimes \mathbf{V}^\top) \\
 &= (\mathbf{W} \otimes \mathbf{U})(\mathbf{S} \otimes \mathbf{D})(\mathbf{Q} \otimes \mathbf{V})^\top
 \end{aligned} \tag{4.24}$$

4.2.3 求逆矩阵

对于任意可逆矩阵 $\mathbf{X} \in \mathbb{R}^{m \times m}$ 与 $\mathbf{Y} \in \mathbb{R}^{n \times n}$, 由于

$$(\mathbf{X} \otimes \mathbf{Y})(\mathbf{X}^{-1} \otimes \mathbf{Y}^{-1}) = (\mathbf{X}\mathbf{X}^{-1}) \otimes (\mathbf{Y}\mathbf{Y}^{-1}) = \mathbf{I}_m \otimes \mathbf{I}_n = \mathbf{I}_{mn} \tag{4.25}$$

故有

$$(\mathbf{X} \otimes \mathbf{Y})^{-1} = \mathbf{X}^{-1} \otimes \mathbf{Y}^{-1} \tag{4.26}$$

恒成立。这意味着：若计算 $\mathbf{X} \otimes \mathbf{Y}$ 的逆矩阵, 可先对 \mathbf{X} 与 \mathbf{Y} 分别求逆矩阵, 再对得到的逆矩阵进行 Kronecker 积运算。一般而言, 直接计算 $\mathbf{X} \otimes \mathbf{Y}$ 的逆矩阵, 时间复杂度为 $\mathcal{O}(m^3n^3)$, 而对 \mathbf{X} 与 \mathbf{Y} 分别求逆矩阵的时间复杂度为 $\mathcal{O}(m^3)$ 与 $\mathcal{O}(n^3)$, 由此可看出: 这一性质可降低 Kronecker 积在大规模问题上的计算复杂度。

例 15. 给定矩阵 $\mathbf{X} = \begin{bmatrix} 1 & 2 \\ 3 & 4 \end{bmatrix}$ 与 $\mathbf{Y} = \begin{bmatrix} 5 & 6 \\ 7 & 8 \end{bmatrix}$, 试写出 $(\mathbf{X} \otimes \mathbf{Y})^{-1}$ 与 $\mathbf{X}^{-1} \otimes \mathbf{Y}^{-1}$ 。

解. 根据 *Kronecker* 积定义, 有

$$\mathbf{X} \otimes \mathbf{Y} = \begin{bmatrix} 5 & 6 & 10 & 12 \\ 7 & 8 & 14 & 16 \\ 15 & 18 & 20 & 24 \\ 21 & 24 & 28 & 32 \end{bmatrix} \quad (4.27)$$

对该矩阵求逆矩阵, 得到

$$(\mathbf{X} \otimes \mathbf{Y})^{-1} = \begin{bmatrix} 8 & -6 & -4 & 3 \\ -7 & 5 & 3.5 & -2.5 \\ -6 & 4.5 & 2 & -1.5 \\ 5.25 & -3.75 & -1.75 & 1.25 \end{bmatrix} \quad (4.28)$$

对矩阵 \mathbf{X} 与 \mathbf{Y} 分别求逆矩阵:

$$\mathbf{X}^{-1} = \begin{bmatrix} -2 & 1 \\ 1.5 & -0.5 \end{bmatrix} \quad \mathbf{Y}^{-1} = \begin{bmatrix} -4 & 3 \\ 3.5 & -2.5 \end{bmatrix} \quad (4.29)$$

再对得到的逆矩阵进行 *Kronecker* 积运算, 有

$$\mathbf{X}^{-1} \otimes \mathbf{Y}^{-1} = \begin{bmatrix} 8 & -6 & -4 & 3 \\ -7 & 5 & 3.5 & -2.5 \\ -6 & 4.5 & 2 & -1.5 \\ 5.25 & -3.75 & -1.75 & 1.25 \end{bmatrix} \quad (4.30)$$

对于任意矩阵 $\mathbf{X} \in \mathbb{R}^{m \times m}$ 与 $\mathbf{Y} \in \mathbb{R}^{p \times q}$, 由上述 *Kronecker* 积性质同样可得到如下性质:

$$(\mathbf{X} \otimes \mathbf{Y})^\dagger = \mathbf{X}^\dagger \otimes \mathbf{Y}^\dagger \quad (4.31)$$

其中, \cdot^\dagger 表示伪逆 (Moore-Penrose pseudoinverse)。

4.2.4 向量化

对于任意矩阵 $\mathbf{A} \in \mathbb{R}^{m \times n}$ 、 $\mathbf{X} \in \mathbb{R}^{n \times p}$ 与 $\mathbf{B} \in \mathbb{R}^{p \times q}$, 三者相乘满足:

$$\text{vec}(\mathbf{AXB}) = (\mathbf{B}^\top \otimes \mathbf{A}) \text{vec}(\mathbf{X}) \quad (4.32)$$

由此, 也可得到

$$\begin{cases} \text{vec}(\mathbf{AX}) = (\mathbf{I}_p \otimes \mathbf{A}) \text{vec}(\mathbf{X}) \\ \text{vec}(\mathbf{XB}) = (\mathbf{B}^\top \otimes \mathbf{I}_n) \text{vec}(\mathbf{X}) \end{cases} \quad (4.33)$$

例 16. 试证明公式(4.32)。

解.

$$\begin{aligned} \text{vec}(\mathbf{AXB}) &= \mathbf{Ax}_1 b_{11} + \mathbf{Ax}_2 b_{21} + \cdots + \mathbf{Ax}_p b_{p1} \\ &\quad + \mathbf{Ax}_1 b_{12} + \mathbf{Ax}_2 b_{22} + \cdots + \mathbf{Ax}_p b_{p2} \\ &\quad + \cdots + \mathbf{Ax}_1 b_{1q} + \mathbf{Ax}_2 b_{2q} + \cdots + \mathbf{Ax}_p b_{pq} \\ &= \begin{bmatrix} \mathbf{A}b_{11} & \mathbf{A}b_{21} & \cdots & \mathbf{A}b_{p1} \\ \mathbf{A}b_{12} & \mathbf{A}b_{22} & \cdots & \mathbf{A}b_{p2} \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{A}b_{1q} & \mathbf{A}b_{2q} & \cdots & \mathbf{A}b_{pq} \end{bmatrix} \begin{bmatrix} \mathbf{x}_1 \\ \mathbf{x}_2 \\ \vdots \\ \mathbf{x}_p \end{bmatrix} \\ &= (\mathbf{B}^\top \otimes \mathbf{A}) \text{vec}(\mathbf{X}) \end{aligned} \quad (4.34)$$

其中, $\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_p \in \mathbb{R}^n$ 表示矩阵 \mathbf{X} 的列向量。

例 17. 对于任意矩阵 $\mathbf{A} \in \mathbb{R}^{n \times n}$ 、 $\mathbf{x} \in \mathbb{R}^n$ 与 $\mathbf{B} \in \mathbb{R}^{n \times n}$ ，试证明三者相乘满足：

$$\text{vec}(\mathbf{A} \text{diag}(\mathbf{x}) \mathbf{B}) = (\mathbf{B}^\top \odot \mathbf{A}) \mathbf{x} \quad (4.35)$$

解. 根据 Kronecker 积与 Khatri-Rao 积性质，有

$$\begin{aligned} \text{vec}(\mathbf{A} \text{diag}(\mathbf{x}) \mathbf{B}) &= (\mathbf{B}^\top \otimes \mathbf{A}) \text{vec}(\text{diag}(\mathbf{x})) \\ &= (\mathbf{B}^\top \odot \mathbf{A}) \mathbf{x} \end{aligned} \quad (4.36)$$

例 18. Sylvester 方程是一种著名的矩阵方程，由英国数学家 James Joseph Sylvester 于 1884 年提出。时至今日，Sylvester 方程已在控制理论中具有极为广泛的应用。具体而言，已知矩阵 $\mathbf{A} \in \mathbb{R}^{m \times m}$ 、 $\mathbf{B} \in \mathbb{R}^{n \times n}$ 与 $\mathbf{C} \in \mathbb{R}^{m \times n}$ ，则 Sylvester 方程的一般形式为

$$\mathbf{A}\mathbf{X} + \mathbf{X}\mathbf{B} = \mathbf{C} \quad (4.37)$$

其中， $\mathbf{X} \in \mathbb{R}^{m \times n}$ 为待定参数。试根据 Kronecker 积性质写出 Sylvester 方程的解析解。

解. 首先将 Sylvester 方程写成

$$\mathbf{A}\mathbf{X}\mathbf{I}_n + \mathbf{I}_m\mathbf{X}\mathbf{B} = \mathbf{C} \quad (4.38)$$

根据 Kronecker 积性质，Sylvester 方程可写成如下形式：

$$(\mathbf{I}_n \otimes \mathbf{A} + \mathbf{B}^\top \otimes \mathbf{I}_m) \text{vec}(\mathbf{X}) = \text{vec}(\mathbf{C}) \quad (4.39)$$

因此，Sylvester 方程的解析解¹为

$$\text{vec}(\mathbf{X}) = (\mathbf{I}_n \otimes \mathbf{A} + \mathbf{B}^\top \otimes \mathbf{I}_m)^{-1} \text{vec}(\mathbf{C}) \quad (4.40)$$

尽管该解析解形式简洁，但复杂度却很高。在实际问题中，往往需要借助更为高效的数值计算方法（如 Bartels-Stewart 算法）对 Sylvester 方程进行求解。

4.3 Kronecker 积特殊性质

4.3.1 矩阵的迹

在线性代数中，矩阵的迹 (trace) 表示方阵对角线元素之和，数学符号为 $\text{tr}(\cdot)$ 。对于任意矩阵 $\mathbf{X} \in \mathbb{R}^{m \times m}$ 与 $\mathbf{Y} \in \mathbb{R}^{n \times n}$ ，矩阵 $\mathbf{X} \otimes \mathbf{Y}$ 的迹等于矩阵 \mathbf{X} 的迹乘以矩阵 \mathbf{Y} 的迹，即

$$\text{tr}(\mathbf{X} \otimes \mathbf{Y}) = \text{tr}(\mathbf{X}) \cdot \text{tr}(\mathbf{Y}) \quad (4.41)$$

恒成立。

例 19. 给定矩阵 $\mathbf{X} = \begin{bmatrix} 1 & 2 \\ 3 & 4 \end{bmatrix}$ 与 $\mathbf{Y} = \begin{bmatrix} 5 & 6 \\ 7 & 8 \end{bmatrix}$ ，试写出 $\text{tr}(\mathbf{X})$ 、 $\text{tr}(\mathbf{Y})$ 与 $\text{tr}(\mathbf{X} \otimes \mathbf{Y})$ 。

解. 根据定义，矩阵 \mathbf{X} 的迹与矩阵 \mathbf{Y} 的迹分别为

$$\text{tr}(\mathbf{X}) = 1 + 4 = 5 \quad \text{tr}(\mathbf{Y}) = 5 + 8 = 13 \quad (4.42)$$

由于

$$\mathbf{X} \otimes \mathbf{Y} = \begin{bmatrix} 5 & 6 & 10 & 12 \\ 7 & 8 & 14 & 16 \\ 15 & 18 & 20 & 24 \\ 21 & 24 & 28 & 32 \end{bmatrix} \quad (4.43)$$

故 $\text{tr}(\mathbf{X} \otimes \mathbf{Y}) = 5 + 8 + 20 + 32 = 65$ 。

¹有时候，可定义 Kronecker 和 (Kronecker sum，数学符号通常为 \oplus) 令 $\mathbf{A} \oplus \mathbf{B}^\top = \mathbf{I}_n \otimes \mathbf{A} + \mathbf{B}^\top \otimes \mathbf{I}_m$ ，将该解析解简记为 $\text{vec}(\mathbf{X}) = (\mathbf{A} \oplus \mathbf{B}^\top)^{-1} \text{vec}(\mathbf{C})$ 。

在矩阵计算中, 矩阵的迹有两条重要性质, 给定任意矩阵 $\mathbf{X} \in \mathbb{R}^{m \times n}$ 与 $\mathbf{Y} \in \mathbb{R}^{n \times m}$, 满足

$$\text{tr}(\mathbf{AB}) = \text{tr}(\mathbf{BA}) \quad (4.44)$$

及

$$\text{tr}(\mathbf{AB}) = \text{vec}(\mathbf{A}^\top)^\top \text{vec}(\mathbf{B}) \quad (4.45)$$

例 20. 给定矩阵 $\mathbf{A} \in \mathbb{R}^{m \times n}$ 、 $\mathbf{B} \in \mathbb{R}^{n \times p}$ 、 $\mathbf{C} \in \mathbb{R}^{p \times q}$ 与 $\mathbf{D} \in \mathbb{R}^{q \times m}$, 试证明

$$\text{tr}(\mathbf{ABCD}) = \text{vec}(\mathbf{B})^\top (\mathbf{C} \otimes \mathbf{A}) \text{vec}(\mathbf{D}^\top) \quad (4.46)$$

解. 根据矩阵的迹与 Kronecker 积性质, 有

$$\begin{aligned} \text{tr}(\mathbf{ABCD}) &= \text{tr}(\mathbf{D(ABC)}) \\ &= \text{vec}(\mathbf{D}^\top)^\top \text{vec}(\mathbf{ABC}) \\ &= \text{vec}(\mathbf{D}^\top)^\top (\mathbf{C}^\top \otimes \mathbf{A}) \text{vec}(\mathbf{B}) \\ &= \text{vec}(\mathbf{B})^\top (\mathbf{C} \otimes \mathbf{A}) \text{vec}(\mathbf{D}^\top) \end{aligned} \quad (4.47)$$

4.3.2 矩阵的 Frobenius 范数

从定义出发, 矩阵的 Frobenius 范数表示矩阵元素的平方和开根号, 一般用 $\|\cdot\|_F$ 表示。对于任意矩阵 $\mathbf{X} \in \mathbb{R}^{m \times n}$, 其 Frobenius 范数为

$$\|\mathbf{X}\|_F = \sqrt{\sum_{i=1}^m \sum_{j=1}^n x_{ij}^2} \quad (4.48)$$

据此定义, 给定任意矩阵 $\mathbf{X} \in \mathbb{R}^{m \times n}$ 与 $\mathbf{Y} \in \mathbb{R}^{p \times q}$, 有

$$\|\mathbf{X} \otimes \mathbf{Y}\|_F = \|\mathbf{X}\|_F \cdot \|\mathbf{Y}\|_F \quad (4.49)$$

恒成立。

例 21. 给定矩阵 $\mathbf{X} = \begin{bmatrix} 1 & 2 \\ 3 & 4 \end{bmatrix}$ 与 $\mathbf{Y} = \begin{bmatrix} 5 & 6 \\ 7 & 8 \end{bmatrix}$, 试写出 $\|\mathbf{X}\|_F$ 、 $\|\mathbf{Y}\|_F$ 与 $\|\mathbf{X} \otimes \mathbf{Y}\|_F$ 。

解. 根据定义, 矩阵 \mathbf{X} 与 \mathbf{Y} 的 Frobenius 范数分别为

$$\|\mathbf{X}\|_F = \sqrt{1^2 + 2^2 + 3^2 + 4^2} = \sqrt{30} \quad \|\mathbf{Y}\|_F = \sqrt{5^2 + 6^2 + 7^2 + 8^2} = \sqrt{174} \quad (4.50)$$

由于

$$\mathbf{X} \otimes \mathbf{Y} = \begin{bmatrix} 5 & 6 & 10 & 12 \\ 7 & 8 & 14 & 16 \\ 15 & 18 & 20 & 24 \\ 21 & 24 & 28 & 32 \end{bmatrix} \quad (4.51)$$

故 $\|\mathbf{X} \otimes \mathbf{Y}\|_F = \sqrt{5220}$ 。

Frobenius 范数这一概念不适用于向量, 对于任意向量 $\mathbf{x} \in \mathbb{R}^m$, 其元素的平方和开根号是 ℓ_2 范数, 即

$$\|\mathbf{x}\|_2 = \sqrt{\sum_{i=1}^m x_i^2} \quad (4.52)$$

例 22. 给定向量 $\mathbf{x} = (1, 2)^\top$ 与 $\mathbf{y} = (3, 4)^\top$, 试写出 $\|\mathbf{x}\|_2$ 、 $\|\mathbf{y}\|_2$ 与 $\|\mathbf{x} \otimes \mathbf{y}\|_2$ 。

解. 根据定义, 向量 \mathbf{x} 与 \mathbf{y} 的 ℓ_2 范数分别为

$$\|\mathbf{x}\|_2 = \sqrt{1^2 + 2^2} = \sqrt{5} \quad \|\mathbf{y}\|_2 = \sqrt{3^2 + 4^2} = 5 \quad (4.53)$$

由于 $\mathbf{x} \otimes \mathbf{y} = (3, 4, 6, 8)^\top$, 故 $\|\mathbf{x} \otimes \mathbf{y}\|_2 = \sqrt{3^2 + 4^2 + 6^2 + 8^2} = 5\sqrt{5}$ 。

4.3.3 矩阵的行列式

矩阵的行列式 (determinant) 是线性代数中非常重要的一个概念, 贯穿线性代数的几乎所有内容, 一般使用符号 $\det(\cdot)$ 表示。若给定矩阵 $\mathbf{X} \in \mathbb{R}^{m \times m}$ 与 $\mathbf{Y} \in \mathbb{R}^{n \times n}$, 则

$$\det(\mathbf{X} \otimes \mathbf{Y}) = \det(\mathbf{X})^n \cdot \det(\mathbf{Y})^m \quad (4.54)$$

恒成立。

例 23. 给定矩阵 $\mathbf{X} = \begin{bmatrix} 1 & 2 \\ 3 & 4 \end{bmatrix}$ 与 $\mathbf{Y} = \begin{bmatrix} 1 & 3 & 2 \\ 4 & 1 & 3 \\ 2 & 5 & 2 \end{bmatrix}$, 试写出矩阵的行列式 $\det(\mathbf{X})$ 、 $\det(\mathbf{Y})$ 与 $\det(\mathbf{X} \otimes \mathbf{Y})$ 。

解. 矩阵 \mathbf{X} 与 \mathbf{Y} 的行列式分别为

$$\det(\mathbf{X}) = \begin{vmatrix} 1 & 2 \\ 3 & 4 \end{vmatrix} = -2 \quad \det(\mathbf{Y}) = \begin{vmatrix} 1 & 3 & 2 \\ 4 & 1 & 3 \\ 2 & 5 & 2 \end{vmatrix} = 17 \quad (4.55)$$

故 $\det(\mathbf{X})^3 \cdot \det(\mathbf{Y})^2 = -2312$ 。

矩阵 $\mathbf{X} \otimes \mathbf{Y}$ 的行列式为

$$\det(\mathbf{X} \otimes \mathbf{Y}) = \begin{vmatrix} 1 & 3 & 2 & 2 & 6 & 4 \\ 4 & 1 & 3 & 8 & 2 & 6 \\ 2 & 5 & 2 & 4 & 10 & 4 \\ 3 & 9 & 6 & 4 & 12 & 8 \\ 12 & 3 & 9 & 16 & 4 & 12 \\ 6 & 15 & 6 & 8 & 20 & 8 \end{vmatrix} = -2312 \quad (4.56)$$

4.3.4 矩阵的秩

矩阵的秩 (rank) 是线性代数中非常重要的一个概念, 在信号处理、图像处理等领域中应用广泛, 一般使用符号 $\text{rank}(\cdot)$ 表示。若给定矩阵 $\mathbf{X} \in \mathbb{R}^{m \times n}$ 与 $\mathbf{Y} \in \mathbb{R}^{p \times q}$, 则

$$\text{rank}(\mathbf{X} \otimes \mathbf{Y}) = \text{rank}(\mathbf{X}) \cdot \text{rank}(\mathbf{Y}) \quad (4.57)$$

恒成立。

例 24. 给定矩阵 $\mathbf{X} = \begin{bmatrix} 1 & 2 \\ 2 & 4 \end{bmatrix}$ 与 $\mathbf{Y} = \begin{bmatrix} 5 & 6 & 7 \\ 8 & 9 & 10 \end{bmatrix}$, 试写出 $\text{rank}(\mathbf{X})$ 、 $\text{rank}(\mathbf{Y})$ 与 $\text{rank}(\mathbf{X} \otimes \mathbf{Y})$ 。

解. 在这里, $\text{rank}(\mathbf{X}) = 1$, $\text{rank}(\mathbf{Y}) = 2$ 。

由于

$$\mathbf{X} \otimes \mathbf{Y} = \begin{bmatrix} 5 & 6 & 7 & 10 & 12 & 14 \\ 8 & 9 & 10 & 16 & 18 & 20 \\ 10 & 12 & 14 & 20 & 24 & 28 \\ 16 & 18 & 20 & 32 & 36 & 40 \end{bmatrix} \quad (4.58)$$

故 $\text{rank}(\mathbf{X} \otimes \mathbf{Y}) = 2$ 。

4.4 向量外积

4.4.1 定义

在线性代数中, 两向量之间的外积可得到一个矩阵。对于任意向量 $\mathbf{a} \in \mathbb{R}^m$ 与 $\mathbf{b} \in \mathbb{R}^n$, 则两者之间的外积为

$$\mathbf{c} = \mathbf{a} \otimes_{\text{outer}} \mathbf{b} = \mathbf{a}\mathbf{b}^\top \in \mathbb{R}^{m \times n} \quad (4.59)$$

其中, 符号 \otimes_{outer} 表示向量外积。

依此类推, 对任意 d 个向量 $\mathbf{x}^{(k)} \in \mathbb{R}^{n_k}, k = 1, 2, \dots, d$, 其外积 (outer product) 可定义为

$$\mathbf{y} = \mathbf{x}^{(1)} \otimes_{\text{outer}} \mathbf{x}^{(2)} \otimes_{\text{outer}} \cdots \otimes_{\text{outer}} \mathbf{x}^{(d)} \in \mathbb{R}^{n_1 \times n_2 \times \cdots \times n_d} \quad (4.60)$$

其中, 符号 \otimes_{outer} 表示向量外积。在张量 \mathbf{y} 中, 任意第 (i_1, i_2, \dots, i_d) 个元素为

$$y_{i_1, i_2, \dots, i_d} = \prod_{k=1}^d x_{i_k}^{(k)} \quad (4.61)$$

其中, $i_k = 1, 2, \dots, n_k, k = 1, 2, \dots, d$ 。

需要注意的是, 由于张量 \mathbf{y} 是由向量外积得到的, 故常被称为秩一张量 (rank-one tensor)。当 $d = 3$ 时, 向量外积得到的三阶张量 $\mathbf{y} = \mathbf{x}^{(1)} \otimes_{\text{outer}} \mathbf{x}^{(2)} \otimes_{\text{outer}} \mathbf{x}^{(3)} \in \mathbb{R}^{n_1 \times n_2 \times n_3}$ 如图4.1所示, 在这里, 张量 \mathbf{y} 的任意第 (i_1, i_2, i_3) 个元素为

$$y_{i_1, i_2, i_3} = \prod_{k=1}^3 x_{i_k}^{(k)} \quad (4.62)$$

其中, $i_k = 1, 2, \dots, n_k, k = 1, 2, 3$ 。

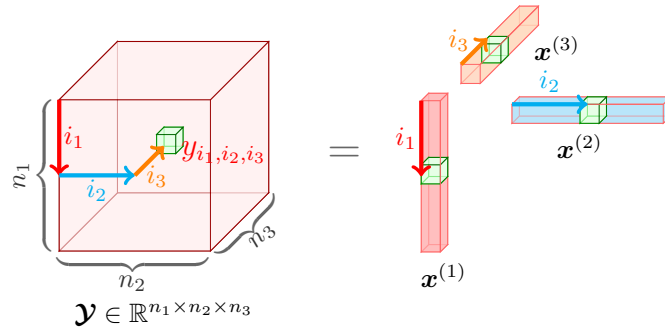


图 4.1: 向量外积得到的三阶张量 $\mathbf{y} \in \mathbb{R}^{n_1 \times n_2 \times n_3}$

当 $d = 2$ 时, 向量外积为

$$\mathbf{Y} = \mathbf{x}^{(1)} \otimes_{\text{outer}} \mathbf{x}^{(2)} = \mathbf{x}^{(1)}(\mathbf{x}^{(2)})^\top \in \mathbb{R}^{n_1 \times n_2} \quad (4.63)$$

在矩阵 \mathbf{Y} 中, 任意第 (i, j) 个元素为

$$y_{i, j} = x_i^{(1)} x_j^{(2)} \quad (4.64)$$

其中, $i = 1, 2, \dots, n_1$ 与 $j = 1, 2, \dots, n_2$ 。

例 25. 给定向量 $\mathbf{x} = (1, 2)^\top$ 与 $\mathbf{y} = (3, 4)^\top$, 试写出 $\mathbf{x} \otimes_{\text{outer}} \mathbf{y}$ 。

解. 根据外积定义, 有

$$\mathbf{x} \otimes_{\text{outer}} \mathbf{y} = \begin{bmatrix} 3 & 4 \\ 6 & 8 \end{bmatrix} \quad (4.65)$$

例 26. 给定向量 $\mathbf{a} = (1, 2)^\top$ 、 $\mathbf{b} = (3, 4, 5)^\top$ 与 $\mathbf{c} = (6, 7, 8, 9)^\top$ ，试写出 $\mathbf{a} \otimes_{\text{outer}} \mathbf{b} \otimes_{\text{outer}} \mathbf{c}$ 。

解. 令 $\mathbf{Y} = \mathbf{a} \otimes_{\text{outer}} \mathbf{b} \otimes_{\text{outer}} \mathbf{c} \in \mathbb{R}^{2 \times 3 \times 4}$ ，根据外积定义，有

$$\mathbf{a} \otimes_{\text{outer}} \mathbf{b} = \begin{bmatrix} 3 & 4 & 5 \\ 6 & 8 & 10 \end{bmatrix} \quad (4.66)$$

由此，可得张量 \mathbf{Y} 的 *frontal* 切片为

$$\begin{aligned} \mathbf{Y}_{:, :, 1} &= (\mathbf{a} \otimes_{\text{outer}} \mathbf{b}) \cdot \mathbf{c}_1 = \begin{bmatrix} 18 & 24 & 30 \\ 36 & 48 & 60 \end{bmatrix} & \mathbf{Y}_{:, :, 2} &= (\mathbf{a} \otimes_{\text{outer}} \mathbf{b}) \cdot \mathbf{c}_2 = \begin{bmatrix} 21 & 28 & 35 \\ 42 & 56 & 70 \end{bmatrix} \\ \mathbf{Y}_{:, :, 3} &= (\mathbf{a} \otimes_{\text{outer}} \mathbf{b}) \cdot \mathbf{c}_3 = \begin{bmatrix} 24 & 32 & 40 \\ 48 & 64 & 80 \end{bmatrix} & \mathbf{Y}_{:, :, 4} &= (\mathbf{a} \otimes_{\text{outer}} \mathbf{b}) \cdot \mathbf{c}_4 = \begin{bmatrix} 27 & 36 & 45 \\ 54 & 72 & 90 \end{bmatrix} \end{aligned} \quad (4.67)$$

4.4.2 性质

张量矩阵化

根据 Khatri-Rao 积定义与张量矩阵化规则，由向量 $\mathbf{x}^{(k)} \in \mathbb{R}^{n_k}, k = 1, 2, \dots, d$ 的外积得到的张量 $\mathbf{Y} \in \mathbb{R}^{n_1 \times n_2 \times \dots \times n_d}$ ，其自第 k 维度展开得到的矩阵可写作如下形式：

$$\begin{aligned} \mathbf{Y}_{(k)} &= \mathbf{x}^{(k)} \otimes_{\text{outer}} (\mathbf{x}^{(d)} \odot \dots \odot \mathbf{x}^{(k+1)} \odot \mathbf{x}^{(k-1)} \odot \dots \odot \mathbf{x}^{(1)}) \\ &= \mathbf{x}^{(k)} (\mathbf{x}^{(d)} \odot \dots \odot \mathbf{x}^{(k+1)} \odot \mathbf{x}^{(k-1)} \odot \dots \odot \mathbf{x}^{(1)})^\top \in \mathbb{R}^{n_k \times \prod_{h \neq k} n_h} \end{aligned} \quad (4.68)$$

其中， \odot 表示 Khatri-Rao 积。

例 27. 给定向量 $\mathbf{a} = (1, 2)^\top$ 、 $\mathbf{b} = (3, 4, 5)^\top$ 与 $\mathbf{c} = (6, 7, 8, 9)^\top$ ，若 $\mathbf{Y} = \mathbf{a} \otimes_{\text{outer}} \mathbf{b} \otimes_{\text{outer}} \mathbf{c}$ ，试写出张量 \mathbf{Y} 的矩阵化形式 $\mathbf{Y}_{(1)}$ 、 $\mathbf{Y}_{(2)}$ 与 $\mathbf{Y}_{(3)}$ 。

解. 根据 Khatri-Rao 积定义，有

$$\begin{cases} \mathbf{c} \odot \mathbf{b} = (18, 24, 30, 21, 28, 35, 24, 32, 40, 27, 36, 45)^\top \\ \mathbf{c} \odot \mathbf{a} = (6, 12, 7, 14, 8, 16, 9, 18)^\top \\ \mathbf{b} \odot \mathbf{a} = (3, 6, 4, 8, 5, 10)^\top \end{cases} \quad (4.69)$$

从而，可得到

$$\mathbf{Y}_{(1)} = \mathbf{a}(\mathbf{c} \odot \mathbf{b})^\top = \begin{bmatrix} 18 & 24 & 30 & 21 & 28 & 35 & 24 & 32 & 40 & 27 & 36 & 45 \\ 36 & 48 & 60 & 42 & 56 & 70 & 48 & 64 & 80 & 54 & 72 & 90 \end{bmatrix} \quad (4.70)$$

$$\mathbf{Y}_{(2)} = \mathbf{a}(\mathbf{c} \odot \mathbf{b})^\top = \begin{bmatrix} 18 & 36 & 21 & 42 & 24 & 48 & 27 & 54 \\ 24 & 48 & 28 & 56 & 32 & 64 & 36 & 72 \\ 30 & 60 & 35 & 70 & 40 & 80 & 45 & 90 \end{bmatrix} \quad (4.71)$$

$$\mathbf{Y}_{(3)} = \mathbf{c}(\mathbf{b} \odot \mathbf{a})^\top = \begin{bmatrix} 18 & 36 & 24 & 48 & 30 & 60 \\ 21 & 42 & 28 & 56 & 35 & 70 \\ 24 & 48 & 32 & 64 & 40 & 80 \\ 27 & 54 & 36 & 72 & 45 & 90 \end{bmatrix} \quad (4.72)$$

张量向量化

根据 Khatri-Rao 积定义与张量向量化规则, 由向量 $\mathbf{x}^{(k)} \in \mathbb{R}^{n_k}$, $k = 1, 2, \dots, d$ 的外积得到的张量 $\mathcal{Y} \in \mathbb{R}^{n_1 \times n_2 \times \dots \times n_d}$, 其向量化形式为

$$\text{vec}(\mathcal{Y}) = \mathbf{x}^{(d)} \odot \mathbf{x}^{(d-1)} \odot \dots \odot \mathbf{x}^{(2)} \odot \mathbf{x}^{(1)} \quad (4.73)$$

