黎曼流形稀疏优化: 理论、算法与拓展

修贤超 著

前言

黎曼流形稀疏优化, 作为一种新兴的大数据分析方法, 通过刻画数据内在的非线性流形几何结构, 同时兼具高效的稀疏压缩优势, 能够从海量高维数据中提取关键特征. 正因如此, 黎曼流形稀疏优化受到了学术界和工业界的广泛关注, 在工程技术、经济分析、金融风控、交通调度、军事应用等多个领域展现出重要的应用价值.

本书秉持 “算法创新为核心、交叉应用为导向” 的编写理念, 共分为三部分、12 章, 系统阐述了黎曼流形稀疏优化的理论、算法与拓展. 第一部分为基础篇, 介绍基于传统优化方法的算法及应用. 第 1 章为绪论, 梳理了相关研究背景与发展现状. 第 2 章聚焦图像处理无监督特征选择问题, 构建了稀疏主成分分析方法. 第 3 章面向物联网数据异常检测问题, 设计了稀疏联邦主成分分析方法. 第 4 章针对工业过程故障检测问题, 探讨了稀疏正交非负矩阵分解方法. 第 5 章聚焦高维多视角聚类问题, 发展了稀疏张量典型相关分析方法. 第 6 章关注无监督特征选择鲁棒性问题, 提出了稀疏低秩对比学习方法. 第二部分为进阶篇, 介绍基于深度学习的算法及应用. 第 7 章聚焦高光谱图像去噪问题, 设计了深度张量低秩表示方法. 第 8 章针对红外小目标检测问题, 构建了深度自适应低秩稀疏方法. 第 9 章研究大规模图像分类问题, 提出了深度注意力引导支持矩阵机方法. 第 10 章针对大语言模型剪枝问题, 发展了梯度引导自适应稀疏优化方法. 第三部分为拓展篇, 介绍本领域的两大研究热点. 第 11 章综述了图像反问题的深度展开方法. 第 12 章综述了大语言模型驱动的优化方法.

诚挚感谢导师北京交通大学数学与统计学院孔令臣教授多年来的精心指导和悉心关怀. 特别感谢中山大学智能工程学院刘万泉教授、上海大学微电子学院刘晶晶副教授等对本书提出的宝贵建议. 课题组研究生王新杰同学在书稿的整理过程中, 投入了大量时间与精力, 在此一并致谢. 本书相关研究工作得到了国家自然科学基金项目 (12371306) 的资助, 哈尔滨工业大学出版社为本书出版提供了全面细致的支持, 谨致谢意.

限于作者的水平, 书中难免存在疏漏与不妥之处, 恳请广大读者批评指正.

主要符号对照表

$x \in \mathbb{R}$	实数
$\boldsymbol{x} \in \mathbb{R}^n$	n 维实向量
$\boldsymbol{X} \in \mathbb{R}^{d\times m}$	d 行 m 列实矩阵
$\boldsymbol{\mathcal{X}} \in \mathbb{R}^{n_1\times n_2\times n_3}$	三阶实张量
$\boldsymbol{x}^i \in \mathbb{R}^m$	矩阵 $\boldsymbol{X}$ 的第 i 行向量
$\boldsymbol{x}_j \in \mathbb{R}^d$	矩阵 $\boldsymbol{X}$ 的第 j 列向量
X_{ij}	矩阵 $\boldsymbol{X}$ 的第 i 行 j 列元素
$\|\boldsymbol{X}\|_1$	矩阵 $\boldsymbol{X}$ 的 ℓ_1 范数, 即 $\|\boldsymbol{X}\|_1 = \sum_{i=1}^{d}\sum_{j=1}^{m}\|X_{ij}\|$
$\|\boldsymbol{X}\|_F$	矩阵 $\boldsymbol{X}$ 的 Frobenius 范数, 即 $\|\boldsymbol{X}\|_F = (\sum_{i=1}^{d}\sum_{j=1}^{m}X_{ij}^2)^{1/2}$
$\|\boldsymbol{X}\|_0$	矩阵 $\boldsymbol{X}$ 的 ℓ_0 范数, 即 $\boldsymbol{X}$ 中非零元素的个数
$\|\boldsymbol{X}\|_{2,0}$	矩阵 $\boldsymbol{X}$ 的 $\ell_{2,0}$ 范数, 即 $\boldsymbol{X}$ 中非零行的个数
$\|\boldsymbol{X}\|_{2,1}$	矩阵 $\boldsymbol{X}$ 的 $\ell_{2,1}$ 范数, 即 $\|\boldsymbol{X}\|_{2,1} = \sum_{i=1}^{d}\sqrt{\sum_{j=1}^{m}X_{ij}^2}$
$\|\boldsymbol{X}\|_{2,p}$	矩阵 $\boldsymbol{X}$ 的 $\ell_{2,p}$ 范数, 即 $\|\boldsymbol{X}\|_{2,p} = (\sum_{i=1}^{d}(\sqrt{\sum_{j=1}^{m}X_{ij}^2})^p)^{1/p}$
$\boldsymbol{X}^T$	矩阵 $\boldsymbol{X}$ 的转置
$\mathrm{tr}(\boldsymbol{X})$	矩阵 $\boldsymbol{X}$ 的迹
$\mathrm{rank}(\boldsymbol{X})$	矩阵 $\boldsymbol{X}$ 的秩
$\mathrm{vec}(\boldsymbol{X})$	矩阵 $\boldsymbol{X}$ 的向量化
$\mathrm{Diag}(\boldsymbol{X})$	矩阵 $\boldsymbol{X}$ 对角元素组成的对角矩阵
$\mathrm{diag}(\boldsymbol{X})$	矩阵 $\boldsymbol{X}$ 对角元素组成的向量
$\nabla f(\boldsymbol{X})$	函数 f 在 $\boldsymbol{X}$ 处的梯度
$\partial f(\boldsymbol{X})$	函数 f 在 $\boldsymbol{X}$ 处的次微分
$\mathrm{prox}_f(X)$	关于函数 f 的近端算子
$\mathrm{sgn}(\boldsymbol{X})$	符号函数
$\boldsymbol{I}_m$	m 维单位矩阵
$\mathrm{St}(d,m)$	Stiefel 流形, 即 $\{\boldsymbol{X} \in \mathbb{R}^{d\times m} \mid \boldsymbol{X}^T\boldsymbol{X} = \boldsymbol{I}_m\}$
$\mathcal{M}$	黎曼流形
$\mathrm{T}_{\boldsymbol{X}}\mathcal{M}$	流形 $\mathcal{M}$ 在点 $\boldsymbol{X}$ 处的切空间
$\mathrm{grad} f(\boldsymbol{X})$	函数 f 在 $\boldsymbol{X}$ 处的黎曼梯度
$\mathrm{Hess} f(\boldsymbol{X})$	函数 f 在 $\boldsymbol{X}$ 处的黎曼海森
$\mathrm{Retr}_{\boldsymbol{X}}(\xi)$	点 $\boldsymbol{X}$ 处沿切向量 ξ 的收缩算子

目录

第一部分

基础篇

第 1 章 绪论

黎曼流形稀疏优化是指决策变量带有稀疏性特征且满足黎曼流形约束的一类优化问题. 它将稀疏优化与黎曼流形优化有机结合, 为大数据分析提供了新的理论与算法框架. 作为当前运筹优化领域的研究热点, 黎曼流形稀疏优化不仅在图像处理、无线通信、故障诊断等场景中得到了广泛应用, 更有力推动了数学、计算机、自动化等多学科的交叉发展.

1.1 稀疏优化

2006 年, 美国数学家 David Donoho、Emmanuel Candès 与陶哲轩等人提出了压缩感知 (compressed sensing, CS) 理论, 即若原始信号具有稀疏性的特征, 则可通过少量的观测信息就能够恢复原始信号. 该理论突破了香农定理对信号采样频率的限制, 能够以较少的采样资源与较高的采样速度获得原始信号, 被评为 2007 年度美国十大科技进展之一.

稀疏优化是指具有稀疏性特征的优化问题. 这里, “稀疏性” 一词原指最优解向量的绝大多数元素为零, 从而实现信号的高效压缩、存储与传输. 当最优解为矩阵或高阶张量时, 稀疏性则延伸为低秩性. 合理利用稀疏性不仅能对数据进行充分地压缩, 还可以从海量的数据中提取本质特征, 让复杂优化问题得以简化. 例如, 无线传感器网络定位问题中, 即便节点规模可达到成千上万, 其位置仍分布在低维欧氏空间, 因此可借助低秩性设计快速高效算法.

经过近二十年的发展, 稀疏优化的研究重心已从早期以 ℓ_1 范数为代表的凸松弛, 逐步拓展至结构稀疏、非凸松弛及 ℓ_0 范数原始问题. 在理论层面, 研究者借助变分分析工具, 系统刻画了稀疏优化问题的变分性质, 包括次微分、邻近算子、切锥与法锥等. 在算法层面, 已形成一系列成熟高效的求解框架, 一阶算法如交替最小化、交替方向乘子法、近端交替最小化等, 二阶算法如半光滑牛顿法、子空间牛顿法等. 相关综述与最新进展可参见文献[1-2].

1.2 黎曼流形优化

黎曼几何起源于 19 世纪中叶德国数学家 Bernhard Riemann, 将高斯的曲面内蕴微分几何推广至任意有限维空间, 并在就职演讲《论奠定几何学基础的假设》中首次严格提出了流形 (manifold) 的概念. 黎曼几何不仅在数学领域产生了深远影响, 还为物理学和人工智能提供了强大的理论工具. 例如, Albert Einstein 在广义相对论中, 借助黎曼几何的思想, 成功描述了宇宙时空的弯曲结构. 而在人工智能领域, DeepSeek 通过流形结构重构大模型残差连接, 大幅提升了模型的性能与稳定性.

黎曼流形优化的理论雏形最早由美国数学家 David Luenberger 提出, 旨在通过基本的微分

几何概念探讨黎曼流形的优化策略. 2008 年, Pierre-Antoine Absil 系统阐释了 “收缩” (Retraction) 的概念, 为黎曼流形上的迭代优化提供了有效的数学工具. 近年来, 国际运筹优化领域权威学者 Jong-Shi Pang、Kim-Chuan Toh, 以及人工智能领域泰斗 Michael Jordan 等均高度重视, 并深入开展黎曼流形优化的理论与算法研究. 中国科学院袁亚湘院士团队、厦门大学黄文教授团队对黎曼流形优化进行了深入的研究, 并开发了高性能算法工具包, 为国内相关研究提供了重要支撑。

现有黎曼流形优化算法大致可分为可行方法与不可行方法, 二者的区别在于迭代过程中是否要求迭代点满足黎曼流形约束. 其中, 可行方法通过收缩算子与向量移动策略, 在每一步迭代中均寻找函数值下降的可行点作为下一轮迭代的初始点, 典型代表包括黎曼梯度法、黎曼信赖域法、黎曼牛顿法等. 此类方法本质上是将经典欧几里得空间上的优化方法推广至黎曼流形, 既继承了欧几里得空间上算法的简洁性, 又充分适配了黎曼流形的几何特性. 与可行方法不同, 不可行方法无需迭代点满足黎曼流形约束, 迭代过程中可灵活调整搜索方向与步长, 因此更适用于大规模优化问题. 关于两类方法的详细介绍, 感兴趣的读者可参考文献[3-4].

1.3 黎曼流形稀疏优化

黎曼流形稀疏优化是稀疏优化与黎曼流形优化交叉融合的前沿研究方向, 其目的是在具有非欧几何结构的黎曼流形 (如 Stiefel 流形、Grassmann 流形、低秩矩阵流形等) 限制上, 求解包含非凸、非光滑甚至非连续稀疏项的优化模型. 然而, 黎曼流形稀疏优化问题的求解并不容易, 欧氏空间中稀疏项可通过近端算子高效处理, 但流形上近端算子通常无解析表达式, 且高度依赖内蕴距离与黎曼度量, 经典工具难以直接迁移.

目前常见的做法是采用松弛方法近似求解, 即对稀疏则寻找不同形式的凸函数或非凸函数近似, 然后通过松弛问题进行近似求解. 例如, Chen 等[5] 探讨了带 ℓ_1 范数正则的 Stiefel 流形优化问题, 通过引入新的变量将非光滑项分开, 然后基于增广拉格朗日函数提出了近端交替极小化算法. Chen 等[6] 针对偏最小二乘回归问题, 构造了带 $\ell_{2,1}$ 范数正则的 Stiefel 流形和 Grassmann 流形优化问题, 进而利用 Manopt 进行求解. Xiao 等[7] 研究了带 $\ell_{2,1}$ 范数正则的 Stiefel 流形优化问题, 利用拉格朗日乘子的显示表达式建立了精确罚理论, 设计了非精确的近端梯度方法. 由于子问题都可以显示求解, 算法的效率得到了保障. 此外, Breloy 等[8] 细致分析了不同类型的稀疏主成分分析模型, 考虑了 ℓ_0 范数的非凸松弛近似, 设计了有效的 Majorization-Minimization 算法. Li 等[9] 针对无监督特征提取问题, 研究了带 $\ell_{2,p}$ 范数正则的 Stiefel 流形优化模型, 利用光滑化技巧进行求解. 最近, Zhou 等[10] 充分考虑了 $\ell_{2,1}$ 范数正则和 Stiefel 流形的二阶信息, 基于半光滑牛顿技巧提出了更快速的增广拉格朗日算法, 每一步迭代都自动满足黎曼流形约束, 同时严格证明了全局收敛性和局部超线性收敛率. Huang 等[11] 针对带 ℓ_1 范数正则的 Stiefel 流形优化问题, 提出了一种非精确加速黎曼近端梯度法, 该方法允许

自适应步长, 并证明了其全局收敛性. 此外, Qu 等[12] 探讨带有 $\ell_{2,0}$ 范数和 Stiefel 流形约束的分布式优化问题, 提出了有效的交替子空间牛顿算法.