其中, \odot 表示 Khatri-Rao 积; $\text{vec}(\cdot)$ 表示向量化操作。

例 28. 给定向量 $\mathbf{a} = (1, 2)^\top$ 、 $\mathbf{b} = (3, 4, 5)^\top$ 与 $\mathbf{c} = (6, 7, 8, 9)^\top$, 若 $\mathcal{Y} = \mathbf{a} \otimes_{\text{outer}} \mathbf{b} \otimes_{\text{outer}} \mathbf{c}$, 试写出张量 \mathcal{Y} 的向量化形式 $\text{vec}(\mathcal{Y})$ 。

解. 根据 Khatri-Rao 积定义, 有

$$\mathbf{c} \odot \mathbf{b} = (18, 24, 30, 21, 28, 35, 24, 32, 40, 27, 36, 45)^\top \quad (4.74)$$

从而, 可得到

$$\begin{aligned} \text{vec}(\mathcal{Y}) = \mathbf{c} \odot \mathbf{b} \odot \mathbf{a} = & (18, 24, 30, 21, 28, 35, 24, 32, 40, 27, 36, 45, \\ & 36, 48, 60, 42, 56, 70, 48, 64, 80, 54, 72, 90)^\top \end{aligned} \quad (4.75)$$

4.5 CP 张量分解

4.5.1 CP 分解形式

CP 分解全称为 CANDECOMP/PARAFAC 分解, 它是以向量外积构成的分解形式, 本质上是矩阵分解的高阶泛化 [Kolda and Bader, 2009]。给定任意张量 $\mathcal{Y} \in \mathbb{R}^{n_1 \times n_2 \times \dots \times n_d}$, 若令其秩为 R , 则 CP 分解可写作如下形式:

$$\mathcal{Y} = \sum_{r=1}^R \mathbf{u}_r^{(1)} \otimes_{\text{outer}} \mathbf{u}_r^{(2)} \otimes_{\text{outer}} \dots \otimes_{\text{outer}} \mathbf{u}_r^{(d)} \quad (4.76)$$

其中, 因子矩阵为

$$\mathbf{U}^{(k)} = \begin{bmatrix} | & | & & | \\ \mathbf{u}_1^{(k)} & \mathbf{u}_2^{(k)} & \dots & \mathbf{u}_R^{(k)} \\ | & | & & | \end{bmatrix} \in \mathbb{R}^{n_k \times R}, \quad k = 1, 2, \dots, d \quad (4.77)$$

在张量 \mathcal{Y} 中, 任意第 (i_1, i_2, \dots, i_d) 个元素为

$$y_{i_1, i_2, \dots, i_d} = \sum_{r=1}^R u_{i_1, r}^{(1)} \times u_{i_2, r}^{(2)} \times \dots \times u_{i_d, r}^{(d)} = \sum_{r=1}^R \prod_{k=1}^d u_{i_k, r}^{(k)} \quad (4.78)$$

其中, $i_k = 1, 2, \dots, n_k$, $k = 1, 2, \dots, d$ 。

解说 1 (三阶张量的 CP 分解). 给定三阶张量 \mathcal{Y} , 其 CP 分解可写作如下形式:

$$\mathcal{Y} = \sum_{r=1}^R \mathbf{u}_r^{(1)} \otimes_{\text{outer}} \mathbf{u}_r^{(2)} \otimes_{\text{outer}} \mathbf{u}_r^{(3)} \quad (4.79)$$

任意第 (i_1, i_2, \dots, i_d) 个元素为

$$y_{i_1, i_2, i_3} = \sum_{r=1}^R u_{i_1, r}^{(1)} \times u_{i_2, r}^{(2)} \times u_{i_3, r}^{(3)} = \sum_{r=1}^R \prod_{k=1}^3 u_{i_k, r}^{(k)} \quad (4.80)$$

图 4.2 给出了三阶张量的 CP 分解示意图。

图 4.2: 三阶张量 $\mathcal{Y} \in \mathbb{R}^{n_1 \times n_2 \times n_3}$ 的 CP 分解

由于 CP 分解可写成因子矩阵列向量外积的形式，因此，CP 分解具有以下性质：

- **张量矩阵化**。由因子矩阵 $\mathbf{U}^{(k)} \in \mathbb{R}^{n_k \times R}$, $k = 1, 2, \dots, d$ 相乘得到的张量 $\mathcal{Y} \in \mathbb{R}^{n_1 \times n_2 \times \dots \times n_d}$ ，其自第 k 维度展开得到的矩阵可写作如下形式：

$$\mathbf{Y}_{(k)} = \mathbf{U}^{(k)} (\mathbf{U}^{(d)} \odot \dots \odot \mathbf{U}^{(k+1)} \odot \mathbf{U}^{(k-1)} \odot \dots \odot \mathbf{U}^{(1)})^\top \in \mathbb{R}^{n_k \times \prod_{h \neq k} n_h} \quad (4.81)$$

- **张量向量化**。由因子矩阵 $\mathbf{U}^{(k)} \in \mathbb{R}^{n_k \times R}$, $k = 1, 2, \dots, d$ 相乘得到的张量 $\mathcal{Y} \in \mathbb{R}^{n_1 \times n_2 \times \dots \times n_d}$ ，其向量化形式为

$$\text{vec}(\mathcal{Y}) = \sum_{r=1}^R \mathbf{u}_r^{(d)} \odot \mathbf{u}_r^{(d-1)} \odot \dots \odot \mathbf{u}_r^{(2)} \odot \mathbf{u}_r^{(1)} \quad (4.82)$$

例 29. 给定矩阵 $\mathbf{U}^{(1)}$

4.5.2 低秩逼近问题

对于任意张量 $\mathcal{Y} \in \mathbb{R}^{n_1 \times n_2 \times \dots \times n_d}$ ，可借助 CP 分解结构得到一个低秩张量用于逼近原张量 \mathcal{Y} ，给定低秩张量的秩为 R ，低秩逼近过程的优化问题为

$$\min_{\{\mathbf{U}^{(k)}\}} \frac{1}{2} \left\| \mathcal{Y} - \sum_{r=1}^R \mathbf{u}_r^{(1)} \otimes_{\text{outer}} \mathbf{u}_r^{(2)} \otimes_{\text{outer}} \dots \otimes_{\text{outer}} \mathbf{u}_r^{(d)} \right\|_F^2 + \frac{\rho}{2} \sum_{h=1}^d \|\mathbf{U}^{(h)}\|_F^2 \quad (4.83)$$

其中， $\mathbf{U}^{(k)} \in \mathbb{R}^{n_k \times R}$, $k = 1, 2, \dots, d$ 为因子矩阵； ρ 为正则项的权重系数。

不妨将优化问题的目标函数进行改写：

$$\begin{aligned} f &= \frac{1}{2} \left\| \mathcal{Y} - \sum_{r=1}^R \mathbf{u}_r^{(1)} \otimes_{\text{outer}} \mathbf{u}_r^{(2)} \otimes_{\text{outer}} \dots \otimes_{\text{outer}} \mathbf{u}_r^{(d)} \right\|_F^2 + \frac{\rho}{2} \sum_{h=1}^d \|\mathbf{U}^{(h)}\|_F^2 \\ &= \frac{1}{2} \left\| \mathbf{Y}_{(k)} - \mathbf{U}^{(k)} (\mathbf{U}^{(d)} \odot \dots \odot \mathbf{U}^{(k+1)} \odot \mathbf{U}^{(k-1)} \odot \dots \odot \mathbf{U}^{(1)})^\top \right\|_F^2 + \frac{\rho}{2} \sum_{h=1}^d \|\mathbf{U}^{(h)}\|_F^2 \end{aligned} \quad (4.84)$$

其中， $k = 1, 2, \dots, d$ 。

对因子矩阵 $\mathbf{U}^{(k)}$ 求偏导数，有

$$\frac{\partial f}{\partial \mathbf{U}^{(k)}} = -(\mathbf{Y}_{(k)} - \mathbf{U}^{(k)} \mathbf{A}^{(k)}) (\mathbf{A}^{(k)})^\top + \rho \mathbf{U}^{(k)} \quad (4.85)$$

其中，

$$\mathbf{A}^{(k)} = (\mathbf{U}^{(d)} \odot \dots \odot \mathbf{U}^{(k+1)} \odot \mathbf{U}^{(k-1)} \odot \dots \odot \mathbf{U}^{(1)})^\top \quad (4.86)$$

此时，令 $\frac{\partial f}{\partial \mathbf{U}^{(k)}} = \mathbf{0}$ ，则因子矩阵 $\mathbf{U}^{(k)}$ 的最小二乘解为

$$\mathbf{U}^{(k)} = \mathbf{Y}_{(k)} (\mathbf{A}^{(k)})^\top (\mathbf{A}^{(k)} (\mathbf{A}^{(k)})^\top + \rho \mathbf{I}_R)^{-1} \quad (4.87)$$

因此,可采用交替最小二乘法对 CP 分解的低秩逼近进行求解,即在迭代过程中,通过交替更新因子矩阵的最小二乘解最终达到收敛。算法1给出了基于交替最小二乘法的 CP 分解算法。

Algorithm 1 CP 分解算法

Input: 张量 $\mathcal{Y} \in \mathbb{R}^{n_1 \times n_2 \times \cdots \times n_d}$, 低秩张量的秩 R , 超参数 ρ 。

Output: 重构出来的低秩张量 $\hat{\mathcal{Y}}$ 。

- 1: 对因子矩阵 $\{\mathbf{U}^{(k)}\}$ 进行初始化;
 - 2: **for** $i = 0$ to 最大迭代次数 **do**
 - 3: **for** $k = 1$ to d **do**
 - 4: 根据公式(4.87)对因子矩阵 $\mathbf{U}^{(k)}$ 的最小二乘解进行更新;
 - 5: **end for**
 - 6: 计算低秩张量 $\hat{\mathcal{Y}} = \sum_{r=1}^R \mathbf{u}_r^{(1)} \otimes_{\text{outer}} \mathbf{u}_r^{(2)} \otimes_{\text{outer}} \cdots \otimes_{\text{outer}} \mathbf{u}_r^{(d)}$;
 - 7: **end for**
-

第五章 时序矩阵分解

时序矩阵分解是矩阵分解中的一个重要模型，主要用于对时间序列数据进行建模 [Chen et al., 2022a]。当多元时间序列数据存在缺失值时，时序矩阵分解中的时序建模技术如向量自回归便会起到不可忽视的作用。在时序矩阵分解中，矩阵分解可从部分观测数据中学习出低秩模式，而时序建模则可刻画时序关联特征。本章分别介绍考虑时空平滑的矩阵分解与时序矩阵分解。

5.1 矩阵分解

5.1.1 模型表达式

对于多元时间序列，若任意时刻 t 对应的观测数据为向量 $\mathbf{y}_t \in \mathbb{R}^N$ ，则多元时间序列可写作矩阵形式：

$$\mathbf{Y} = \begin{bmatrix} | & | & \cdots & | \\ \mathbf{y}_1 & \mathbf{y}_2 & \cdots & \mathbf{y}_T \\ | & | & & | \end{bmatrix} \in \mathbb{R}^{N \times T} \quad (5.1)$$

当矩阵中存在缺失值时，可用 Ω 表示被观测元素的索引集合。一般而言，可定义作用于集合 Ω 上的正交映射 (orthogonal projection) $\mathcal{P}_\Omega : \mathbb{R}^{N \times T} \rightarrow \mathbb{R}^{N \times T}$ ，对于矩阵 \mathbf{Y} 任意第 (i, t) 个元素，有

$$[\mathcal{P}_\Omega(\mathbf{Y})]_{i,t} = \begin{cases} \mathbf{y}_{i,t} & \text{if } (i, t) \in \Omega \\ 0 & \text{otherwise} \end{cases} \quad (5.2)$$

同时，可定义作用于集合 Ω 补集上的正交映射 $\mathcal{P}_\Omega^\perp : \mathbb{R}^{N \times T} \rightarrow \mathbb{R}^{N \times T}$ 。

例 30. 给定矩阵 $\mathbf{X} = \begin{bmatrix} 1 & 2 \\ 3 & 4 \end{bmatrix}$ ，若 $\Omega = \{(1, 1), (2, 2)\}$ ，则

$$\mathcal{P}_\Omega(\mathbf{X}) = \begin{bmatrix} 1 & 0 \\ 0 & 4 \end{bmatrix} \quad \mathcal{P}_\Omega^\perp(\mathbf{X}) = \begin{bmatrix} 0 & 2 \\ 3 & 0 \end{bmatrix} \quad (5.3)$$

通常来说，对于矩阵 \mathbf{Y} ，矩阵分解的优化问题为

$$\min_{\mathbf{W}, \mathbf{X}} \frac{1}{2} \|\mathcal{P}_\Omega(\mathbf{Y} - \mathbf{W}^\top \mathbf{X})\|_F^2 + \frac{\rho}{2} (\|\mathbf{W}\|_F^2 + \|\mathbf{X}\|_F^2) \quad (5.4)$$

其中， $\mathbf{W} \in \mathbb{R}^{R \times N}$ 与 $\mathbf{X} \in \mathbb{R}^{R \times T}$ 为因子矩阵，在时空交通数据中， \mathbf{W} 与 \mathbf{X} 通常被分别称为空间因子矩阵 (spatial factor matrix) 与时序因子矩阵 (temporal factor matrix)； ρ 为正则项的权重系数。图5.1直观展示了矩阵分解。

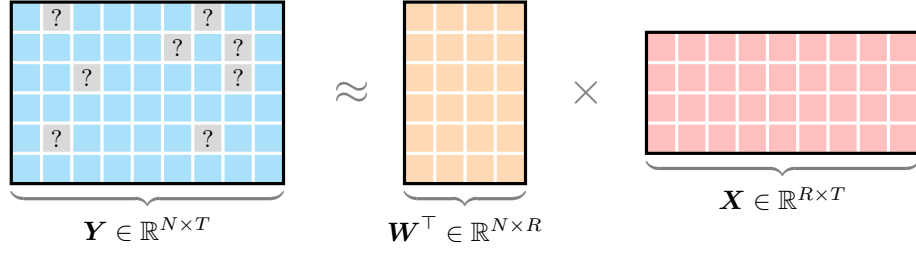


图 5.1: 矩阵分解示意图

5.1.2 梯度下降法

为了对公式(5.4)中的优化问题进行求解,不妨使用梯度下降法 [Chi et al., 2019]。将目标函数记作

$$f = \frac{1}{2} \|\mathcal{P}_\Omega(\mathbf{Y} - \mathbf{W}^\top \mathbf{X})\|_F^2 + \frac{\rho}{2} (\|\mathbf{W}\|_F^2 + \|\mathbf{X}\|_F^2) \quad (5.5)$$

则梯度下降法在迭代过程中的更新规则为

$$\begin{cases} \mathbf{W} := \mathbf{W} - \alpha \frac{\partial f}{\partial \mathbf{W}} \\ \mathbf{X} := \mathbf{X} - \alpha \frac{\partial f}{\partial \mathbf{X}} \end{cases} \quad (5.6)$$

其中, α 为梯度下降的步长 (step size); 目标函数关于因子矩阵 \mathbf{W} 的偏导数为

$$\frac{\partial f}{\partial \mathbf{W}} = -\mathbf{X} \mathcal{P}_\Omega^\top(\mathbf{Y} - \mathbf{W}^\top \mathbf{X}) + \rho \mathbf{W} \quad (5.7)$$

目标函数关于因子矩阵 \mathbf{X} 的偏导数为

$$\frac{\partial f}{\partial \mathbf{X}} = -\mathbf{W} \mathcal{P}_\Omega(\mathbf{Y} - \mathbf{W}^\top \mathbf{X}) + \rho \mathbf{X} \quad (5.8)$$

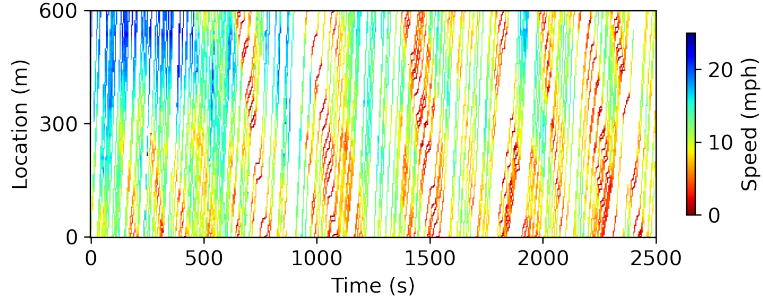
例 31. 给定如图5.2所示的稀疏速度场, 试采用梯度下降法对矩阵分解进行求解并重构速度场。

解. 将矩阵分解的秩设置为 $R = 10$ 、正则项系数设置为 $\rho = 10$ 、固定步长设置为 $\alpha = 10^{-4}$ 、迭代次数为 1000 代, 使用 Python 实现算法即可得到如图5.3所示的重构速度场, 在图中, 矩阵分解的重构误差为 $MAPE = 50.66\%$ 、 $RMSE = 2.33$ 。

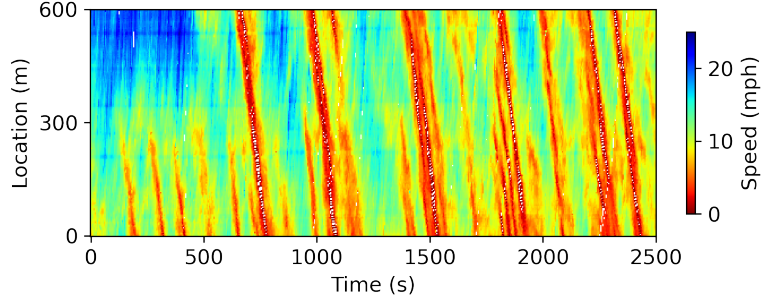
5.1.3 最速梯度下降法

在上述梯度下降法的迭代过程中, 步长 α 始终是固定不变的。为了加速梯度下降的收敛, 最速梯度下降法 (steepest gradient descent, 参见附录B.1.2) 可在每次迭代中自动计算最优的步长。令最速梯度下降法的更新规则为

$$\begin{cases} \alpha := \arg \min_{\alpha} f(\mathbf{W} - \alpha \frac{\partial f}{\partial \mathbf{W}}, \mathbf{X}) \\ \mathbf{W} := \mathbf{W} - \alpha \frac{\partial f}{\partial \mathbf{W}} \\ \beta := \arg \min_{\beta} f(\mathbf{W}, \mathbf{X} - \beta \frac{\partial f}{\partial \mathbf{X}}) \\ \mathbf{X} := \mathbf{X} - \beta \frac{\partial f}{\partial \mathbf{X}} \end{cases} \quad (5.9)$$



(a) 稀疏速度场



(b) 真实速度场

图 5.2: NGSIM 数据集中的速度场

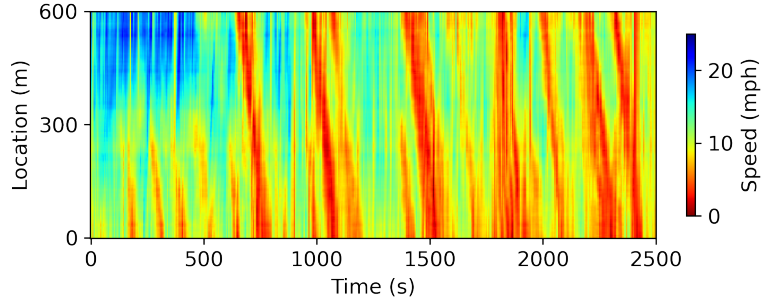


图 5.3: 使用梯度下降法求解矩阵分解所得到的重构速度场

在这里，若对步长 α 进行优化，则有

$$\begin{aligned}
 \alpha &:= \arg \min_{\alpha} f(\mathbf{W} - \alpha \frac{\partial f}{\partial \mathbf{W}}, \mathbf{X}) \\
 &= \arg \min_{\alpha} \frac{1}{2} \left\| \mathcal{P}_{\Omega} \left(\mathbf{Y} - (\mathbf{W} - \alpha \frac{\partial f}{\partial \mathbf{W}})^{\top} \mathbf{X} \right) \right\|_F^2 + \frac{\rho}{2} \left\| \mathbf{W} - \alpha \frac{\partial f}{\partial \mathbf{W}} \right\|_F^2 \\
 &= \arg \min_{\alpha} \frac{1}{2} \left\| \mathcal{P}_{\Omega}(\mathbf{Y} - \mathbf{W}^{\top} \mathbf{X}) + \alpha \mathcal{P}_{\Omega} \left(\left(\frac{\partial f}{\partial \mathbf{W}} \right)^{\top} \mathbf{X} \right) \right\|_F^2 + \frac{\rho}{2} \left\| \mathbf{W} - \alpha \frac{\partial f}{\partial \mathbf{W}} \right\|_F^2 \\
 &= \frac{a_2}{a_1}
 \end{aligned} \tag{5.10}$$

其中，

$$\begin{cases} a_1 = \left\| \mathcal{P}_{\Omega} \left(\left(\frac{\partial f}{\partial \mathbf{W}} \right)^{\top} \mathbf{X} \right) \right\|_F^2 + \rho \left\| \frac{\partial f}{\partial \mathbf{W}} \right\|_F^2 \\ a_2 = - \left\langle \mathcal{P}_{\Omega}(\mathbf{Y} - \mathbf{W}^{\top} \mathbf{X}), \mathcal{P}_{\Omega} \left(\left(\frac{\partial f}{\partial \mathbf{W}} \right)^{\top} \mathbf{X} \right) \right\rangle + \rho \left\langle \mathbf{W}, \frac{\partial f}{\partial \mathbf{W}} \right\rangle \end{cases} \tag{5.11}$$

解说 2 (矩阵内积 (inner product)). 给定任意矩阵 $\mathbf{X} \in \mathbb{R}^{m \times n}$ 与 $\mathbf{Y} \in \mathbb{R}^{m \times n}$ ，其内积为

$$\langle \mathbf{X}, \mathbf{Y} \rangle = \sum_{i=1}^m \sum_{j=1}^n x_{ij} y_{ij} \in \mathbb{R} \tag{5.12}$$

其中, 符号 $\langle \cdot, \cdot \rangle$ 表示内积。

此时, 矩阵 \mathbf{X} 的 *Frobenius* 范数也可写作如下形式:

$$\|\mathbf{X}\|_F = \sqrt{\langle \mathbf{X}, \mathbf{X} \rangle} = \sqrt{\sum_{i=1}^m \sum_{j=1}^n x_{ij}^2} \quad (5.13)$$

这里矩阵内积的定义同样可推广到向量与张量上。

若对步长 β 进行优化, 则有

$$\begin{aligned} \beta &:= \arg \min_{\beta} f(\mathbf{W}, \mathbf{X} - \beta \frac{\partial f}{\partial \mathbf{X}}) \\ &= \arg \min_{\beta} \frac{1}{2} \left\| \mathcal{P}_{\Omega}(\mathbf{Y} - \mathbf{W}^{\top}(\mathbf{X} - \beta \frac{\partial f}{\partial \mathbf{X}})) \right\|_F^2 + \frac{\rho}{2} \left\| \mathbf{X} - \beta \frac{\partial f}{\partial \mathbf{X}} \right\|_F^2 \\ &= \arg \min_{\beta} \frac{1}{2} \left\| \mathcal{P}_{\Omega}(\mathbf{Y} - \mathbf{W}^{\top} \mathbf{X}) + \beta \mathcal{P}_{\Omega}(\mathbf{W}^{\top} \frac{\partial f}{\partial \mathbf{X}}) \right\|_F^2 + \frac{\rho}{2} \left\| \mathbf{X} - \beta \frac{\partial f}{\partial \mathbf{X}} \right\|_F^2 \\ &= \frac{b_2}{b_1} \end{aligned} \quad (5.14)$$

其中,

$$\begin{cases} b_1 = \left\| \mathcal{P}_{\Omega}(\mathbf{W}^{\top} \frac{\partial f}{\partial \mathbf{X}}) \right\|_F^2 + \rho \left\| \frac{\partial f}{\partial \mathbf{X}} \right\|_F^2 \\ b_2 = -\left\langle \mathcal{P}_{\Omega}(\mathbf{Y} - \mathbf{W}^{\top} \mathbf{X}), \mathcal{P}_{\Omega}(\mathbf{W}^{\top} \frac{\partial f}{\partial \mathbf{X}}) \right\rangle + \rho \left\langle \mathbf{X}, \frac{\partial f}{\partial \mathbf{X}} \right\rangle \end{cases} \quad (5.15)$$

例 32. 给定如图 5.2 所示的稀疏速度场, 试采用最速梯度下降法对矩阵分解进行求解并重构速度场。

解. 将矩阵分解的秩设置为 $R = 10$ 、正则项系数设置为 $\rho = 10$ 、迭代次数为 1000 代, 使用 *Python* 实现算法即可得到如图 5.4 所示的重构速度场, 在图中, 矩阵分解的重构误差为 $MAPE = 45.13\%$ 、 $RMSE = 2.79$ 。

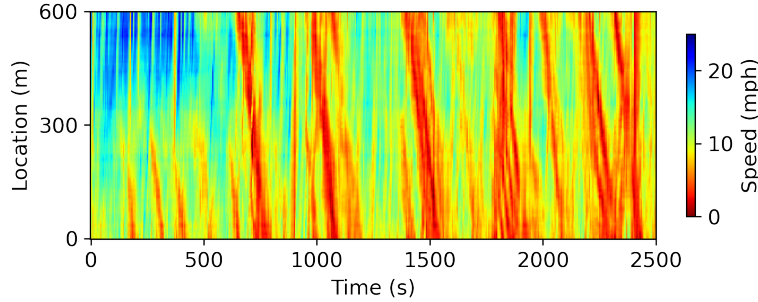


图 5.4: 使用最速梯度下降法求解矩阵分解所得到的重构速度场

5.1.4 交替优化算法

除了梯度下降法, 矩阵分解中常用的另一种求解算法是交替优化算法 (alternating minimization algorithm)。该方法也是迭代算法, 在每次迭代时, 需求解出因子矩阵 \mathbf{W} 与 \mathbf{X} 的最小二乘解或者近似解。

令因子矩阵 \mathbf{W} 与 \mathbf{X} 的偏导数分别为 $\mathbf{0}$, 即

$$\begin{aligned} \frac{\partial f}{\partial \mathbf{W}} &= -\mathbf{X} \mathcal{P}_{\Omega}^{\top}(\mathbf{Y} - \mathbf{W}^{\top} \mathbf{X}) + \rho \mathbf{W} = \mathbf{0} \\ \frac{\partial f}{\partial \mathbf{X}} &= -\mathbf{W} \mathcal{P}_{\Omega}(\mathbf{Y} - \mathbf{W}^{\top} \mathbf{X}) + \rho \mathbf{X} = \mathbf{0} \end{aligned} \quad (5.16)$$

相应地, 因子矩阵 \mathbf{W} 的列向量 $\mathbf{w}_i \in \mathbb{R}^R, i = 1, 2, \dots, N$ 存在最小二乘解, 即

$$\mathbf{w}_i = \left(\sum_{t:(i,t) \in \Omega} \mathbf{x}_t \mathbf{x}_t^\top + \rho \mathbf{I}_R \right)^{-1} \sum_{t:(i,t) \in \Omega} \mathbf{x}_t y_{i,t} \quad (5.17)$$

其中, 向量 $\mathbf{x}_t \in \mathbb{R}^R, \forall t \in \{1, 2, \dots, T\}$ 是因子矩阵 \mathbf{X} 的第 t 列; 符号 $\sum_{t:(i,t) \in \Omega}$ 表示固定索引 i 后对索引集合 Ω 内所有索引 t 进行求和。

类似地, 向量 \mathbf{x}_t 的最小二乘解为

$$\mathbf{x}_t = \left(\sum_{i:(i,t) \in \Omega} \mathbf{w}_i \mathbf{w}_i^\top + \rho \mathbf{I}_R \right)^{-1} \sum_{i:(i,t) \in \Omega} \mathbf{w}_i y_{i,t} \quad (5.18)$$

其中, 符号 $\sum_{i:(i,t) \in \Omega}$ 表示固定索引 t 后对索引集合 Ω 内所有索引 i 进行求和。

由于因子矩阵的列向量有最小二乘解, 因此, 这里的交替优化算法也被称为交替最小二乘法。

例 33. 给定如图 5.2 所示的稀疏速度场, 试采用交替最小二乘法对矩阵分解进行求解并重构速度场, 与此同时, 试分析梯度下降法、最速梯度下降法与交替最小二乘法在迭代过程中的目标函数变化情况。

解. 将矩阵分解的秩设置为 $R = 10$ 、正则项系数设置为 $\rho = 10$ 、迭代次数为 200 代, 使用 *Python* 实现算法即可得到如图 5.5 所示的重构速度场, 在图中, 矩阵分解的重构误差为 $MAPE = 45.84\%$ 、 $RMSE = 2.80$ 。

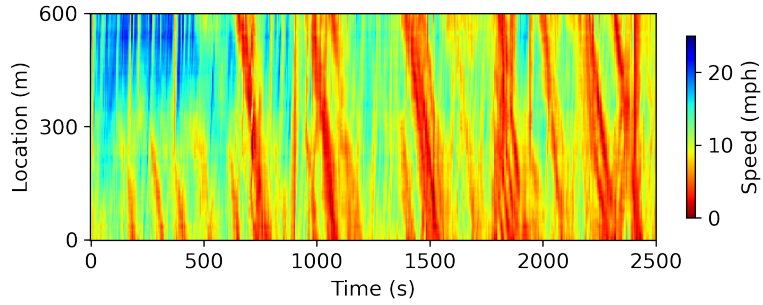


图 5.5: 使用交替最小二乘法求解矩阵分解所得到的重构速度场

在图 5.6 的目标函数变化曲线中, *GD* 代表梯度下降法, 即 *gradient descent* 的简称; *SGD* 代表最速梯度下降法, 即 *steepest gradient descent* 的简称; *ALS* 代表交替最小二乘法, 即 *alternating least squares* 的简称。对于求解矩阵分解的优化问题, 三种方法具有不同的收敛率, 不难看出: 梯度下降法的收敛速度远低于最速梯度下降法与交替最小二乘法; 最速梯度下降法的收敛速度稍逊于交替最小二乘法。

例 34. 以西雅图高速公路车速数据集为例, 随机生成 60% 的缺失值, 求解矩阵分解时, 试对比梯度下降法、最速梯度下降法与交替最小二乘法的收敛速度与矩阵分解的重构误差。

解. 将矩阵分解的秩设置为 $R = 10$ 、正则项系数设置为 $\rho = 10^2$ 、梯度下降法的固定步长设置为 $\alpha = 2 \times 10^{-5}$ 、梯度下降法与最速梯度下降法的迭代次数为 1000 代、交替最小二乘法的迭代次数为 200 代, 使用 *Python* 实现相应的算法。矩阵分解的重构误差为

- 梯度下降法: $MAPE = 9.14\%$ 、 $RMSE = 5.24$;
- 最速梯度下降法: $MAPE = 9.12\%$ 、 $RMSE = 5.24$;
- 交替最小二乘法: $MAPE = 9.13\%$ 、 $RMSE = 5.24$ 。