1.4 本章小结

本书系统整理了作者近三年来在相关研究方向上取得的部分成果, 研究范围不仅涵盖黎曼流形稀疏优化领域, 也包含若干仅聚焦于稀疏优化的内容. 应用场景广泛, 涉及图像特征选择、物联网异常检测、工业故障诊断、红外小目标检测、大模型剪枝等多个领域, 具体安排如图 1.1 所示. 衷心希望本书能够为从事运筹优化、机器学习、数据科学及相关工程领域的科研人员、高校师生及行业从业者提供有益的参考, 为相关领域的发展贡献一份力量.

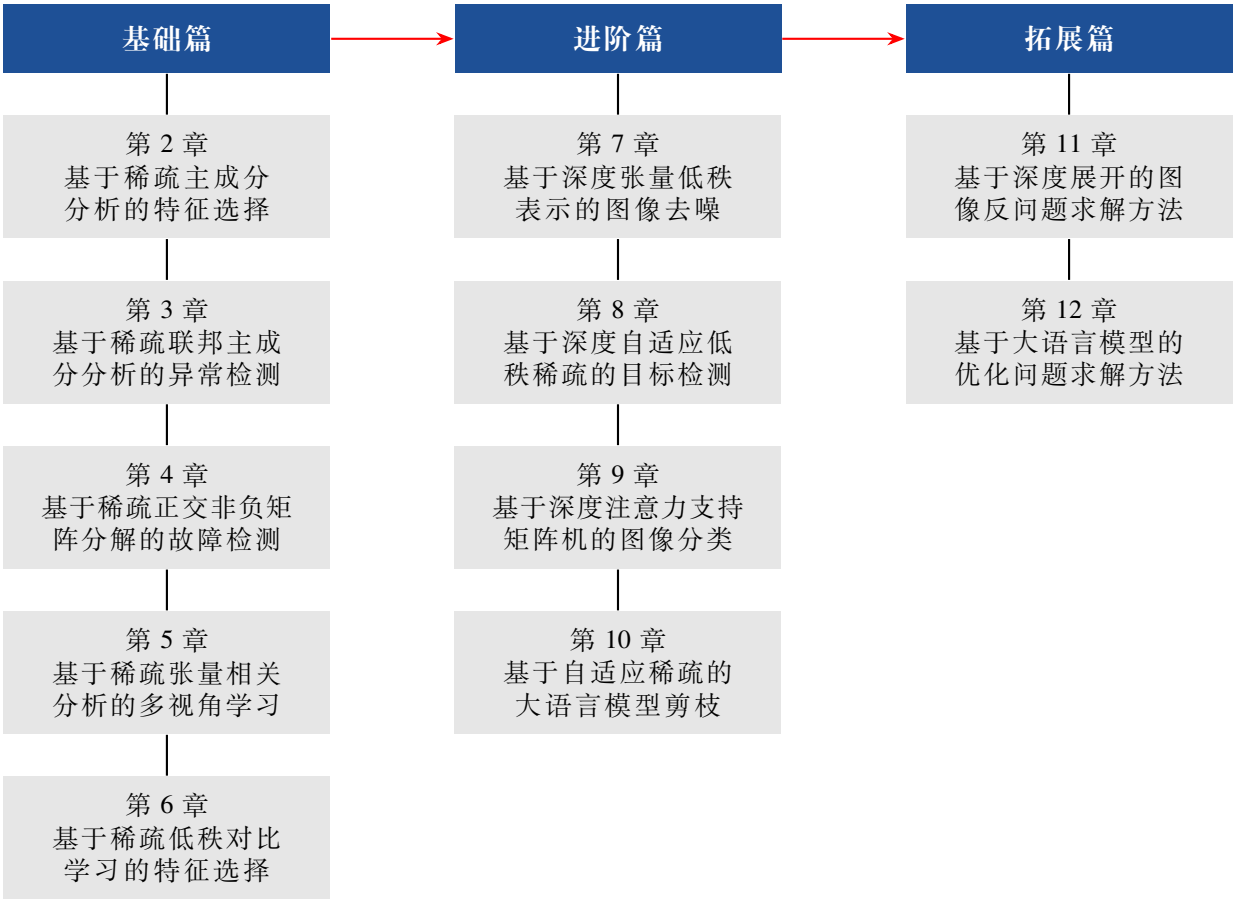


图 1.1: 本书后续章节安排

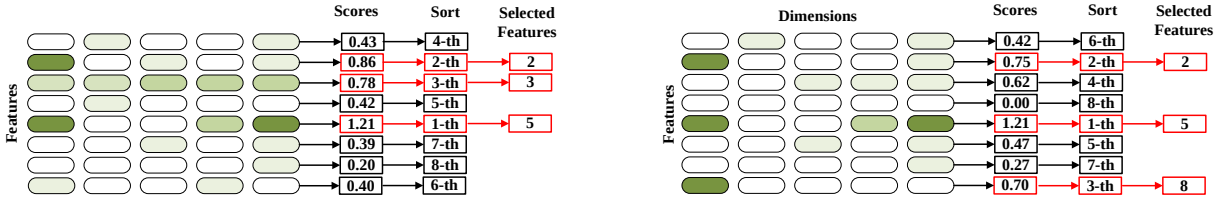
第 2 章 基于稀疏主成分分析的特征选择

主成分分析已发展为一类重要的无监督特征选择方法. 然而, 现有多数基于主成分分析的无监督特征选择方法, 通常仅在变换矩阵上引入单一稀疏正则项, 难以充分刻画复杂数据的内在结构. 为此, 本章提出了一种双稀疏无监督特征选择 (bi-sparse unsupervised feature selection, BSUFS), 其核心思想是在经典主成分分析模型中同时引入 $\ell_{2,p}$ 范数与 ℓ_q 范数, 能够在提取关键特征的同时有效剔除无关噪声. 这里, 参数 $p, q \in [0, 1)$. 因此, BSUFS 不仅构建了统一的双稀疏优化框架, 还可将若干现有相关方法纳入其特例范畴. 为求解由此产生的非凸模型, 本章融合 Stiefel 流形优化与稀疏优化技术, 设计了一种高效的近端交替最小化算法, 并对计算复杂度进行了分析. 大量的数值实验验证了所提 BSUFS 在无监督特征选择中的有效性和鲁棒性.

2.1 引言

高维数据中普遍存在冗余信息与噪声干扰, 给数据分析带来了严峻的挑战. 特征选择技术应运而生, 并成为机器学习与数据挖掘领域的重要研究方向, 其相关前沿进展可详见综述文献[13]. 在特征选择的研究中, 无监督特征选择是一类重要的分支, 其任务是在无标签数据集上实现有效特征筛选. 值得注意的是, 无标签数据在实际应用场景中成本更低、获取更为便捷. 因此, 无监督特征选择受到学术界和工业界的关注, 已广泛应用于图像处理、基因分析、无线通信及芯片设计等领域.

无监督特征选择方法可以根据模型训练和特征选择的结合方式分为三种类型: 过滤式、包裹式和嵌入式. LapScore[14] 是一种典型的过滤式方法. 它首先通过维持数据局部结构的能力来评估特征的重要性, 其次对特征按照重要性排序, 最后根据需求选取前列的特征用于后续的模型训练. 然而, 过滤式无监督特征选择方法由于脱离模型性能进行选择, 可能会选出对特定模型而言不理想的特征. 相反地, 包裹式方法通过使用特定的模型来评估特征子集的优劣. 它把特征选择 “包裹” 在一个机器学习模型中, 选择出最契合模型性能的特征. 与其他两类不同, 嵌入式方法在训练模型的过程中对特征进行一定的处理 (如 ℓ_1 范数正则), 从而实现特征的自动选择. Yang 等[15] 考虑到局部判别信息比全局判别信息更重要, 将构造的局部判别分析得分矩阵和 $\ell_{2,1}$ 范数相结合, 提出了无监督判别特征选择 (unsupervised discriminative feature selection, UDFS). 随后, Liu 等[16] 通过局部线性嵌入算法获得特征权重矩阵, 并使用 ℓ_1 范数描述损失函数, 提出了鲁棒邻域嵌入 (robust neighborhood embedding, RNE). 然而, 上述基于谱分析的方法将图的构造和特征选择的过程分离, 使得它们对冗余特征和噪声特征十分敏感. 为了解决此问题, Nie 等[17] 在低维空间中学习自适应的图并嵌入 $\ell_{2,p}$ $(0 < p \leq 1)$ 范数来表征稀疏性, 进而将局部结构的学习融合到特征选择的过程中, 最终提出了结构化最优图特征选择



(a) 现有 SPCAFS　　**(b)** 所提 BSUFS

图 2.1: 与现有方法的特征选择结果对比

(structured optimal graph feature selection, SOGFS). 事实上, 这些方法各有优缺点, 实际选择时应根据数据集的特性、计算资源的限制以及任务的需求来决定.

主成分分析 (principal component analysis, PCA) 通过构造变换矩阵实现特征提取, 是一类实现简洁且应用极为广泛的嵌入式方法[18]. 但主成分分析提取出来的特征通常不具备直观可解释性, 极大限制了其在高维数据分析中的应用[19]. 针对上述问题, Li 等[9] 将 $\ell_{2,p}$ 范数正则项引入主成分分析变换矩阵的学习过程, 构建了稀疏主成分分析特征选择 (sparse PCA for feature selection, SPCAFS). 该工作中 $p \in (0,1)$, 因此 SPCAFS 可视为基于 $\ell_{2,1}$ 范数的稀疏主成分分析的非凸推广形式. 此外, Nie 等[20] 直接采用 $\ell_{2,0}$ 范数对变换矩阵施加约束以实现特征选择, 提出了特征稀疏约束主成分分析 (feature-sparsity constrained PCA, FSPCA). 值得注意的是, $\ell_{2,0}$ 范数能够直接刻画特征维度的稀疏性, 使得 FSPCA 可高效筛选出数据中最具代表性的关键特征. 近期, Zheng 等[21] 提出了基于半正定投影的稀疏主成分分析 (sparse PCA via positive semidefinite projection, SPCA-PSD), 在聚类任务中展现出优异的计算效率. Gao 等[22] 则将 $\ell_{2,p}$ 范数与模糊弹性网络相结合, 构建了模糊弹性网络主成分分析特征选择 (PCA with fuzzy elastic net for feature selection, FEN-PCAFS). 上述研究中采用的 $\ell_{2,1}$ 范数、$\ell_{2,p}$ 范数与 $\ell_{2,0}$ 范数均能实现变换矩阵的行稀疏约束, 而行稀疏性恰好对应特征维度的选择. 这类结构化稀疏正则项有效提升了主成分分析模型的特征可解释性, 为高维数据结构挖掘提供了重要的技术支撑.

然而, 现有稀疏主成分分析相关工作大多仅对变换矩阵施加单一的行稀疏正则, 忽略变换矩阵内部元素级的稀疏, 难以同时实现特征筛选与噪声抑制. 为此, Zhu 等[23] 构建了融合 $\ell_{2,1}$ 范数与 ℓ_1 范数的双稀疏优化模型, 其中 $\ell_{2,1}$ 范数用于实现全局特征选择, ℓ_1 范数用于剔除冗余噪声分量. 尽管双稀疏正则思想已在其他图像处理领域得到应用, 但现有成果通常将两类稀疏项分别作用于不同变量, 缺乏对同一变换矩阵的联合稀疏约束. 注意到 $p \in [0,1)$ 的 $\ell_{2,p}$ 范数可统一涵盖文献[9] 中的 $\ell_{2,p}$ 范数与文献[20] 中的 $\ell_{2,0}$ 范数, 是更具一般性的结构化稀疏度量. 由此自然引出一个关键的科学问题: 能否在继承 $p \in [0,1)$ 的 $\ell_{2,p}$ 范数与 $q \in [0,1)$ 的 ℓ_q 范数优势的基础上, 构建统一的非凸双稀疏特征学习框架?

基于上述分析, 本章提出了双稀疏无监督特征选择 (bi-sparse unsupervised feature selection, BSUFS). 即在主成分分析框架中创新性引入 $\ell_{2,p}$ 范数与 ℓ_q 范数正则, 其中 p 与 q 的取值范围均为 $[0,1)$. 如图 2.1 所示, 与仅采用 $\ell_{2,p}$ 范数且 $p \in (0,1)$ 的 SPCAFS 相比, BSUFS 因额外引

入元素级 ℓ_q 范数正则, 在特征选择结果上呈现出显著的差异. 从模型通用性来看, 所提 BSUFS 可将 SPCAFS 与 FSPCA 均纳入其特例框架之中, 且相较于文献[23] 中的凸松弛双稀疏模型, 具备更高的灵活性. 当然, 参数 p 与 q 的取值效果依赖于具体数据结构, 后续实验部分将详细阐明参数取值范围从 $(0,1)$ 扩展至 $[0,1)$ 的实际应用价值. 本章的主要贡献为

(1) 通过同时引入 $\ell_{2,p}$ 范数与 ℓ_q 范数正则, 提出了一种新的无监督特征选择方法. 特别, 首次将参数 p 与 q 的取值范围扩展至 $[0,1)$ 区间.

(2) 设计了高效的近端交替最小化 (proximal alternating minimization, PAM) 算法, 其所有子问题或具备解析形式的近端算子, 或可依托 Stiefel 流形优化实现快速求解.

(3) 数值实验评估了所提方法的性能, 分析了 $p, q \in [0,1)$ 的取值对特征选择结果的影响, 并表明在特征选择中 p 起主导作用, 而 q 起补充作用, 二者协同不可或缺.

2.2 数学模型

2.2.1 预备知识

给定数据 $\boldsymbol{X} = (\boldsymbol{x}_1, \boldsymbol{x}_2, \ldots, \boldsymbol{x}_n) \in \mathbb{R}^{d\times n}$, 其中 $\boldsymbol{x}_i \in \mathbb{R}^d$ 为第 i 个列向量. 记变换矩阵 $\boldsymbol{W} = (\boldsymbol{w}_1, \boldsymbol{w}_2, \ldots, \boldsymbol{w}_m) \in \mathbb{R}^{d\times m}$ $(m < n)$, 其中 $\boldsymbol{w}_i \in \mathbb{R}^d$, 且满足 $\|\boldsymbol{w}_i\| = 1$. 当 $i \neq j$ 时, $\boldsymbol{w}_i^T \boldsymbol{w}_j = 0$, 即 $\boldsymbol{W}$ 的列向量两两正交. 假设数据已中心化, 主成分分析可表示为

$$\begin{aligned} \min_{\boldsymbol{W}\in\mathbb{R}^{d\times m}} \quad & -\operatorname{tr}(\boldsymbol{W}^T \boldsymbol{X}\boldsymbol{X}^T \boldsymbol{W}) \\ \text{s.t.} \quad & \boldsymbol{W}^T \boldsymbol{W} = \boldsymbol{I}_m. \end{aligned} \tag{2.1}$$

对于一般的非中心化数据情形, 主成分分析可进一步表示为

$$\begin{aligned} \min_{\boldsymbol{W}\in\mathbb{R}^{d\times m}} \quad & -\operatorname{tr}(\boldsymbol{W}^T \boldsymbol{S}\boldsymbol{W}) \\ \text{s.t.} \quad & \boldsymbol{W}^T \boldsymbol{W} = \boldsymbol{I}_m, \end{aligned} \tag{2.2}$$

其中 $\boldsymbol{S} = \boldsymbol{X}\boldsymbol{H}\boldsymbol{X}^T$ 为协方差矩阵, $\boldsymbol{H} = \boldsymbol{I}_n - \frac{1}{n}\mathbf{1}\mathbf{1}^T$ 为中心化矩阵, $\mathbf{1} \in \mathbb{R}^n$ 为全为 1 的向量.

不失一般性, 记 $(\boldsymbol{w}^i)^T$ 为向量 $\boldsymbol{w}^i$ 的转置, 其中 $\boldsymbol{w}^i$ 为变换矩阵 $\boldsymbol{W}$ 的第 i 行. 在无监督特征选择中, $(\boldsymbol{w}^i)^T$ 可视为与输入数据 $\boldsymbol{X}$ 的第 i 个特征对应的权重向量, 用于衡量该特征的贡献度. 具体地, 矩阵 $\boldsymbol{W}$ 可表示为

$$\boldsymbol{W} = (\boldsymbol{w}_1, \boldsymbol{w}_2, \ldots, \boldsymbol{w}_m) = \begin{pmatrix} \boldsymbol{w}^1 \\ \boldsymbol{w}^2 \\ \vdots \\ \boldsymbol{w}^d \end{pmatrix} \in \mathbb{R}^{d\times m}. \tag{2.3}$$

将样本 $\boldsymbol{x}_i$ 通过变换矩阵 $\boldsymbol{W}$ 进行线性映射, 得到变换后的向量为

$$\boldsymbol{z}_i = \boldsymbol{W}^T\boldsymbol{x}_i = ((\boldsymbol{w}^1)^T, (\boldsymbol{w}^2)^T, \ldots, (\boldsymbol{w}^d)^T)\begin{pmatrix} x_{1i} \\ x_{2i} \\ \vdots \\ x_{di} \end{pmatrix}. \tag{2.4}$$

由式 (2.4) 可知, $\|\boldsymbol{w}^i\|$ 的大小可直接用于衡量 $\boldsymbol{X}$ 的第 i 个特征在降维过程中的重要性, $\|\boldsymbol{w}^i\|$ 越大, 对应特征的贡献度越高.

2.2.2 稀疏主成分分析

近年来, 非凸优化理论与算法得到了快速发展, 相较于传统凸优化方法, 其在解的灵活性、模型拟合精度等方面具有显著的优势[24]. 基于此, Li 等[9] 提出了 SPCAFS, 具体形式为

$$\begin{aligned} \min_{\boldsymbol{W}\in\mathbb{R}^{d\times m}} \quad & -\mathrm{tr}(\boldsymbol{W}^T\boldsymbol{S}\boldsymbol{W}) + \lambda\|\boldsymbol{W}\|_{2,p}^p \\ \text{s.t.} \quad & \boldsymbol{W}^T\boldsymbol{W} = \boldsymbol{I}_m, \end{aligned} \tag{2.5}$$

其中 $\lambda \geq 0$ 为正则参数, $p \in (0,1)$. SPCAFS 通过在式 (2.2) 的目标函数中引入 $\ell_{2,p}$ 范数正则项, 可诱导变换矩阵 $\boldsymbol{W}$ 产生行稀疏性, 进而实现有效的特征选择. 数值结果还表明, 当 $p = 1/2$ 时, 该模型的特征选择性能优于基于凸松弛的稀疏主成分分析方法.

FSPCA[20] 是另一种广泛应用的无监督特征选择方法, 其数学模型为

$$\begin{aligned} \min_{\boldsymbol{W}\in\mathbb{R}^{d\times m}} \quad & -\mathrm{tr}(\boldsymbol{W}^T\boldsymbol{S}\boldsymbol{W}) \\ \text{s.t.} \quad & \|\boldsymbol{W}\|_{2,0} \leq s,\ \boldsymbol{W}^T\boldsymbol{W} = \boldsymbol{I}_m, \end{aligned} \tag{2.6}$$

其中 $s > 0$ 为预设的稀疏度水平, $\|\boldsymbol{W}\|_{2,0}$ 表示 $\boldsymbol{W}$ 的行稀疏度 (即非零行的数量). 结合图 2.1 与式 (2.4) 可知, 稀疏度水平 s 对应于最终选择的特征数量.

如前文所述, 现有稀疏主成分分析方法大多仅引入单一结构的稀疏正则项. 文献[23] 通过在变换矩阵上同时引入 $\ell_{2,1}$ 范数与 ℓ_1 范数正则项, 提出了双稀疏无监督特征选择模型, 即

$$\begin{aligned} \min_{\boldsymbol{W}\in\mathbb{R}^{d\times m}} \quad & -\mathrm{tr}(\boldsymbol{W}^T\boldsymbol{S}\boldsymbol{W}) + \lambda_1\|\boldsymbol{W}\|_{2,1} + \lambda_2\|\boldsymbol{W}\|_1 \\ \text{s.t.} \quad & \boldsymbol{W}^T\boldsymbol{W} = \boldsymbol{I}_m, \end{aligned} \tag{2.7}$$

其中 $\lambda_1, \lambda_2 \geq 0$ 为正则参数, 分别用于调节变换矩阵 $\boldsymbol{W}$ 的行稀疏与元素稀疏结构.

2.2.3 构建模型

为兼顾非凸优化的灵活性与双稀疏正则的特征筛选能力, 本章构建了如下 BSUFS 模型

$$\begin{aligned} \min_{\boldsymbol{W}\in\mathbb{R}^{d\times m}} \quad & -\mathrm{tr}(\boldsymbol{W}^T\boldsymbol{S}\boldsymbol{W}) + \lambda_1\|\boldsymbol{W}\|_{2,p}^p + \lambda_2\|\boldsymbol{W}\|_q^q \\ \text{s.t.} \quad & \boldsymbol{W}^T\boldsymbol{W} = \boldsymbol{I}_m, \end{aligned} \tag{2.8}$$

其中 $p, q \in [0, 1)$. 与现有基于主成分分析的无监督特征选择方法相比, 当 $\lambda_2 = 0$ 时, 式 (2.8) 可退化为式 (2.5) 与式 (2.6) 的拉格朗日形式. 当 p 与 q 均趋于 1 时, 即 $\ell_{2,p}$ 范数退化为 $\ell_{2,1}$ 范数, ℓ_q 范数退化为 ℓ_1 范数, 式 (2.8) 等价于式 (2.7).

综上, 所提 BSUFS 通过在 $[0, 1)$ 范围内选择不同的 p 与 q, 可产生更具灵活性的稀疏解, 能够满足不同类型数据的特征选择需求, 进而提升无监督特征选择的性能.

2.3 优化算法

受 Stiefel 流形约束 $\boldsymbol{W}^T\boldsymbol{W} = \boldsymbol{I}_m$、非光滑正则 $\ell_{2,p}$ 范数与 ℓ_q 范数的影响, 式 (2.8) 属于非凸、非光滑的复合优化问题. 本节给出一种基于近端交替最小化技术的优化算法.

首先, 引入辅助变量 $\boldsymbol{V}, \boldsymbol{U}$ 满足等价约束 $\boldsymbol{W} = \boldsymbol{V}$ 与 $\boldsymbol{W} = \boldsymbol{U}$, 将式 (2.8) 等价改写为

$$\begin{aligned} \min_{\boldsymbol{W},\boldsymbol{U},\boldsymbol{V}\in\mathbb{R}^{d\times m}} \quad & -\mathrm{tr}(\boldsymbol{W}^T\boldsymbol{S}\boldsymbol{W}) + \lambda_1\|\boldsymbol{V}\|_{2,p}^p + \lambda_2\|\boldsymbol{U}\|_q^q \\ \text{s.t.} \quad & \boldsymbol{W}^T\boldsymbol{W} = \boldsymbol{I}_m,\ \boldsymbol{W} = \boldsymbol{V},\ \boldsymbol{W} = \boldsymbol{U}. \end{aligned} \tag{2.9}$$

定义 Stiefel 流形约束的指示函数

$$\Phi(\boldsymbol{W}) = \begin{cases} 0, & \boldsymbol{W}^T\boldsymbol{W} = \boldsymbol{I}_m, \\ +\infty, & \text{其他}, \end{cases} \tag{2.10}$$

结合二次罚方法, 将式 (2.9) 转化为无约束优化问题

$$\begin{aligned} \min_{\boldsymbol{W},\boldsymbol{U},\boldsymbol{V}\in\mathbb{R}^{d\times m}} \quad & -\mathrm{tr}(\boldsymbol{W}^T\boldsymbol{S}\boldsymbol{W}) + \lambda_1\|\boldsymbol{V}\|_{2,p}^p + \lambda_2\|\boldsymbol{U}\|_q^q \\ & + \frac{\beta_1}{2}\|\boldsymbol{W} - \boldsymbol{U}\|_F^2 + \frac{\beta_2}{2}\|\boldsymbol{W} - \boldsymbol{V}\|_F^2 + \Phi(\boldsymbol{W}), \end{aligned} \tag{2.11}$$

其中 $\beta_1, \beta_2 > 0$ 为等式约束对应的罚参数. 便于描述, 记式 (2.11) 的目标函数为 $f(\boldsymbol{W}, \boldsymbol{U}, \boldsymbol{V})$. 基于近端交替最小化思想, 可按如下方式进行变量更新

$$\boldsymbol{W}^{k+1} \in \underset{\boldsymbol{W}\in\mathbb{R}^{d\times m}}{\mathrm{argmin}}\ f(\boldsymbol{W}, \boldsymbol{U}^k, \boldsymbol{V}^k) + \frac{\tau_1}{2}\|\boldsymbol{W} - \boldsymbol{W}^k\|_F^2, \tag{2.12}$$

$$\boldsymbol{U}^{k+1} \in \underset{\boldsymbol{U}\in\mathbb{R}^{d\times m}}{\mathrm{argmin}}\ f(\boldsymbol{W}^{k+1}, \boldsymbol{U}, \boldsymbol{V}^k) + \frac{\tau_2}{2}\|\boldsymbol{U} - \boldsymbol{U}^k\|_F^2, \tag{2.13}$$

$$\boldsymbol{V}^{k+1} \in \underset{\boldsymbol{V}\in\mathbb{R}^{d\times m}}{\mathrm{argmin}}\ f(\boldsymbol{W}^{k+1}, \boldsymbol{U}^{k+1}, \boldsymbol{V}) + \frac{\tau_3}{2}\|\boldsymbol{V} - \boldsymbol{V}^k\|_F^2, \tag{2.14}$$

其中 $\tau_1, \tau_2, \tau_3 > 0$ 为近端参数, k 为迭代次数. 整体迭代框架如算法 1 所示, 其中停止准则为

算法 1 求解式 (2.8) 的近端交替最小化算法

输入: 数据 $\boldsymbol{X} \in \mathbb{R}^{d\times n}$, 参数 $p,\ q,\ \lambda_1,\ \lambda_2,\ \beta_1,\ \beta_2,\ \tau_1,\ \tau_2,\ \tau_3$

初始化: 令 $k = 0$, 取 $(\boldsymbol{W}^0, \boldsymbol{U}^0, \boldsymbol{V}^0)$

当 未收敛 **时**

1: 通过式 (2.12) 更新 $\boldsymbol{W}^{k+1}$
2: 通过式 (2.13) 更新 $\boldsymbol{U}^{k+1}$
3: 通过式 (2.14) 更新 $\boldsymbol{V}^{k+1}$
4: 通过式 (2.15) 检验收敛性

结束循环

输出: $\boldsymbol{V}$

$$\frac{|f^{k+1} - f^k|}{\max\{|f^k|, 1\}} < 10^{-4} \text{ 或 } k > 500. \tag{2.15}$$

后续小节依次推导 $\boldsymbol{W}, \boldsymbol{U}, \boldsymbol{V}$ 的迭代更新规则. 在特征选择任务中, 建议优先更新 $\boldsymbol{U}^{k+1}$ 以抑制数据噪声干扰, 再更新 $\boldsymbol{V}^{k+1}$ 以选择更具判别性的特征.

2.3.1 更新 $\boldsymbol{W}$

固定变量 $\boldsymbol{U}$ 和 $\boldsymbol{V}$, 式 (2.12) 可简化为

$$\begin{aligned}\min_{\boldsymbol{W}^T\boldsymbol{W}=\boldsymbol{I}_m} g(\boldsymbol{W}) = -\mathrm{tr}(\boldsymbol{W}^T\boldsymbol{S}\boldsymbol{W}) + \frac{\beta_1}{2}\|\boldsymbol{W} - \boldsymbol{U}^k\|_F^2 \\ + \frac{\beta_2}{2}\|\boldsymbol{W} - \boldsymbol{V}^k\|_F^2 + \frac{\tau_1}{2}\|\boldsymbol{W} - \boldsymbol{W}^k\|_F^2.\end{aligned} \tag{2.16}$$

关于目标函数 $g(\boldsymbol{W})$, 其欧氏梯度为

$$\nabla g(\boldsymbol{W}) = -2\boldsymbol{S}\boldsymbol{W} + \beta_1(\boldsymbol{W} - \boldsymbol{U}^k) + \beta_2(\boldsymbol{W} - \boldsymbol{V}^k) + \tau_1(\boldsymbol{W} - \boldsymbol{W}^k), \tag{2.17}$$

欧氏海森为

$$\nabla^2 g(\boldsymbol{W}) = -2\boldsymbol{I}_m \otimes \boldsymbol{S} + (\beta_1 + \beta_2 + \tau_1)\boldsymbol{I}_{dm}, \tag{2.18}$$

其中 $\otimes$ 表示克罗内克积 (Kronecker product).