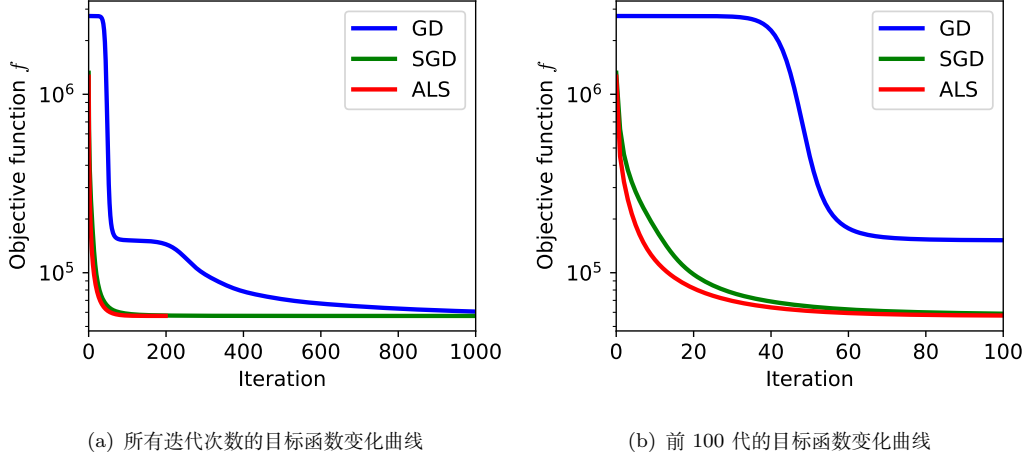


图 5.6: 在速度场数据集上, 三种迭代法所对应的矩阵分解目标函数变化曲线

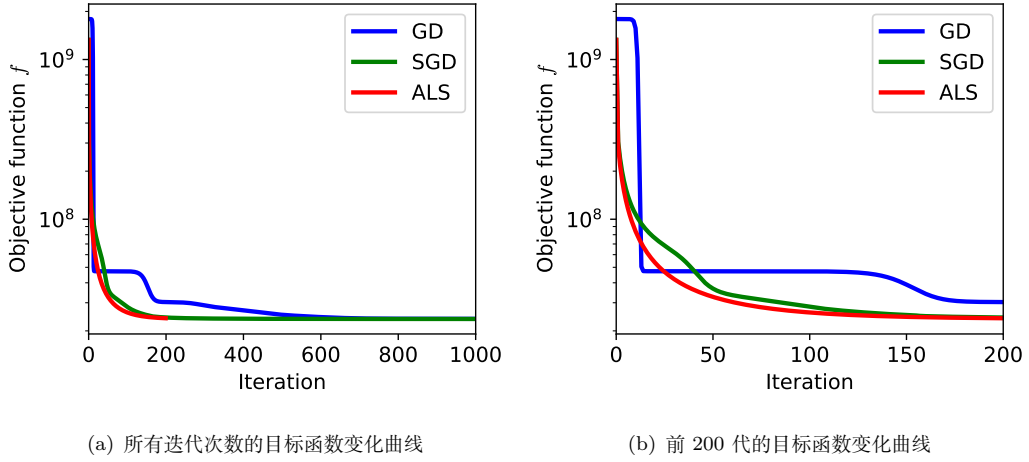


图 5.7: 在西雅图高速公路车速数据集上, 三种迭代法所对应的矩阵分解目标函数变化曲线

5.2 平滑矩阵分解

5.2.1 模型表达式

在公式(5.4)所示的矩阵分解中, 低秩结构能捕捉数据的全局模式, 但却往往忽略了局部的关联特征。为了分别对空间局部信息与时序局部信息进行建模, 不妨引入如下形式的矩阵分解, 即平滑矩阵分解¹:

$$\min_{\mathbf{W}, \mathbf{X}} \frac{1}{2} \|\mathcal{P}_{\Omega}(\mathbf{Y} - \mathbf{W}^{\top} \mathbf{X})\|_F^2 + \frac{\rho}{2} (\|\mathbf{W}\|_F^2 + \|\mathbf{X}\|_F^2) + \frac{\lambda}{2} (\|\mathbf{W} \Psi_1^{\top}\|_F^2 + \|\mathbf{X} \Psi_2^{\top}\|_F^2) \quad (5.19)$$

其中, 目标函数的最后两项为平滑正则项 (smoothing regularization), 正则项的权重系数为 λ , 用于平滑处理的矩阵被定义为

$$\begin{aligned} \Psi_1 &= \begin{bmatrix} \mathbf{0}_{(N-1) \times 1} & \mathbf{I}_{N-1} \end{bmatrix} - \begin{bmatrix} \mathbf{I}_{N-1} & \mathbf{0}_{(N-1) \times 1} \end{bmatrix} \in \mathbb{R}^{(N-1) \times N} \\ \Psi_2 &= \begin{bmatrix} \mathbf{0}_{(T-1) \times 1} & \mathbf{I}_{T-1} \end{bmatrix} - \begin{bmatrix} \mathbf{I}_{T-1} & \mathbf{0}_{(T-1) \times 1} \end{bmatrix} \in \mathbb{R}^{(T-1) \times T} \end{aligned} \quad (5.20)$$

¹当 $\lambda = 0$ 时, 则考虑平滑处理的矩阵分解变为标准的矩阵分解。

即

$$\begin{aligned}\Psi_1 &= \begin{bmatrix} -1 & 1 & 0 & \cdots & 0 & 0 \\ 0 & -1 & 1 & \cdots & 0 & 0 \\ 0 & 0 & -1 & \cdots & 0 & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & 0 & \cdots & -1 & 1 \end{bmatrix} \in \mathbb{R}^{(N-1) \times N} \\ \Psi_2 &= \begin{bmatrix} -1 & 1 & 0 & \cdots & 0 & 0 \\ 0 & -1 & 1 & \cdots & 0 & 0 \\ 0 & 0 & -1 & \cdots & 0 & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & 0 & \cdots & -1 & 1 \end{bmatrix} \in \mathbb{R}^{(T-1) \times T}\end{aligned}\quad (5.21)$$

根据定义，恒有

$$\begin{aligned}\mathbf{W}\Psi_1^\top &= \begin{bmatrix} | & | & & | \\ \mathbf{w}_2 & \mathbf{w}_3 & \cdots & \mathbf{w}_N \\ | & | & & | \end{bmatrix} - \begin{bmatrix} | & | & & | \\ \mathbf{w}_1 & \mathbf{w}_2 & \cdots & \mathbf{w}_{N-1} \\ | & | & & | \end{bmatrix} \\ \mathbf{X}\Psi_2^\top &= \begin{bmatrix} | & | & & | \\ \mathbf{x}_2 & \mathbf{x}_3 & \cdots & \mathbf{x}_T \\ | & | & & | \end{bmatrix} - \begin{bmatrix} | & | & & | \\ \mathbf{x}_1 & \mathbf{x}_2 & \cdots & \mathbf{x}_{T-1} \\ | & | & & | \end{bmatrix}\end{aligned}\quad (5.22)$$

分别表示作用于空间因子矩阵与时序因子矩阵的一阶差分 (first-order differencing)。

图5.8直观展示了平滑矩阵分解的建模思路：对两个因子矩阵的列向量分别进行局部关联，并以平滑正则项的形式写进矩阵分解的目标函数。

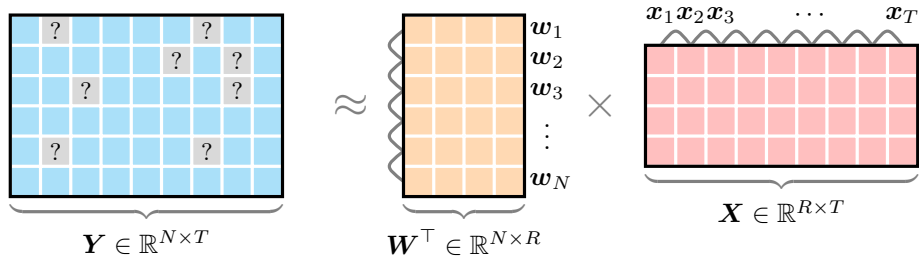


图 5.8: 平滑矩阵分解示意图

5.2.2 求解过程

为了估计优化问题中的待定参数，即 \mathbf{W} 与 \mathbf{X} ，可采用交替优化算法 (alternating minimization algorithm)。交替优化算法（如交替最小二乘法）是求解矩阵分解中非凸优化问题的常用方法，该方法采用迭代过程，可通过交替更新待估计变量的最优解（如最小二乘解）最终达到收敛。

在平滑矩阵分解的优化问题中，令目标函数为

$$f = \frac{1}{2} \|\mathcal{P}_\Omega(\mathbf{Y} - \mathbf{W}^\top \mathbf{X})\|_F^2 + \frac{\rho}{2} (\|\mathbf{W}\|_F^2 + \|\mathbf{X}\|_F^2) + \frac{\lambda}{2} (\|\mathbf{W}\Psi_1^\top\|_F^2 + \|\mathbf{X}\Psi_2^\top\|_F^2) \quad (5.23)$$

更新变量 \mathbf{W}

对变量 \mathbf{W} 求偏导数, 有

$$\frac{\partial f}{\partial \mathbf{W}} = -\mathbf{X}\mathcal{P}_{\Omega}^{\top}(\mathbf{Y} - \mathbf{W}^{\top}\mathbf{X}) + \rho\mathbf{W} + \lambda\mathbf{W}\Psi_1^{\top}\Psi_1 \quad (5.24)$$

此时, 令 $\frac{\partial f}{\partial \mathbf{W}} = \mathbf{0}$, 则可得到如下矩阵方程:

$$\mathbf{X}\mathcal{P}_{\Omega}^{\top}(\mathbf{W}^{\top}\mathbf{X}) + \rho\mathbf{W} + \lambda\mathbf{W}\Psi_1^{\top}\Psi_1 = \mathbf{X}\mathcal{P}_{\Omega}^{\top}(\mathbf{Y}) \quad (5.25)$$

在这里, 可采用共轭梯度法对该矩阵方程进行求解。

解说 3 (共轭梯度法求解线性方程组). 共轭梯度法是一种经典的数值计算方法, 主要用于求解线性方程组, 在机器学习中有诸多应用。从算法结构上来看, 共轭梯度法是一种迭代算法, 当线性方程组的规模较大且存在稀疏性问题时, 共轭梯度法便可派上用场。

在线性代数中, 线性方程组的表达式为

$$\mathbf{A}\mathbf{x} = \mathbf{b} \quad (5.26)$$

其中, 矩阵 $\mathbf{A} \in \mathbb{R}^{n \times n}$ 与向量 $\mathbf{b} \in \mathbb{R}^n$ 已知, $\mathbf{x} \in \mathbb{R}^n$ 为待求解变量。

不妨采用共轭梯度法对该线性方程组进行求解:

1. 明确线性方程组中的矩阵 \mathbf{A} 为实对称矩阵和正定矩阵, 对变量 \mathbf{x} 进行初始化 (可令向量的所有元素均为 0), 记作 \mathbf{x}_0 ;
2. 计算残差向量 $\mathbf{r}_0 := \mathbf{b} - \mathbf{A}\mathbf{x}_0$;
3. 令 $\mathbf{q}_0 := \mathbf{r}_0$;
4. 定义迭代过程, 令 $\ell = 0$:

- 计算系数

$$\alpha_{\ell} := \frac{\mathbf{r}_{\ell}^{\top}\mathbf{r}_{\ell}}{\mathbf{q}_{\ell}^{\top}\mathbf{A}\mathbf{q}_{\ell}} \quad (5.27)$$

- 更新变量

$$\mathbf{x}_{\ell+1} := \mathbf{x}_{\ell} + \alpha_{\ell}\mathbf{q}_{\ell} \quad (5.28)$$

- 更新变量

$$\mathbf{r}_{\ell+1} := \mathbf{r}_{\ell} - \alpha_{\ell}\mathbf{A}\mathbf{q}_{\ell} \quad (5.29)$$

- 判断: 若此时残差 $\mathbf{r}_{\ell+1}$ 足够小, 可终止循环;

- 计算系数

$$\beta_{\ell} := \frac{\mathbf{r}_{\ell+1}^{\top}\mathbf{r}_{\ell+1}}{\mathbf{r}_{\ell}^{\top}\mathbf{r}_{\ell}} \quad (5.30)$$

- 更新变量

$$\mathbf{q}_{\ell+1} := \mathbf{r}_{\ell+1} + \beta_{\ell}\mathbf{q}_{\ell} \quad (5.31)$$

- $\ell := \ell + 1$

5. 输出最终迭代结果 $\mathbf{x}_{\ell+1}$ 作为线性方程组的近似解。

假设线性方程组 $Ax = b$ 中已知 $A = \begin{bmatrix} 4 & 1 \\ 1 & 3 \end{bmatrix}$ 与 $b = \begin{bmatrix} 1 \\ 2 \end{bmatrix}$ ，即如下二元一次方程组：

$$\begin{cases} 4x_1 + x_2 = 1 \\ x_1 + 3x_2 = 2 \end{cases} \quad (5.32)$$

图5.9给出了共轭梯度法求解二元一次方程组的迭代过程，其中，两条直线的交点 $(\frac{1}{11}, \frac{7}{11})$ 为方程组的解。在迭代过程中，变量 $\mathbf{0}$ 的初始值为坐标原点，第二次迭代结果恰好落在交点上，结果为

$$\mathbf{x} = \begin{bmatrix} 0.09090909 \\ 0.63636364 \end{bmatrix} \approx \begin{bmatrix} \frac{1}{11} \\ \frac{7}{11} \end{bmatrix} \quad (5.33)$$

需要注意的是，共轭梯度法具有快速收敛的性质，一般仅需少量迭代次数就可得到精准的近似解。

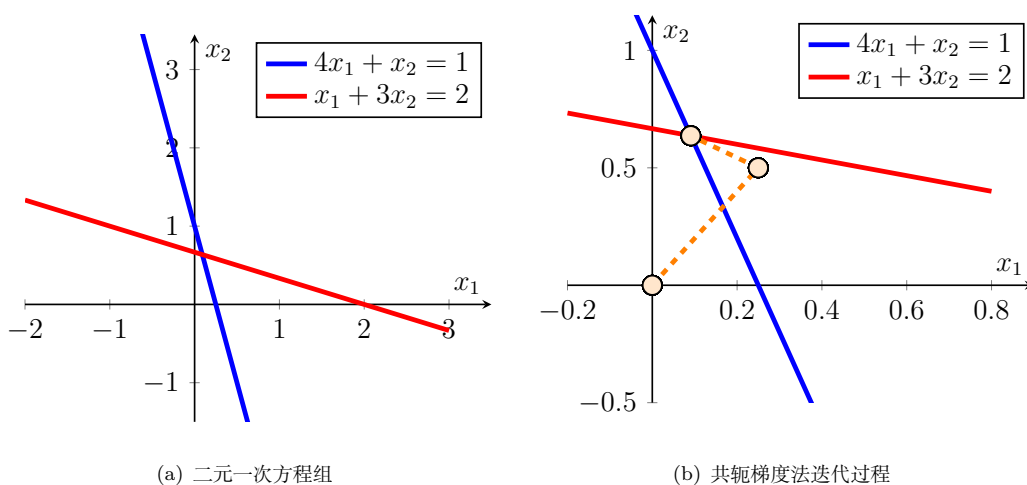


图 5.9: 使用共轭梯度法求解二元一次方程组

```
import numpy as np

def conjugate_grad(A, b, maxiter = 5):
    n = A.shape[0]
    x = np.zeros(n)
    r = b - A @ x
    q = r.copy()
    r_old = np.inner(r, r)
    for it in range(maxiter):
        alpha = r_old / np.inner(q, A @ q)
        x += alpha * q
        r -= alpha * A @ q
        r_new = np.inner(r, r)
        if np.sqrt(r_new) < 1e-10:
            break
        beta = r_new / r_old
        q = r + beta * q
        r_old = r_new.copy()
    return x

A = np.array([[4, 1], [1, 3]])
b = np.array([1, 2])
x = conjugate_grad(A, b)
```

例 35. 试写出求解公式(5.25)所示矩阵方程的共轭梯度法。

解. 对公式(5.25)左边进行向量化操作, 并记作

$$g(\mathbf{W}) = \text{vec}(\mathbf{X}\mathcal{P}_{\Omega}^{\top}(\mathbf{W}^{\top}\mathbf{X}) + \rho\mathbf{W} + \lambda\mathbf{W}\Psi_1^{\top}\Psi_1) \quad (5.34)$$

共轭梯度法的具体过程如下:

1. 对变量 \mathbf{W} 进行向量化操作, 并初始化为 \mathbf{w}_0 ;
2. 计算残差向量 $\mathbf{r}_0 := \text{vec}(\mathbf{X}\mathcal{P}_{\Omega}^{\top}(\mathbf{Y})) - g(\mathbf{W}_0)$;
3. 令 $\mathbf{q}_0 := \mathbf{r}_0$;
4. 定义迭代过程, 令 $\ell = 0$:

- 计算系数

$$\alpha_{\ell} := \frac{\mathbf{r}_{\ell}^{\top}\mathbf{r}_{\ell}}{\mathbf{q}_{\ell}^{\top}g(\mathbf{Q}_{\ell})} \quad (5.35)$$

其中, \mathbf{Q}_{ℓ} 是 \mathbf{q}_{ℓ} 的矩阵化结果;

- 更新变量

$$\mathbf{w}_{\ell+1} := \mathbf{w}_{\ell} + \alpha_{\ell}\mathbf{q}_{\ell} \quad (5.36)$$

- 更新变量

$$\mathbf{r}_{\ell+1} := \mathbf{r}_{\ell} - \alpha_{\ell}g(\mathbf{Q}_{\ell}) \quad (5.37)$$

- 判断: 若此时残差 $\mathbf{r}_{\ell+1}$ 足够小, 可终止循环;
- 计算系数

$$\beta_{\ell} := \frac{\mathbf{r}_{\ell+1}^{\top}\mathbf{r}_{\ell+1}}{\mathbf{r}_{\ell}^{\top}\mathbf{r}_{\ell}} \quad (5.38)$$

- 更新变量

$$\mathbf{q}_{\ell+1} := \mathbf{r}_{\ell+1} + \beta_{\ell}\mathbf{q}_{\ell} \quad (5.39)$$

- $\ell := \ell + 1$

5. 输出最终迭代结果 $\mathbf{w}_{\ell+1}$ 并进行矩阵化, 得到的矩阵 $\mathbf{W}_{\ell+1}$ 即作为矩阵方程的近似解。

更新变量 \mathbf{X}

对变量 \mathbf{X} 求偏导数, 有

$$\frac{\partial f}{\partial \mathbf{X}} = -\mathbf{W}\mathcal{P}_{\Omega}(\mathbf{Y} - \mathbf{W}^{\top}\mathbf{X}) + \rho\mathbf{X} + \lambda\mathbf{X}\Psi_2^{\top}\Psi_2 \quad (5.40)$$

此时, 令 $\frac{\partial f}{\partial \mathbf{X}} = \mathbf{0}$, 则可得到如下矩阵方程:

$$\mathbf{W}\mathcal{P}_{\Omega}(\mathbf{W}^{\top}\mathbf{X}) + \rho\mathbf{X} + \lambda\mathbf{X}\Psi_2^{\top}\Psi_2 = \mathbf{W}\mathcal{P}_{\Omega}(\mathbf{Y}) \quad (5.41)$$

在这里, 同样可采用共轭梯度法对该矩阵方程进行求解。

例 36. 试写出求解公式(5.41)所示矩阵方程的共轭梯度法。

解. 对公式(5.41)左边进行向量化操作, 并记作

$$g(\mathbf{X}) = \text{vec}(\mathbf{W}\mathcal{P}_{\Omega}(\mathbf{W}^{\top}\mathbf{X}) + \rho\mathbf{X} + \lambda\mathbf{X}\Psi_2^{\top}\Psi_2) \quad (5.42)$$

共轭梯度法的具体过程如下:

1. 对变量 \mathbf{X} 进行向量化操作, 并初始化为 \mathbf{x}_0 ;
2. 计算残差向量 $\mathbf{r}_0 := \text{vec}(\mathbf{W}\mathcal{P}_\Omega(\mathbf{Y})) - g(\mathbf{X}_0)$;
3. 令 $\mathbf{q}_0 := \mathbf{r}_0$;
4. 定义迭代过程, 令 $\ell = 0$:

- 计算系数

$$\alpha_\ell := \frac{\mathbf{r}_\ell^\top \mathbf{r}_\ell}{\mathbf{q}_\ell^\top g(\mathbf{Q}_\ell)} \quad (5.43)$$

其中, \mathbf{Q}_ℓ 是 \mathbf{q}_ℓ 的矩阵化结果;

- 更新变量

$$\mathbf{x}_{\ell+1} := \mathbf{x}_\ell + \alpha_\ell \mathbf{q}_\ell \quad (5.44)$$

- 更新变量

$$\mathbf{r}_{\ell+1} := \mathbf{r}_\ell - \alpha_\ell g(\mathbf{Q}_\ell) \quad (5.45)$$

- 判断: 若此时残差 $\mathbf{r}_{\ell+1}$ 足够小, 可终止循环;

- 计算系数

$$\beta_\ell := \frac{\mathbf{r}_{\ell+1}^\top \mathbf{r}_{\ell+1}}{\mathbf{r}_\ell^\top \mathbf{r}_\ell} \quad (5.46)$$

- 更新变量

$$\mathbf{q}_{\ell+1} := \mathbf{r}_{\ell+1} + \beta_\ell \mathbf{q}_\ell \quad (5.47)$$

- $\ell := \ell + 1$

5. 输出最终迭代结果 $\mathbf{x}_{\ell+1}$ 并进行矩阵化, 得到的矩阵 $\mathbf{X}_{\ell+1}$ 即作为矩阵方程的近似解。

算法2给出了考虑平滑处理的矩阵分解算法的具体实现过程。

Algorithm 2 考虑平滑处理的矩阵分解算法

Input: 观测矩阵 $\mathbf{Y} \in \mathbb{R}^{N \times T}$, 被观测元素的索引集合 Ω , 超参数 $\{\rho, \lambda\}$ 。

Output: 重构出来的矩阵 $\hat{\mathbf{Y}} \in \mathbb{R}^{N \times T}$ 。

- 1: 对变量 $\{\mathbf{W}, \mathbf{X}\}$ 进行初始化;
 - 2: **for** $i = 0$ to 最大迭代次数 **do**
 - 3: 使用共轭梯度法对公式(5.25)中的变量 \mathbf{W} 进行求解;
 - 4: 使用共轭梯度法对公式(5.41)中的变量 \mathbf{X} 进行求解;
 - 5: 计算 $\hat{\mathbf{Y}} = \mathbf{W}^\top \mathbf{X}$;
 - 6: **end for**
-

例 37. 给定如图 5.2 所示的稀疏速度场, 试使用平滑矩阵分解重构速度场。

解. 将平滑矩阵分解的秩设置为 $R = 10$ 、正则项系数设置为 $\rho = 10$ 、迭代次数为 200 代, 令平滑正则项系数分别为 $\lambda = 10, 10^2$, 使用 *Python* 实现算法即可得到如图 5.10 所示的重构速度场, 在图中, 平滑矩阵分解的误差为

- 当 $\lambda = 10$ 时, 有 $MAPE = 44.06\%$ 、 $RMSE = 2.16$;
- 当 $\lambda = 10^2$ 时, 有 $MAPE = 48.00\%$ 、 $RMSE = 1.60$ 。

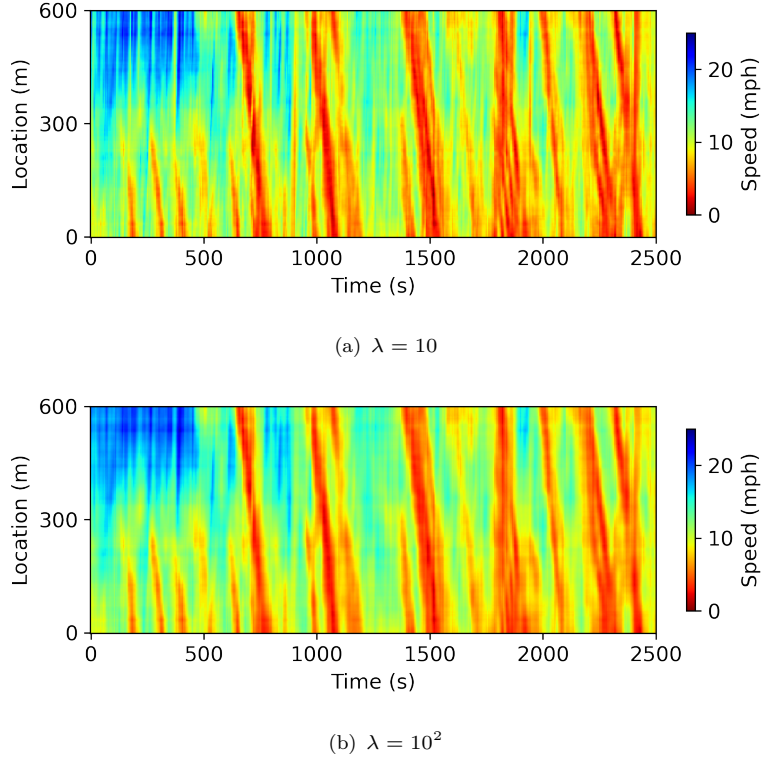


图 5.10: 平滑矩阵分解所得到的重构速度场

5.3 时序矩阵分解

5.3.1 模型表达式

在该矩阵分解的优化问题中, 可通过增加向量自回归过程使得模型具备时序建模能力。不妨对时序因子矩阵构造向量自回归过程, 可得到时序矩阵分解的优化问题为

$$\min_{\mathbf{W}, \mathbf{X}, \{\mathbf{A}_k\}} \frac{1}{2} \|\mathcal{P}_\Omega(\mathbf{Y} - \mathbf{W}^\top \mathbf{X})\|_F^2 + \frac{\rho}{2} (\|\mathbf{W}\|_F^2 + \|\mathbf{X}\|_F^2) + \frac{\lambda}{2} \sum_{t=d+1}^T \left\| \mathbf{x}_t - \sum_{k=1}^d \mathbf{A}_k \mathbf{x}_{t-k} \right\|_2^2 \quad (5.48)$$

其中, $\mathbf{A}_1, \mathbf{A}_2, \dots, \mathbf{A}_d \in \mathbb{R}^{R \times R}$ 为向量自回归过程的系数矩阵; $\{\rho, \lambda\}$ 为正则项的权重系数。

根据向量自回归的定义, 时序矩阵分解的优化问题可写作如下形式:

$$\min_{\mathbf{W}, \mathbf{X}, \{\mathbf{A}_k\}} \frac{1}{2} \|\mathcal{P}_\Omega(\mathbf{Y} - \mathbf{W}^\top \mathbf{X})\|_F^2 + \frac{\rho}{2} (\|\mathbf{W}\|_F^2 + \|\mathbf{X}\|_F^2) + \frac{\lambda}{2} \left\| \mathbf{X} \mathbf{\Psi}_0^\top - \sum_{k=1}^d \mathbf{A}_k \mathbf{X} \mathbf{\Psi}_k^\top \right\|_F^2 \quad (5.49)$$

其中, $\mathbf{\Psi}_k \in \mathbb{R}^{(T-d) \times T}$, $k = 0, 1, \dots, d$ 为构造出来的矩阵 (参见公式(5.52))。

解说 4 (向量自回归). 对于多元时间序列, 若任意时刻 t 对应的观测数据为向量 $\mathbf{x}_t \in \mathbb{R}^N$, 则向量自回归的表达式为

$$\mathbf{x}_t = \sum_{k=1}^d \mathbf{A}_k \mathbf{x}_{t-k} + \boldsymbol{\epsilon}_t, \quad t = d+1, d+2, \dots, T \quad (5.50)$$

其中, $\mathbf{A}_1, \mathbf{A}_2, \dots, \mathbf{A}_d \in \mathbb{R}^{N \times N}$ 为自回归过程的系数矩阵 (*coefficient matrix*); d 为自回归过程的阶数 (*order*); $\boldsymbol{\epsilon}_t \in \mathbb{R}^N$ 为残差向量。

令

$$\mathbf{X} = \begin{bmatrix} | & | & \cdots & | \\ \mathbf{x}_1 & \mathbf{x}_2 & \cdots & \mathbf{x}_T \\ | & | & \cdots & | \end{bmatrix} \in \mathbb{R}^{N \times T} \quad (5.51)$$

若构造分块矩阵

$$\begin{aligned}\Psi_k &= \begin{bmatrix} 0 & \cdots & 0 & 1 & \cdots & 0 & 0 & \cdots & 0 \\ \vdots & \ddots & \vdots & \vdots & \ddots & \vdots & \vdots & \ddots & \vdots \\ 0 & \cdots & 0 & 0 & \cdots & 1 & 0 & \cdots & 0 \end{bmatrix} \\ &= \underbrace{\begin{bmatrix} 0 & \cdots & 0 \end{bmatrix}}_{d-k} \underbrace{\begin{bmatrix} 1 & \cdots & 0 \end{bmatrix}}_{T-d} \underbrace{\begin{bmatrix} 0 & \cdots & 0 \end{bmatrix}}_k \\ &= \begin{bmatrix} \mathbf{0}_{(T-d) \times (d-k)} & \mathbf{I}_{T-d} & \mathbf{0}_{(T-d) \times k} \end{bmatrix} \in \mathbb{R}^{(T-d) \times T}, k = 0, 1, \dots, d\end{aligned}\quad (5.52)$$

则向量自回归可写作如下形式:

$$\mathbf{X}\Psi_0^\top = \sum_{k=1}^d \mathbf{A}_k \mathbf{X}\Psi_k^\top + \mathbf{E} \quad (5.53)$$

其中, $\mathbf{E} \in \mathbb{R}^{N \times (T-d)}$ 为残差矩阵。

5.3.2 求解过程

为了估计优化问题中的待定参数, 即变量 \mathbf{W} 、 \mathbf{X} 以及 $\mathbf{A}_1, \mathbf{A}_2, \dots, \mathbf{A}_d$, 可采用交替优化算法。在时序矩阵分解中, 每次更新特定变量时, 可令其他变量固定不变, 仅求解当前变量的最优解 (如最小二乘解) 或近似解。

更新变量 \mathbf{W}

不妨将时序矩阵分解优化问题的目标函数记作 f , 对变量 \mathbf{W} 求偏导数, 有

$$\frac{\partial f}{\partial \mathbf{W}} = -\mathbf{X}\mathcal{P}_\Omega^\top(\mathbf{Y} - \mathbf{W}^\top \mathbf{X}) + \rho \mathbf{W} \quad (5.54)$$

此时, 令 $\frac{\partial f}{\partial \mathbf{W}} = \mathbf{0}$, 则矩阵方程为

$$\mathbf{X}\mathcal{P}_\Omega^\top(\mathbf{W}^\top \mathbf{X}) + \rho \mathbf{W} = \mathbf{X}\mathcal{P}_\Omega^\top(\mathbf{Y}) \quad (5.55)$$

对于变量 \mathbf{W} , 该矩阵方程的最小二乘解为

$$\mathbf{w}_i = \left(\sum_{t:(i,t) \in \Omega} \mathbf{x}_t \mathbf{x}_t^\top + \rho \mathbf{I}_R \right)^{-1} \sum_{t:(i,t) \in \Omega} \mathbf{x}_t y_{i,t} \quad (5.56)$$

其中, $i = 1, 2, \dots, N$; 符号 $\sum_{t:(i,t) \in \Omega}$ 表示固定索引 i 后对索引集合 Ω 内所有索引 t 进行求和。

更新变量 \mathbf{X}

令 $\mathbf{A}_0 = -\mathbf{I}_R$, 则时序矩阵分解优化问题的目标函数可写作如下形式:

$$f = \frac{1}{2} \|\mathcal{P}_\Omega(\mathbf{Y} - \mathbf{W}^\top \mathbf{X})\|_F^2 + \frac{\rho}{2} (\|\mathbf{W}\|_F^2 + \|\mathbf{X}\|_F^2) + \frac{\lambda}{2} \left\| \sum_{k=0}^d \mathbf{A}_k \mathbf{X}\Psi_k^\top \right\|_F^2 \quad (5.57)$$