记 Stiefel 流形集合为 $\mathrm{St}(d, m) = \{\boldsymbol{W} \in \mathbb{R}^{d\times m} \mid \boldsymbol{W}^T\boldsymbol{W} = \boldsymbol{I}_m\}$, 则式 (2.16) 本质为一个 Stiefel 流形优化问题, 可重写为

$$\min_{\boldsymbol{W}\in\mathrm{St}(d,m)} g(\boldsymbol{W}). \tag{2.19}$$

本节采用信赖域黎曼流形优化算法[4] 求解该流形优化模型. 黎曼信赖域迭代需要两个基本要素: 搜索方向与信赖域比率. 搜索方向依赖于式 (2.19) 中目标函数 $g(\boldsymbol{W})$ 的黎曼梯度与黎曼海森. 具体而言, 黎曼梯度可通过将欧氏梯度投影到 Stiefel 流形的切空间来获得, 即

$$\begin{aligned}\operatorname{grad} g(\boldsymbol{W}) &= \mathcal{P}_{\boldsymbol{W}}(\nabla g(\boldsymbol{W})) \\ &= \nabla g(\boldsymbol{W}) - \boldsymbol{W}\operatorname{sym}(\boldsymbol{W}^T\nabla g(\boldsymbol{W})),\end{aligned} \tag{2.20}$$

其中 $\operatorname{sym}(\boldsymbol{X}) = (\boldsymbol{X}+\boldsymbol{X}^T)/2$ 表示提取方阵 $\boldsymbol{X}$ 的对称部分. 同理, 黎曼海森可通过将欧氏海森投影到 Stiefel 流形切空间得到, 即

$$\begin{aligned}\operatorname{Hess} g(\boldsymbol{W})[\boldsymbol{M}] = \mathcal{P}_{\boldsymbol{W}}(&\nabla^2 g(\boldsymbol{W})[\boldsymbol{M}] - \boldsymbol{M}\operatorname{sym}(\boldsymbol{W}^T\nabla g(\boldsymbol{W})) \\ &- \boldsymbol{W}\operatorname{sym}(\boldsymbol{M}^T\nabla g(\boldsymbol{W})) - \boldsymbol{W}\operatorname{sym}(\boldsymbol{W}^T\nabla^2 g(\boldsymbol{W})[\boldsymbol{M}])),\end{aligned} \tag{2.21}$$

其中 $\nabla^2 g(\boldsymbol{W})[\boldsymbol{M}]$ 为欧氏海森与切向量的乘积. 于是, 信赖域算法的搜索方向可通过求解如下问题得到

$$\begin{aligned}\min_{\boldsymbol{M}\in \mathrm{T}_{\boldsymbol{W}}\mathrm{St}(d,m)} \ & m_{\boldsymbol{W}}(\boldsymbol{M}) = g(\boldsymbol{W}) + \langle \operatorname{grad} g(\boldsymbol{W}), \boldsymbol{M}\rangle_{\boldsymbol{W}} + \frac{1}{2}\langle \operatorname{Hess} g(\boldsymbol{W})[\boldsymbol{M}], \boldsymbol{M}\rangle_{\boldsymbol{W}} \\ \text{s.t.} \quad & \langle \boldsymbol{M}, \boldsymbol{M}\rangle_{\boldsymbol{W}} \le \Delta^2,\end{aligned} \tag{2.22}$$

其中 Δ 为当前信赖域半径, $\mathrm{T}_{\boldsymbol{W}}\mathrm{St}(d,m)$ 表示流形在点 $\boldsymbol{W}$ 处的切空间.

最后, 信赖域比率定义为

$$\rho = \frac{g(\boldsymbol{W}) - g(\operatorname{Retr}_{\boldsymbol{W}}(\boldsymbol{M}))}{m_{\boldsymbol{W}}(0) - m_{\boldsymbol{W}}(\boldsymbol{M})}, \tag{2.23}$$

其中 $\operatorname{Retr}_{\boldsymbol{W}}(\boldsymbol{M})$ 表示将 $\boldsymbol{M}$ 收缩到 Stiefel 流形上的算子. 完整的黎曼信赖域迭代流程如算法 2 所示, 其中停止准则为

$$\operatorname{grad} g(\boldsymbol{W}_{i+1}^k) < 10^{-6} \text{ 或 } i > 100. \tag{2.24}$$

2.3.2 更新 $\boldsymbol{U}$

在更新 $\boldsymbol{W}$ 之后, 式 (2.13) 可通过求解下式得到

$$\min_{\boldsymbol{U}\in\mathbb{R}^{d\times m}} \ \lambda_2\|\boldsymbol{U}\|_q^q + \frac{\beta_1}{2}\|\boldsymbol{W}^{k+1} - \boldsymbol{U}\|_F^2 + \frac{\tau_2}{2}\|\boldsymbol{U} - \boldsymbol{U}^k\|_F^2. \tag{2.25}$$

合并式 (2.25) 中的二次项并整理, 得到

$$\min_{\boldsymbol{U}\in\mathbb{R}^{d\times m}} \ \lambda_2\|\boldsymbol{U}\|_q^q + \frac{\beta_1+\tau_2}{2}\|\boldsymbol{U} - \boldsymbol{Y}\|_F^2, \tag{2.26}$$

其中

$$\boldsymbol{Y} = \frac{\beta_1}{\beta_1+\tau_2}\boldsymbol{W}^{k+1} + \frac{\tau_2}{\beta_1+\tau_2}\boldsymbol{U}^k. \tag{2.27}$$

由于 ℓ_q 范数具备元素可分性, 式 (2.26) 可分解为一系列关于元素 u_{ij} 的子问题, 即

算法 2 求解式 (2.12) 的信赖域黎曼流形优化算法

输入: 数据 $\boldsymbol{S} \in \mathbb{R}^{d\times d}$, $\boldsymbol{U}^k \in \mathbb{R}^{d\times m}$, $\boldsymbol{V}^k \in \mathbb{R}^{d\times m}$, 参数 β_1, β_2, τ_1, ε, 以及 $\Delta' > 0$, $\rho' \in [0, \frac{1}{4})$

初始化: 令 $i = 0$, 取 $\boldsymbol{W}_i^k \in \mathrm{St}(d, m)$, $\Delta_i \in (0, \Delta')$

当 未收敛 **时**

通过式 (2.22) 得到 $\boldsymbol{M}_i$
由式 (2.23) 计算 ρ_i
if $\rho_i < \frac{1}{4}$ **then**
    $\Delta_{i+1} = \frac{1}{4}\Delta_i$
else if $\rho_i > \frac{3}{4}$ 且 $\|\boldsymbol{M}_i\| = \Delta_i$ **then**
    $\Delta_{i+1} = \min(2\Delta_i, \Delta')$
else
    $\Delta_{i+1} = \Delta_i$
end if
if $\rho_i > \rho'$ **then**
    $\boldsymbol{W}_{i+1}^k = \mathrm{Retr}_{\boldsymbol{W}}(\boldsymbol{M}_i)$
else
    $\boldsymbol{W}_{i+1}^k = \boldsymbol{W}_i^k$
end if
根据式 (2.24) 检验收敛性

结束循环

输出: $\boldsymbol{W}^k$

$$\min_{u_{ij}\in\mathbb{R}} \lambda_2|u_{ij}|^q + \frac{\beta_1+\tau_2}{2}(u_{ij}-y_{ij})^2. \tag{2.28}$$

引理 2.1 *考虑 ℓ_q 范数对应的近端算子*

$$\begin{aligned}\mathrm{prox}_{\lambda|\cdot|^q}(a) &= \operatorname*{argmin}_{x\in\mathbb{R}} \lambda|x|^q + \frac{1}{2}(x-a)^2 \\ &= \begin{cases} \{0\}, & |a| < \kappa(\lambda, q), \\ \{0, \mathrm{sgn}(a)c(\lambda,q)\}, & |a| = \kappa(\lambda,q), \\ \left\{\mathrm{sgn}(a)\varpi_q(|a|)\right\}, & |a| > \kappa(\lambda,q), \end{cases}\end{aligned} \tag{2.29}$$

其中参数满足

$$\begin{aligned} c(\lambda,p) &= (2\lambda(1-q))^{\frac{1}{2-q}} > 0, \\ \kappa(\lambda,q) &= (2-q)\lambda^{\frac{1}{2-q}}(2(1-q))^{\frac{q+1}{q-2}}, \\ \varpi_q(a) &\in \{x \mid x - a + \lambda q\,\mathrm{sgn}(x)x^{q-1} = 0,\ x > 0\}. \end{aligned} \tag{2.30}$$

更多细节可参见文献[25].

依据引理 2.1, 式 (2.28) 的解可由下式直接给出

$$u_{ij} \in \operatorname{prox}_{\frac{\lambda_2}{\beta_1+\tau_2}|\cdot|^q}(y_{ij}). \tag{2.31}$$

根据文献[26], 当 $q = 0$ 时, 式 (2.29) 对应于硬阈值算子. 根据文献[27], 当 $q = 1/2$ 与 $q = 2/3$ 时, 式 (2.29) 也存在解析表达. 对于一般 $q \in (0, 1)$, 可采用文献[25] 中提出的快速迭代策略求解.

2.3.3 更新 V

当 $\boldsymbol{W}$ 与 $\boldsymbol{U}$ 迭代更新后, 式 (2.14) 可通过下式计算

$$\min_{\boldsymbol{V}\in\mathbb{R}^{d\times m}} \lambda_1\|\boldsymbol{V}\|_{2,p}^p + \frac{\beta_2}{2}\|\boldsymbol{W}^{k+1} - \boldsymbol{V}\|_F^2 + \frac{\tau_3}{2}\|\boldsymbol{V} - \boldsymbol{V}^k\|_F^2. \tag{2.32}$$

同理合并二次惩罚项与近端项, 整理得

$$\min_{\boldsymbol{V}\in\mathbb{R}^{d\times m}} \lambda_1\|\boldsymbol{V}\|_{2,p}^p + \frac{\beta_2+\tau_3}{2}\|\boldsymbol{V} - \boldsymbol{Z}\|_F^2, \tag{2.33}$$

其中

$$\boldsymbol{Z} = \frac{\beta_2}{\beta_2+\tau_3}\boldsymbol{W}^{k+1} + \frac{\tau_3}{\beta_2+\tau_3}\boldsymbol{V}^k. \tag{2.34}$$

注意到 $\ell_{2,p}$ 范数具备行可分性, 于是该问题可分解为一系列关于行 $\boldsymbol{v}^i$ 的子问题, 即

$$\min_{\boldsymbol{v}^i\in\mathbb{R}^m} \lambda_1\|\boldsymbol{v}^i\|^p + \frac{\beta_2+\tau_3}{2}\|\boldsymbol{v}^i - \boldsymbol{z}^i\|^2, \tag{2.35}$$

其中 $i \in \{1, 2, \ldots, d\}$.

引理 2.2 *考虑 $\ell_{2,p}$ 范数对应的近端算子*

$$\begin{aligned}\operatorname{prox}_{\lambda\|\cdot\|^p}(\boldsymbol{a}) &= \operatorname*{argmin}_{\boldsymbol{x}\in\mathbb{R}^m} \lambda\|\boldsymbol{x}\|^p + \frac{1}{2}\|\boldsymbol{x} - \boldsymbol{a}\|^2 \\ &= \begin{cases} \operatorname{prox}_{\lambda|\cdot|^p}(\|\boldsymbol{a}\|)\cdot\frac{\boldsymbol{a}}{\|\boldsymbol{a}\|}, & \boldsymbol{a} \neq \boldsymbol{0}, \\ \{\boldsymbol{0}\}, & \boldsymbol{a} = \boldsymbol{0}. \end{cases}\end{aligned} \tag{2.36}$$

特别地, 约定当 $\boldsymbol{x} \neq \boldsymbol{0}$ 时 $\|\boldsymbol{x}\|^0 = 1$, 当 $\boldsymbol{x} = \boldsymbol{0}$ 时 $\|\boldsymbol{x}\|^0 = 0$.

结合引理 2.2, 式 (2.35) 的解为

$$\boldsymbol{v}^i \in \begin{cases} \operatorname{prox}_{\frac{\lambda_1}{\beta_2+\tau_3}|\cdot|^p}(\|\boldsymbol{z}^i\|)\cdot\frac{\boldsymbol{z}^i}{\|\boldsymbol{z}^i\|}, & \boldsymbol{z}^i \neq \boldsymbol{0}, \\ \{0\}, & \boldsymbol{z}^i = \boldsymbol{0}. \end{cases} \tag{2.37}$$

值得注意的是, 一般近端交替最小化算法的收敛性可依托 Kurdyka-Łojasiewicz (K-L) 性质结合充分下降引理开展论证[28]. 然而, 本章流形约束变量 $\boldsymbol{W}^{k+1}$ 的更新依赖于算法 2. 文献[4] 已严格证明, 黎曼信赖域算法可收敛至流形约束下的临界点, 即满足 $\operatorname{grad} g(\boldsymbol{W}) = 0$ 的一阶稳

表 2.1: 所选数据集信息

类型	数据集	特征数	样本数	类别数
仿真数据	Dartboard1	9	1,000	4
	Diamond9	9	3,000	9
真实数据	COIL20	1,024	1,440	20
	ISOLET	617	1,560	26
	USPS	256	1,000	10
	UMIST	644	575	20
	GLIOMA	4,434	50	4
	PIE	1,024	1,166	53
	LUNG	325	73	7
	MSTAR	1,024	2,425	10

定点. 但若要建立算法 1 的收敛性, 还需进一步证明算法 2 收敛到全局最优解, 这是一个更强的结论.

2.3.4 计算复杂度

初始化算法 1 时, 计算 $\boldsymbol{H}$ 与 $\boldsymbol{S}$ 分别需要 $O(n^2)$ 与 $O(dn^2)$, 因此初始化的计算复杂度为 $O(dn^2)$. 使用算法 2 更新 $\boldsymbol{W}$ 时, 计算复杂度由求解式 (2.22) 与式 (2.23) 决定, 计算复杂度分别为 $O(d^2m+dm^2)$ 和 $O(dm^2)$. 综上, 算法 2 每次迭代的计算复杂度为 $O(d^2m+dm^2)$. 更新 $\boldsymbol{U}$ 与 $\boldsymbol{V}$ 时, 计算复杂度仅取决于近端算子, 其计算复杂度为 $O(dm)$. 收敛性检查基于损失函数 f, 其计算复杂度为 $O(d^2m)$. 因此, 算法 1 每次迭代的总体计算复杂度为 $O((K+1)d^2m+Kdm^2+dm)$, 其中 K 为算法 2 的迭代次数.

2.4 数值实验

为验证所提 BSUFS 的有效性, 本节将其与多种主流无监督特征选择方法开展对比实验, 包含 LapScore[14]、SOGFS[17]、RNE[16]、UDFS[15]、SPCAFS[9]、FSPCA[29]、SPCA-PSD[21] 及 FEN-PCAFS[22]. 其中, LapScore、SOGFS、RNE 与 UDFS 依托开源工具箱 AutoUFSTool[1] 完成复现, SPCAFS[2]、FSPCA[3]、SPCA-PSD[4] 与 FEN-PCAFS[5] 均采用作者公开的 GitHub 原始代码, 保证对比实验的公平性. 此外, 所提方法开源代码见链接 https://github.com/xianchaoxiu/BSUFS.

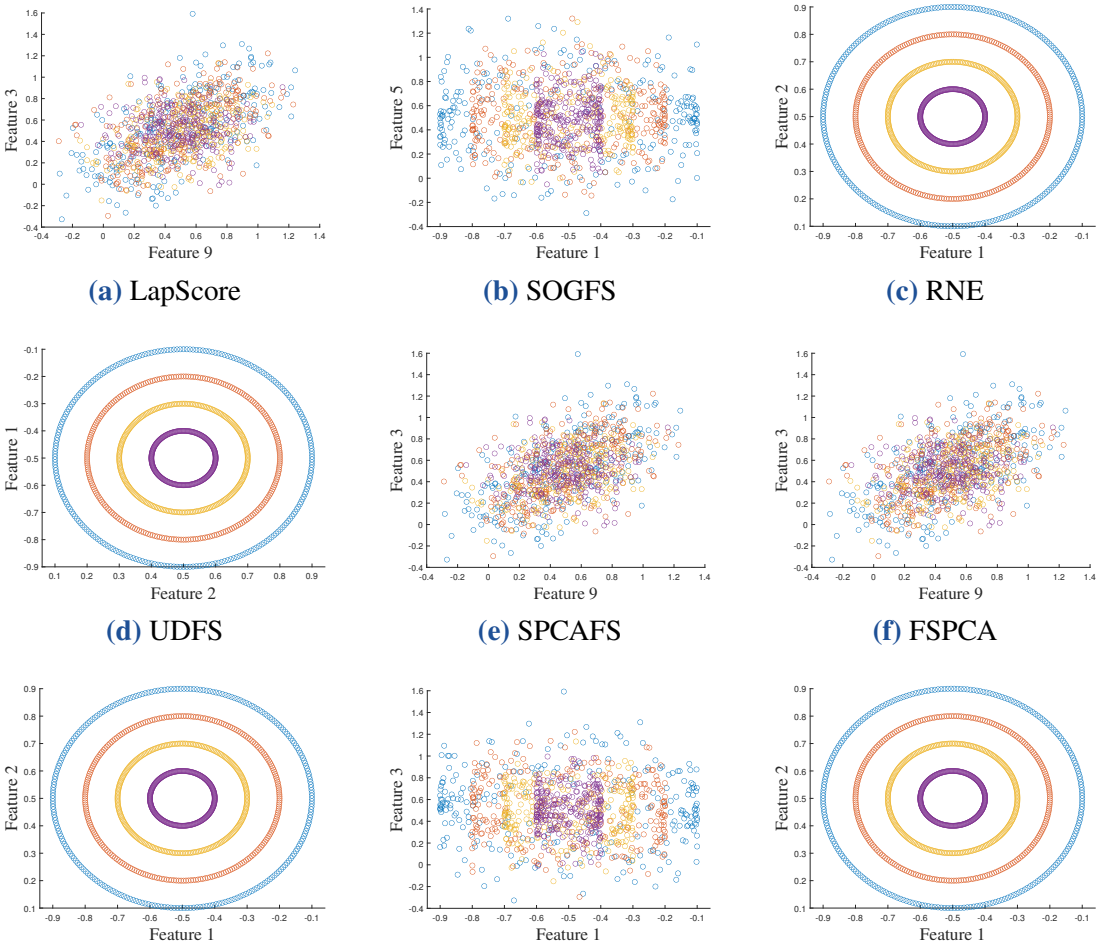
[1]https://github.com/farhadabedinzadeh/AutoUFSTool

[2]https://github.com/quiter2005/algorithm

[3]https://github.com/tianlai09/FSPCA

[4]https://github.com/zjj20212035/SPCA-PSD

[5]https://github.com/gaoyl-group/FEN-PCAFS



(a) LapScore **(b)** SOGFS **(c)** RNE

(d) UDFS **(e)** SPCAFS **(f)** FSPCA

(g) SPCA-PSD **(h)** FEN-PCAFS **(i)** BSUFS

图 2.2: Dartboard1 数据集可视化结果

2.4.1 实验设置

(1) 数据集

本节实验采用两个仿真数据集与八个真实数据集, 以验证所提 BSUFS 的性能. 两个仿真数据集[6]均对前 2 个特征赋予特定分布, 而其余 7 个特征用高斯噪声填充. 八个真实数据集覆盖多个领域, 例如字母识别数据集 ISOLET[7]、深度学习数据集 MSTAR[8]、生物信息数据集 GLIOMA[7] 与 LUNG[7], 以及图像处理数据集 COIL20[7]、USPS[7]、PIE[9]、UMIST[10]. 所有数据集的特征数、样本数、类别数如表 2.1 所示.

(2) 参数设置

对于 LapScore、SOGFS 与 RNE, 统一将 k 近邻 (k-nearest neighbor, k-NN) 的取值设为 5. 对

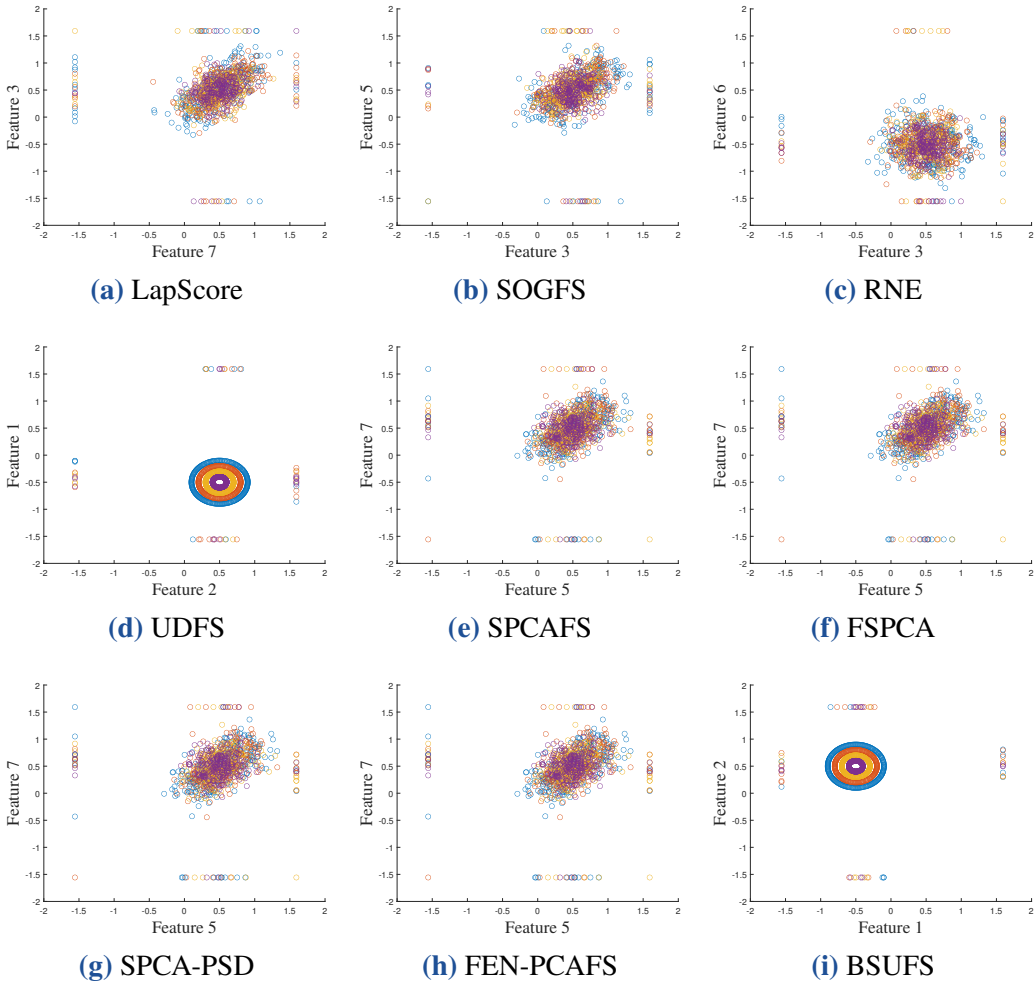
[6]https://github.com/milaan9/Clustering-Datasets

[7]https://jundongl.github.io/scikit-feature/datasets.html

[8]https://github.com/zjj20212035/SPCA-PSD

[9]https://data.nvision2.eecs.yorku.ca/PIE_dataset/

[10]https://github.com/saining/PPSL/blob/master/Platform/Data/UMIST/UMIST.mat



(a) LapScore **(b)** SOGFS **(c)** RNE

(d) UDFS **(e)** SPCAFS **(f)** FSPCA

(g) SPCA-PSD **(h)** FEN-PCAFS **(i)** BSUFS

图 2.3: Dartboard1 噪声数据集可视化结果

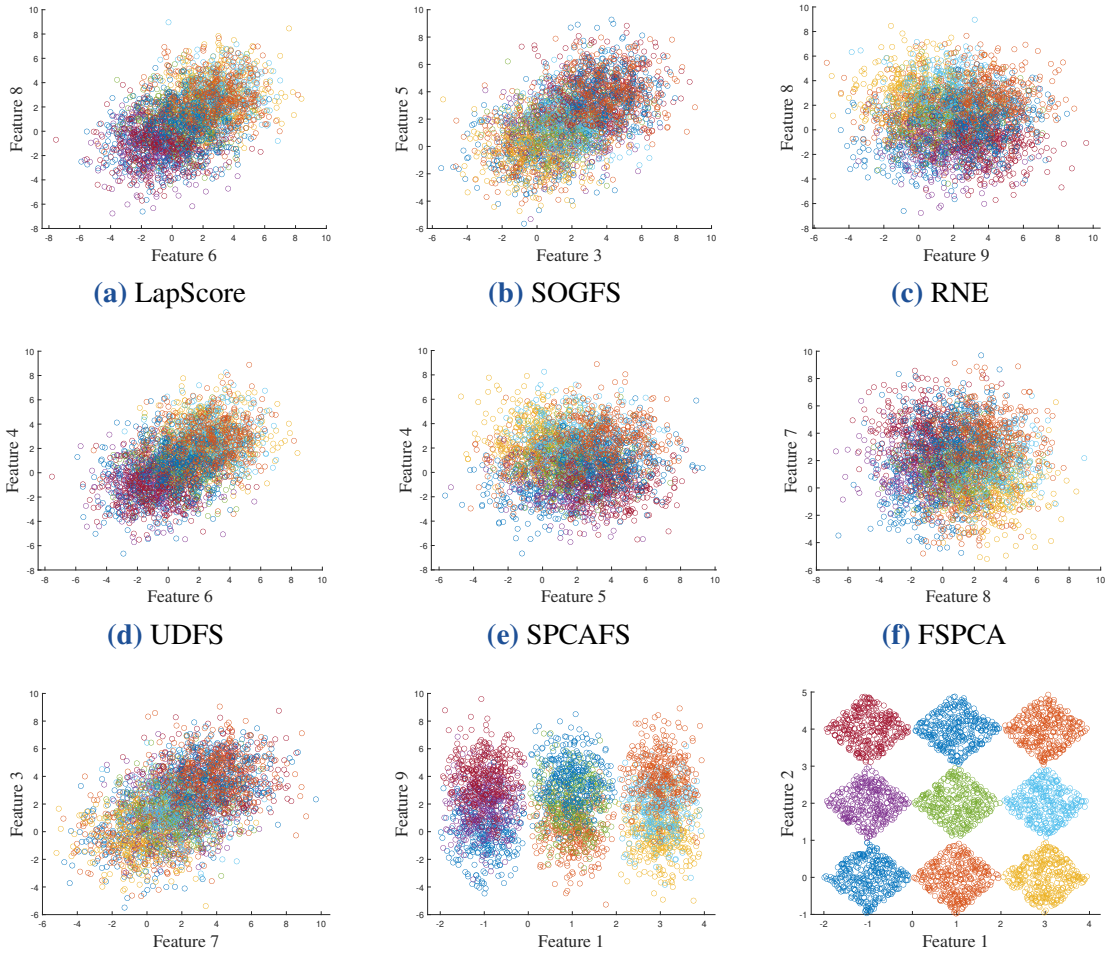
于 SOGFS、SPCAFS、SPCA-PSD、FEN-PCAFS 与 BSUFS, 正则参数在集合 $\{10^{-6}, 10^{-4}, \ldots, 10^{6}\}$ 内网格寻优. 参照文献[25] 的建议, BSUFS 的 p 与 q 从 $\{0, 1/2, 2/3\}$ 中选取. 尽管理论上 p, q 可取区间 $[0, 1)$ 内任意数值, 但区间内多数取值对应的近端算子无解析形式, 需依赖迭代数值求解. 与文献[9] 保持一致, 所有数据集的筛选特征数以 10 为步长, 取值范围设定为 $[10, 100]$. 为降低不同初始化带来的波动、保证实验可复现性与公平性, 所有数据集上的 k 均值 (k-means) 聚类算法重复运行 50 次, 并报告 50 次结果的均值与标准差.