对变量 \mathbf{X} 求偏导数, 有

$$\frac{\partial f}{\partial \mathbf{X}} = -\mathbf{W}\mathcal{P}_\Omega(\mathbf{Y} - \mathbf{W}^\top \mathbf{X}) + \rho \mathbf{X} + \lambda \sum_{k=0}^d \mathbf{A}_k^\top \left(\sum_{h=0}^d \mathbf{A}_h \mathbf{X}\Psi_h^\top \right) \Psi_k \quad (5.58)$$

此时, 令 $\frac{\partial f}{\partial \mathbf{X}} = \mathbf{0}$, 关于变量 \mathbf{X} 的矩阵方程为

$$\mathbf{W}\mathcal{P}_\Omega(\mathbf{Y} - \mathbf{W}^\top \mathbf{X}) + \rho \mathbf{X} + \lambda \sum_{k=0}^d \mathbf{A}_k^\top \left(\sum_{h=0}^d \mathbf{A}_h \mathbf{X}\Psi_h^\top \right) \Psi_k = \mathbf{W}\mathcal{P}_\Omega(\mathbf{Y}) \quad (5.59)$$

例 38. 试写出求解公式(5.59)所示矩阵方程的共轭梯度法。

解. 对公式(5.41)左边进行向量化操作, 并记作

$$g(\mathbf{X}) = \text{vec} \left(\mathbf{W} \mathcal{P}_\Omega(\mathbf{W}^\top \mathbf{X}) + \rho \mathbf{X} + \lambda \sum_{k=0}^d \mathbf{A}_k^\top \left(\sum_{h=0}^d \mathbf{A}_h \mathbf{X} \Psi_h^\top \right) \Psi_k \right) \quad (5.60)$$

共轭梯度法的具体过程可参考例36。

更新系数矩阵 $\mathbf{A}_1, \mathbf{A}_2, \dots, \mathbf{A}_d$

若令

$$\begin{aligned} \mathbf{A} &= \begin{bmatrix} \mathbf{A}_1 & \mathbf{A}_2 & \cdots & \mathbf{A}_d \end{bmatrix} \in \mathbb{R}^{R \times (dR)} \\ \Psi &= \begin{bmatrix} \Psi_1 & \Psi_2 & \cdots & \Psi_d \end{bmatrix} \in \mathbb{R}^{(T-d) \times (dT)} \end{aligned} \quad (5.61)$$

则对于系数矩阵 \mathbf{A} 的子问题可改写为

$$\min_{\mathbf{A}} \frac{1}{2} \|\mathbf{X} \Psi_0^\top - \mathbf{A}(\mathbf{I}_d \otimes \mathbf{X}) \Psi^\top\|_F^2 \quad (5.62)$$

因此, 系数矩阵 \mathbf{A} 的最小二乘解为

$$\mathbf{A} = \mathbf{X} \Psi_0^\top ((\mathbf{I}_d \otimes \mathbf{X}) \Psi^\top)^\dagger \quad (5.63)$$

例 39. 试证明

$$\sum_{k=1}^d \mathbf{A}_k \mathbf{X} \Psi_k^\top = \mathbf{A}(\mathbf{I}_d \otimes \mathbf{X}) \Psi^\top \quad (5.64)$$

解. 根据 *Kronecker* 定义, 有

$$\begin{aligned} \sum_{k=1}^d \mathbf{A}_k \mathbf{X} \Psi_k^\top &= \mathbf{A}_1 \mathbf{X} \Psi_1^\top + \mathbf{A}_2 \mathbf{X} \Psi_2^\top + \cdots + \mathbf{A}_d \mathbf{X} \Psi_d^\top \\ &= \begin{bmatrix} \mathbf{A}_1 & \mathbf{A}_2 & \cdots & \mathbf{A}_d \end{bmatrix} \begin{bmatrix} \mathbf{X} & \mathbf{0} & \cdots & \mathbf{0} \\ \mathbf{0} & \mathbf{X} & \cdots & \mathbf{0} \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{0} & \mathbf{0} & \cdots & \mathbf{X} \end{bmatrix} \begin{bmatrix} \Psi_1^\top \\ \Psi_2^\top \\ \vdots \\ \Psi_d^\top \end{bmatrix} \\ &= \begin{bmatrix} \mathbf{A}_1 & \mathbf{A}_2 & \cdots & \mathbf{A}_d \end{bmatrix} (\mathbf{I}_d \otimes \mathbf{X}) \begin{bmatrix} \Psi_1^\top \\ \Psi_2^\top \\ \vdots \\ \Psi_d^\top \end{bmatrix} \\ &= \mathbf{A}(\mathbf{I}_d \otimes \mathbf{X}) \Psi^\top \end{aligned} \quad (5.65)$$

恒成立。

算法3给出了时序矩阵分解算法的具体实现过程。在这里, 不妨使用共轭梯度法对变量 \mathbf{W} 与 \mathbf{X} 进行求解。

5.3.3 时间序列预测

Algorithm 3 时序矩阵分解算法

Input: 观测矩阵 $\mathbf{Y} \in \mathbb{R}^{N \times T}$, 被观测元素的索引集合 Ω , 向量自回归的阶数 d , 超参数 $\{\rho, \lambda\}$ 。

Output: 重构出来的矩阵 $\hat{\mathbf{Y}} \in \mathbb{R}^{N \times T}$ 。

- 1: 对变量 $\{\mathbf{W}, \mathbf{X}, \mathbf{A}\}$ 进行初始化;
 - 2: **for** $i = 0$ to 最大迭代次数 **do**
 - 3: 使用共轭梯度法对公式(5.55)中的变量 \mathbf{W} 进行求解;
 - 4: 使用共轭梯度法对公式(5.59)中的变量 \mathbf{X} 进行求解;
 - 5: 根据公式(5.63)计算系数矩阵 \mathbf{A} ;
 - 6: 计算 $\hat{\mathbf{Y}} = \mathbf{W}^\top \mathbf{X}$;
 - 7: **end for**
-

第六章 贝叶斯张量分解

本章将要介绍的内容包括：

- 基于高斯分布假设的 CP 分解表达式
- 根据共轭先验的设置规则构建贝叶斯张量分解的贝叶斯网络
- 采用贝叶斯推断方法对贝叶斯张量分解进行求解

6.1 基于高斯假设的 CP 分解表达式

对于任意三阶张量 $\mathcal{Y} \in \mathbb{R}^{M \times N \times T}$ ，若 Ω 表示被观测元素的索引集合，则 CP 张量分解的优化问题可写作如下形式：

$$\min_{\mathbf{U}, \mathbf{V}, \mathbf{X}} \frac{1}{2} \left\| \mathcal{P}_{\Omega} \left(\mathcal{Y} - \sum_{r=1}^R \mathbf{u}_r \otimes_{\text{outer}} \mathbf{v}_r \otimes_{\text{outer}} \mathbf{x}_r \right) \right\|_F^2 + \frac{\rho}{2} (\|\mathbf{U}\|_F^2 + \|\mathbf{V}\|_F^2 + \|\mathbf{X}\|_F^2) \quad (6.1)$$

其中， $\mathbf{U} \in \mathbb{R}^{M \times R}$ 、 $\mathbf{V} \in \mathbb{R}^{N \times R}$ 与 $\mathbf{X} \in \mathbb{R}^{T \times R}$ 为因子矩阵； $\mathbf{u}_r \in \mathbb{R}^M$ 、 $\mathbf{v}_r \in \mathbb{R}^N$ 与 $\mathbf{x}_r \in \mathbb{R}^T$ 分别为这些因子矩阵的列向量。

需要注意的是，这一形式等价于

$$\min_{\mathbf{U}, \mathbf{V}, \mathbf{X}} \frac{1}{2} \sum_{(i,j,t) \in \Omega} \left(y_{i,j,t} - \sum_{r=1}^R u_{i,r} v_{j,r} x_{t,r} \right)^2 + \frac{\rho}{2} (\|\mathbf{U}\|_F^2 + \|\mathbf{V}\|_F^2 + \|\mathbf{X}\|_F^2) \quad (6.2)$$

其中， $y_{i,j,t}$ 为张量 \mathcal{Y} 的第 (i, j, t) 个元素； $u_{i,r}$ 为因子矩阵 \mathbf{U} 的第 (i, r) 个元素； $v_{j,r}$ 为因子矩阵 \mathbf{V} 的第 (j, r) 个元素； $x_{t,r}$ 为因子矩阵 \mathbf{X} 的第 (t, r) 个元素。

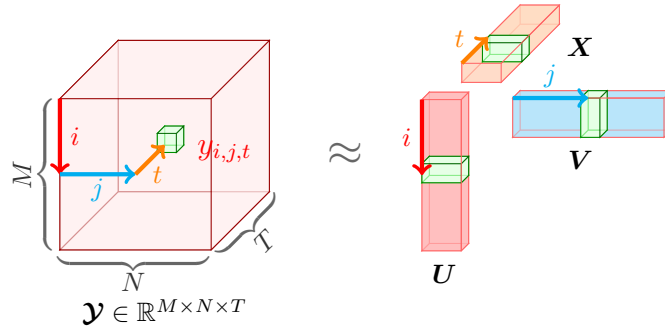


图 6.1: 三阶张量 $\mathcal{Y} \in \mathbb{R}^{M \times N \times T}$ 的 CP 分解

实际上，贝叶斯张量分解的发展在很大程度上得益于先前的各类贝叶斯矩阵分解模型 [Salakhutdinov and Mnih, 2008, Xiong et al., 2010, Zhao et al., 2015, Chen et al., 2019]。通常来说，贝叶斯张量分解需要建立在特定分布假设的基础上，若以高斯分布作为张量元素

的分布假设, 则 CP 分解为

$$y_{i,j,t} \sim \mathcal{N}\left(\sum_{r=1}^R u_{i,r} v_{j,r} x_{t,r}, \tau^{-1}\right), \forall (i, j, t) \quad (6.3)$$

其中, 符号 $\mathcal{N}(\cdot)$ 是高斯分布的简写; 低秩张量 $\sum_{r=1}^R u_{i,r} v_{j,r} x_{t,r}$ 作用于高斯分布的均值项, 恰好符合对 CP 分解逼近模型的定义; 参数 τ 是高斯分布中方差的倒数, 在这里表示精度项。

解说 5 (高斯分布). 高斯分布是统计学中最为常用的分布形式之一, 对于随机变量 x , 其概率密度函数为

$$\mathcal{N}(x | \mu, \tau^{-1}) = \sqrt{\frac{\tau}{2\pi}} \exp\left(-\frac{1}{2}\tau(x - \mu)^2\right) \quad (6.4)$$

其中, μ 是高斯分布的均值项; τ 是高斯分布的精度项。

对于多元随机变量 $\mathbf{x} \in \mathbb{R}^n$, 多元高斯分布的概率密度函数为

$$\mathcal{N}(\mathbf{x} | \boldsymbol{\mu}, \boldsymbol{\Lambda}^{-1}) = \sqrt{\frac{\det(\boldsymbol{\Lambda})}{2\pi}} \exp\left(-\frac{1}{2}(\mathbf{x} - \boldsymbol{\mu})^\top \boldsymbol{\Lambda}(\mathbf{x} - \boldsymbol{\mu})\right) \quad (6.5)$$

其中, $\boldsymbol{\mu} \in \mathbb{R}^n$ 是刻画随机变量 \mathbf{x} 多元高斯分布的均值向量; $\boldsymbol{\Lambda} \in \mathbb{R}^{n \times n}$ 是刻画随机变量 \mathbf{x} 多元高斯分布中协方差矩阵的逆矩阵; 符号 $\det(\cdot)$ 表示矩阵的行列式。

在这里, 高斯分布一方面能刻画实际数据的随机性, 另一方面也能契合张量分解优化问题的目标函数。在公式(6.3)中, 高斯分布对应的概率密度函数的指数项为

$$-\frac{1}{2}\tau\left(y_{i,j,t} - \sum_{r=1}^R u_{i,r} v_{j,r} x_{t,r}\right)^2 \quad (6.6)$$

这与 CP 张量分解的目标函数中的第一项

$$\frac{1}{2} \sum_{(i,j,t) \in \Omega} \left(y_{i,j,t} - \sum_{r=1}^R u_{i,r} v_{j,r} x_{t,r}\right)^2 \quad (6.7)$$

在形式上存在一致性。所不同的是, 采用贝叶斯框架可将优化问题转化为后验推断问题。

6.2 贝叶斯网络

在张量分解的优化问题中, 为了从数据张量中学习出因子矩阵 $\{\mathbf{U}, \mathbf{V}, \mathbf{X}\}$, 根据贝叶斯准则, 需要对这些因子矩阵设置共轭先验, 从而可得到这些参数的后验分布, 其中, 后验分布正比于先验分布与似然函数的乘积。

6.2.1 共轭分布

解说 6 (共轭分布). 共轭分布是贝叶斯推断中十分重要的概念, 结合贝叶斯定理, 可将“共轭”理解为后验与先验是同一种分布。以高斯分布为例, 假设一组观测样本 x_1, x_2, \dots, x_n 独立同分布于高斯分布, 即 $x_i \sim \mathcal{N}(\mu, \sigma^2)$, $i = 1, 2, \dots, n$, 其中, 均值 μ 未知, 方差 σ^2 已知, 此时, 似然函数为

$$\begin{aligned} \mathcal{L}(x_1, x_2, \dots, x_n | \mu) &= \prod_{i=1}^n \frac{1}{\sqrt{2\pi\sigma}} \exp\left(-\frac{1}{2\sigma^2}(x_i - \mu)^2\right) \\ &\propto \exp\left(-\frac{1}{2\sigma^2}(x_i - \mu)^2\right) \end{aligned} \quad (6.8)$$

若要对均值 μ 进行推断, 不妨假设 μ 的先验分布为

$$\mu \sim \mathcal{N}(\mu_0, \sigma_0^2) = \frac{1}{\sqrt{2\pi}\sigma_0} \exp\left(-\frac{1}{2\sigma_0^2}(\mu - \mu_0)^2\right) \quad (6.9)$$

从而, 可推导出后验分布为

$$\begin{aligned} p(\mu \mid x_1, x_2, \dots, x_n, \mu_0, \sigma_0) &\propto p(\mu \mid \mu_0, \sigma_0) \mathcal{L}(x_1, x_2, \dots, x_n \mid \mu) \\ &\propto \exp\left(-\frac{1}{2\sigma^2} \sum_{i=1}^n (x_i - \mu)^2 + \frac{1}{2\sigma_0^2} (\mu - \mu_0)^2\right) \\ &\propto \exp\left(-\frac{1}{2} \left(\frac{1}{\sigma_0^2} + \frac{n}{\sigma^2}\right) \left(\mu - \frac{\frac{\mu_0}{\sigma_0^2} + \frac{n\bar{x}}{\sigma^2}}{\frac{1}{\sigma_0^2} + \frac{n}{\sigma^2}}\right)^2\right) \end{aligned} \quad (6.10)$$

其中, $\bar{x} = \frac{1}{n} \sum_{i=1}^n x_i$ 。

因此, 后验分布满足高斯分布形式, 其均值与方差分别为

$$\frac{\frac{\mu_0}{\sigma_0^2} + \frac{n\bar{x}}{\sigma^2}}{\frac{1}{\sigma_0^2} + \frac{n}{\sigma^2}} \quad (6.11)$$

与

$$\left(\frac{1}{\sigma_0^2} + \frac{n}{\sigma^2}\right) \quad (6.12)$$

6.2.2 因子矩阵的先验分布

假设张量 \mathcal{Y} 的任意元素服从独立的高斯分布, 依据共轭先验设置准则, 因子矩阵的先验分布为高斯分布。考虑到因子矩阵是由 R 个列向量构成, 这些向量被假设服从多元高斯分布, 具体形式为

$$\begin{cases} \mathbf{u}_i \sim \mathcal{N}(\boldsymbol{\mu}_u, \boldsymbol{\Lambda}_u^{-1}), \forall i \in \{1, 2, \dots, M\} \\ \mathbf{v}_j \sim \mathcal{N}(\boldsymbol{\mu}_v, \boldsymbol{\Lambda}_v^{-1}), \forall j \in \{1, 2, \dots, N\} \\ \mathbf{x}_t \sim \mathcal{N}(\boldsymbol{\mu}_x, \boldsymbol{\Lambda}_x^{-1}), \forall t \in \{1, 2, \dots, T\} \end{cases} \quad (6.13)$$

其中, 向量 $\mathbf{u}_i \in \mathbb{R}^R$ 为因子矩阵 $\mathbf{U} \in \mathbb{R}^{M \times R}$ 的第 i 行, 并被假设服从独立的多元高斯分布; 类似地, 向量 $\mathbf{v}_j \in \mathbb{R}^R$ 为因子矩阵 $\mathbf{V} \in \mathbb{R}^{N \times R}$ 的第 j 行, 向量 \mathbf{x}_t 为因子矩阵 $\mathbf{X} \in \mathbb{R}^{T \times R}$ 的第 t 行, 两者都服从多元高斯分布。

在因子矩阵的先验分布中, 参数 $\{(\boldsymbol{\mu}_u, \boldsymbol{\Lambda}_u), (\boldsymbol{\mu}_v, \boldsymbol{\Lambda}_v), (\boldsymbol{\mu}_x, \boldsymbol{\Lambda}_x)\}$ 是贝叶斯模型的超参数。为了避免贝叶斯模型出现过拟合现象, 同时增加贝叶斯模型的灵活性, 依据共轭先验设置准则, 可假设超参数 $\{\boldsymbol{\mu}_u, \boldsymbol{\mu}_v, \boldsymbol{\mu}_x\}$ 服从多元高斯分布、超参数 $\{\boldsymbol{\Lambda}_u, \boldsymbol{\Lambda}_v, \boldsymbol{\Lambda}_x\}$ 服从 Wishart 分布, 即

$$\begin{cases} \boldsymbol{\mu}_u \sim \mathcal{N}(\boldsymbol{\mu}_0, (\beta_0 \boldsymbol{\Lambda}_u)^{-1}) & \boldsymbol{\Lambda}_u \sim \mathcal{W}(\mathbf{W}_0, \nu_0) \\ \boldsymbol{\mu}_v \sim \mathcal{N}(\boldsymbol{\mu}_0, (\beta_0 \boldsymbol{\Lambda}_v)^{-1}) & \boldsymbol{\Lambda}_v \sim \mathcal{W}(\mathbf{W}_0, \nu_0) \\ \boldsymbol{\mu}_x \sim \mathcal{N}(\boldsymbol{\mu}_0, (\beta_0 \boldsymbol{\Lambda}_x)^{-1}) & \boldsymbol{\Lambda}_x \sim \mathcal{W}(\mathbf{W}_0, \nu_0) \end{cases} \quad (6.14)$$

其中, 符号 $\mathcal{W}(\cdot)$ 表示 Wishart 分布。在这里, 超参数的先验分布是由高斯分布与 Wishart 分布共同构成的, 先验分布中的 $\{\boldsymbol{\mu}_0, \beta_0, \mathbf{W}_0, \nu_0\}$ 为初始化参数, 在贝叶斯网络中无需设置共轭先验。

解说 7 (Wishart 分布). 在共轭分布中, Wishart 分布可用来描述多元高斯分布中协方差矩阵 $\boldsymbol{\Lambda} \in \mathbb{R}^{R \times R}$ 的先验分布, 其概率密度函数为

$$\mathcal{W}(\boldsymbol{\Lambda} \mid \mathbf{W}, \nu) = \frac{1}{C} \det(\boldsymbol{\Lambda})^{(\nu-r-1)/2} \exp\left(-\frac{1}{2} \text{tr}(\mathbf{W}^{-1} \boldsymbol{\Lambda})\right) \quad (6.15)$$

6.3 Gibbs 采样与后验推断

6.3.1 Gibbs 采样

Gibbs 采样是一种特殊的马尔可夫链蒙特卡洛 (Markov chain Monte Carlo, 通常记作 MCMC) 算法, 也被称为交替条件采样, 隶属于随机型贝叶斯推断算法。对于具体统计学习问题而言, Gibbs 采样的核心是贝叶斯准则, 围绕先验分布与似然函数推导后验分布, 相应的参数在迭代过程中会被交替使用。

解说 9 (Gibbs 采样的原理). 已知观测值为 \mathbf{y} , 给定参数 $\boldsymbol{\theta} = (\theta_1, \theta_2, \dots, \theta_d)^\top$, 若参数 $\theta_k^\ell, k = 1, 2, \dots, d$ 表示第 k 个参数 θ_k 在第 ℓ 次迭代的采样值, 则该采样值取自概率分布

$$p(\theta_k^\ell \mid \theta_1^\ell, \dots, \theta_{k-1}^\ell, \theta_{k+1}^{\ell-1}, \dots, \theta_d^{\ell-1}, \mathbf{y})$$

例 40. 假设观测值 $\mathbf{y} = (y_1, y_2)^\top$ 服从均值为 $\boldsymbol{\theta} = (\theta_1, \theta_2)^\top$ (未知)、协方差为 $\boldsymbol{\Sigma} = \begin{bmatrix} 1 & \rho \\ \rho & 1 \end{bmatrix}$ (已知), 且变量 $\boldsymbol{\theta}$ 服从均匀分布, 如何利用 Gibbs 采样估计 $\boldsymbol{\theta}$?

解. 根据贝叶斯准则, 后验分布 $p(\boldsymbol{\theta} \mid \mathbf{y})$ 通常被定义为

$$p(\boldsymbol{\theta} \mid \mathbf{y}) \propto p(\boldsymbol{\theta}) \times p(\mathbf{y} \mid \boldsymbol{\theta}) \quad (6.18)$$

其中, $p(\boldsymbol{\theta})$ 为先验分布, $p(\mathbf{y} \mid \boldsymbol{\theta})$ 是关于观测值的似然函数, 符号 \propto 表示正比于。

因此, 这里的后验分布为

$$p(\boldsymbol{\theta} \mid \mathbf{y}) = \mathcal{N}(\mathbf{y}, \boldsymbol{\Sigma}) \quad (6.19)$$

尽管 $\boldsymbol{\theta}$ 可直接通过后验分布得到, 但需要注意的是, Gibbs 采样要求参数 θ_1 与 θ_2 在迭代过程中交替采样, 相应的后验分布分别为

$$\begin{aligned} p(\theta_1^\ell \mid \theta_2^{\ell-1}, \mathbf{y}) &= \mathcal{N}(y_1 + \rho(\theta_2^{\ell-1} - y_2), 1 - \rho^2) \\ p(\theta_2^\ell \mid \theta_1^\ell, \mathbf{y}) &= \mathcal{N}(y_2 + \rho(\theta_1^\ell - y_1), 1 - \rho^2) \end{aligned} \quad (6.20)$$

6.3.2 因子矩阵的后验分布

在贝叶斯高斯张量分解模型中, 为了从部分观测的张量中学习出合理的分解结构, 得到因子矩阵 $\{\mathbf{U}, \mathbf{V}, \mathbf{X}\}$, 关键在于推导出这些因子矩阵的后验分布。

因子矩阵 \mathbf{U} 的后验分布

由于因子矩阵 $\mathbf{U} \in \mathbb{R}^{M \times R}$ 的先验分布直接作用于向量 $\mathbf{u}_i \in \mathbb{R}^R, i = 1, 2, \dots, M$, 而非整个因子矩阵, 因此, 推导因子矩阵 \mathbf{U} 的后验分布实际上是通过单独推导向量 \mathbf{u}_i 的后验分布完成的。根据张量元素的高斯分布假设, 即公式(6.3), 可得到关于向量 \mathbf{u}_i 的似然函数为

$$\begin{aligned} \mathcal{L}(\mathcal{Y} \mid \mathbf{u}_i, \mathbf{V}, \mathbf{X}, \tau) &\propto \prod_{j,t:(i,j,t) \in \Omega} \exp\left(-\frac{1}{2}\tau(y_{i,j,t} - \hat{y}_{i,j,t})^2\right) \\ &= \exp\left(-\frac{1}{2}\tau \sum_{j,t:(i,j,t) \in \Omega} \left(y_{i,j,t} - \sum_{r=1}^R u_{i,r} v_{j,r} x_{t,r}\right)^2\right) \\ &= \exp\left(-\frac{1}{2}\tau \sum_{j,t:(i,j,t) \in \Omega} \left(y_{i,j,t} - \mathbf{u}_i^\top (\mathbf{v}_j \circ \mathbf{x}_t)\right)^2\right) \end{aligned} \quad (6.21)$$

其中, $\mathcal{L}(\cdot)$ 用来表示似然函数; $\hat{y}_{i,j,t}$ 表示张量分解得到的估计值; 求积符号 $\prod_{j,t:(i,j,t) \in \Omega}$ 表示在特定索引 i 下, 索引集合 Ω 中任意 $j \in \{1, 2, \dots, N\}$ 与 $t \in \{1, 2, \dots, T\}$ 所对应的求积

公式；求和符号 $\sum_{j,t:(i,j,t) \in \Omega}$ 表示在特定索引 i 下，索引集合 Ω 中任意 $j \in \{1, 2, \dots, N\}$ 与 $t \in \{1, 2, \dots, T\}$ 所对应的求和公式；符号 \circ 表示元素间的点乘 (Hadamard product)。

对该似然函数左右两边同时取对数，有

$$\begin{aligned} \ln \mathcal{L}(\mathcal{Y} | \mathbf{u}_i, \mathbf{V}, \mathbf{X}, \tau) &= -\frac{1}{2}\tau \sum_{j,t:(i,j,t) \in \Omega} (y_{i,j,t} - \mathbf{u}_i^\top (\mathbf{v}_j \circ \mathbf{x}_t))^2 + C \\ &= -\frac{1}{2}\mathbf{u}_i^\top \left(\tau \sum_{j,t:(i,j,t) \in \Omega} (\mathbf{v}_j \circ \mathbf{x}_t) (\mathbf{v}_j \circ \mathbf{x}_t)^\top \right) \mathbf{u}_i \\ &\quad + \frac{1}{2}\mathbf{u}_i^\top \left(\tau \sum_{j,t:(i,j,t) \in \Omega} y_{i,j,t} (\mathbf{v}_j \circ \mathbf{x}_t) \right) + C \end{aligned} \quad (6.22)$$

其中， C 表示常数项，在后续内容中不再赘述。

向量 \mathbf{u}_i 的先验分布为多元高斯分布，其概率密度函数的对数形式为

$$\ln p(\mathbf{u}_i | \boldsymbol{\mu}_u, \boldsymbol{\Lambda}_u) = -\frac{1}{2}\mathbf{u}_i^\top \boldsymbol{\Lambda}_u \mathbf{u}_i + \frac{1}{2}\mathbf{u}_i^\top \boldsymbol{\Lambda}_u \boldsymbol{\mu}_u + C \quad (6.23)$$

根据贝叶斯准则，后验分布正比于先验分布与似然函数的乘积。由于向量 \mathbf{u}_i 的后验分布与先验分布都为多元高斯分布，因此，有

$$\begin{aligned} p(\mathbf{u}_i | \mathcal{Y}, \mathbf{V}, \mathbf{X}, \tau, \boldsymbol{\mu}_u, \boldsymbol{\Lambda}_u) &\propto \mathcal{N}(\mathbf{u}_i | \boldsymbol{\mu}_u, \boldsymbol{\Lambda}_u^{-1}) \times \mathcal{L}(\mathcal{Y} | \mathbf{u}_i, \mathbf{V}, \mathbf{X}, \tau) \\ &\propto \mathcal{N}(\mathbf{u}_i | \tilde{\boldsymbol{\mu}}_{\mathbf{u}_i}, \tilde{\boldsymbol{\Lambda}}_{\mathbf{u}_i}^{-1}) \end{aligned} \quad (6.24)$$

可推导出向量 \mathbf{u}_i 的后验分布参数分别为

$$\begin{cases} \tilde{\boldsymbol{\mu}}_{\mathbf{u}_i} = \tilde{\boldsymbol{\Lambda}}_{\mathbf{u}_i}^{-1} \left(\tau \sum_{j,t:(i,j,t) \in \Omega} y_{i,j,t} (\mathbf{v}_j \circ \mathbf{x}_t) + \boldsymbol{\Lambda}_u \boldsymbol{\mu}_u \right) \\ \tilde{\boldsymbol{\Lambda}}_{\mathbf{u}_i} = \tau \sum_{j,t:(i,j,t) \in \Omega} (\mathbf{v}_j \circ \mathbf{x}_t) (\mathbf{v}_j \circ \mathbf{x}_t)^\top + \boldsymbol{\Lambda}_u \end{cases} \quad (6.25)$$

在 Gibbs 采样中，有了向量 \mathbf{u}_i 的后验分布参数，即可对向量 \mathbf{u}_i 进行采样更新。

因子矩阵 \mathbf{V} 的后验分布

实际上，因子矩阵 \mathbf{V} 的后验分布推导过程与上述因子矩阵 \mathbf{U} 的后验分布推导过程如出一辙、原理一致。根据张量元素的高斯分布假设，即公式(6.3)，可得到关于向量 $\mathbf{v}_j \in \mathbb{R}^R, \forall j \in \{1, 2, \dots, N\}$ 的似然函数为

$$\begin{aligned} \mathcal{L}(\mathcal{Y} | \mathbf{U}, \mathbf{v}_j, \mathbf{X}, \tau) &\propto \prod_{i,t:(i,j,t) \in \Omega} \exp\left(-\frac{1}{2}\tau (y_{i,j,t} - \hat{y}_{i,j,t})^2\right) \\ &= \exp\left(-\frac{1}{2}\tau \sum_{i,t:(i,j,t) \in \Omega} (y_{i,j,t} - \mathbf{v}_j^\top (\mathbf{u}_i \circ \mathbf{x}_t))^2\right) \end{aligned} \quad (6.26)$$

其中，求积符号 $\prod_{i,t:(i,j,t) \in \Omega}$ 表示在特定索引 j 下，索引集合 Ω 中任意 $i \in \{1, 2, \dots, M\}$ 与 $t \in \{1, 2, \dots, T\}$ 所对应的求积公式；求和符号 $\sum_{i,t:(i,j,t) \in \Omega}$ 表示在特定索引 j 下，索引集合 Ω 中任意 $i \in \{1, 2, \dots, M\}$ 与 $t \in \{1, 2, \dots, T\}$ 所对应的求和公式。

对该似然函数左右两边同时取对数，有

$$\begin{aligned} \ln \mathcal{L}(\mathcal{Y} | \mathbf{U}, \mathbf{v}_j, \mathbf{X}, \tau) &= -\frac{1}{2}\tau \sum_{i,t:(i,j,t) \in \Omega} (y_{i,j,t} - \mathbf{v}_j^\top (\mathbf{u}_i \circ \mathbf{x}_t))^2 + C \\ &= -\frac{1}{2}\mathbf{v}_j^\top \left(\tau \sum_{i,t:(i,j,t) \in \Omega} (\mathbf{u}_i \circ \mathbf{x}_t) (\mathbf{u}_i \circ \mathbf{x}_t)^\top \right) \mathbf{v}_j \\ &\quad + \frac{1}{2}\mathbf{v}_j^\top \left(\tau \sum_{i,t:(i,j,t) \in \Omega} y_{i,j,t} (\mathbf{u}_i \circ \mathbf{x}_t) \right) + C \end{aligned} \quad (6.27)$$