(3) 评估指标

为评估无监督特征选择方法，采用聚类准确率 (accuracy, ACC) 与归一化互信息 (normalized mutual information, NMI) 作为指标. ACC 的定义为

$$\mathrm{ACC} = \frac{1}{n} \sum_{i=1}^{n} \delta(y_i, c_i) \times 100\%, \tag{2.38}$$

其中 n 为样本数, y_i 为第 i 个样本的真实标签, c_i 为第 i 个样本的聚类标签. 函数 $\delta(y_i, c_i)$ 表示若 $y_i = c_i$ 则取 1, 否则取 0. NMI 的定义为



(a) LapScore **(b)** SOGFS **(c)** RNE

(d) UDFS **(e)** SPCAFS **(f)** FSPCA

(g) SPCA-PSD **(h)** FEN-PCAFS **(i)** BSUFS

图 2.4: Diamond9 数据集可视化结果

$$\mathrm{NMI} = \frac{I(\boldsymbol{y}, \boldsymbol{c})}{\sqrt{H(\boldsymbol{y})H(\boldsymbol{c})}} \times 100\%, \tag{2.39}$$

其中 $\boldsymbol{y} = (y_1, y_2, \ldots, y_n) \in \mathbb{R}^n$ 为真实标签向量, $\boldsymbol{c} = (c_1, c_2, \ldots, c_n) \in \mathbb{R}^n$ 为聚类标签向量, $I(\boldsymbol{y}, \boldsymbol{c})$ 表示真实标签与聚类标签之间的互信息, $H(\boldsymbol{y})$ 与 $H(\boldsymbol{c})$ 分别对应两类标签的信息熵.

2.4.2 仿真数据结果

本节基于两组仿真数据集开展特征选择可视化实验. 针对所有方法, 先对全部 9 个特征进行打分并选取得分最高的 2 个特征, 随后将选出的特征与样本一起用散点图进行可视化展示.

图 2.2 展示了 Dartboard1 数据集的特征选择结果. 相比其他方法, RNE、UDFS、SPCA-PSD 与 BSUFS 均能够识别出合适的特征. 在此基础上, 向该数据集添加强度为 0.03 的椒盐噪声, 对应结果如图 2.3 所示. 可以看出, 仅 UDFS 与 BSUFS 能够准确筛选出判别性特征. 图 2.4 为 Diamond9 数据集的特征选择可视化结果. 对比可见, 仅有 BSUFS 能够成功识别出 2 个最具判别性特征的方法. 上述结果表明, 所提 BSUFS 能够稳定地选择出具有判别性的特征, 在仿真数据集上具有良好的鲁棒性.

表 2.2: 平均 ACC (均值 ± 标准差) 及所选特征数的结果 (%)

数据集	ALLfea	LapScore	SOGFS	RNE	UDFS	SPCAFS	FSPCA	SPCA-PSD	FEN-PCAFS	BSUFS
COIL20	58.97±4.99 (10)	53.91±3.61 (100)	56.77±3.09 (70)	49.66±3.63 (100)	55.16±3.35 (20)	51.71±3.05 (50)	54.63±3.64 (100)	56.57±4.08 (80)	**60.41±4.41 (70)**	59.18±3.49 (100)
ISOLET	59.18±3.19 (10)	52.55±2.83 (100)	41.11±1.71 (100)	48.93±2.69 (100)	47.39±2.91 (80)	54.15±2.69 (70)	52.26±2.81 (100)	53.45±2.82 (100)	56.04±3.50 (100)	**61.34±3.33 (80)**
USPS	67.79±4.96 (10)	61.76±4.52 (100)	62.83±3.79 (100)	56.00±3.48 (100)	61.28±3.46 (100)	65.43±4.90 (90)	66.98±3.92 (100)	68.38±3.85 (100)	68.36±4.62 (90)	**70.77±3.73 (50)**
UMIST	41.68±2.46 (10)	39.71±3.28 (100)	38.64±1.61 (40)	43.81±2.98 (60)	41.01±2.25 (90)	46.58±2.34 (100)	47.32±3.48 (80)	48.08±3.06 (90)	48.61±3.23 (100)	**52.29±3.61 (20)**
GLIOMA	57.44±6.40 (10)	57.36±3.60 (100)	56.64±6.47 (70)	57.32±6.47 (20)	57.80±2.98 (20)	48.04±5.26 (90)	52.08±3.64 (80)	59.32±6.27 (90)	57.24±8.16 (80)	**61.28±9.01 (100)**
PIE	25.79±1.39 (10)	34.86±1.43 (60)	26.82±1.32 (100)	23.78±1.19 (100)	17.49±0.76 (40)	30.39±1.43 (100)	41.16±2.46 (60)	43.16±2.38 (90)	**44.21±2.03 (100)**	42.45±1.74 (80)
LUNG	66.03±7.23 (10)	60.93±8.02 (70)	65.89±7.43 (90)	67.53±7.73 (90)	66.68±8.32 (100)	63.62±5.45 (20)	70.16±7.71 (100)	**73.53±8.91 (80)**	70.58±6.88 (90)	73.51±6.80 (90)
MSTAR	80.81±8.76 (10)	68.21±4.57 (100)	81.25±7.48 (100)	73.46±5.61 (100)	77.82±6.16 (100)	78.74±5.20 (30)	78.63±8.68 (90)	79.53±6.75 (90)	79.03±6.02 (50)	**81.43±6.89 (100)**
平均	57.21±4.92	53.66±3.98	53.74±4.11	52.56±4.22	53.08±3.77	54.83±3.79	57.90±4.54	60.25±4.76	60.56±4.86	**62.78±4.83**

表 2.3: 平均 NMI (均值 ± 标准差) 及所选特征数的结果 (%)

数据集	ALLfea	LapScore	SOGFS	RNE	UDFS	SPCAFS	FSPCA	SPCA-PSD	FEN-PCAFS	BSUFS
COIL20	76.04±1.69 (10)	69.01±1.53 (100)	69.12±1.17 (80)	68.03±1.59 (100)	70.76±2.07 (100)	68.41±1.60 (100)	70.29±1.31 (100)	69.21±1.41 (100)	73.23±1.31 (90)	**74.78±1.79 (100)**
ISOLET	76.09±1.77 (10)	69.86±1.26 (100)	56.73±1.05 (100)	67.15±1.45 (100)	64.74±1.28 (90)	71.12±1.11 (80)	69.18±1.33 (100)	70.11±1.11 (100)	70.14±1.56 (100)	**75.32±1.22 (100)**
USPS	62.11±2.24 (10)	59.37±1.98 (100)	57.76±2.02 (100)	53.36±1.83 (100)	52.77±2.01 (100)	61.14±1.87 (100)	60.28±2.17 (100)	63.93±2.06 (90)	**63.96±2.24 (100)**	60.16±1.68 (50)
UMIST	64.07±1.76 (10)	61.23±2.15 (100)	55.43±1.50 (80)	61.46±2.03 (60)	56.08±1.80 (70)	64.94±1.65 (100)	66.26±1.74 (100)	66.39±1.93 (90)	67.51±1.92 (100)	**67.62±1.91 (70)**
GLIOMA	49.59±6.76 (10)	48.96±3.59 (100)	45.86±8.08 (20)	46.51±9.11 (100)	**54.21±2.23 (20)**	22.17±5.17 (90)	22.01±4.88 (80)	50.31±6.65 (80)	41.16±7.66 (100)	45.14±8.66 (100)
PIE	51.01±1.02 (10)	57.53±0.73 (90)	50.55±1.03 (100)	48.05±0.76 (100)	40.45±0.79 (100)	56.21±0.90 (100)	64.94±1.30 (100)	67.40±1.21 (90)	**68.47±1.15 (100)**	66.66±1.14 (80)
LUNG	63.18±5.48 (10)	57.44±6.44 (70)	64.27±5.35 (90)	63.62±5.41 (90)	63.74±5.30 (40)	62.23±4.80 (20)	67.91±6.23 (100)	71.36±6.71 (80)	68.40±5.34 (90)	**72.64±4.69 (90)**
MSTAR	83.96±3.14 (10)	73.90±1.62 (100)	78.18±3.64 (90)	76.56±1.54 (100)	78.26±2.51 (100)	78.87±2.52 (90)	79.62±2.30 (100)	80.44±2.04 (90)	79.34±3.27 (100)	**80.66±2.68 (100)**
平均	65.76±2.98	62.16±2.41	59.74±2.98	60.59±2.96	60.13±2.25	60.64±2.45	62.56±2.66	67.39±2.89	66.53±3.06	**67.87±2.97**

2.4.3 真实数据结果

本节给出八个真实数据集上的数值实验结果, 同时引入全特征聚类方法 ALLfea 作为基准参照, 将全部特征的聚类结果视为无特征压缩下的参考标准. 表 2.2 与表 2.3 分别汇总了所有方法的均值与标准差, 括号内为取得最优性能时对应的筛选特征数量.

对于 ACC 指标, 所提 BSUFS 在绝大多数真实数据集上取得最优或次优结果, 甚至优于最新的 FEN-PCAFS 与 SPCA-PSD. 由平均指标可知, BSUFS 平均 ACC 得分位列所有对比算法首位, 其次为 FEN-PCAFS 与 SPCA-PSD. 此外, 相比 RNE 与 UDFS 等图方法, 基于主成分分析的 SPCAFS 与 SPCA-PSD 整体表现更佳. 由于引入了双稀疏正则项, BSUFS 相比 SPCAFS 的 ACC 平均提升 7.95%. 当然, 在部分情形下, BSUFS 略逊于 SPCA-PSD, 说明低秩先验同样能够提升无监督特征选择的性能. 对于 ISOLET 数据集, BSUFS 的 ACC 提升效果尤为突出, 原因可能在于引入的 ℓ_q 范数抑制了语音数据中的细粒度噪声与说话人差异, 从而提取更有效的特征.

对于 NMI 指标, 可得到类似与上述 ACC 的结论. 需要说明的是, 此处 NMI 结果均采用最优 ACC 对应的参数设置, 未针对 NMI 单独寻优, 因此部分数据集上 NMI 未达到理论最优值. 平均而言, BSUFS 相比其他方法至少提升 0.48%. 对于样本数较少而特征数较多的 GLIOMA 数据集, 虽然其 ACC 更高, 但 NMI 相对较低, 这可能是因为该数据集仅有 4 个类别, 容易造成 ACC 与 NMI 之间的不均衡.

总体而言, 所提 BSUFS 在多个真实数据集上取得了更高的 ACC 与 NMI. 另一方面, 在较少的特征数量下仍能取得更高的准确率, 使得 BSUFS 在真实应用场景中更具实用性.

2.4.4 消融实验

为研究 BSUFS 中双稀疏正则项的作用, 本节设置如下四组消融对比方案: (I) 去掉 $\ell_{2,p}$ 范数与 ℓ_q 范数的 BSUFS, (II) 去掉 $\ell_{2,p}$ 范数的 BSUFS, (III) 去掉 ℓ_q 范数的 BSUFS, (IV) 完整的

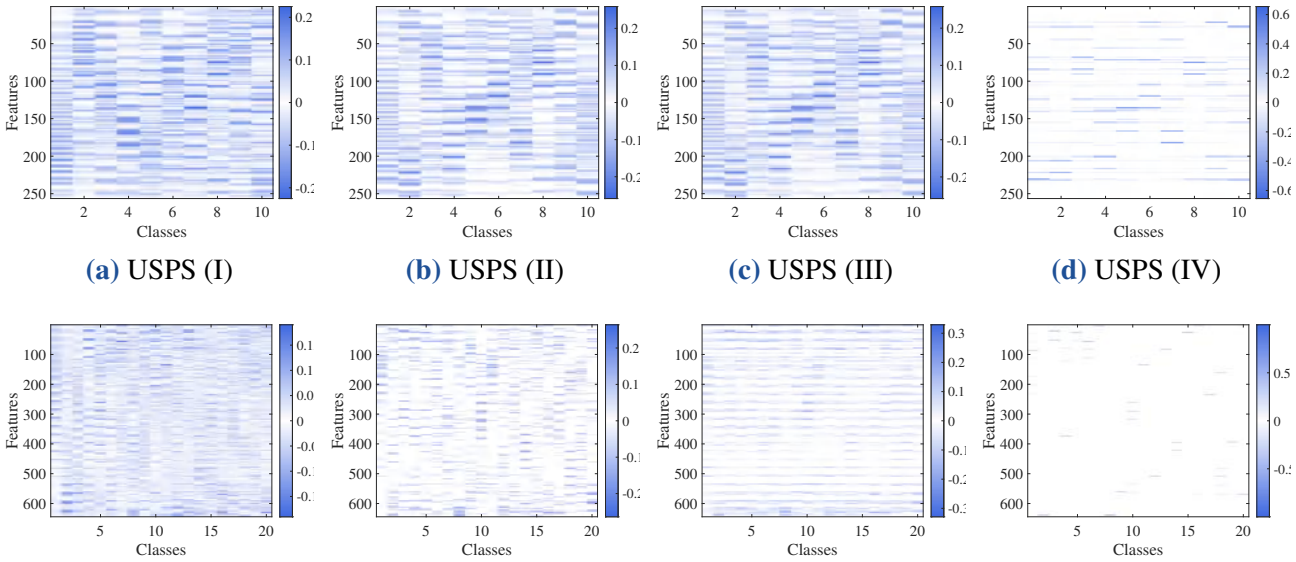
表 2.4: 所提方法消融实验结果 (%)

数据集	ACC				NMI			
	I	II	III	IV	I	II	III	IV
COIL20	54.09	57.33	58.76	**59.18**	69.94	72.12	74.57	**74.78**
ISOLET	51.77	58.78	56.19	**61.34**	66.84	73.30	72.73	**75.32**
USPS	67.06	66.42	68.11	**70.77**	58.86	35.66	**61.14**	60.16
UMIST	47.16	47.57	49.23	**52.29**	66.48	67.77	**69.45**	67.62
GLIOMA	49.76	58.40	60.12	**61.28**	20.64	**52.11**	43.13	45.14
PIE	40.98	41.05	41.15	**42.45**	65.02	65.13	65.23	**66.66**
LUNG	71.34	71.51	72.33	**73.51**	69.31	69.17	71.94	**72.64**
MSTAR	79.25	74.67	80.08	**81.43**	79.92	73.14	79.97	**80.66**

BSUFS. 此处统一取 $p, q \in [0, 1)$.