向量 \mathbf{v}_j 的先验分布为多元高斯分布，其概率密度函数的对数形式为

$$\ln p(\mathbf{v}_j | \boldsymbol{\mu}_v, \boldsymbol{\Lambda}_v) = -\frac{1}{2} \mathbf{v}_j^\top \boldsymbol{\Lambda}_v \mathbf{v}_j + \frac{1}{2} \mathbf{v}_j^\top \boldsymbol{\Lambda}_v \boldsymbol{\mu}_v + C \quad (6.28)$$

将先验分布与似然函数进行相乘，由于 \mathbf{v}_j 的后验分布是多元高斯分布，与先验分布的分布形式保持一致，因此，有

$$\begin{aligned} p(\mathbf{v}_j | \mathcal{Y}, \mathbf{U}, \mathbf{X}, \tau, \boldsymbol{\mu}_v, \boldsymbol{\Lambda}_v) &\propto \mathcal{N}(\mathbf{v}_j | \boldsymbol{\mu}_v, \boldsymbol{\Lambda}_v^{-1}) \times \mathcal{L}(\mathcal{Y} | \mathbf{U}, \mathbf{v}_j, \mathbf{X}, \tau) \\ &\propto \mathcal{N}(\mathbf{v}_j | \tilde{\boldsymbol{\mu}}_{\mathbf{v}_j}, \tilde{\boldsymbol{\Lambda}}_{\mathbf{v}_j}^{-1}) \end{aligned} \quad (6.29)$$

可推导出向量 \mathbf{v}_j 的后验分布参数分别为

$$\begin{cases} \tilde{\boldsymbol{\mu}}_{\mathbf{v}_j} = \tilde{\boldsymbol{\Lambda}}_{\mathbf{v}_j}^{-1} \left(\tau \sum_{i,t:(i,j,t) \in \Omega} y_{i,j,t} (\mathbf{u}_i \circ \mathbf{x}_t) + \boldsymbol{\Lambda}_v \boldsymbol{\mu}_v \right) \\ \tilde{\boldsymbol{\Lambda}}_{\mathbf{v}_j} = \tau \sum_{i,t:(i,j,t) \in \Omega} (\mathbf{u}_i \circ \mathbf{x}_t) (\mathbf{u}_i \circ \mathbf{x}_t)^\top + \boldsymbol{\Lambda}_v \end{cases} \quad (6.30)$$

因子矩阵 \mathbf{X} 的后验分布

根据张量元素的高斯分布假设，即公式(6.3)，可得到关于向量 $\mathbf{x}_t, \forall t \in \{1, 2, \dots, T\}$ 的似然函数为

$$\begin{aligned} \mathcal{L}(\mathcal{Y} | \mathbf{U}, \mathbf{V}, \mathbf{x}_t, \tau) &\propto \prod_{i,j:(i,j,t) \in \Omega} \exp\left(-\frac{1}{2} \tau (y_{i,j,t} - \hat{y}_{i,j,t})^2\right) \\ &= \exp\left(-\frac{1}{2} \tau \sum_{i,j:(i,j,t) \in \Omega} (y_{i,j,t} - \mathbf{x}_t^\top (\mathbf{u}_i \circ \mathbf{v}_j))^2\right) \end{aligned} \quad (6.31)$$

其中，求积符号 $\prod_{i,j:(i,j,t) \in \Omega}$ 表示在特定索引 t 下，索引集合 Ω 中任意 $i \in \{1, 2, \dots, M\}$ 与 $j \in \{1, 2, \dots, N\}$ 所对应的求积公式；求和符号 $\sum_{i,j:(i,j,t) \in \Omega}$ 表示在特定索引 t 下，索引集合 Ω 中任意 $i \in \{1, 2, \dots, M\}$ 与 $j \in \{1, 2, \dots, N\}$ 所对应的求和公式。

对该似然函数左右两边同时取对数，有

$$\begin{aligned} \ln \mathcal{L}(\mathcal{Y} | \mathbf{U}, \mathbf{V}, \mathbf{x}_t, \tau) &= -\frac{1}{2} \tau \sum_{i,j:(i,j,t) \in \Omega} (y_{i,j,t} - \mathbf{x}_t^\top (\mathbf{u}_i \circ \mathbf{v}_j))^2 + C \\ &= -\frac{1}{2} \mathbf{x}_t^\top \left(\tau \sum_{i,j:(i,j,t) \in \Omega} (\mathbf{u}_i \circ \mathbf{v}_j) (\mathbf{u}_i \circ \mathbf{v}_j)^\top \right) \mathbf{x}_t \\ &\quad + \frac{1}{2} \mathbf{x}_t^\top \left(\tau \sum_{i,j:(i,j,t) \in \Omega} y_{i,j,t} (\mathbf{u}_i \circ \mathbf{v}_j) \right) + C \end{aligned} \quad (6.32)$$

向量 \mathbf{x}_t 的先验分布为多元高斯分布，其概率密度函数为

$$\ln p(\mathbf{x}_t | \boldsymbol{\mu}_x, \boldsymbol{\Lambda}_x) = -\frac{1}{2} \mathbf{x}_t^\top \boldsymbol{\Lambda}_x \mathbf{x}_t + \frac{1}{2} \mathbf{x}_t^\top \boldsymbol{\Lambda}_x \boldsymbol{\mu}_x + C \quad (6.33)$$

根据贝叶斯准则，可得

$$\begin{aligned} p(\mathbf{x}_t | \mathcal{Y}, \mathbf{U}, \mathbf{V}, \tau, \boldsymbol{\mu}_x, \boldsymbol{\Lambda}_x) &\propto \mathcal{N}(\mathbf{x}_t | \boldsymbol{\mu}_x, \boldsymbol{\Lambda}_x^{-1}) \times \mathcal{L}(\mathcal{Y} | \mathbf{U}, \mathbf{V}, \mathbf{x}_t, \tau) \\ &\propto \mathcal{N}(\mathbf{x}_t | \tilde{\boldsymbol{\mu}}_{\mathbf{x}_t}, \tilde{\boldsymbol{\Lambda}}_{\mathbf{x}_t}^{-1}) \end{aligned} \quad (6.34)$$

从而可推导出向量 \mathbf{x}_t 的后验分布参数分别为

$$\begin{cases} \tilde{\boldsymbol{\mu}}_{\mathbf{x}_t} = \tilde{\boldsymbol{\Lambda}}_{\mathbf{x}_t}^{-1} \left(\tau \sum_{i,j:(i,j,t) \in \Omega} y_{i,j,t} (\mathbf{u}_i \circ \mathbf{v}_j) + \boldsymbol{\Lambda}_x \boldsymbol{\mu}_x \right) \\ \tilde{\boldsymbol{\Lambda}}_{\mathbf{x}_t} = \tau \sum_{i,j:(i,j,t) \in \Omega} (\mathbf{u}_i \circ \mathbf{v}_j) (\mathbf{u}_i \circ \mathbf{v}_j)^\top + \boldsymbol{\Lambda}_x \end{cases} \quad (6.35)$$

6.3.3 超参数的后验分布

在图6.2所示的贝叶斯网络中, 待估计的超参数包括 $\{(\boldsymbol{\mu}_u, \boldsymbol{\Lambda}_u), (\boldsymbol{\mu}_v, \boldsymbol{\Lambda}_v), (\boldsymbol{\mu}_x, \boldsymbol{\Lambda}_x)\}$, 这些超参数作用于因子矩阵, 可保证因子矩阵在贝叶斯后验推断过程中学到尽可能多的隐性信息, 使贝叶斯高斯张量分解保持较高的灵活性。

超参数 $(\boldsymbol{\mu}_u, \boldsymbol{\Lambda}_u)$ 的后验分布

在如图6.2所示的贝叶斯网络中, 贝叶斯高斯张量分解的因子矩阵 $\mathbf{U} \in \mathbb{R}^{M \times R}$ 的所有行向量均被假设服从超参数为 $(\boldsymbol{\mu}_u, \boldsymbol{\Lambda}_u)$ 的多元高斯分布, 因此, 超参数 $(\boldsymbol{\mu}_u, \boldsymbol{\Lambda}_u)$ 的似然函数为

$$\begin{aligned} \mathcal{L}(\mathbf{U} \mid \boldsymbol{\mu}_u, \boldsymbol{\Lambda}_u) &\propto \det(\boldsymbol{\Lambda}_u)^{M/2} \prod_{i=1}^M \exp\left(-\frac{1}{2}(\mathbf{u}_i - \boldsymbol{\mu}_u)^\top \boldsymbol{\Lambda}_u (\mathbf{u}_i - \boldsymbol{\mu}_u)\right) \\ &\propto \det(\boldsymbol{\Lambda}_u)^{M/2} \exp\left(-\frac{1}{2} \sum_{i=1}^M (\mathbf{u}_i - \boldsymbol{\mu}_u)^\top \boldsymbol{\Lambda}_u (\mathbf{u}_i - \boldsymbol{\mu}_u)\right) \end{aligned} \quad (6.36)$$

在贝叶斯高斯张量分解的贝叶斯网络中, 超参数 $(\boldsymbol{\mu}_u, \boldsymbol{\Lambda}_u)$ 的共轭先验为高斯分布与 Wishart 分布构成的组合分布, 将两者写在一起, 则先验分布的概率密度函数为

$$\begin{aligned} p(\boldsymbol{\mu}_u, \boldsymbol{\Lambda}_u \mid \boldsymbol{\mu}_0, \mathbf{W}_0, \nu_0, \beta_0) &= \mathcal{N}(\boldsymbol{\mu}_u \mid \boldsymbol{\mu}_0, (\beta_0 \boldsymbol{\Lambda}_u)^{-1}) \times \mathcal{W}(\boldsymbol{\Lambda}_u \mid \mathbf{W}_0, \nu_0) \\ &\propto \det(\beta_0 \boldsymbol{\Lambda}_u)^{1/2} \exp\left(-\frac{1}{2} \beta_0 (\boldsymbol{\mu}_u - \boldsymbol{\mu}_0)^\top \boldsymbol{\Lambda}_u (\boldsymbol{\mu}_u - \boldsymbol{\mu}_0)\right) \\ &\quad \times \det(\boldsymbol{\Lambda}_u)^{(\nu_0 - R - 1)/2} \exp\left(-\frac{1}{2} \text{tr}(\mathbf{W}_0^{-1} \boldsymbol{\Lambda}_u)\right) \end{aligned} \quad (6.37)$$

由此, 将先验分布与似然函数进行相乘便可写出超参数 $(\boldsymbol{\mu}_u, \boldsymbol{\Lambda}_u)$ 的后验分布形式, 即

$$\begin{aligned} p(\boldsymbol{\mu}_u, \boldsymbol{\Lambda}_u \mid \mathbf{U}, \boldsymbol{\mu}_0, \mathbf{W}_0, \nu_0, \beta_0) &\propto \mathcal{N}(\boldsymbol{\mu}_u \mid \boldsymbol{\mu}_0, (\beta_0 \boldsymbol{\Lambda}_u)^{-1}) \times \mathcal{W}(\boldsymbol{\Lambda}_u \mid \mathbf{W}_0, \nu_0) \times \mathcal{L}(\mathbf{U} \mid \boldsymbol{\mu}_u, \boldsymbol{\Lambda}_u) \\ &\propto \mathcal{N}(\boldsymbol{\mu}_u \mid \tilde{\boldsymbol{\mu}}_u^*, (\tilde{\beta}_u^* \boldsymbol{\Lambda}_u)^{-1}) \times \mathcal{W}(\boldsymbol{\Lambda}_u \mid \tilde{\mathbf{W}}_u^*, \tilde{\nu}_u^*) \end{aligned} \quad (6.38)$$

在这里, 对超参数 $(\boldsymbol{\mu}_u, \boldsymbol{\Lambda}_u)$ 的后验分布取对数, 有

$$\begin{aligned} \ln p(\boldsymbol{\mu}_u, \boldsymbol{\Lambda}_u \mid \mathbf{U}, \boldsymbol{\mu}_0, \mathbf{W}_0, \nu_0, \beta_0) &= \frac{1}{2} M \ln \det(\boldsymbol{\Lambda}_u) - \frac{1}{2} \sum_{i=1}^M (\boldsymbol{\mu}_u - \boldsymbol{\mu}_0)^\top \boldsymbol{\Lambda}_u (\boldsymbol{\mu}_u - \boldsymbol{\mu}_0) \\ &\quad + \frac{1}{2} \ln \det(\beta_0 \boldsymbol{\Lambda}_u) - \frac{1}{2} \beta_0 (\boldsymbol{\mu}_u - \boldsymbol{\mu}_0)^\top \boldsymbol{\Lambda}_u (\boldsymbol{\mu}_u - \boldsymbol{\mu}_0) \\ &\quad + \frac{1}{2} (\nu_0 - R - 1) \ln \det(\boldsymbol{\Lambda}_u) - \frac{1}{2} \text{tr}(\mathbf{W}_0^{-1} \boldsymbol{\Lambda}_u) + C \end{aligned} \quad (6.39)$$

可将其改写成如下形式:

$$\begin{aligned} \ln p(\boldsymbol{\mu}_u, \boldsymbol{\Lambda}_u \mid \mathbf{U}, \boldsymbol{\mu}_0, \mathbf{W}_0, \nu_0, \beta_0) &= \frac{1}{2} \ln \det(\boldsymbol{\Lambda}_u) - \frac{1}{2} \left(\boldsymbol{\mu}_u - \frac{M \bar{\mathbf{u}} + \beta_0 \boldsymbol{\mu}_0}{M + \beta_0} \right)^\top ((M + \beta_0) \boldsymbol{\Lambda}_u) \left(\boldsymbol{\mu}_u - \frac{M \bar{\mathbf{u}} + \beta_0 \boldsymbol{\mu}_0}{M + \beta_0} \right) \\ &\quad + \frac{1}{2} (M + \nu_0 - R - 1) \ln \det(\boldsymbol{\Lambda}_u) \\ &\quad - \frac{1}{2} \text{tr} \left(\left(\mathbf{W}_0^{-1} + \sum_{i=1}^M (\mathbf{u}_i - \bar{\mathbf{u}})(\mathbf{u}_i - \bar{\mathbf{u}})^\top + \frac{M \beta_0}{M + \beta_0} (\bar{\mathbf{u}} - \boldsymbol{\mu}_0)(\bar{\mathbf{u}} - \boldsymbol{\mu}_0)^\top \right) \boldsymbol{\Lambda}_u \right) + C \end{aligned} \quad (6.40)$$

因此, 超参数超参数 $(\boldsymbol{\mu}_u, \boldsymbol{\Lambda}_u)$ 的后验分布参数为

$$\begin{cases} \tilde{\boldsymbol{\mu}}_u^* = (M\bar{\mathbf{u}} + \beta_0\boldsymbol{\mu}_0)/(M + \beta_0) \\ \tilde{\beta}_u^* = M + \beta_0 \\ \tilde{\nu}_u^* = M + \nu_0 \\ \tilde{\mathbf{W}}_u^* = (\mathbf{W}_0^{-1} + M\mathbf{S}_u + M\beta_0(\bar{\mathbf{u}} - \boldsymbol{\mu}_0)(\bar{\mathbf{u}} - \boldsymbol{\mu}_0)^\top/(M + \beta_0))^{-1} \end{cases} \quad (6.41)$$

其中, $\bar{\mathbf{u}} \in \mathbb{R}^R$ 与 $\mathbf{S}_u \in \mathbb{R}^{R \times R}$ 分别表示因子矩阵 \mathbf{U} 的统计量, 即

$$\bar{\mathbf{u}} = \frac{1}{M} \sum_{i=1}^M \mathbf{u}_i \quad \mathbf{S}_u = \frac{1}{M} \sum_{i=1}^M (\mathbf{u}_i - \bar{\mathbf{u}})(\mathbf{u}_i - \bar{\mathbf{u}})^\top \quad (6.42)$$

超参数 $(\boldsymbol{\mu}_v, \boldsymbol{\Lambda}_v)$ 的后验分布

与超参数 $(\boldsymbol{\mu}_u, \boldsymbol{\Lambda}_u)$ 的后验分布推导过程类似, 超参数 $(\boldsymbol{\mu}_v, \boldsymbol{\Lambda}_v)$ 的后验分布形式为

$$\begin{aligned} & p(\boldsymbol{\mu}_v, \boldsymbol{\Lambda}_v \mid \mathbf{V}, \boldsymbol{\mu}_0, \mathbf{W}_0, \nu_0, \beta_0) \\ & \propto \mathcal{N}(\boldsymbol{\mu}_v \mid \boldsymbol{\mu}_0, (\beta_0\boldsymbol{\Lambda}_v)^{-1}) \times \mathcal{W}(\boldsymbol{\Lambda}_v \mid \mathbf{W}_0, \nu_0) \times \mathcal{L}(\mathbf{U} \mid \boldsymbol{\mu}_v, \boldsymbol{\Lambda}_v) \\ & \propto \mathcal{N}(\boldsymbol{\mu}_v \mid \tilde{\boldsymbol{\mu}}_v^*, (\tilde{\beta}_v^*\boldsymbol{\Lambda}_v)^{-1}) \times \mathcal{W}(\boldsymbol{\Lambda}_v \mid \tilde{\mathbf{W}}_v^*, \tilde{\nu}_v^*) \end{aligned} \quad (6.43)$$

在这里, 对超参数 $(\boldsymbol{\mu}_v, \boldsymbol{\Lambda}_v)$ 的后验分布取对数, 有

$$\begin{aligned} & \ln p(\boldsymbol{\mu}_v, \boldsymbol{\Lambda}_v \mid \mathbf{V}, \boldsymbol{\mu}_0, \mathbf{W}_0, \nu_0, \beta_0) \\ & = \frac{1}{2}N \ln \det(\boldsymbol{\Lambda}_v) - \frac{1}{2} \sum_{j=1}^N (\boldsymbol{\mu}_v - \boldsymbol{\mu}_0)^\top \boldsymbol{\Lambda}_v (\boldsymbol{\mu}_v - \boldsymbol{\mu}_0) \\ & \quad + \frac{1}{2} \ln \det(\beta_0\boldsymbol{\Lambda}_v) - \frac{1}{2} \beta_0 (\boldsymbol{\mu}_v - \boldsymbol{\mu}_0)^\top \boldsymbol{\Lambda}_v (\boldsymbol{\mu}_v - \boldsymbol{\mu}_0) \\ & \quad + \frac{1}{2}(\nu_0 - R - 1) \ln \det(\boldsymbol{\Lambda}_v) - \frac{1}{2} \text{tr}(\mathbf{W}_0^{-1}\boldsymbol{\Lambda}_v) + C \end{aligned} \quad (6.44)$$

可将其改写成如下形式:

$$\begin{aligned} & \ln p(\boldsymbol{\mu}_v, \boldsymbol{\Lambda}_v \mid \mathbf{V}, \boldsymbol{\mu}_0, \mathbf{W}_0, \nu_0, \beta_0) \\ & = \frac{1}{2} \ln \det(\boldsymbol{\Lambda}_v) - \frac{1}{2} \left(\boldsymbol{\mu}_v - \frac{N\bar{\mathbf{v}} + \beta_0\boldsymbol{\mu}_0}{N + \beta_0} \right)^\top ((N + \beta_0)\boldsymbol{\Lambda}_v) \left(\boldsymbol{\mu}_v - \frac{N\bar{\mathbf{v}} + \beta_0\boldsymbol{\mu}_0}{N + \beta_0} \right) \\ & \quad + \frac{1}{2}(N + \nu_0 - R - 1) \ln \det(\boldsymbol{\Lambda}_v) \\ & \quad - \frac{1}{2} \text{tr} \left(\left(\mathbf{W}_0^{-1} + \sum_{j=1}^N (\mathbf{v}_j - \bar{\mathbf{v}})(\mathbf{v}_j - \bar{\mathbf{v}})^\top + \frac{N\beta_0}{N + \beta_0} (\bar{\mathbf{v}} - \boldsymbol{\mu}_0)(\bar{\mathbf{v}} - \boldsymbol{\mu}_0)^\top \right) \boldsymbol{\Lambda}_v \right) + C \end{aligned} \quad (6.45)$$

因此, 超参数超参数 $(\boldsymbol{\mu}_v, \boldsymbol{\Lambda}_v)$ 的后验分布参数为

$$\begin{cases} \tilde{\boldsymbol{\mu}}_v^* = (N\bar{\mathbf{v}} + \beta_0\boldsymbol{\mu}_0)/(N + \beta_0) \\ \tilde{\beta}_v^* = N + \beta_0 \\ \tilde{\nu}_v^* = N + \nu_0 \\ \tilde{\mathbf{W}}_v^* = (\mathbf{W}_0^{-1} + N\mathbf{S}_v + N\beta_0(\bar{\mathbf{v}} - \boldsymbol{\mu}_0)(\bar{\mathbf{v}} - \boldsymbol{\mu}_0)^\top/(N + \beta_0))^{-1} \end{cases} \quad (6.46)$$

其中, $\bar{\mathbf{v}} \in \mathbb{R}^R$ 与 $\mathbf{S}_v \in \mathbb{R}^{R \times R}$ 分别表示因子矩阵 \mathbf{V} 的统计量, 即

$$\bar{\mathbf{v}} = \frac{1}{N} \sum_{j=1}^N \mathbf{v}_j \quad \mathbf{S}_v = \frac{1}{N} \sum_{j=1}^N (\mathbf{v}_j - \bar{\mathbf{v}})(\mathbf{v}_j - \bar{\mathbf{v}})^\top \quad (6.47)$$

超参数 $(\boldsymbol{\mu}_x, \boldsymbol{\Lambda}_x)$ 的后验分布

与超参数 $(\boldsymbol{\mu}_u, \boldsymbol{\Lambda}_u)$ 的后验分布推导过程类似, 超参数 $(\boldsymbol{\mu}_x, \boldsymbol{\Lambda}_x)$ 的后验分布形式为

$$\begin{aligned} p(\boldsymbol{\mu}_x, \boldsymbol{\Lambda}_x \mid \mathbf{X}, \boldsymbol{\mu}_0, \mathbf{W}_0, \nu_0, \beta_0) \\ \propto \mathcal{N}(\boldsymbol{\mu}_x \mid \boldsymbol{\mu}_0, (\beta_0 \boldsymbol{\Lambda}_x)^{-1}) \times \mathcal{W}(\boldsymbol{\Lambda}_x \mid \mathbf{W}_0, \nu_0) \times \mathcal{L}(\mathbf{X} \mid \boldsymbol{\mu}_x, \boldsymbol{\Lambda}_x) \\ \propto \mathcal{N}(\boldsymbol{\mu}_x \mid \tilde{\boldsymbol{\mu}}_x^*, (\tilde{\beta}_x^* \boldsymbol{\Lambda}_x)^{-1}) \times \mathcal{W}(\boldsymbol{\Lambda}_x \mid \tilde{\mathbf{W}}_x^*, \tilde{\nu}_x^*) \end{aligned} \quad (6.48)$$

在这里, 对超参数 $(\boldsymbol{\mu}_x, \boldsymbol{\Lambda}_x)$ 的后验分布取对数, 有

$$\begin{aligned} \ln p(\boldsymbol{\mu}_x, \boldsymbol{\Lambda}_x \mid \mathbf{X}, \boldsymbol{\mu}_0, \mathbf{W}_0, \nu_0, \beta_0) \\ = \frac{1}{2} T \ln \det(\boldsymbol{\Lambda}_x) - \frac{1}{2} \sum_{t=1}^T (\boldsymbol{\mu}_x - \boldsymbol{\mu}_0)^\top \boldsymbol{\Lambda}_x (\boldsymbol{\mu}_x - \boldsymbol{\mu}_0) \\ + \frac{1}{2} \ln \det(\beta_0 \boldsymbol{\Lambda}_x) - \frac{1}{2} \beta_0 (\boldsymbol{\mu}_x - \boldsymbol{\mu}_0)^\top \boldsymbol{\Lambda}_x (\boldsymbol{\mu}_x - \boldsymbol{\mu}_0) \\ + \frac{1}{2} (\nu_0 - R - 1) \ln \det(\boldsymbol{\Lambda}_x) - \frac{1}{2} \text{tr}(\mathbf{W}_0^{-1} \boldsymbol{\Lambda}_x) + C \end{aligned} \quad (6.49)$$

可将其改写成如下形式:

$$\begin{aligned} \ln p(\boldsymbol{\mu}_x, \boldsymbol{\Lambda}_x \mid \mathbf{X}, \boldsymbol{\mu}_0, \mathbf{W}_0, \nu_0, \beta_0) \\ = \frac{1}{2} \ln \det(\boldsymbol{\Lambda}_x) - \frac{1}{2} \left(\boldsymbol{\mu}_x - \frac{T\bar{\mathbf{x}} + \beta_0 \boldsymbol{\mu}_0}{T + \beta_0} \right)^\top ((T + \beta_0) \boldsymbol{\Lambda}_x) \left(\boldsymbol{\mu}_x - \frac{T\bar{\mathbf{x}} + \beta_0 \boldsymbol{\mu}_0}{T + \beta_0} \right) \\ + \frac{1}{2} (T + \nu_0 - R - 1) \ln \det(\boldsymbol{\Lambda}_x) \\ - \frac{1}{2} \text{tr} \left(\left(\mathbf{W}_0^{-1} + \sum_{t=1}^T (\mathbf{x}_t - \bar{\mathbf{x}})(\mathbf{x}_t - \bar{\mathbf{x}})^\top + \frac{T\beta_0}{T + \beta_0} (\bar{\mathbf{x}} - \boldsymbol{\mu}_0)(\bar{\mathbf{x}} - \boldsymbol{\mu}_0)^\top \right) \boldsymbol{\Lambda}_x \right) + C \end{aligned} \quad (6.50)$$

因此, 超参数超参数 $(\boldsymbol{\mu}_x, \boldsymbol{\Lambda}_x)$ 的后验分布参数为

$$\begin{cases} \tilde{\boldsymbol{\mu}}_x^* = (T\bar{\mathbf{x}} + \beta_0 \boldsymbol{\mu}_0) / (T + \beta_0) \\ \tilde{\beta}_x^* = T + \beta_0 \\ \tilde{\nu}_x^* = T + \nu_0 \\ \tilde{\mathbf{W}}_x^* = (\mathbf{W}_0^{-1} + T\mathbf{S}_x + T\beta_0(\bar{\mathbf{x}} - \boldsymbol{\mu}_0)(\bar{\mathbf{x}} - \boldsymbol{\mu}_0)^\top / (T + \beta_0))^{-1} \end{cases} \quad (6.51)$$

其中, $\bar{\mathbf{x}} \in \mathbb{R}^R$ 与 $\mathbf{S}_x \in \mathbb{R}^{R \times R}$ 分别表示因子矩阵 \mathbf{X} 的统计量, 即

$$\bar{\mathbf{x}} = \frac{1}{T} \sum_{t=1}^T \mathbf{x}_t \quad \mathbf{S}_x = \frac{1}{T} \sum_{t=1}^T (\mathbf{x}_t - \bar{\mathbf{x}})(\mathbf{x}_t - \bar{\mathbf{x}})^\top \quad (6.52)$$

6.3.4 精度参数的后验分布

根据张量元素的高斯分布假设, τ 是作为高斯分布的精度参数, 可用于刻画真实数据所带有的随机性与噪声。就精度参数 τ 而言, 其似然函数为

$$\mathcal{L}(\mathcal{Y} \mid \mathbf{U}, \mathbf{V}, \mathbf{X}, \tau) \propto \prod_{(i,j,t) \in \Omega} \tau^{1/2} \exp \left(-\frac{1}{2} \tau (y_{i,j,t} - \hat{y}_{i,j,t})^2 \right) \quad (6.53)$$

根据共轭先验设置准则, 由于精度参数 τ 的先验分布为伽马分布, 因此, 后验分布也为伽马分布, 即

$$\begin{aligned} p(\tau \mid \mathcal{Y}, \mathbf{U}, \mathbf{V}, \mathbf{X}, a_0, b_0) &\propto \text{Gamma}(\tau \mid a_0, b_0) \times \mathcal{L}(\mathcal{Y} \mid \mathbf{U}, \mathbf{V}, \mathbf{X}, \tau) \\ &\propto \text{Gamma}(\tau \mid \tilde{a}_\tau, \tilde{b}_\tau) \end{aligned} \quad (6.54)$$

对该后验分布左右两边同时取对数，有

$$\begin{aligned}
& \ln p(\tau \mid \mathbf{Y}, \mathbf{U}, \mathbf{V}, \mathbf{X}, a_0, b_0) \\
&= \frac{1}{2} |\Omega| \ln \tau - \frac{1}{2} \tau \sum_{(i,j,t) \in \Omega} (y_{i,j,t} - \hat{y}_{i,j,t})^2 - b_0 \tau + (a_0 - 1) \ln \tau + \text{const} \\
&= \left(a_0 + \frac{1}{2} |\Omega| - 1 \right) \ln \tau - \left(b_0 + \frac{1}{2} \sum_{(i,j,t) \in \Omega} (y_{i,j,t} - \hat{y}_{i,j,t})^2 \right) \tau + \text{const}
\end{aligned} \tag{6.55}$$

因此，精度参数 τ 的后验分布参数分别为

$$\begin{cases} \tilde{a}_\tau = \frac{1}{2} |\Omega| + a_0 \\ \tilde{b}_\tau = \frac{1}{2} \sum_{(i,j,t) \in \Omega} (y_{i,j,t} - \hat{y}_{i,j,t})^2 + b_0 \end{cases} \tag{6.56}$$

6.4 算法实现

根据上述推导出的参数与超参数的后验分布，可采用 Gibbs 采样方法对贝叶斯高斯张量分解进行参数估计，具体实现流程如算法4所示。在算法中， $L_1, L_2 \in \mathbb{N}^+$ 分别表示 burn-in 阶段与 Gibbs 采样阶段的迭代次数。

Algorithm 4 贝叶斯高斯张量分解算法

Input: 张量 $\mathbf{Y} \in \mathbb{R}^{M \times N \times T}$ ，被观测元素的索引集合 Ω ，低秩张量的秩 R 。

Output: 重构出来的张量 $\hat{\mathbf{Y}} \in \mathbb{R}^{M \times N \times T}$ 。

- 1: 设定 $\boldsymbol{\mu}_0 = \mathbf{0}, \beta_0 = 1, \mathbf{W}_0 = \mathbf{I}, \nu_0 = R, a_0 = 10^{-6}, b_0 = 10^{-6}$;
 - 2: 对因子矩阵 $\{\mathbf{U}, \mathbf{V}, \mathbf{X}\}$ 进行初始化，并令 $\mathbf{U} = \mathbf{0}, \mathbf{V} = \mathbf{0}, \mathbf{X} = \mathbf{0}$;
 - 3: **for** $\ell = 1$ to $L_1 + L_2$ **do**
 - 4: 根据公式(6.38)、(6.43)与(6.48)中的后验分布分别对 $\{(\boldsymbol{\mu}_u, \boldsymbol{\Lambda}_u), (\boldsymbol{\mu}_v, \boldsymbol{\Lambda}_v), (\boldsymbol{\mu}_x, \boldsymbol{\Lambda}_x)\}$ 进行采样;
 - 5: **for** $i = 1, 2, \dots, M$ **do**
 - 6: 根据公式(6.24)中的后验分布对 \mathbf{u}_i 进行采样;
 - 7: **end for**
 - 8: **for** $j = 1, 2, \dots, N$ **do**
 - 9: 根据公式(6.29)中的后验分布对 \mathbf{v}_j 进行采样;
 - 10: **end for**
 - 11: **for** $t = 1, 2, \dots, T$ **do**
 - 12: 根据公式(6.34)中的后验分布对 \mathbf{x}_t 进行采样;
 - 13: **end for**
 - 14: 根据公式(6.54)中的后验分布对 τ 进行采样;
 - 15: **if** $\ell > L_1$ **then**
 - 16: 执行 $\tilde{\mathbf{U}} += \mathbf{U}, \tilde{\mathbf{V}} += \mathbf{V}, \tilde{\mathbf{X}} += \mathbf{X}$;
 - 17: **end if**
 - 18: **end for**
 - 19: 计算因子矩阵 $\mathbf{U} := \tilde{\mathbf{U}}/L_2, \mathbf{V} := \tilde{\mathbf{V}}/L_2, \mathbf{X} := \tilde{\mathbf{X}}/L_2$ ，并根据因子矩阵计算 CP 分解重构出来的张量 $\hat{\mathbf{Y}}$ 。
-

6.5 Python 实现代码

表6.1罗列了使用 Python 实现贝叶斯高斯张量分解算法的一系列函数。

表 6.1: 贝叶斯高斯张量分解算法一览表

序号	函数名	功能介绍
1	mvnrnd	生成多元高斯分布 $\mathcal{N}(\mu, \Lambda^{-1})$ 的随机数
2	cp_combine	三个因子矩阵相乘得到一个三阶张量
3	ten2mat	将张量自特定唯独展开成矩阵，即张量矩阵化过程

6.5.1 贝叶斯高斯张量分解算法的函数

```
import numpy as np
from scipy.stats import wishart
from numpy.random import normal as normrnd
from scipy.linalg import khatri_rao as kr_prod
from numpy.linalg import inv as inv
from numpy.linalg import solve as solve
from scipy.linalg import cholesky as cholesky_upper
from scipy.linalg import solve_triangular as solve_ut

def mvnrnd(mu, Lambda):
    src = normrnd(size = (mu.shape[0],))
    return solve_ut(cholesky_upper(Lambda, overwrite_a = True,
                                   check_finite = False),
                    src, lower = False, check_finite = False,
                    overwrite_b = True) + mu

def cp_combine(var):
    return np.einsum('is, js, ts -> ijt', var[0], var[1], var[2])

def ten2mat(tensor, mode):
    return np.reshape(np.moveaxis(tensor, mode, 0),
                      (tensor.shape[mode], -1), order = 'F')

def cov_mat(mat, mat_bar):
    mat = mat - mat_bar
    return mat.T @ mat

def sample_factor(tau_sparse_tensor, tau_ind, factor, k, beta0 = 1):
    dim, rank = factor[k].shape
    dim = factor[k].shape[0]
    factor_bar = np.mean(factor[k], axis = 0)
    temp = dim / (dim + beta0)
    var_mu_hyper = temp * factor_bar
    var_W_hyper = (inv(np.eye(rank) + cov_mat(factor[k], factor_bar)
    + temp * beta0 * np.outer(factor_bar, factor_bar)))
    var_Lambda_hyper = wishart.rvs(df = dim + rank,
                                   scale = var_W_hyper)
    var_mu_hyper = mvnrnd(var_mu_hyper,
                          (dim + beta0) * var_Lambda_hyper)

    idx = list(filter(lambda x: x != k, range(len(factor))))
    var1 = kr_prod(factor[idx[1]], factor[idx[0]].T
    var2 = kr_prod(var1, var1)
    var3 = ((var2 @ ten2mat(tau_ind, k).T).reshape([rank, rank, dim]))
```



```

        + var_Lambda_hyper[:, :, np.newaxis])
var4 = (var1 @ ten2mat(tau_sparse_tensor, k).T
        + (var_Lambda_hyper @ var_mu_hyper)[:, np.newaxis])
for i in range(dim):
    factor[k][i, :] = mvnrnd(solve(var3[:, :, i], var4[:, i]),
                             var3[:, :, i])

return factor[k]

def sample_precision_tau(sparse_tensor, tensor_hat, ind):
    var_alpha = 1e-6 + 0.5 * np.sum(ind)
    var_beta = 1e-6 + 0.5 * np.sum(((sparse_tensor
                                     - tensor_hat) ** 2) * ind)

    return np.random.gamma(var_alpha, 1 / var_beta)

def compute_mape(var, var_hat):
    return np.sum(np.abs(var - var_hat) / var) / var.shape[0]

def compute_rmse(var, var_hat):
    return np.sqrt(np.sum((var - var_hat) ** 2) / var.shape[0])

def BGCP(dense_tensor, sparse_tensor, factor, burn_iter, gibbs_iter):
    """Bayesian Gaussian CP (BGCP) decomposition."""

    dim = np.array(sparse_tensor.shape)
    rank = factor[0].shape[1]
    if np.isnan(sparse_tensor).any() == False:
        ind = sparse_tensor != 0
        pos_obs = np.where(ind)
        pos_test = np.where((dense_tensor != 0) & (sparse_tensor == 0))
    elif np.isnan(sparse_tensor).any() == True:
        pos_test = np.where((dense_tensor != 0) & (np.isnan(sparse_tensor)))
        ind = ~np.isnan(sparse_tensor)
        pos_obs = np.where(ind)
        sparse_tensor[np.isnan(sparse_tensor)] = 0
    show_iter = 200
    tau = 1
    factor_plus = []
    for k in range(len(dim)):
        factor_plus.append(np.zeros((dim[k], rank)))
    temp_hat = np.zeros(dim)
    tensor_hat_plus = np.zeros(dim)
    for it in range(burn_iter + gibbs_iter):
        tau_ind = tau * ind
        tau_sparse_tensor = tau * sparse_tensor
        for k in range(len(dim)):
            factor[k] = sample_factor(tau_sparse_tensor, tau_ind, factor, k)
        tensor_hat = cp_combine(factor)
        temp_hat += tensor_hat
        tau = sample_precision_tau(sparse_tensor, tensor_hat, ind)
        if it + 1 > burn_iter:
            factor_plus = [factor_plus[k] + factor[k] for k in range(len(dim))]
            tensor_hat_plus += tensor_hat
        if (it + 1) % show_iter == 0 and it < burn_iter:
            temp_hat = temp_hat / show_iter
            print('Iter: {}'.format(it + 1))
            print('MAPE: {:.6}'.format(compute_mape(dense_tensor[pos_test],
                                                    temp_hat[pos_test])))
            print('RMSE: {:.6}'.format(compute_rmse(dense_tensor[pos_test],
                                                    temp_hat[pos_test])))

            temp_hat = np.zeros(sparse_tensor.shape)
            print()
    factor = [i / gibbs_iter for i in factor_plus]

```

```
tensor_hat = tensor_hat_plus / gibbs_iter
print('Final MAPE: {:.6}'.format(compute_mape(dense_tensor[pos_test],
                                              tensor_hat[pos_test])))

print('Final RMSE: {:.6}'.format(compute_rmse(dense_tensor[pos_test],
                                              tensor_hat[pos_test])))

print()

return tensor_hat, factor
```

6.5.2 实验：时空交通数据修复

第七章 低秩张量填充

低秩张量填充 (low-rank tensor completion) 是一种经典的张量模型。

7.1 基于核范数最小化的矩阵填充

7.1.1 核范数最小化

给定任意矩阵 $\mathbf{Y} \in \mathbb{R}^{N \times T}$ ，若其观测元素的索引集合为 Ω ，则以矩阵核范数为目标函数的矩阵填充模型可被描述为

$$\begin{aligned} \min_{\mathbf{X}} \quad & \|\mathbf{X}\|_* \\ \text{s.t.} \quad & \|\mathcal{P}_{\Omega}(\mathbf{X} - \mathbf{Y})\|_F \leq \epsilon \end{aligned} \tag{7.1}$$

例 41. 卷积矩阵核范数最小化。

7.1.2 截断核范数最小化

7.1.3 加权核范数最小化

7.2 基于多重核范数的张量填充

7.3 基于多重截断核范数的张量填充

7.4 基于张量核范数的张量填充

引入多维傅立叶变换

第八章 低秩拉普拉斯卷积模型

本章将要介绍的内容包括：

- 循环卷积的定义与性质、卷积定理
- 循环矩阵核范数及其最小化问题
- 低秩拉普拉斯卷积模型

8.1 循环卷积与离散傅立叶变换

8.1.1 循环卷积定义

给定向量 $\mathbf{x} = (x_1, x_2, \dots, x_T)^\top \in \mathbb{R}^T$ 与 $\mathbf{y} = (y_1, y_2, \dots, y_\tau)^\top \in \mathbb{R}^\tau$ ，其中 $\tau \leq T$ ，若两者之间的循环卷积为 $\mathbf{z} = \mathbf{x} \star \mathbf{y} \in \mathbb{R}^T$ ，则向量 \mathbf{z} 的任意元素为

$$z_t = \sum_{k=1}^{\tau} x_{t-k+1} y_k, \forall t \in \{1, 2, \dots, T\} \quad (8.1)$$

其中，当 $t+1 \leq k$ 时，则令 $x_{t-k+1} = x_{t-k+1+T}$ 。

例 42. 给定向量 $\mathbf{x} = (0, 1, 2, 3, 4)^\top$ 与 $\mathbf{y} = (2, -1, 3)^\top$ ，试写出循环卷积 $\mathbf{z} = \mathbf{x} \star \mathbf{y}$ 。

解. 根据定义，有

$$\begin{aligned} \mathbf{z} = \mathbf{x} \star \mathbf{y} &= \begin{bmatrix} \sum_{k=1}^3 x_{1-k+1} y_k \\ \sum_{k=1}^3 x_{2-k+1} y_k \\ \sum_{k=1}^3 x_{3-k+1} y_k \\ \sum_{k=1}^3 x_{4-k+1} y_k \\ \sum_{k=1}^3 x_{5-k+1} y_k \end{bmatrix} = \begin{bmatrix} x_1 y_1 + x_5 y_2 + x_4 y_3 \\ x_2 y_1 + x_1 y_2 + x_5 y_3 \\ x_3 y_1 + x_2 y_2 + x_1 y_3 \\ x_4 y_1 + x_3 y_2 + x_2 y_3 \\ x_5 y_1 + x_4 y_2 + x_3 y_3 \end{bmatrix} \\ &= \begin{bmatrix} 0 \times 2 + 4 \times (-1) + 3 \times 3 \\ 1 \times 2 + 0 \times (-1) + 4 \times 3 \\ 2 \times 2 + 1 \times (-1) + 0 \times 3 \\ 3 \times 2 + 2 \times (-1) + 1 \times 3 \\ 4 \times 2 + 3 \times (-1) + 2 \times 3 \end{bmatrix} = \begin{bmatrix} 5 \\ 14 \\ 3 \\ 7 \\ 11 \end{bmatrix} \end{aligned} \quad (8.2)$$

由于循环卷积在计算过程中呈现出线性结构。根据循环卷积的运算规则，可定义一种线性变换，使得

$$\mathbf{x} \star \mathbf{y} = \mathcal{C}_\tau(\mathbf{x})\mathbf{y} \quad (8.3)$$

其中， $\mathcal{C}_\tau: \mathbb{R}^T \rightarrow \mathbb{R}^{T \times \tau}$ 表示构造卷积矩阵的算子；在这里，卷积矩阵为

$$\mathcal{C}_\tau(\mathbf{x}) = \begin{bmatrix} x_1 & x_T & x_{T-1} & \cdots & x_{T-\tau+2} \\ x_2 & x_1 & x_T & \cdots & x_{T-\tau+3} \\ x_3 & x_2 & x_1 & \cdots & x_{T-\tau+4} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ x_T & x_{T-1} & x_{T-2} & \cdots & x_{T-\tau+1} \end{bmatrix} \in \mathbb{R}^{T \times \tau} \quad (8.4)$$

其中，卷积矩阵的列数为 τ 。

例 43. 给定向量 $\mathbf{x} = (0, 1, 2, 3, 4)^\top$ 与 $\mathbf{y} = (2, -1, 3)^\top$ ，试写出卷积矩阵 $\mathcal{C}_3(\mathbf{x})$ 与循环卷积 $\mathbf{z} = \mathcal{C}_3(\mathbf{x})\mathbf{y}$ 。

解. 根据定义，向量 \mathbf{x} 对应的卷积矩阵为

$$\mathcal{C}_3(\mathbf{x}) = \begin{bmatrix} 0 & 4 & 3 \\ 1 & 0 & 4 \\ 2 & 1 & 0 \\ 3 & 2 & 1 \\ 4 & 3 & 2 \end{bmatrix} \quad (8.5)$$

由此，向量 \mathbf{x} 与 \mathbf{y} 的循环卷积为

$$\mathbf{z} = \mathcal{C}_3(\mathbf{x})\mathbf{y} = \begin{bmatrix} 0 & 4 & 3 \\ 1 & 0 & 4 \\ 2 & 1 & 0 \\ 3 & 2 & 1 \\ 4 & 3 & 2 \end{bmatrix} \begin{bmatrix} 2 \\ -1 \\ 3 \end{bmatrix} = \begin{bmatrix} 5 \\ 14 \\ 3 \\ 7 \\ 11 \end{bmatrix} \quad (8.6)$$

8.1.2 一维卷积定理

离散傅立叶变换 (discrete Fourier transform) 是数学中非常重要的一个概念，被应用于众多领域，如信号处理与图像处理。由于离散傅立叶变换通常采用快速傅立叶变换 (fast Fourier transform) 进行高效求解，所以两者经常出现在一起。

实际上，循环卷积与离散傅立叶变换联系紧密。卷积定理 (convolution theorem) 可用于描述两者之间的这种关系，给定任意向量 $\mathbf{x}, \mathbf{y} \in \mathbb{R}^T$ ，有

$$\mathbf{x} \star \mathbf{y} = \mathcal{F}^{-1}(\mathcal{F}(\mathbf{x}) \circ \mathcal{F}(\mathbf{y})) \quad (8.7)$$

恒成立。在这里， $\mathcal{F}(\cdot)$ 表示离散傅立叶变换； $\mathcal{F}^{-1}(\cdot)$ 表示离散傅立叶逆变换；符号 \circ 表示元素间的点乘 (Hadamard product)。将向量做离散傅立叶变换的意义在于利用快速傅立叶变换进行高效计算。

假设向量 $\mathbf{x} \in \mathbb{R}^T$ ，给定向量 $\mathbf{y} \in \mathbb{R}^\tau, \tau < T$ ，使用离散傅立叶变换计算循环卷积时需首先令

$$\mathbf{y} = (y_1, y_2, \cdots, y_\tau, \underbrace{0, \cdots, 0}_{T-\tau})^\top \in \mathbb{R}^T \quad (8.8)$$

然后分别对向量 \mathbf{x} 与 \mathbf{y} 进行离散傅立叶变换。

例 44. 给定向量 $\mathbf{x} = (0, 1, 2, 3, 4)^\top$ 与 $\mathbf{y} = (2, -1, 3)^\top$, 试根据卷积定理计算循环卷积 $\mathbf{z} = \mathbf{x} \star \mathbf{y}$ 。

解. 分别对向量 \mathbf{x} 与 \mathbf{y} 进行离散傅立叶变换, 有

$$\begin{cases} \mathcal{F}(\mathbf{x}) = (10, -2.5 + 3.44i, -2.5 + 0.81i, -2.5 - 0.81i, -2.5 - 3.44i)^\top \\ \mathcal{F}(\mathbf{y}) = (4, -0.74 - 0.81i, 3.74 + 3.44i, 3.74 - 3.44i, -0.74 + 0.81i)^\top \end{cases} \quad (8.9)$$

其中, $i = \sqrt{-1}$ 表示复数的虚部。

根据卷积定理, 有

$$\mathbf{z} = \mathcal{F}^{-1}(\mathcal{F}(\mathbf{x}) \circ \mathcal{F}(\mathbf{y})) = (5, 14, 3, 7, 11)^\top \quad (8.10)$$

根据卷积定理, 若 $\mathbf{z} = \mathbf{x} \star \mathbf{y}$, 则

$$\mathbf{x} = \mathcal{F}^{-1}(\mathcal{F}(\mathbf{z}) \oslash \mathcal{F}(\mathbf{y})) \quad \mathbf{y} = \mathcal{F}^{-1}(\mathcal{F}(\mathbf{z}) \oslash \mathcal{F}(\mathbf{x})) \quad (8.11)$$

其中, 符号 \oslash 表示元素间的点除 (Hadamard division)。

例 45. 给定向量 $\mathbf{x} = (0, 1, 2, 3, 4)^\top$ 与 $\mathbf{z} = (5, 14, 3, 7, 11)^\top$, 若 $\mathbf{z} = \mathbf{x} \star \mathbf{y}$, 试根据卷积定理计算 \mathbf{y} 。

解. 根据卷积定理, 有

$$\mathbf{y} = \mathcal{F}^{-1}(\mathcal{F}(\mathbf{z}) \oslash \mathcal{F}(\mathbf{x})) = (2, -1, 3, 0, 0)^\top \quad (8.12)$$

8.1.3 二维卷积定理

对于任意矩阵 $\mathbf{X} \in \mathbb{R}^{M \times N}$ 与 $\mathbf{K} \in \mathbb{R}^{\nu_1 \times \nu_2}$, 其中, $\nu_1 \leq M, \nu_2 \leq N$, 若两者之间的循环卷积为 $\mathbf{Y} = \mathbf{K} \star \mathbf{X} \in \mathbb{R}^{M \times N}$, 则矩阵 \mathbf{Y} 的任意元素为

$$y_{m,n} = \sum_{i=1}^{\nu_1} \sum_{j=1}^{\nu_2} \kappa_{i,j} x_{m-i+1, n-j+1}, \quad \forall (m, n) \quad (8.13)$$

其中, $\kappa_{i,j}$ 为矩阵 \mathbf{K} 的第 (i, j) 个元素。

根据循环卷积定义, 矩阵 \mathbf{Y} 的第 m 行为

$$\begin{aligned} \mathbf{y}_{m,:} &= \sum_{i=1}^{\nu_1} \kappa_{i,:} \star \mathbf{x}_{m-i+1,:} \\ &= \sum_{i=1}^{\nu_1} \mathcal{F}^{-1}(\mathcal{F}(\kappa_{i,:}) \circ \mathcal{F}(\mathbf{x}_{m-i+1,:})) \end{aligned} \quad (8.14)$$

其中, $\kappa_{i,:}$ 为矩阵 \mathbf{K} 的第 i 行。

矩阵 \mathbf{Y} 的第 n 列为

$$\begin{aligned} \mathbf{y}_{:,n} &= \sum_{j=1}^{\nu_2} \kappa_{:,j} \star \mathbf{x}_{:,n-j+1} \\ &= \sum_{j=1}^{\nu_2} \mathcal{F}^{-1}(\mathcal{F}(\kappa_{:,j}) \circ \mathcal{F}(\mathbf{x}_{:,n-j+1})) \end{aligned} \quad (8.15)$$

其中, $\kappa_{:,j}$ 为矩阵 \mathbf{K} 的第 j 列。

在这里, 二维循环卷积与离散傅立叶变换之间的卷积定理也可写作如下形式:

$$\mathbf{Y} = \mathbf{K} \star \mathbf{X} = \mathcal{F}^{-1}(\mathcal{F}(\mathbf{K}) \circ \mathcal{F}(\mathbf{X})) \quad (8.16)$$

其中, $\mathcal{F}(\cdot)$ 表示二维离散傅立叶变换; $\mathcal{F}^{-1}(\cdot)$ 表示二维离散傅立叶逆变换。

例 46. 给定矩阵 $\mathbf{X} = \begin{bmatrix} 1 & 2 & 3 & 4 \\ 4 & 5 & 6 & 7 \\ 7 & 8 & 9 & 10 \\ 10 & 11 & 12 & 13 \end{bmatrix}$ 与 $\mathbf{K} = \begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \\ 7 & 8 & 9 \end{bmatrix}$, 试根据卷积定理计算二维循环卷积 $\mathbf{Y} = \mathbf{K} \star \mathbf{X}$ 。

解. 根据卷积定理, 二维循环卷积为

$$\mathbf{Y} = \mathcal{F}^{-1}(\mathcal{F}(\mathbf{K}) \circ \mathcal{F}(\mathbf{X})) = \begin{bmatrix} 405 & 390 & 363 & 408 \\ 360 & 345 & 318 & 363 \\ 207 & 192 & 165 & 210 \\ 342 & 327 & 300 & 345 \end{bmatrix} \quad (8.17)$$

8.1.4 Parseval 定理

Parseval 定理 (Parseval's theorem) 表明信号的能量在时域与频域相等。在离散傅立叶变换中, 对于任意向量 $\mathbf{x} = (x_1, x_2, \dots, x_T) \in \mathbb{R}^T$, 离散形式的 Parseval 定理为

$$\|\mathbf{x}\|_2^2 = \frac{1}{T} \|\mathcal{F}(\mathbf{x})\|_2^2 \quad (8.18)$$

例 47. 给定向量 $\mathbf{x} = (0, 1, 2, 3, 4)^\top$, 试写出 $\|\mathbf{x}\|_2^2$ 与 $\|\mathcal{F}(\mathbf{x})\|_2^2$ 。

解. 根据 ℓ_2 范数定义, 有

$$\|\mathbf{x}\|_2^2 = 0^2 + 1^2 + 2^2 + 3^2 + 4^2 = 30 \quad (8.19)$$

由于

$$\mathcal{F}(\mathbf{x}) = (10, -2.5 + 3.44i, -2.5 + 0.81i, -2.5 - 0.81i, -2.5 - 3.44i)^\top \quad (8.20)$$

其中, $i = \sqrt{-1}$ 表示复数的虚部。故 $\|\mathcal{F}(\mathbf{x})\|_2^2 = 150$ 。

针对二维离散傅立叶变量, 给定任意矩阵 $\mathbf{X} \in \mathbb{R}^{M \times N}$, Parseval 定理为

$$\|\mathbf{X}\|_F^2 = \frac{1}{MN} \|\mathcal{F}(\mathbf{X})\|_F^2 \quad (8.21)$$

例 48. 给定矩阵 $\mathbf{X} = \begin{bmatrix} 5 & 6 & 7 \\ 8 & 9 & 10 \end{bmatrix}$, 试写出 $\|\mathbf{X}\|_F^2$ 与 $\|\mathcal{F}(\mathbf{X})\|_F^2$ 。

解. 根据 F 范数定义, 有

$$\|\mathbf{X}\|_F^2 = 5^2 + 6^2 + 7^2 + 8^2 + 9^2 + 10^2 = 355 \quad (8.22)$$

由于

$$\mathcal{F}(\mathbf{X}) = \begin{bmatrix} 45 & -3 + 1.73i & -3 - 1.73i \\ -9 & 0 & 0 \end{bmatrix} \quad (8.23)$$

其中, $i = \sqrt{-1}$ 表示复数的虚部。故 $\|\mathcal{F}(\mathbf{X})\|_F^2 = 2130$ 。

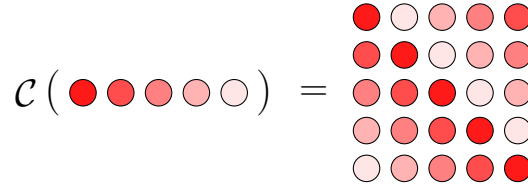


图 8.1: 循环矩阵示意图

8.2 离散傅立叶变换与循环矩阵核范数

8.2.1 循环矩阵定义

循环矩阵 (circulant matrix) 是一种特殊的代数结构, 广泛应用于信号处理等。从定义出发, 给定任意向量 $\mathbf{x} = (x_1, x_2, \dots, x_T)^\top \in \mathbb{R}^T$, 其对应的循环矩阵可写作如下形式:

$$\mathcal{C}(\mathbf{x}) = \begin{bmatrix} x_1 & x_T & x_{T-1} & \cdots & x_2 \\ x_2 & x_1 & x_T & \cdots & x_3 \\ x_3 & x_2 & x_1 & \cdots & x_4 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ x_T & x_{T-1} & x_{T-2} & \cdots & x_1 \end{bmatrix} \in \mathbb{R}^{T \times T} \quad (8.24)$$

其中, $\mathcal{C}: \mathbb{R}^T \rightarrow \mathbb{R}^{T \times T}$ 表示循环算子 (circulant operator)。该循环矩阵的第一列为向量 \mathbf{x} 本身, 对角线元素均为 x_1 。

例 49. 给定任意向量 $\mathbf{x} = (x_1, x_2, x_3, x_4, x_5)^\top \in \mathbb{R}^5$, 试写出其对应的循环矩阵。

解. 向量 \mathbf{x} 对应的循环矩阵为

$$\mathcal{C}(\mathbf{x}) = \begin{bmatrix} x_1 & x_5 & x_4 & x_3 & x_2 \\ x_2 & x_1 & x_5 & x_4 & x_3 \\ x_3 & x_2 & x_1 & x_5 & x_4 \\ x_4 & x_3 & x_2 & x_1 & x_5 \\ x_5 & x_4 & x_3 & x_2 & x_1 \end{bmatrix} \in \mathbb{R}^{5 \times 5} \quad (8.25)$$

图 8.1 直观描述了循环矩阵的构造规则。

例 50. 给定任意向量 $\mathbf{x} = (x_1, x_2, \dots, x_T)^\top \in \mathbb{R}^T$ 与 $\mathbf{y} = (y_1, y_2, \dots, y_T)^\top \in \mathbb{R}^T$, 若两者之间的循环卷积 (circular convolution) 为 $\mathbf{z} = \mathbf{x} \star \mathbf{y} \in \mathbb{R}^T$, 其中, 符号 \star 表示卷积运算, 则向量 \mathbf{z} 的任意元素为

$$z_t = \sum_{k=1}^T x_{t-k+1} y_k, \forall t \in \{1, 2, \dots, T\} \quad (8.26)$$

其中, 当 $t+1 \leq k$ 时, 则令 $x_{t-k+1} = x_{t-k+1+T}$ 。试根据循环矩阵的定义写出循环卷积。

解. 在这里, 循环卷积可写作如下形式:

$$\mathbf{z} = \mathbf{x} \star \mathbf{y} = \begin{bmatrix} x_1 y_1 + x_T y_2 + \cdots + x_2 y_T \\ x_2 y_1 + x_1 y_2 + \cdots + x_3 y_T \\ \vdots \\ x_T y_1 + x_{T-1} y_2 + \cdots + x_1 y_T \end{bmatrix} = \begin{bmatrix} x_1 & x_T & \cdots & x_2 \\ x_2 & x_1 & \cdots & x_3 \\ \vdots & \vdots & \ddots & \vdots \\ x_T & x_{T-1} & \cdots & x_1 \end{bmatrix} \begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_T \end{bmatrix} = \mathcal{C}(\mathbf{x}) \mathbf{y} \quad (8.27)$$

8.2.2 循环矩阵核范数

在线性代数中, 矩阵的核范数为奇异值之和。对于任意矩阵 $\mathbf{X} \in \mathbb{R}^{M \times N}$, 其奇异值分解为

$$\mathbf{X} = \sum_{r=1}^{\min\{M,N\}} \sigma_r \mathbf{u}_r \mathbf{v}_r^\top \quad (8.28)$$

其中, 奇异值为 $\sigma_1 \geq \sigma_2 \geq \cdots \geq \sigma_{\min\{M,N\}}$; 矩阵的核范数为

$$\|\mathbf{X}\|_* = \sum_{r=1}^{\min\{M,N\}} \sigma_r \quad (8.29)$$

给定向量 $\mathbf{x} \in \mathbb{R}^T$, 其循环矩阵为

$$\mathcal{C}(\mathbf{x}) = \begin{bmatrix} x_1 & x_T & x_{T-1} & \cdots & x_2 \\ x_2 & x_1 & x_T & \cdots & x_3 \\ x_3 & x_2 & x_1 & \cdots & x_4 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ x_T & x_{T-1} & x_{T-2} & \cdots & x_1 \end{bmatrix} \in \mathbb{R}^{T \times T} \quad (8.30)$$

对该循环矩阵进行特征值分解, 有

$$\mathcal{C}(\mathbf{x}) = \mathbf{U} \text{diag}(\mathcal{F}(\mathbf{x})) \mathbf{U}^H \quad (8.31)$$

其中, $\mathbf{U} \in \mathbb{C}^{T \times T}$ 为酉矩阵 (unitary matrix); \cdot^H 表示共轭转置 (conjugate transpose)。

因此, 循环矩阵的核范数可写作如下形式:

$$\begin{aligned} \|\mathcal{C}(\mathbf{x})\|_* &= \|\mathbf{U} \text{diag}(\mathcal{F}(\mathbf{x})) \mathbf{U}^H\|_* \\ &= \|\text{diag}(\mathcal{F}(\mathbf{x}))\|_* \\ &= \|\mathcal{F}(\mathbf{x})\|_1 \end{aligned} \quad (8.32)$$

由此可见, 循环矩阵的核范数可转化为离散傅立叶变换的 ℓ_1 范数。在这里, 快速傅立叶变换的计算复杂度为 $\mathcal{O}(T \log T)$, 可大大提高求解循环矩阵的核范数最小化问题的计算效率。

例 51. 给定向量 $\mathbf{x} = (0, 1, 2, 3, 4)^\top$, 试写出循环矩阵 $\mathcal{C}(\mathbf{x})$ 的奇异值与 $\|\mathcal{F}(\mathbf{x})\|_1$ 。

解. 根据循环矩阵定义, 有

$$\mathcal{C}(\mathbf{x}) = \begin{bmatrix} 0 & 4 & 3 & 2 & 1 \\ 1 & 0 & 4 & 3 & 2 \\ 2 & 1 & 0 & 4 & 3 \\ 3 & 2 & 1 & 0 & 4 \\ 4 & 3 & 2 & 1 & 0 \end{bmatrix} \quad (8.33)$$

对其进行奇异值分解, 奇异值构成的向量为

$$\boldsymbol{\sigma} = (10, 4.25, 4.25, 2.63, 2.63)^\top \quad (8.34)$$

另外, 直接对向量 \mathbf{x} 作离散傅立叶变换, 有

$$\mathcal{F}(\mathbf{x}) = (10, -2.5 + 3.44i, -2.5 + 0.81i, -2.5 - 0.81i, -2.5 - 3.44i)^\top \quad (8.35)$$

其中, $i = \sqrt{-1}$ 表示复数的虚部。

由此, 可得到

$$\|\mathcal{F}(\mathbf{x})\|_1 = 10 + 2\sqrt{2.5^2 + 3.44^2} + 2\sqrt{2.5^2 + 0.81^2} = 10 + 8.50 + 5.26 = 23.76 \quad (8.36)$$

8.2.3 ℓ_1 范数最小化问题

一般而言, 假设图优化问题的目标函数为 $f(x) = g(x) + h(x)$ 是由 $g(x)$ 与 $h(x)$ 叠加而成的, 其中, 限定 $g(x)$ 是不可微的凸函数、 $h(x)$ 是可微的凸函数, 则这类优化问题可通过近端梯度下降法 (proximal gradient descent) 进行求解。

对于向量 \mathbf{x} , 令 $g(\mathbf{x}) = \|\mathbf{x}\|_1$ 表示不可微函数¹、 $h(\mathbf{x}) = \frac{\lambda}{2}\|\mathbf{x} - \mathbf{w}\|_2^2$ (\mathbf{w} 已知) 表示可微函数, 则 ℓ_1 范数最小化问题可归纳为

$$\min_{\mathbf{x}} \|\mathbf{x}\|_1 + \frac{\lambda}{2}\|\mathbf{x} - \mathbf{w}\|_2^2 \quad (8.37)$$