表 2.4 给出了四组方案在 ACC 与 NMI 指标下的聚类结果. 实验表明, 方案 IV 在绝大多数数据集上均取得最优表现. 相较于方案 I, 方案 IV 在 USPS 与 GLIOMA 数据集上的 ACC 分别由 67.06%、49.76% 提升至 70.77%、61.28%, 性能提升显著. 对比方案 III 与方案 IV 可以发现, 引入 ℓ_q 范数正则后, 所提方法的聚类 ACC 与 NMI 指标均有所提升, 这表明 ℓ_q 范数对无监督特征选择是有意义的.

图 2.5 展示了 USPS 与 UMIST 数据集上变换矩阵 $\boldsymbol{W}$ 的热力图可视化结果. 由于图像数据普遍存在背景噪声、纹理冗余等干扰, 稀疏性强弱直接决定特征选择的不同. 显然, 方案 IV 融合了 $\ell_{2,p}$ 范数行稀疏与 ℓ_q 范数元素级稀疏的双重优, 获得了更稀疏的变换矩阵 $\boldsymbol{W}$, 即更聚焦于有效特征.



(a) USPS (I) **(b)** USPS (II) **(c)** USPS (III) **(d)** USPS (IV)

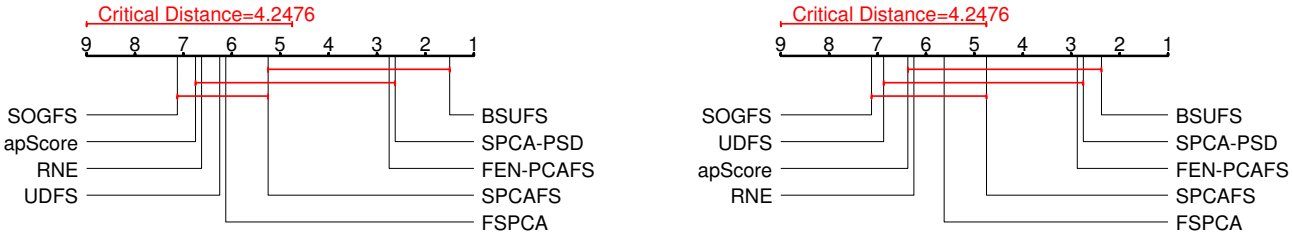
(e) UMIST (I) **(f)** UMIST (II) **(g)** UMIST (III) **(h)** UMIST (IV)

图 2.5: 变换矩阵的可视化结果

综合上述消融实验结果可知, 所提 BSUFS 通过引入双稀疏正则项能够提升主成分分析在特征选择任务中的性能, 这表明双稀疏优化框架具备合理性、有效性与可拓展性.

2.4.5 统计分析

为评估所有对比方法之间的两两差异, 本节采用事后 Nemenyi 检验开展统计显著性分析. 引入临界差值 (critical difference, CD) 作为差异性判定依据, 检验结果如图 2.6. 可以发现, 相较于 SOGFS、LapScore、UDFS 与 RNE 等基于图学习特征选择方法, 所提 BSUFS 在统计意义上存在显著差异. 而与 FSPCA、SPCAFS、SPCA-PSD、PEN-PCAFS 等基于主成分分析的特征选择方法相比, BSUFS 的表现并无显著性差异.



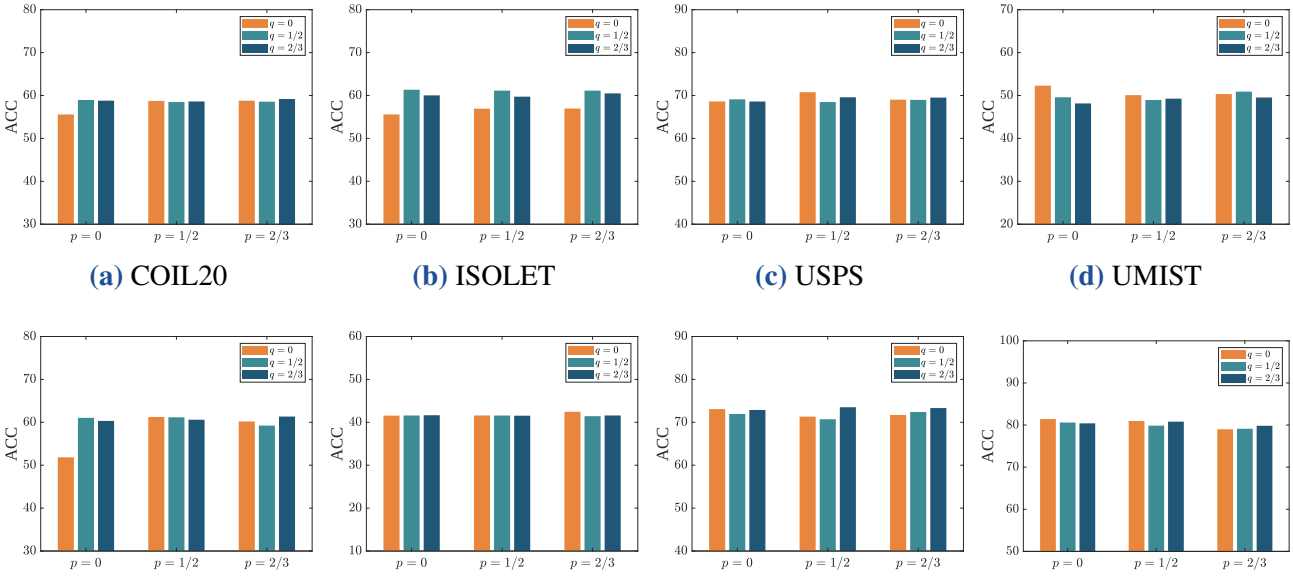
(a) ACC 指标　　**(b)** NMI 指标

图 2.6: Nemenyi 检验结果

2.4.6 参数 p 与 q 分析

在所提 BSUFS 双稀疏优化模型中, p 与 q 常用的三种取值为 $\{0, 1/2, 2/3\}$. 为探究 p, q 对模型聚类性能的影响, 图 2.7 与图 2.8 分别可视化了不同参数组合下 ACC 与 NMI 的结果. 图中横轴表示参数 p 的不同取值, 柱状图的颜色变化表示参数 q 的不同取值.

首先, 不同数据集对应的最优 p 与 q 选择不同. 具体而言, 对于 ISOLET 数据集, 最优的 p 与 q 分别为 0 与 1/2, 而对于 LUNG 数据集, 最优取值分别为 1/2 与 2/3. 其次, 在不同 p 与 q 取值下, ACC 与对应的 NMI 并不总是一致. 例如, 对于 GLIOMA 数据集, 当 $p = 0$ 时, ACC 的变化幅度较大, 这说明 q 的选择会影响聚类的结果. 最后, 对于 UMIST 与 MSTAR 数据集, 当 p 与 q 同时取 0 时, 可取得最优聚类效果, 这表明将取值范围从 $(0, 1)$ 扩展到 $[0, 1)$ 具有重要意义. 由此可以看出, 参数 p 与 q 需结合数据特性谨慎调节.



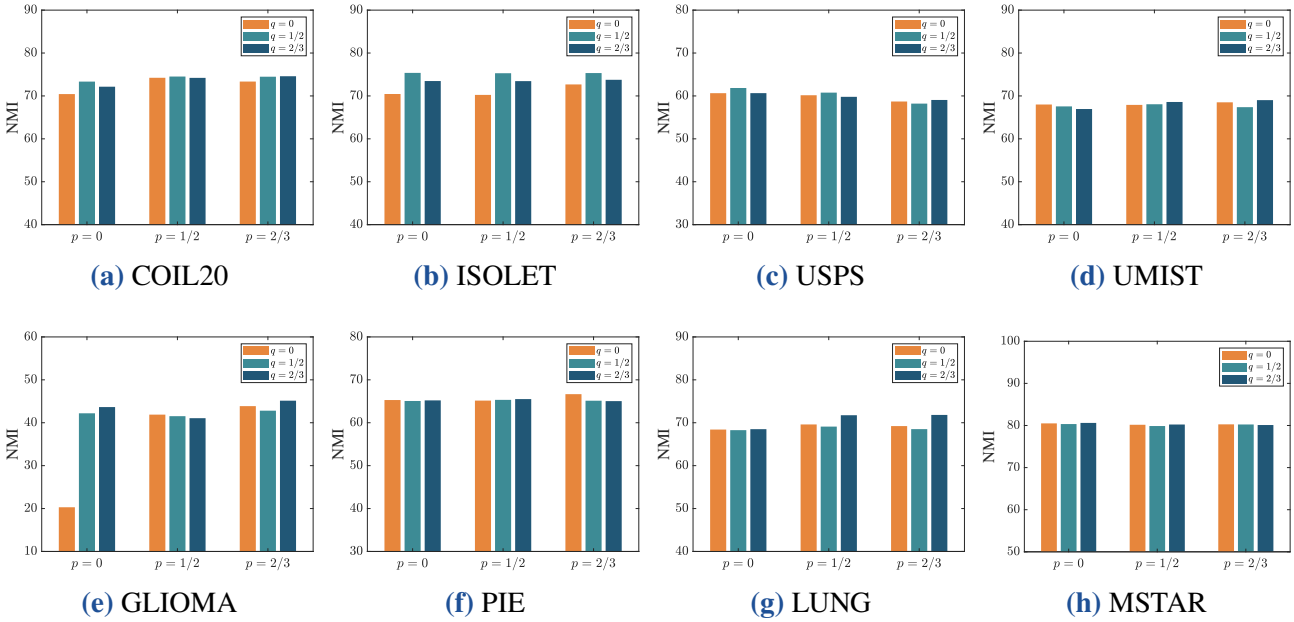
(a) COIL20　**(b)** ISOLET　**(c)** USPS　**(d)** UMIST

(e) GLIOMA　**(f)** PIE　**(g)** LUNG　**(h)** MSTAR

图 2.7: 参数 p 与 q 对 ACC 的影响

2.4.7 讨论

(1) 特征相关性

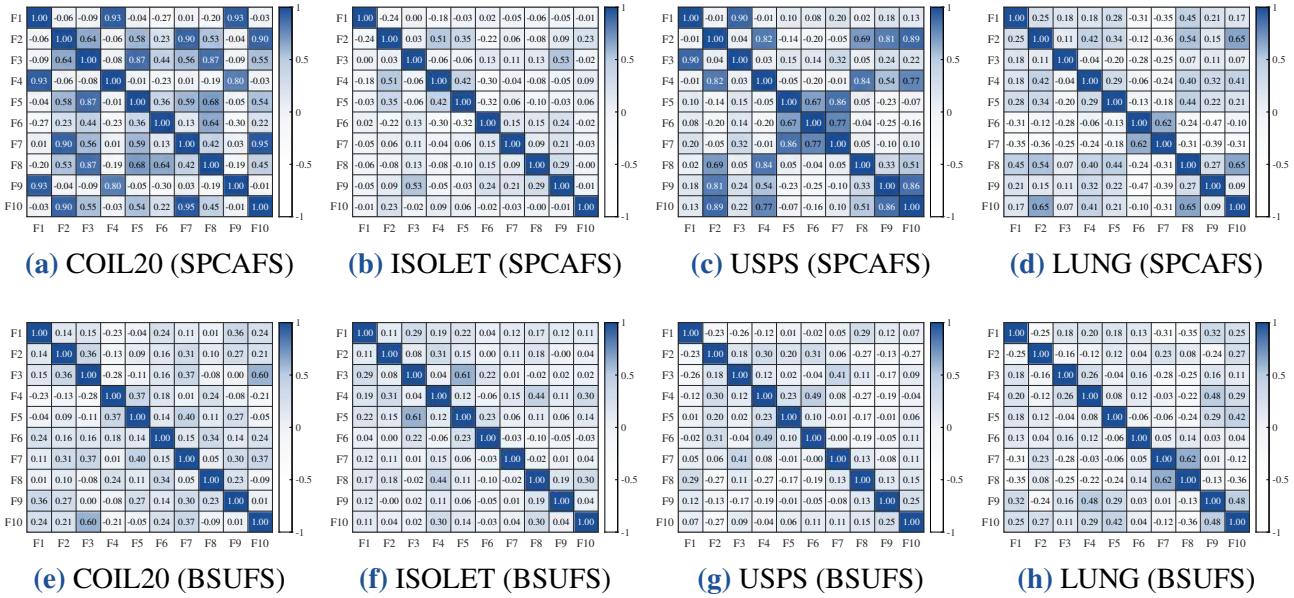


(a) COIL20 (b) ISOLET (c) USPS (d) UMIST

(e) GLIOMA (f) PIE (g) LUNG (h) MSTAR

图 2.8: 参数 p 与 q 对 NMI 的影响

图 2.9 可视化了 COIL20 与 USPS 数据集上 SPCAFS 与 BSUFS 的特征相关性结果. 实验统一选取前 10 维关键特征, 记为 $F_1, F_2, \ldots, F_{10}$, 并考察这些特征之间的相关性. 显然, 与代表性方法 SPCAFS 相比, 所提 BSUFS 的特征相关性更低, 这意味着引入 ℓ_q 范数能够消除冗余特征并改善特征选择结果.