在 ℓ_1 范数最小化问题中, 若 $\mathbf{w} \in \mathbb{R}^T$ 已知、 $\mathbf{x} \in \mathbb{R}^T$ 未知, 则近端算子 (proximal operator) 可写作如下的软阈值函数 (soft thresholding):

$$\mathcal{S}_{1/\lambda}(\mathbf{w}) = \arg \min_{\mathbf{x}} \|\mathbf{x}\|_1 + \frac{\lambda}{2}\|\mathbf{x} - \mathbf{w}\|_2^2 \quad (8.38)$$

对于向量 \mathbf{x} 中的任意元素 x_t , 有

$$x_t := \mathcal{S}_{1/\lambda}(w_t) = \begin{cases} w_t - 1/\lambda & \text{if } w_t > 1/\lambda \\ w_t + 1/\lambda & \text{if } w_t \leq -1/\lambda \\ 0 & \text{otherwise} \end{cases} \quad (8.39)$$

在写法上, 软阈值函数可进一步写作如下形式²:

$$x_t := \frac{w_t}{|w_t|} \cdot \max\{0, |w_t| - 1/\lambda\}, \quad t = 1, 2, \dots, T \quad (8.40)$$

故向量 \mathbf{x} 的解析解为

$$\mathbf{x} := \mathcal{S}_{1/\lambda}(\mathbf{w}) \quad (8.41)$$

其中, $\mathcal{S}_{1/\lambda}(\cdot)$ 表示超参数为 λ 的软阈值函数; $\max\{x, y\}$ 表示取 x 与 y 之间较大的数值。

8.2.4 循环矩阵核范数最小化问题

对于任意观测向量 $\mathbf{y} \in \mathbb{R}^T$, 若被观测元素的索引集合为 Ω , 则循环矩阵核范数最小化问题 [Liu and Zhang, 2022] 可描述为

$$\begin{aligned} \min_{\mathbf{x}} \quad & \|\mathcal{C}(\mathbf{x})\|_* \\ \text{s.t.} \quad & \|\mathcal{P}_{\Omega}(\mathbf{x} - \mathbf{y})\|_2 \leq \epsilon \end{aligned} \quad (8.42)$$

其中, 约束条件中的 ϵ 表示容许误差。为便于求解, 可将上述核范数最小化问题中的约束条件进行改写, 令约束条件作为目标函数的正则项, 则构造出来的优化问题为

$$\begin{aligned} \min_{\mathbf{x}, \mathbf{z}} \quad & \|\mathcal{C}(\mathbf{x})\|_* + \frac{\eta}{2}\|\mathcal{P}_{\Omega}(\mathbf{z} - \mathbf{y})\|_2^2 \\ \text{s.t.} \quad & \mathbf{x} = \mathbf{z} \end{aligned} \quad (8.43)$$

其中, η 为正则项的权重系数。由于观测向量存在缺失值, 为便于求解优化问题, 可令中间变量的等价关系作为约束条件。

¹对于任意向量 $\mathbf{x} \in \mathbb{R}^T$, 其 ℓ_1 范数为 $\|\mathbf{x}\|_1 = \sum_{t=1}^T |x_t|$, 即元素绝对值之和。

²这种写法可确保当 $\mathbf{w}, \mathbf{x} \in \mathbb{C}^T$ 时, 软阈值函数依然适用。

通常来说, 求解上述优化问题可采用 ADMM 算法。使用 ADMM 求解过程中, 需首先构造增广拉格朗日函数, 即

$$\mathcal{L}(\mathbf{x}, \mathbf{z}, \mathbf{w}) = \|\mathcal{C}(\mathbf{x})\|_* + \frac{\lambda}{2}\|\mathbf{x} - \mathbf{z}\|_2^2 + \langle \mathbf{w}, \mathbf{x} - \mathbf{z} \rangle + \frac{\eta}{2}\|\mathcal{P}_\Omega(\mathbf{z} - \mathbf{y})\|_2^2 \quad (8.44)$$

其中, $\mathbf{w} \in \mathbb{R}^T$ 为拉格朗日乘子; λ 为权重系数。符号 $\langle \cdot, \cdot \rangle$ 表示内积 (inner product), 满足如下关系:

$$\langle \mathbf{w}, \mathbf{x} - \mathbf{z} \rangle = \mathbf{w}^\top (\mathbf{x} - \mathbf{z}) \in \mathbb{R} \quad (8.45)$$

然后, 可采用如下 ADMM 算法:

$$\begin{cases} \mathbf{x} := \arg \min_{\mathbf{x}} \mathcal{L}(\mathbf{x}, \mathbf{z}, \mathbf{w}) \\ \mathbf{z} := \arg \min_{\mathbf{z}} \mathcal{L}(\mathbf{x}, \mathbf{z}, \mathbf{w}) \\ \mathbf{w} := \mathbf{w} + \lambda(\mathbf{x} - \mathbf{z}) \end{cases} \quad (8.46)$$

对于变量 \mathbf{x} , 有

$$\begin{aligned} \mathbf{x} &:= \arg \min_{\mathbf{x}} \|\mathcal{C}(\mathbf{x})\|_* + \frac{\lambda}{2}\|\mathbf{x} - \mathbf{z}\|_2^2 + \langle \mathbf{w}, \mathbf{x} - \mathbf{z} \rangle \\ &= \arg \min_{\mathbf{x}} \|\mathcal{C}(\mathbf{x})\|_* + \frac{\lambda}{2}\mathbf{x}^\top \mathbf{x} - \lambda \langle \mathbf{z}, \mathbf{x} \rangle + \langle \mathbf{w}, \mathbf{x} \rangle \\ &= \arg \min_{\mathbf{x}} \|\mathcal{C}(\mathbf{x})\|_* + \frac{\lambda}{2}\mathbf{x}^\top \mathbf{x} - \lambda \langle \mathbf{z} - \mathbf{w}/\lambda, \mathbf{x} \rangle \\ &= \arg \min_{\mathbf{x}} \|\mathcal{C}(\mathbf{x})\|_* + \frac{\lambda}{2}\|\mathbf{x} - \mathbf{z} + \mathbf{w}/\lambda\|_2^2 \\ &= \arg \min_{\mathbf{x}} \|\mathcal{F}(\mathbf{x})\|_1 + \frac{\lambda}{2T}\|\mathcal{F}(\mathbf{x} - \mathbf{z} + \mathbf{w}/\lambda)\|_2^2 \end{aligned} \quad (8.47)$$

令

$$\mathbf{h} = \mathbf{z} - \mathbf{w}/\lambda \quad (8.48)$$

若 $\{\hat{\mathbf{x}}, \hat{\mathbf{h}}\} = \{\mathcal{F}(\mathbf{x}), \mathcal{F}(\mathbf{h})\}$ 记作离散傅立叶变换之后的变量, 根据公式(8.38)中给出的软阈值过程, 则变量 $\hat{\mathbf{x}}$ 的解析解为

$$\hat{x}_t = \frac{\hat{h}_t}{|\hat{h}_t|} \cdot \max\{0, |\hat{h}_t| - T/\lambda\}, \quad t = 1, 2, \dots, T \quad (8.49)$$

因此, 通过离散傅立叶逆变换, 则变量 \mathbf{x} 的更新公式为

$$\mathbf{x} := \mathcal{F}^{-1}(\hat{\mathbf{x}}) \quad (8.50)$$

例 52. 现有循环矩阵核范数最小化问题为

$$\min_{\mathbf{x}} \|\mathcal{C}(\mathbf{x})\|_* + \frac{\lambda}{2}\|\mathbf{x} - \mathbf{z}\|_2^2 \quad (8.51)$$

其中, $\mathbf{z} \in \mathbb{R}^T$ 为已知变量; $\mathbf{x} \in \mathbb{R}^T$ 为待优化变量。

通常来说, 可将公式(8.51)中的优化问题写作如下形式:

$$\min_{\mathbf{x}} \|\mathcal{F}(\mathbf{x})\|_1 + \frac{\lambda}{2T}\|\mathcal{F}(\mathbf{x} - \mathbf{z})\|_2^2 \quad (8.52)$$

其中, $\mathcal{F}(\cdot)$ 表示离散傅立叶变换。令

$$\hat{\mathbf{h}} = \mathcal{F}(\mathbf{z}) \quad (8.53)$$

则该 ℓ_1 范数最小化问题的解析解为

$$\mathbf{x} := \mathcal{F}^{-1}(\hat{\mathbf{x}}) \quad \hat{x}_t = \frac{\hat{h}_t}{|\hat{h}_t|} \cdot \max\{0, |\hat{h}_t| - T/\lambda\}, t = 1, 2, \dots, T \quad (8.54)$$

其中, $\mathcal{F}^{-1}(\cdot)$ 表示离散傅立叶逆变换。

不妨令 $\mathbf{z} = (0, 1, 2, 3, 4)^\top$ 与 $\lambda = 2$, 试写出公式(8.51)的最优解。

解. 对已知变量 \mathbf{z} 进行离散傅立叶变换, 有

$$\hat{\mathbf{h}} = \mathcal{F}(\mathbf{z}) = (10, -2.5 + 3.44i, -2.5 + 0.81i, -2.5 - 0.81i, -2.5 - 3.44i)^\top \quad (8.55)$$

其中, $i = \sqrt{-1}$ 表示复数的虚部。向量 $\hat{\mathbf{h}}$ 的绝对值为

$$|\hat{\mathbf{h}}| = (10, 4.25, 2.63, 2.63, 4.25)^\top \quad (8.56)$$

根据公式(8.54), 可得到

$$\mathbf{x} = (1.04, 0.86, 1.5, 2.14, 1.96)^\top \quad (8.57)$$

在这里, 公式(8.51)中的优化问题目标函数为

$$\|\mathcal{C}(\mathbf{x})\|_* + \frac{\lambda}{2} \|\mathbf{x} - \mathbf{z}\|_2^2 = 17.51 \quad (8.58)$$

对于变量 \mathbf{z} , 可分别对 $\mathcal{P}_\Omega(\mathbf{z})$ 与 $\mathcal{P}_\Omega^\perp(\mathbf{z})$ 求偏导数:

$$\begin{cases} \frac{\partial \mathcal{L}(\mathbf{x}, \mathbf{z}, \mathbf{w})}{\partial \mathcal{P}_\Omega(\mathbf{z})} = \lambda \mathcal{P}_\Omega(\mathbf{z} - \mathbf{x} - \mathbf{w}/\lambda) + \eta \mathcal{P}_\Omega(\mathbf{z} - \mathbf{y}) \\ \frac{\partial \mathcal{L}(\mathbf{x}, \mathbf{z}, \mathbf{w})}{\partial \mathcal{P}_\Omega^\perp(\mathbf{z})} = \lambda \mathcal{P}_\Omega^\perp(\mathbf{z} - \mathbf{x} - \mathbf{w}/\lambda) \end{cases} \quad (8.59)$$

令偏导数为 $\mathbf{0}$, 则变量 \mathbf{z} 的解析解为

$$\mathbf{z} := \frac{1}{\lambda + \eta} \mathcal{P}_\Omega(\lambda \mathbf{x} + \mathbf{w} + \eta \mathbf{y}) + \mathcal{P}_\Omega^\perp(\mathbf{x} + \mathbf{w}/\lambda) \quad (8.60)$$

算法5给出了循环矩阵核范数最小化算法的具体实现过程。

Algorithm 5 循环矩阵核范数最小化算法

Input: 观测向量 $\mathbf{y} \in \mathbb{R}^T$, 被观测元素的索引集合 Ω , 超参数 $\{\lambda, \eta\}$ 。

Output: 重构出来的向量 $\mathbf{x} \in \mathbb{R}^T$ 。

- 1: 对变量 $\{\mathbf{x}, \mathbf{z}, \mathbf{w}\}$ 进行初始化;
 - 2: **for** $i = 0$ to 最大迭代次数 **do**
 - 3: 对变量 $\{\mathbf{z}, \mathbf{w}\}$ 进行快速傅立叶变换;
 - 4: 根据公式(8.48)计算 \mathbf{h} ;
 - 5: 根据公式(8.49)计算 $\hat{\mathbf{x}}$;
 - 6: 令 $\mathbf{x} := \mathcal{F}^{-1}(\hat{\mathbf{x}})$;
 - 7: 根据公式(8.60)计算 \mathbf{z} ;
 - 8: 计算 $\mathbf{w} := \mathbf{w} + \lambda(\mathbf{x} - \mathbf{z})$;
 - 9: **end for**
-

8.3 低秩拉普拉斯卷积模型

8.3.1 拉普拉斯卷积核

一般而言, 对关系型数据进行建模时, 可对数据之间的关联构造拉普拉斯矩阵。若 $\mathbf{D} \in \mathbb{R}^{T \times T}$ 表示度矩阵 (degree matrix)、 $\mathbf{A} \in \mathbb{R}^{T \times T}$ 表示邻接矩阵 (adjacency matrix), 则对应的拉普拉斯矩阵为

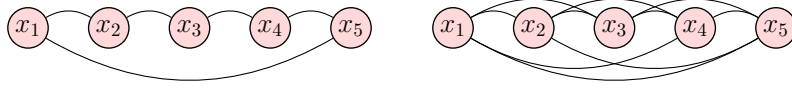
$$\mathbf{L} = \mathbf{D} - \mathbf{A} \quad (8.61)$$

举例来说, 图8.2(a)中的图模型为两两相结、首尾相连的循环图, 对应的图拉普拉斯矩阵为

$$\mathbf{L} = \begin{bmatrix} 2 & -1 & 0 & 0 & -1 \\ -1 & 2 & -1 & 0 & 0 \\ 0 & -1 & 2 & -1 & 0 \\ 0 & 0 & -1 & 2 & -1 \\ -1 & 0 & 0 & -1 & 2 \end{bmatrix} \quad (8.62)$$

图8.2(b)对应的图拉普拉斯矩阵为

$$\mathbf{L} = \begin{bmatrix} 4 & -1 & -1 & -1 & -1 \\ -1 & 4 & -1 & -1 & -1 \\ -1 & -1 & 4 & -1 & -1 \\ -1 & -1 & -1 & 4 & -1 \\ -1 & -1 & -1 & -1 & 4 \end{bmatrix} \quad (8.63)$$



(a) 度为 2 的循环图

(b) 度为 4 的循环图

图 8.2: 基于关系型数据 $\{x_1, x_2, \dots, x_5\}$ 的无向循环图

由于这两个拉普拉斯矩阵均为循环矩阵, 不妨令其第一列为拉普拉斯核 (Laplacian kernel), 则两者的拉普拉斯核分别为

$$\boldsymbol{\ell} = (2, -1, 0, 0, -1)^\top \quad \boldsymbol{\ell} = (4, -1, -1, -1, -1)^\top \quad (8.64)$$

由此, 对于任意向量 $\mathbf{x} = (x_1, x_2, \dots, x_T)^\top \in \mathbb{R}^T$, 令 $\tau \in \mathbb{N}^+$ 为拉普拉斯核的超参数, 其中, $\tau \leq \frac{1}{2}(T-1)$, 则拉普拉斯核可被定义 [Chen et al., 2022b] 为

$$\boldsymbol{\ell} = (2\tau, \underbrace{-1, \dots, -1}_\tau, 0, \dots, 0, \underbrace{-1, \dots, -1}_\tau)^\top \in \mathbb{R}^T \quad (8.65)$$

该拉普拉斯核同时也是拉普拉斯矩阵的第一列。拉普拉斯核的第一个元素为 2τ , 表示拉普拉斯矩阵的度。

8.3.2 拉普拉斯时序正则

在时间序列中, 若数据向量为 $\mathbf{x} = (x_1, x_2, \dots, x_T)^\top \in \mathbb{R}^T$, 则基于拉普拉斯矩阵的时序正则项为

$$\mathcal{R}_\tau(\mathbf{x}) = \frac{1}{2} \|\mathbf{L}\mathbf{x}\|_2^2 \quad (8.66)$$

根据 Parseval 定理, 可得到基于拉普拉斯核的时序正则项为

$$\mathcal{R}_\tau(\mathbf{x}) = \frac{1}{2} \|\ell \star \mathbf{x}\|_2^2 = \frac{1}{2T} \|\mathcal{F}(\ell) \circ \mathcal{F}(\mathbf{x})\|_2^2 \quad (8.67)$$

例 53. 试根据 Parseval 定理证明公式(8.67)。

解. 在公式(8.67)中, 不妨令

$$\begin{cases} \alpha = \ell \star \mathbf{x} \\ \beta = \mathcal{F}(\ell) \circ \mathcal{F}(\mathbf{x}) \end{cases} \quad (8.68)$$

根据卷积定理, 有

$$\mathcal{F}(\alpha) = \beta \quad (8.69)$$

再根据 Parseval 定理, 则

$$\|\alpha\|_2^2 = \frac{1}{T} \|\mathcal{F}(\alpha)\|_2^2 = \frac{1}{T} \|\beta\|_2^2 \quad (8.70)$$

由此, 公式(8.67)得证。

例 54. 对于时间序列 $\mathbf{x} = (x_1, x_2, \dots, x_T)^\top \in \mathbb{R}^T$ 而言, 基于拉普拉斯核的时序正则项可对时间序列的局部趋势 (local trend) 进行建模, 以拉普拉斯核 $\ell = (2, -1, 0, \dots, 0, -1)^\top \in \mathbb{R}^T$ 为例, 试解释基于拉普拉斯核的时序正则项在局部趋势建模中所起的作用。

解. 根据定义, 时序正则项 $\mathcal{R}_\tau(\mathbf{x})$ 可写作如下形式:

$$\mathcal{R}_\tau(\mathbf{x}) = \frac{1}{2}(2x_1 - (x_2 + x_T))^2 + \frac{1}{2}(2x_2 - (x_3 + x_1))^2 + \dots + \frac{1}{2}(2x_T - (x_1 + x_{T-1}))^2 \quad (8.71)$$

由此可见, 该正则项实际上对时间序列进行了平滑处理。

上述定义的拉普拉斯卷积核对于有向图依然适用, 若令拉普拉斯卷积核为 $\ell = (1, -1, 0, \dots, 0)^\top$, 则相应的拉普拉斯矩阵 [Takayama and Yokota, 2022] 为

$$\mathbf{L} = \begin{bmatrix} 1 & 0 & 0 & \dots & 0 & -1 \\ -1 & 1 & 0 & \dots & 0 & 0 \\ 0 & -1 & 1 & \dots & 0 & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots & \\ 0 & 0 & 0 & \dots & 1 & 0 \\ 0 & 0 & 0 & \dots & -1 & 1 \end{bmatrix} \quad (8.72)$$

除去拉普拉斯矩阵的第一行, 可构造如下矩阵:

$$\Psi = \begin{bmatrix} \mathbf{0}_{(T-1) \times 1} & \mathbf{I}_{T-1} \end{bmatrix} - \begin{bmatrix} \mathbf{I}_{T-1} & \mathbf{0}_{(T-1) \times 1} \end{bmatrix} \quad (8.73)$$

由此, 相应的时序正则项为 $\frac{1}{2} \|\Psi \mathbf{x}\|_2^2$, 该正则项对时间序列建模时可起到平滑的作用。

8.3.3 一维低秩拉普拉斯卷积模型

在时间序列缺失值重构任务中, 对全局趋势 (global trend) 与局部趋势建模往往缺一不可。对于任意时间序列 $\mathbf{x} = (x_1, x_2, \dots, x_T)^\top \in \mathbb{R}^T$ 而言, 可采用循环矩阵核范数捕捉低秩信息、借助拉普拉斯时序正则项刻画局部趋势 (如图8.3所示), 由此, 得到的低秩拉普拉斯卷积模型的目标函数兼具循环矩阵核范数与循环卷积 [Chen et al., 2022b], 即

$$\begin{aligned} \min_{\mathbf{x}} \quad & \|\mathcal{C}(\mathbf{x})\|_* + \frac{\gamma}{2} \|\ell \star \mathbf{x}\|_2^2 \\ \text{s.t.} \quad & \|\mathcal{P}_\Omega(\mathbf{x} - \mathbf{y})\|_2 \leq \epsilon \end{aligned} \quad (8.74)$$

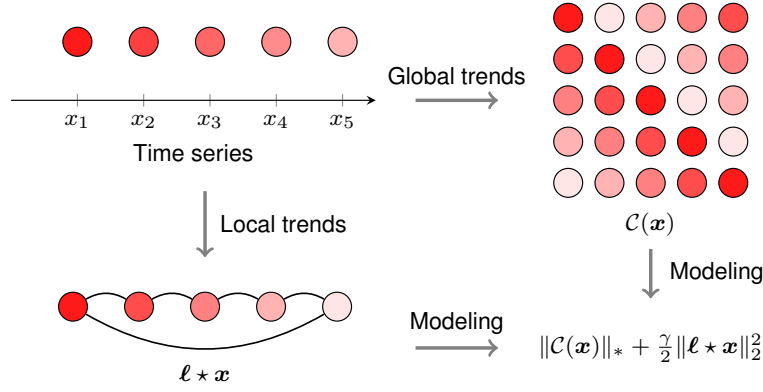


图 8.3: 低秩拉普拉斯卷积模型的示意图

其中, $\ell \in \mathbb{R}^T$ 为表征时序关联的拉普拉斯卷积核; γ 为拉普拉斯时序正则项的权重系数; 约束条件中的 ϵ 表示容许误差。

为便于求解, 可将上述优化问题中的约束条件进行改写, 令约束条件作为目标函数的正则项, 则构造出来的优化问题为

$$\begin{aligned} \min_{\mathbf{x}, \mathbf{z}} \quad & \|C(\mathbf{x})\|_* + \frac{\gamma}{2} \|\ell \star \mathbf{x}\|_2^2 + \frac{\eta}{2} \|\mathcal{P}_\Omega(\mathbf{z} - \mathbf{y})\|_2^2 \\ \text{s.t.} \quad & \mathbf{x} = \mathbf{z} \end{aligned} \quad (8.75)$$

其中, η 为正则项的权重系数。由于观测向量存在缺失值, 为便于求解优化问题, 可令中间变量的等价关系作为约束条件。

使用 ADMM 算法求解时, 需首先构造增广拉格朗日函数, 即

$$\mathcal{L}(\mathbf{x}, \mathbf{z}, \mathbf{w}) = \|C(\mathbf{x})\|_* + \frac{\gamma}{2} \|\ell \star \mathbf{x}\|_2^2 + \frac{\lambda}{2} \|\mathbf{x} - \mathbf{z}\|_2^2 + \langle \mathbf{w}, \mathbf{x} - \mathbf{z} \rangle + \frac{\eta}{2} \|\mathcal{P}_\Omega(\mathbf{z} - \mathbf{y})\|_2^2 \quad (8.76)$$

对于变量 \mathbf{x} , 子问题为

$$\begin{aligned} \mathbf{x} &:= \arg \min_{\mathbf{x}} \|C(\mathbf{x})\|_* + \frac{\gamma}{2} \|\ell \star \mathbf{x}\|_2^2 + \frac{\lambda}{2} \|\mathbf{x} - \mathbf{z} + \mathbf{w}/\lambda\|_2^2 \\ &= \arg \min_{\mathbf{x}} \|\mathcal{F}(\mathbf{x})\|_1 + \frac{\gamma}{2T} \|\mathcal{F}(\ell) \circ \mathcal{F}(\mathbf{x})\|_2^2 + \frac{\lambda}{2T} \|\mathcal{F}(\mathbf{x} - \mathbf{z} + \mathbf{w}/\lambda)\|_2^2 \end{aligned} \quad (8.77)$$

若 $\{\hat{\ell}, \hat{\mathbf{x}}, \hat{\mathbf{z}}, \hat{\mathbf{w}}\} = \{\mathcal{F}(\ell), \mathcal{F}(\mathbf{x}), \mathcal{F}(\mathbf{z}), \mathcal{F}(\mathbf{w})\}$, 令子问题的正则项为

$$f = \frac{\gamma}{2T} \|\mathcal{F}(\ell) \circ \mathcal{F}(\mathbf{x})\|_2^2 + \frac{\lambda}{2T} \|\mathcal{F}(\mathbf{x} - \mathbf{z} + \mathbf{w}/\lambda)\|_2^2 \quad (8.78)$$

相应地, 对于变量 $\hat{\mathbf{x}}$, 函数 f 的偏导数为

$$\begin{aligned} \frac{\partial f}{\partial \hat{\mathbf{x}}} &= \frac{\gamma}{T} \hat{\ell}^* \circ \hat{\ell} \circ \hat{\mathbf{x}} + \frac{\lambda}{T} (\hat{\mathbf{x}} - \hat{\mathbf{z}} + \hat{\mathbf{w}}/\lambda) \\ &= \frac{1}{T} (\gamma \hat{\ell}^* \circ \hat{\ell} + \lambda \mathbf{1}_T) \circ \hat{\mathbf{x}} - \frac{1}{T} (\lambda \hat{\mathbf{z}} - \hat{\mathbf{w}}) \end{aligned} \quad (8.79)$$

其中, 向量 $\mathbf{1}_T \in \mathbb{R}^T$ 的所有元素均为 1; 符号 \cdot^* 表示共轭复数。

此时, 子问题的正则项为

$$f = \frac{\lambda}{2T} \|\hat{\mathbf{x}} - (\lambda \hat{\mathbf{z}} - \hat{\mathbf{w}}) \odot (\gamma \hat{\ell}^* \circ \hat{\ell} + \lambda \mathbf{1}_T)\|_2^2 \quad (8.80)$$

不妨令

$$\hat{\mathbf{h}} = (\lambda \hat{\mathbf{z}} - \hat{\mathbf{w}}) \odot (\gamma \hat{\ell}^* \circ \hat{\ell} + \lambda \mathbf{1}_T) \quad (8.81)$$

对应于 $\frac{\partial f}{\partial \hat{\mathbf{x}}} = \mathbf{0}$ 。

由此，变量 $\hat{\mathbf{x}}$ 的解析解可根据公式(8.49)计算得到，通过离散傅立叶逆变换，变量 \mathbf{x} 的更新公式为 $\mathbf{x} := \mathcal{F}^{-1}(\hat{\mathbf{x}})$ 。

在 ADMM 算法中，变量 \mathbf{z} 的解析解与公式(8.60)一致；变量 \mathbf{w} 的更新公式参见公式(8.46)。算法6给出了一维低秩拉普拉斯卷积模型的具体实现过程。

Algorithm 6 一维低秩拉普拉斯卷积模型

Input: 观测向量 $\mathbf{y} \in \mathbb{R}^T$ ，被观测元素的索引集合 Ω ，超参数 $\{\gamma, \lambda, \eta, \tau\}$ 。

Output: 重构出来的向量 $\mathbf{x} \in \mathbb{R}^T$ 。

- 1: 对变量 $\{\mathbf{x}, \mathbf{z}, \mathbf{w}\}$ 进行初始化；
 - 2: **for** $i = 0$ to 最大迭代次数 **do**
 - 3: 对变量 $\{\mathbf{z}, \mathbf{w}\}$ 进行快速傅立叶变换；
 - 4: 根据公式(8.81)计算 $\hat{\mathbf{h}}$ ；
 - 5: 根据公式(8.49)计算 $\hat{\mathbf{x}}$ ；
 - 6: 令 $\mathbf{x} := \mathcal{F}^{-1}(\hat{\mathbf{x}})$ ；
 - 7: 根据公式(8.60)计算 \mathbf{z} ；
 - 8: 计算 $\mathbf{w} := \mathbf{w} + \lambda(\mathbf{x} - \mathbf{z})$ ；
 - 9: **end for**
-

例 55. 给定某高速公路断面交通流的车速时间序列如图8.4(a)所示，采集数据的时间粒度为 15 分钟，即每天预期可获取 96 个观测值；采集时长为 3 天，预期产生 288 个观测值，即 $T = 288$ 。现假设该车速时间序列存在 90% 的缺失值，如图8.4(b)所示，试使用一维低秩拉普拉斯卷积模型对部分观测的车速时间序列进行重构、修复缺失值。

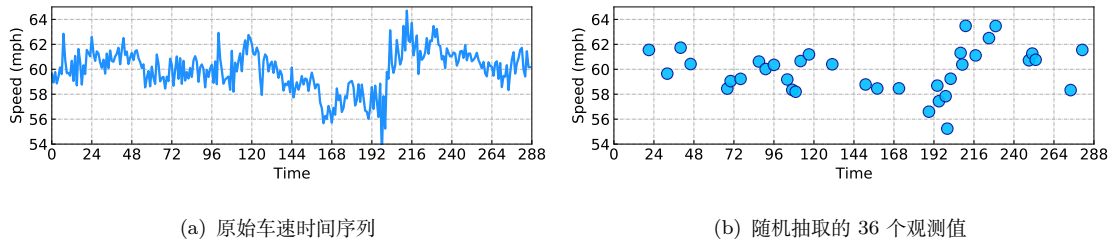


图 8.4: 某高速公路断面交通流的车速时间序列，其中，蓝色曲线表示车速时间序列；蓝色圆圈表示抽取的部分观测值。

解. 在图8.5所示的重构时间序列中，将一维低秩拉普拉斯卷积模型中的超参数设置为 $\lambda = 5 \times 10^{-3}T$, $\gamma = 2\lambda$, $\eta = 100\lambda$, $\tau = 2$ 。从中不难发现，该模型重构出来的时间序列与真实时间序列趋势吻合。

8.3.4 二维低秩拉普拉斯卷积模型

8.4 Python 实现代码

Table of functions

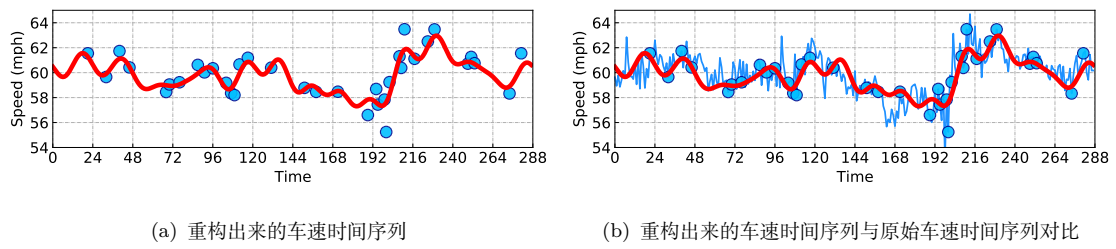


图 8.5: 基于一维低秩拉普拉斯卷积模型的 90% 缺失率的车速时间序列重构, 其中, 红色曲线表示重构出来的车速时间序列。

第九章 基于 Hankel 结构的张量分解

第十章 低秩深度学习时空预测模型

在深度学习中，张量是最为基本的数学结构，张量是神经网络中的核心数据结构，用于表示输入数据、权重参数、中间层结果和输出结果。神经网络中的所有操作，如卷积、池化、非线性激活函数等都是基于张量的运算。目前已有大量的研究采用各种深度学习结构来进行时空交通数据预测，这些研究处理的任务如下式所示：

$$\{\mathbf{X}_{t+1}, \mathbf{X}_{t+2}, \dots, \mathbf{X}_{t+h}\} = f_{\theta}(\{\mathbf{X}_{t-p}, \dots, \mathbf{X}_{t-1}, \mathbf{X}_t\}), \quad (10.1)$$

其中 $\mathbf{X}_{t+i} \in \mathbb{R}^{N \times C}$ ， N 为空间点的数目， C 为变量的数目， h 和 p 分别为预测区间和历史区间的长度， θ 是神经网络的参数。因此，时空交通预测问题的实质为求一个历史三维张量 $\mathcal{X} \in \mathbb{R}^{p \times N \times C}$ 到未来张量 $\mathcal{Y} \in \mathbb{R}^{h \times N \times C}$ 的映射。

图神经网络是当前时空预测领域最为常用的深度学习模型，其主要采用图卷积的框架来捕捉交通数据在空间上的相关性。下面简单介绍一下基础图卷积的概念，及其在时空预测框架下的张量表达形式。

首先，我们需要定义一些基本的图论概念。一个图 \mathcal{G} 可以表示为一个二元组 $(\mathcal{V}, \mathcal{E})$ ，其中 \mathcal{V} 是节点集合， \mathcal{E} 是边集合。每条边连接两个交通节点，可以表示为一个有序对 (u, v) ，其中 $u, v \in \mathcal{V}$ 。我们可以使用邻接矩阵 \mathbf{A} 来表示图的连接关系。对于无向图来说，邻接矩阵是对称的，即 $A_{ij} = A_{ji}$ ，表示节点 i 和节点 j 之间是否有连接。对于有向图来说，邻接矩阵是不对称的， A_{ij} 表示从节点 i 到节点 j 是否有连接。 A_{ij} 的数值大小表示两个节点之间的关联强度，在时空交通预测领域， A_{ij} 一般通过距离的高斯核函数来获取：

$$A_{ij} = \begin{cases} \exp\left(-\frac{\text{dist}_{ij}^2}{\sigma^2}\right) & \text{otherwise} \\ 0 & \text{if } \exp\left(-\frac{\text{dist}_{ij}^2}{\sigma^2}\right) < \epsilon \end{cases} \quad (10.2)$$

其中 σ 为所有距离 dist_{ij} 的方差， ϵ 为设定的某个阈值。在基础的图卷积神经网络之中，常采用时间上共享的图卷积层获取空间关联性，这等价于在时间模式上进行张量矩阵乘积。下面给出基础图卷积操作下的张量表达形式：

$$\mathcal{H}^{(l+1)} = \mathcal{H}^{(l)} \times_2 \hat{\mathbf{A}} \times_3 \mathbf{W}^{(l)}, \quad (10.3)$$

$\mathcal{H}^{(l)} \in \mathbb{R}^{p \times N \times C^{*(l)}}$ 为第 l 层神经网络的特征， $\hat{\mathbf{A}} = \mathbf{D}^{-1/2} \mathbf{A} \mathbf{D}^{-1/2}$ ， \mathbf{D} 为对焦矩阵，有 $D_{ii} = \sum_j A_{ij}$ ， $\mathbf{W}^{(l)} \in \mathbb{R}^{C^{*(l+1)} \times C^{*(l)}}$ 为第 l 层图神经网络的参数。

附录 A 公开交通数据集

A.1 波特兰高速公路交通流量数据集

A.2 西雅图高速公路交通速度数据集

附录 B 算法详解

B.1 共轭梯度法

共轭梯度法 (conjugate gradient) 也被称为共轭梯度下降法 (conjugate gradient descent), 是一种经典的迭代优化算法, 可用于求解特定的非约束优化问题, 是由数学家 Magnus Hestenes 与 Eduard Stiefel 于 1952 年为求解线性系统而提出来的。¹在凸优化问题中, 二次型 (quadratic form, 即关于向量的二次函数型函数) 通常可写作如下形式:

$$f(\mathbf{x}) = \frac{1}{2} \mathbf{x}^\top \mathbf{A} \mathbf{x} - \mathbf{b}^\top \mathbf{x} + c \quad (\text{B.1})$$

其中, $\mathbf{A} \in \mathbb{R}^{n \times n}$ 、 $\mathbf{b} \in \mathbb{R}^n$ 与 $c \in \mathbb{R}$ 已知, 矩阵 \mathbf{A} 为实对称矩阵和正定矩阵; $\mathbf{x} \in \mathbb{R}^n$ 为函数的自变量。由于 \mathbf{A} 为实对称矩阵, 因此, 函数 $f(\mathbf{x})$ 的导数为

$$\frac{df(\mathbf{x})}{d\mathbf{x}} = \mathbf{A}\mathbf{x} - \mathbf{b} \quad (\text{B.2})$$

通常也被称为梯度 (gradient), 可简写成 $\nabla f(\mathbf{x})$ 。

在这里, 令 $\frac{df(\mathbf{x})}{d\mathbf{x}} = \mathbf{0}$ (一阶导数为 $\mathbf{0}$) 即可得到 $f(\mathbf{x})$ 的极值, 对应的最优解为

$$\mathbf{x}^* = \mathbf{A}^{-1}\mathbf{b} \quad (\text{B.3})$$

通常而言, 在二次型中, 求解 $f(\mathbf{x})$ 的极值可转化为对线性方程组 $\mathbf{A}\mathbf{x} - \mathbf{b} = \mathbf{0}$ 的求解, 且不难写出解析解。为了引申出以下内容, 若 \mathbf{A} 为正定矩阵, 相应的线性方程组也可通过共轭梯度法进行快速求解, 共轭梯度法的优点在于算法形式简单且无需精心调整参数, 在速度上也远快于梯度下降法。需要注意的是, 为了与梯度下降法、最速梯度下降法在概念上保持一致, 以下不妨称共轭梯度法为共轭梯度下降法。

B.1.1 梯度下降法

梯度下降法是数值计算中最为常用的一种迭代算法, 被广泛用于各种机器学习模型中, 能求解各类优化问题。在算法实现时, 需设定梯度下降法的迭代过程, 令

$$\mathbf{g}_\ell \triangleq \nabla f(\mathbf{x}_\ell) = \mathbf{A}\mathbf{x}_\ell - \mathbf{b} \quad (\text{B.4})$$

表示第 $\ell \in \mathbb{N}^0 = \{0, 1, 2, \dots\}$ 次迭代的梯度, 可将其理解为使得函数 $f(\mathbf{x})$ 增长最快的方向。事实上, 梯度下降的搜索方向一般是通过梯度 \mathbf{g}_ℓ 来确定的, 变量 \mathbf{x} 的更新规则为

$$\mathbf{x}_{\ell+1} = \mathbf{x}_\ell - \alpha_\ell \mathbf{g}_\ell \quad (\text{B.5})$$

其中, 超参数 α_ℓ 表示梯度下降的步长 (step size), 通常可将其设置为常数, 即 $\alpha_\ell = \alpha$ 。梯度下降法的更新规则可简单归纳为

$$\begin{cases} \mathbf{g}_\ell = \mathbf{A}\mathbf{x}_\ell - \mathbf{b} \\ \mathbf{x}_{\ell+1} = \mathbf{x}_\ell - \alpha \mathbf{g}_\ell \end{cases} \quad (\text{B.6})$$

¹以下内容主要参考<https://gregorygundersen.com/blog/>中的博客《Conjugate Gradient Descent》。

即对梯度 \mathbf{g}_ℓ 与变量 \mathbf{x}_ℓ 进行交替更新。在整个迭代过程中, 需要不断计算梯度 \mathbf{g}_ℓ 与变量 \mathbf{x}_ℓ , 使得算法最终达到收敛。

B.1.2 最速梯度下降法

为了加速梯度下降的收敛, 最速梯度下降法 (steepest gradient descent) 可在每次迭代中自动计算最优的步长 α_ℓ 。在梯度下降法中, 从公式(B.5)给出的变量 \mathbf{x} 更新规则可推知, $\mathbf{x}_{\ell+1}$ 的计算结果会使得目标函数 $f(\mathbf{x}_{\ell+1})$ 接近最小化, 因此, 不妨构造如下优化问题:

$$\begin{aligned}\hat{\alpha}_\ell &:= \arg \min_{\alpha_\ell} f(\mathbf{x}_\ell - \alpha_\ell \mathbf{g}_\ell) \\ &= \arg \min_{\alpha_\ell} \frac{1}{2}(\mathbf{x}_\ell - \alpha_\ell \mathbf{g}_\ell)^\top \mathbf{A}(\mathbf{x}_\ell - \alpha_\ell \mathbf{g}_\ell) - \mathbf{b}^\top (\mathbf{x}_\ell - \alpha_\ell \mathbf{g}_\ell) + c \\ &= \arg \min_{\alpha_\ell} \frac{1}{2}\alpha_\ell^2 \mathbf{g}_\ell^\top \mathbf{A} \mathbf{g}_\ell - \alpha_\ell \mathbf{g}_\ell^\top \mathbf{A} \mathbf{x}_\ell + \alpha_\ell \mathbf{g}_\ell^\top \mathbf{b}\end{aligned}\quad (\text{B.7})$$

将目标函数记作 h , 则关于步长 α_ℓ 的偏导数为

$$\frac{\partial h}{\partial \alpha_\ell} = \alpha_\ell \mathbf{g}_\ell^\top \mathbf{A} \mathbf{g}_\ell - \mathbf{g}_\ell^\top (\mathbf{A} \mathbf{x}_\ell - \mathbf{b}) \quad (\text{B.8})$$

令偏导数 $\frac{\partial h}{\partial \alpha_\ell} = 0$, 可得到

$$\alpha_\ell = \frac{\mathbf{g}_\ell^\top (\mathbf{A} \mathbf{x}_\ell - \mathbf{b})}{\mathbf{g}_\ell^\top \mathbf{A} \mathbf{g}_\ell} = \frac{\mathbf{g}_\ell^\top \mathbf{g}_\ell}{\mathbf{g}_\ell^\top \mathbf{A} \mathbf{g}_\ell} \quad (\text{B.9})$$

因此, 最速梯度下降法的更新规则可归纳为

$$\begin{cases} \mathbf{g}_\ell = \mathbf{A} \mathbf{x}_\ell - \mathbf{b} \\ \alpha_\ell = \frac{\mathbf{g}_\ell^\top \mathbf{g}_\ell}{\mathbf{g}_\ell^\top \mathbf{A} \mathbf{g}_\ell} \\ \mathbf{x}_{\ell+1} = \mathbf{x}_\ell - \alpha_\ell \mathbf{g}_\ell \end{cases} \quad (\text{B.10})$$

相比梯度下降法, 最速梯度下降法的步长在每次迭代都会经过优化, 因此, 优化后的步长会在很大程度上加速梯度下降的收敛。

另外, 由 $\frac{\partial h}{\partial \alpha_\ell} = 0$ 可推导出

$$\begin{aligned}\frac{\partial h}{\partial \alpha_\ell} &= \alpha_\ell \mathbf{g}_\ell^\top \mathbf{A} \mathbf{g}_\ell - \mathbf{g}_\ell^\top (\mathbf{A} \mathbf{x}_\ell - \mathbf{b}) \\ &= -\mathbf{g}_\ell^\top (\mathbf{A} \mathbf{x}_\ell - \mathbf{b} - \alpha_\ell \mathbf{A} \mathbf{g}_\ell) \\ &= -\mathbf{g}_\ell^\top (\mathbf{A}(\mathbf{x}_\ell - \alpha_\ell \mathbf{g}_\ell) - \mathbf{b}) \\ &= -\mathbf{g}_\ell^\top (\mathbf{A} \mathbf{x}_{\ell+1} - \mathbf{b}) \\ &= -\mathbf{g}_\ell^\top \mathbf{g}_{\ell+1} \\ &= 0\end{aligned}\quad (\text{B.11})$$

这意味着梯度 \mathbf{g}_ℓ 与 $\mathbf{g}_{\ell+1}$ 在迭代过程中会保持正交。

例 56. 给定二次型为

$$f(\mathbf{x}) = \frac{1}{2} \mathbf{x}^\top \mathbf{A} \mathbf{x} - \mathbf{b}^\top \mathbf{x} + c \quad (\text{B.12})$$

其中, 已知 $\mathbf{A} = \begin{bmatrix} 4 & 1 \\ 1 & 3 \end{bmatrix}$ 与 $\mathbf{b} = \begin{bmatrix} 1 \\ 2 \end{bmatrix}$, 试使用最速梯度下降法求 $f(\mathbf{x})$ 的极值对应的最优解。

解. 将最速梯度下降法的迭代过程初始值设置为

$$\mathbf{x}_0 = \begin{bmatrix} 0 \\ 0 \end{bmatrix} \quad (\text{B.13})$$

并根据公式(B.10)所示的更新规则进行迭代。

- 第一次迭代:

$$\mathbf{g}_0 = \mathbf{A}\mathbf{x}_0 - \mathbf{b} = \begin{bmatrix} -1 \\ -2 \end{bmatrix} \quad (\text{B.14})$$

$$\alpha_0 = \frac{\mathbf{g}_0^\top \mathbf{g}_0}{\mathbf{g}_0^\top \mathbf{A} \mathbf{g}_0} = \frac{1}{4} \quad (\text{B.15})$$

$$\mathbf{x}_1 = \mathbf{x}_0 - \alpha_0 \mathbf{g}_0 = \begin{bmatrix} \frac{1}{4} \\ \frac{1}{2} \end{bmatrix} \quad (\text{B.16})$$

- 第二次迭代:

$$\mathbf{g}_1 = \mathbf{A}\mathbf{x}_1 - \mathbf{b} = \begin{bmatrix} \frac{1}{2} \\ -\frac{1}{4} \end{bmatrix} \quad (\text{B.17})$$

$$\alpha_1 = \frac{\mathbf{g}_1^\top \mathbf{g}_1}{\mathbf{g}_1^\top \mathbf{A} \mathbf{g}_1} = \frac{1}{3} \quad (\text{B.18})$$

$$\mathbf{x}_2 = \mathbf{x}_1 - \alpha_1 \mathbf{g}_1 = \begin{bmatrix} \frac{1}{12} \\ \frac{7}{12} \end{bmatrix} \quad (\text{B.19})$$

- 第三次迭代:

$$\mathbf{g}_2 = \mathbf{A}\mathbf{x}_2 - \mathbf{b} = \begin{bmatrix} -\frac{1}{12} \\ -\frac{1}{6} \end{bmatrix} \quad (\text{B.20})$$

$$\alpha_2 = \frac{\mathbf{g}_2^\top \mathbf{g}_2}{\mathbf{g}_2^\top \mathbf{A} \mathbf{g}_2} = \frac{1}{4} \quad (\text{B.21})$$

$$\mathbf{x}_3 = \mathbf{x}_2 - \alpha_2 \mathbf{g}_2 = \begin{bmatrix} \frac{5}{48} \\ \frac{15}{24} \end{bmatrix} \quad (\text{B.22})$$

- 第四次迭代:

$$\mathbf{g}_3 = \mathbf{A}\mathbf{x}_3 - \mathbf{b} = \begin{bmatrix} \frac{1}{24} \\ -\frac{1}{48} \end{bmatrix} \quad (\text{B.23})$$

$$\alpha_3 = \frac{\mathbf{g}_3^\top \mathbf{g}_3}{\mathbf{g}_3^\top \mathbf{A} \mathbf{g}_3} = \frac{1}{3} \quad (\text{B.24})$$

$$\mathbf{x}_4 = \mathbf{x}_3 - \alpha_3 \mathbf{g}_3 = \begin{bmatrix} \frac{13}{144} \\ \frac{91}{144} \end{bmatrix} \quad (\text{B.25})$$

继续迭代下去，最速梯度下降法能够找到 \mathbf{x}^* 的近似解。

B.1.3 共轭梯度下降法

在二次型中, 矩阵 $\mathbf{A} \in \mathbb{R}^{n \times n}$ 已知, 由于 \mathbf{A} 是实对称矩阵和正定矩阵, 若向量 $\mathbf{u} \in \mathbb{R}^n$ 与 $\mathbf{v} \in \mathbb{R}^n$ ($\mathbf{u} \neq \mathbf{v}$) 满足

$$\mathbf{u}^\top \mathbf{A} \mathbf{v} = 0 \quad (\text{B.26})$$

则向量 $\mathbf{u} \in \mathbb{R}^n$ 与 $\mathbf{v} \in \mathbb{R}^n$ 被认为关于矩阵 \mathbf{A} 正交, 另一种叫法是 \mathbf{u} 与 \mathbf{v} 相互“共轭”, 共轭梯度下降法中的“共轭”一词也由此而来。

在最速梯度下降法中, 梯度 \mathbf{g}_ℓ 与 $\mathbf{g}_{\ell+1}$ 始终保持正交特性, 意味着最速梯度下降法会选择一系列相互正交的搜索方向, 然而, 这一性质在实际优化问题求解中往往会显现一定的局限性。将搜索方向的正交特性替换成共轭特性, 梯度下降过程的收敛会进一步加快、所需的迭代次数变少。

共轭的意义

对于一组关于矩阵 \mathbf{A} 共轭的向量:

$$\mathcal{D} = \{\mathbf{d}_1, \mathbf{d}_2, \dots, \mathbf{d}_N\} \quad (\text{B.27})$$

由于这些向量可构成唯一的线性组合, 故函数 $f(\mathbf{x})$ 的极值对应的解可写作如下形式:

$$\mathbf{x}^* = \sum_{i=1}^N \alpha_i \mathbf{d}_i \quad (\text{B.28})$$

其中, $\alpha_i, i = 1, 2, \dots, N$ 表示线性组合的系数, 对公式左右两边同时左乘 $\mathbf{d}_\ell^\top \mathbf{A}$, 其中, $\mathbf{d}_\ell \in \mathcal{D}$, 有

$$\mathbf{d}_\ell^\top \mathbf{A} \mathbf{x}^* = \sum_{i=1}^N \alpha_i \mathbf{d}_\ell^\top \mathbf{A} \mathbf{d}_i \quad (\text{B.29})$$

由于 \mathcal{D} 中所包含的向量相互共轭, 故根据

$$\mathbf{d}_i^\top \mathbf{A} \mathbf{d}_j = 0, i \neq j \quad (\text{B.30})$$

可得

$$\mathbf{d}_\ell^\top \mathbf{A} \mathbf{x}^* = \alpha_\ell \mathbf{d}_\ell^\top \mathbf{A} \mathbf{d}_\ell \quad (\text{B.31})$$

从而可推导出

$$\alpha_\ell = \frac{\mathbf{d}_\ell^\top \mathbf{A} \mathbf{x}^*}{\mathbf{d}_\ell^\top \mathbf{A} \mathbf{d}_\ell} = \frac{\mathbf{d}_\ell^\top \mathbf{b}}{\mathbf{d}_\ell^\top \mathbf{A} \mathbf{d}_\ell} \quad (\text{B.32})$$

这条公式表明: 当找到 N 个相互共轭的搜索方向后, 即 \mathcal{D} , 接下来通过计算每个 α_ℓ 便能确保最终计算出 \mathbf{x}^* , 这种做法的精妙之处在于 \mathbf{x}^* 未知的情况下依然能根据 $\mathbf{A} \mathbf{x}^* = \mathbf{b}$ 计算每个 α_ℓ 。相比最速梯度下降法中的正交, 这无不凸显共轭的意义。

共轭梯度下降

令 \mathbf{r}_ℓ 表示残差 (residual) 向量, 即

$$\mathbf{r}_\ell = \mathbf{b} - \mathbf{A} \mathbf{x}_\ell \quad (\text{B.33})$$

同时也是梯度 \mathbf{g}_ℓ 相反数, 两者满足 $\mathbf{r}_\ell = -\mathbf{g}_\ell$ 。在迭代过程中, 残差向量相互正交, 即 $\mathbf{r}_i^\top \mathbf{r}_j = 0, i \neq j$ 。

令 \mathbf{d}_ℓ 表示每一次迭代的搜索方向, 采用 $\mathbf{d}_0 = \mathbf{r}_0 = \mathbf{b} - \mathbf{A}\mathbf{x}_0$ 作为初始的搜索方向, 恰好是梯度的相反数。在后续迭代过程中, 每次构建的搜索方向 $\mathbf{d}_{\ell+1}$ 都与过去的搜索方向共轭, 其更新规则为

$$\mathbf{d}_{\ell+1} = \mathbf{r}_{\ell+1} + \beta_\ell \mathbf{d}_\ell \quad (\text{B.34})$$

此时, 变量 \mathbf{x} 的更新规则为

$$\mathbf{x}_{\ell+1} = \mathbf{x}_\ell + \alpha_\ell \mathbf{d}_\ell \quad (\text{B.35})$$

进一步, 可得到残差向量的更新规则, 即

$$\begin{aligned} \mathbf{A}\mathbf{x}_{\ell+1} &= \mathbf{A}\mathbf{x}_\ell + \alpha_\ell \mathbf{A}\mathbf{d}_\ell \\ &\Downarrow \\ \mathbf{b} - \mathbf{A}\mathbf{x}_{\ell+1} &= \mathbf{b} - \mathbf{A}\mathbf{x}_\ell - \alpha_\ell \mathbf{A}\mathbf{d}_\ell \\ &\Downarrow \\ \mathbf{r}_{\ell+1} &= \mathbf{r}_\ell - \alpha_\ell \mathbf{A}\mathbf{d}_\ell \end{aligned} \quad (\text{B.36})$$

综上所述, 共轭梯度下降法的更新规则有

$$\begin{cases} \mathbf{x}_{\ell+1} = \mathbf{x}_\ell + \alpha_\ell \mathbf{d}_\ell \\ \mathbf{r}_{\ell+1} = \mathbf{r}_\ell - \alpha_\ell \mathbf{A}\mathbf{d}_\ell \\ \mathbf{d}_{\ell+1} = \mathbf{r}_{\ell+1} + \beta_\ell \mathbf{d}_\ell \end{cases} \quad (\text{B.37})$$

在这里, 残差向量 \mathbf{r}_ℓ 相互正交, 搜索方向 \mathbf{d}_ℓ 相互共轭, 对于任意 $i \neq j$, 满足

$$\begin{cases} \mathbf{r}_i^\top \mathbf{r}_j = 0 \\ \mathbf{d}_i^\top \mathbf{A}\mathbf{d}_j = 0 \end{cases} \quad (\text{B.38})$$

根据这里的正交特性与共轭特性, 可写出系数 α_ℓ 与 β_ℓ 的更新公式。

对于 α_ℓ , 已知

$$\mathbf{r}_{\ell+1}^\top \mathbf{r}_\ell = (\mathbf{r}_\ell - \alpha_\ell \mathbf{A}\mathbf{d}_\ell)^\top \mathbf{r}_\ell = 0 \quad (\text{B.39})$$

可推导出

$$\begin{aligned} \alpha_\ell &= \frac{\mathbf{r}_\ell^\top \mathbf{r}_\ell}{\mathbf{r}_\ell^\top \mathbf{A}\mathbf{d}_\ell} \\ &= \frac{\mathbf{r}_\ell^\top \mathbf{r}_\ell}{(\mathbf{d}_\ell - \beta_{\ell-1} \mathbf{d}_{\ell-1})^\top \mathbf{A}\mathbf{d}_\ell} \\ &= \frac{\mathbf{r}_\ell^\top \mathbf{r}_\ell}{\mathbf{d}_\ell^\top \mathbf{A}\mathbf{d}_\ell} \end{aligned} \quad (\text{B.40})$$

对于 β_ℓ , 已知

$$\mathbf{d}_{\ell+1}^\top \mathbf{A}\mathbf{d}_\ell = (\mathbf{r}_{\ell+1} + \beta_\ell \mathbf{d}_\ell)^\top \mathbf{A}\mathbf{d}_\ell = 0 \quad (\text{B.41})$$

可推导出

$$\begin{aligned} \beta_\ell &= -\frac{\mathbf{r}_{\ell+1}^\top \mathbf{A}\mathbf{d}_\ell}{\mathbf{d}_\ell^\top \mathbf{A}\mathbf{d}_\ell} \\ &= -\frac{\mathbf{r}_{\ell+1}^\top (\mathbf{r}_\ell - \mathbf{r}_{\ell+1})}{\alpha_\ell \mathbf{d}_\ell^\top \mathbf{A}\mathbf{d}_\ell} \\ &= \frac{\mathbf{r}_{\ell+1}^\top \mathbf{r}_{\ell+1}}{\mathbf{r}_\ell^\top \mathbf{r}_\ell} \end{aligned} \quad (\text{B.42})$$

因此，共轭梯度下降法的更新规则可归纳为

$$\begin{cases} \alpha_\ell = \frac{\mathbf{r}_\ell^\top \mathbf{r}_\ell}{\mathbf{d}_\ell^\top \mathbf{A} \mathbf{d}_\ell} \\ \mathbf{x}_{\ell+1} = \mathbf{x}_\ell + \alpha_\ell \mathbf{d}_\ell \\ \mathbf{r}_{\ell+1} = \mathbf{r}_\ell - \alpha_\ell \mathbf{A} \mathbf{d}_\ell \\ \beta_\ell = \frac{\mathbf{r}_{\ell+1}^\top \mathbf{r}_{\ell+1}}{\mathbf{r}_\ell^\top \mathbf{r}_\ell} \\ \mathbf{d}_{\ell+1} = \mathbf{r}_{\ell+1} + \beta_\ell \mathbf{d}_\ell \end{cases} \quad (\text{B.43})$$

例 57. 给定二次型为

$$f(\mathbf{x}) = \frac{1}{2} \mathbf{x}^\top \mathbf{A} \mathbf{x} - \mathbf{b}^\top \mathbf{x} + c \quad (\text{B.44})$$

其中，已知 $\mathbf{A} = \begin{bmatrix} 4 & 1 \\ 1 & 3 \end{bmatrix}$ 与 $\mathbf{b} = \begin{bmatrix} 1 \\ 2 \end{bmatrix}$ ，试使用共轭梯度下降法求 $f(\mathbf{x})$ 的极值对应的最优解。

解. 将共轭梯度下降法的迭代过程初始值设置为

$$\mathbf{x}_0 = \begin{bmatrix} 0 \\ 0 \end{bmatrix} \quad \mathbf{r}_0 = \mathbf{b} - \mathbf{A} \mathbf{x}_0 = \begin{bmatrix} 1 \\ 2 \end{bmatrix} \quad \mathbf{d}_0 = \mathbf{r}_0 = \begin{bmatrix} 1 \\ 2 \end{bmatrix} \quad (\text{B.45})$$

并根据公式(B.43)所示的更新规则进行迭代。

• 第一次迭代:

$$\alpha_0 = \frac{\mathbf{r}_0^\top \mathbf{r}_0}{\mathbf{d}_0^\top \mathbf{A} \mathbf{d}_0} = \frac{1}{4} \quad (\text{B.46})$$

$$\mathbf{x}_1 = \mathbf{x}_0 + \alpha_0 \mathbf{d}_0 = \begin{bmatrix} \frac{1}{4} \\ \frac{1}{2} \end{bmatrix} \quad (\text{B.47})$$

$$\mathbf{r}_1 = \mathbf{r}_0 - \alpha_0 \mathbf{A} \mathbf{d}_0 = \begin{bmatrix} -\frac{1}{2} \\ \frac{1}{4} \end{bmatrix} \quad (\text{B.48})$$

$$\beta_0 = \frac{\mathbf{r}_1^\top \mathbf{r}_1}{\mathbf{r}_0^\top \mathbf{r}_0} = \frac{1}{16} \quad (\text{B.49})$$

$$\mathbf{d}_1 = \mathbf{r}_1 + \beta_0 \mathbf{d}_0 = \begin{bmatrix} -\frac{7}{16} \\ \frac{3}{8} \end{bmatrix} \quad (\text{B.50})$$

• 第二次迭代:

$$\alpha_1 = \frac{\mathbf{r}_1^\top \mathbf{r}_1}{\mathbf{d}_1^\top \mathbf{A} \mathbf{d}_1} = \frac{4}{11} \quad (\text{B.51})$$

$$\mathbf{x}_2 = \mathbf{x}_1 + \alpha_1 \mathbf{d}_1 = \begin{bmatrix} \frac{1}{11} \\ \frac{7}{11} \end{bmatrix} \quad (\text{B.52})$$

$$\mathbf{r}_2 = \mathbf{r}_1 - \alpha_1 \mathbf{A} \mathbf{d}_1 = \begin{bmatrix} 0 \\ 0 \end{bmatrix} \quad (\text{B.53})$$

$$\beta_1 = \frac{\mathbf{r}_2^\top \mathbf{r}_2}{\mathbf{r}_1^\top \mathbf{r}_1} = 0 \quad (\text{B.54})$$

$$\mathbf{d}_2 = \mathbf{r}_2 + \beta_1 \mathbf{d}_1 = \begin{bmatrix} 0 \\ 0 \end{bmatrix} \quad (\text{B.55})$$

由于第二次迭代中的残差向量 $\mathbf{r}_2 = \mathbf{0}$ ，因此，共轭梯度下降法仅迭代两次便找到了最优解，即

$$\mathbf{x}^* = \mathbf{x}_2 = \begin{bmatrix} \frac{1}{11} \\ \frac{7}{11} \end{bmatrix} \quad (\text{B.56})$$

由此可见，相比于最速梯度下降法，共轭梯度下降法收敛更快。

参考文献

- Tamara G Kolda and Brett W Bader. Tensor decompositions and applications. *SIAM review*, 51(3):455–500, 2009.
- Nicholas D Sidiropoulos, Lieven De Lathauwer, Xiao Fu, Kejun Huang, Evangelos E Papalexakis, and Christos Faloutsos. Tensor decomposition for signal processing and machine learning. *IEEE Transactions on Signal Processing*, 65(13):3551–3582, 2017.
- Xinyu Chen, Chengyuan Zhang, Xi-Le Zhao, Nicolas Saunier, and Lijun Sun. Nonstationary temporal matrix factorization for multivariate time series forecasting. *arXiv preprint arXiv:2203.10651*, 2022a.
- Yuejie Chi, Yue M Lu, and Yuxin Chen. Nonconvex optimization meets low-rank matrix factorization: An overview. *IEEE Transactions on Signal Processing*, 67(20):5239–5269, 2019.
- Ruslan Salakhutdinov and Andriy Mnih. Bayesian probabilistic matrix factorization using markov chain monte carlo. In *Proceedings of the 25th international conference on Machine learning*, pages 880–887, 2008.
- Liang Xiong, Xi Chen, Tzu-Kuo Huang, Jeff Schneider, and Jaime G Carbonell. Temporal collaborative filtering with bayesian probabilistic tensor factorization. In *Proceedings of the 2010 SIAM international conference on data mining*, pages 211–222. SIAM, 2010.
- Qibin Zhao, Liqing Zhang, and Andrzej Cichocki. Bayesian cp factorization of incomplete tensors with automatic rank determination. *IEEE transactions on pattern analysis and machine intelligence*, 37(9):1751–1763, 2015.
- Xinyu Chen, Zhaocheng He, and Lijun Sun. A bayesian tensor decomposition approach for spatiotemporal traffic data imputation. *Transportation research part C: emerging technologies*, 98:73–84, 2019.
- Guangcan Liu and Wayne Zhang. Recovery of future data via convolution nuclear norm minimization. *IEEE Transactions on Information Theory*, 69(1):650–665, 2022.
- Xinyu Chen, Zhanhong Cheng, Nicolas Saunier, and Lijun Sun. Laplacian convolutional representation for traffic time series imputation. *arXiv preprint arXiv:2212.01529*, 2022b.
- Hiromu Takayama and Tatsuya Yokota. Fast signal completion algorithm with cyclic convolutional smoothing. In *2022 Asia-Pacific Signal and Information Processing Association Annual Summit and Conference (APSIPA ASC)*, pages 364–371. IEEE, 2022.

致谢

感谢以下贡献者（按姓名拼音排序）在本文撰写过程中提供宝贵的建议与反馈，并协助校对数学公式与语言表达。

- 李本正（电子科技大学）
- 王文硕 (McGill University)
- 王旭东 (McGill University)
- 闫章存（同济大学）
- 张程远 (McGill University)

审稿人列表

感谢以下国内外高校及研究机构研究人员（按姓名拼音排序）对本文初稿进行审阅。

-