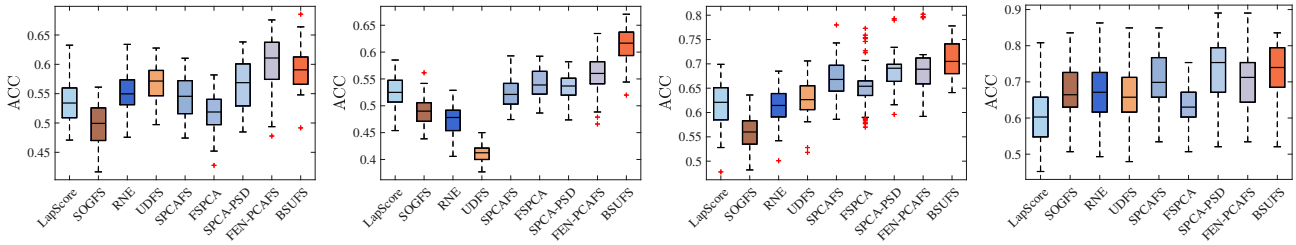
(a) COIL20 (SPCAFS) (b) ISOLET (SPCAFS) (c) USPS (SPCAFS) (d) LUNG (SPCAFS)

(e) COIL20 (BSUFS) (f) ISOLET (BSUFS) (g) USPS (BSUFS) (h) LUNG (BSUFS)

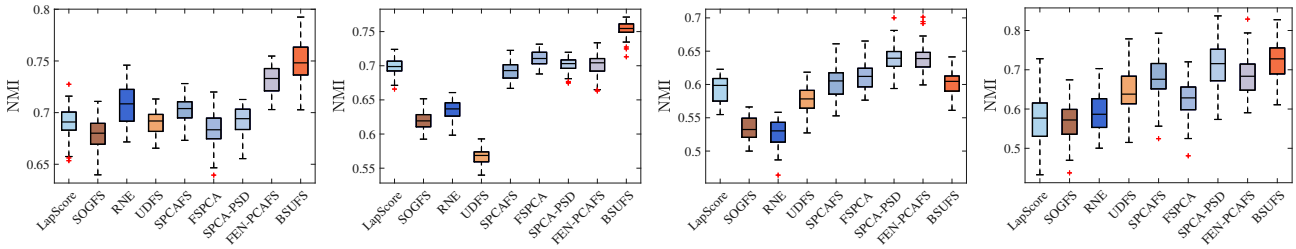
图 2.9: 特征相关性可视化结果

(2) 模型稳定性

为评估模型鲁棒性, 本节开展 50 次聚类实验, 其箱线图如图 2.10 所示. 可以观察到, 无论是 ACC 还是 NMI 指标, BSUFS 的平均值整体上高于其他对比方法, 尤其是在 ISOLET 数据上提升更为明显. 这是因为该数据集具有较强稀疏性, 从而更充分地体现了双稀疏建模的优势.



(a) COIL20 (ACC) (b) ISOLET (ACC) (c) USPS (ACC) (d) LUNG (ACC)

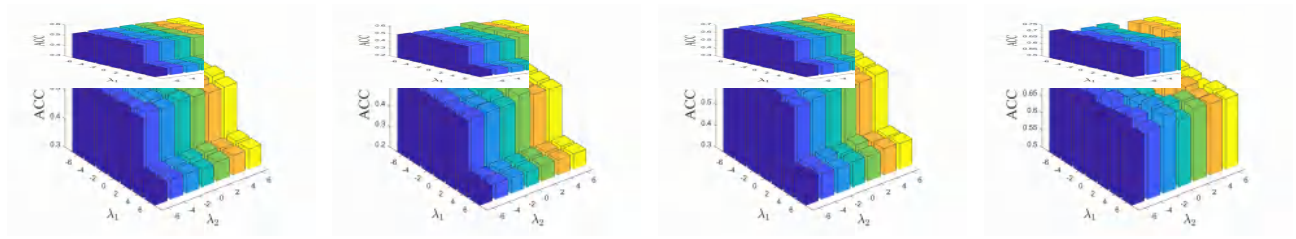


(e) COIL20 (NMI) (f) ISOLET (NMI) (g) USPS (NMI) (h) LUNG (NMI)

图 2.10: 模型稳定性比较

(3) 参数敏感性

图 2.11 探究了正则参数 λ_1 与 λ_2 对 ACC 与 NMI 指标的影响. 尽管不同 λ_2 带来的提升通常不如不同 λ_1 显著, 但仍存在差异, 这也从侧面验证了 BSUFS 中双稀疏项的必要性. 总体而言, $\ell_{2,p}$ 范数在 BSUFS 中起主导作用, 而 ℓ_q 范数在特征选择中起补充作用.



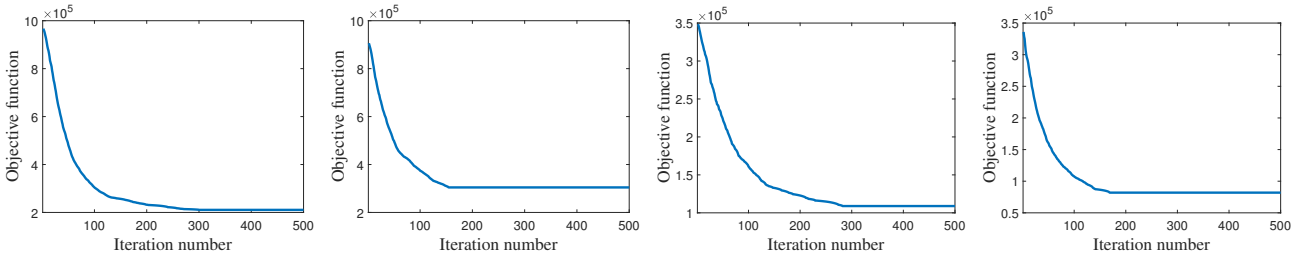
(a) COIL20 (ACC) (b) ISOLET(ACC) (c) USPS(ACC) (d) LUNG (ACC)

(e) COIL20 (NMI) (f) ISOLET (NMI) (g) USPS (NMI) (h) LUNG (NMI)

图 2.11: 参数敏感性分析

(4) 算法收敛性

图 2.12 展示了算法 1 的目标函数值曲线. 可以看出, 所提 BSUFS 的目标函数呈现出持续下降的一致趋势, 并在有限迭代次数内达到稳定. 尽管无法在理论上证明算法的收敛性, 但在数值层面具有良好的收敛性.



(a) COIL20 **(b)** ISOLET **(c)** USPS **(d)** LUNG

图 2.12: 算法收敛性分析

2.5 本章小结

本章针对无监督特征选择问题, 将取值满足 $p, q \in [0, 1)$ 的 $\ell_{2,p}$ 范数与 ℓ_q 范数共同引入主成分分析, 首次构造了统一的双稀疏正则方法. 技术上, $\ell_{2,p}$ 范数用于挖掘特征的稀疏结构, 而 ℓ_q 范数用于抑制冗余特征带来的干扰. 在算法方面, 利用流形优化和稀疏优化技巧设计了高效的近端交替最小化求解策略, 并分析了算法的计算复杂度. 实验结果表明, 所提方法在 ACC 与 NMI 指标上均优于其他各类对比方法. 此外, 还进一步验证, 将参数 p 与 q 的取值范围扩展至 $[0, 1)$ 是必要的, 且二者的最优选择需由具体数据集决定. 总之, 在无监督特征选择中, $\ell_{2,p}$ 范数起主导作用, 而 ℓ_q 范数可进一步强化特征筛选的效果并改善模型的鲁棒性.

第 3 章 基于稀疏联邦主成分分析的异常检测

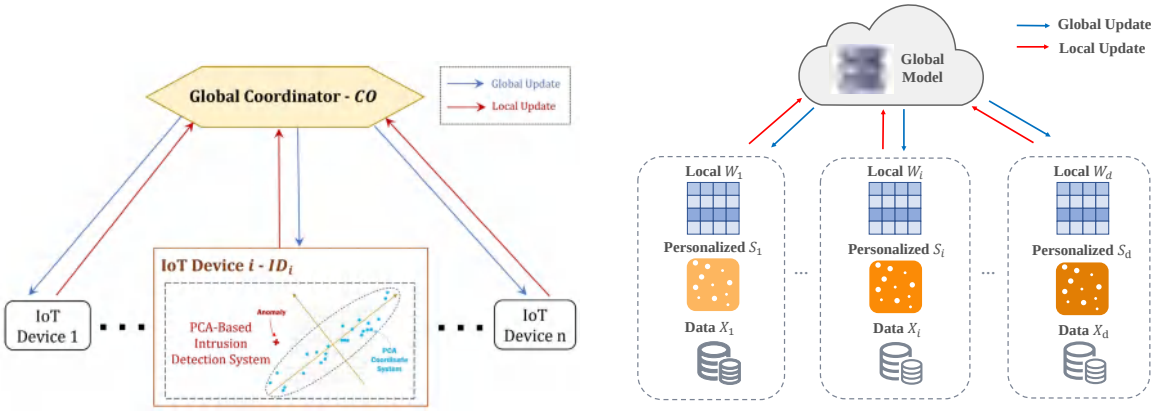
作为一种隐私保护框架, 联邦学习在物联网通讯中受到广泛关注. 然而, 现有联邦主成分分析缺乏对个性化与鲁棒性的有效适配, 严重制约了异常检测任务的实际落地效果. 针对上述不足, 本章提出了一种面向物联网异常检测的高效个性化联邦主成分分析 (efficient personalized federated PCA, FedEP). 该方法通过引入 ℓ_1 范数实现客户端模型的个性化定制, 同时引入 $\ell_{2,1}$ 范数强化全局联邦模型的稳健性. 为求解非凸问题, 设计了基于交替方向乘子法的黎曼流形优化算法, 并给出了收敛性分析. 实验结果表明, 相较于当前主流联邦主成分分析, 所提 FedEP 在关键评价指标上均实现提升.

3.1 引言

物联网 (Internet of things, IoT) 已成为现代无线通信的基础, 成功链接了网络空间与物理世界[30]. 通过赋能数据驱动的自主决策, 物联网技术正推动智慧城市、精准农业及新一代医疗健康等多个领域的数字化转型, 深刻改变着人们的生产生活模式. 然而, 这种无处不在的互联互通特性, 也使得物联网生态系统面临各类复杂网络威胁的侵害[31]. 作为物联网安全防护的第一道防线, 异常检测在构建物联网生态系统安全框架中发挥着关键作用.

过去十年间, 学术界和工业界已提出多种物联网异常检测方法. 集中式方法凭借复杂架构取得了显著的检测精度, 例如用于工业场景检测的双自编码器生成对抗网络 (generative adversarial networks, GAN)[32]、统计分析驱动的自编码器[33], 以及针对复杂物联网网络环境设计的混合深度学习框架[34]. 然而, 对集中式存储库的依赖带来了严峻的隐私泄露风险. 在此背景下, 联邦学习 (federated learning, FL) 作为一种极具前景的分布式架构应运而生, 通过在边缘设备上完成本地模型训练, 仅将模型参数或梯度上传至中央服务器进行聚合, 有效缓解了数据孤岛问题并强化了隐私保护[35]. 尽管已有大量研究将联邦学习应用于物联网入侵检测, 包括基于分组的联邦学习方法 (group-based federated learning, FedGroup)[36], 但这类基于深度学习的联邦学习方法往往伴随着高昂的计算与通信开销. 其庞大的参数量频繁超出资源受限型边缘物联网设备的承载能力, 而资源密集型的反向传播过程, 进一步挑战了现有计算卸载策略的可行性. 因此, 受限于传感器的计算能力与电池续航, 亟需构建更轻量级的模型以维持高效运行[37].

近期, Nguyen 等[38] 将主成分分析 (principal component analysis, PCA) 拓展到联邦学习框架, 提出了黎曼流形上的联邦主成分分析 (federated PCA on Grassmann manifold, FedPG). 用户无需交换原始数据, 即可实现协作式子空间估计. 然而, FedPG 在异构物联网环境部署时, 仍面临两个瓶颈. 一方面, 个性化需求被低估[39]. 物联网数据在不同传感器及部署位置上呈现出固有的非独立同分布 (non-independent and identically distributed, non-IID) 特性, 强制采用单一全



(a) 现有 FedPG　　**(b)** 所提 FedEP

图 3.1: 与现有方法的框架对比

局模型会导致对局部分布的次优拟合, 进而降低网关级别的检测灵敏度[40]. 另一方面, 对数据损坏的鲁棒性不足[41]. 经典主成分分析对噪声和异常值高度敏感, 若缺乏有效的鲁棒机制, 原始数据中的恶意扰动或传感器故障可能严重扭曲估计的子空间结构[42].

基于此, 本章提出了一种面向物联网异常检测的高效个性化联邦主成分分析 (efficient personalized federated PCA, FedEP), 整体结构如图 3.1 所示. 该框架将每个客户端的本地数据分解为低秩分量与稀疏分量, 其中低秩分量用于捕获设备的正常行为模式, 而稀疏分量则用于表征异常信号. FedEP 的独特之处在于, 将个性化与双稀疏正则有机结合, 利用 $\ell_{2,1}$ 范数施加行稀疏以实现结构鲁棒性, 利用 ℓ_1 范数提供逐元素稀疏性以过滤冗余噪声. 需要说明的是, 本章的双稀疏正则作用于两个不同变量, 与第二章的双稀疏设定存在区别[43]. 同时, 也没有采用非凸稀疏正则项, 原因在于非凸正则虽理论上可获得更优的检测性能, 但对超参数选取敏感, 易降低模型的稳定性与实际部署适用性. 本章的主要贡献为

(1) 提出了一种兼具个性化与鲁棒性的物联网异常检测方法, 允许每个局部模型捕获其特定的数据分布, 同时保持全局结构的一致性.

(2) 融合黎曼流形优化和半光滑牛顿技巧, 开发了一种基于交替方向乘子法的高效算法, 并在理论上分析了算法的收敛性.

(3) 通过实验表明, 所提方法相较于先进的 FedPG, 在检测精度、个性化能力及计算效率三个维度均展现出一定的优势.

3.2 数学模型

3.2.1 稀疏主成分分析

给定数据 $\boldsymbol{X} = (\boldsymbol{x}_1, \boldsymbol{x}_2, \ldots, \boldsymbol{x}_p) \in \mathbb{R}^{n\times p}$, 其中第 j 个样本为 $\boldsymbol{x}_j \in \mathbb{R}^n$. 稀疏主成分分析 (sparse PCA, SPCA)[44] 通过引入 ℓ_1 范数正则项实现投影矩阵 $\boldsymbol{W} \in \mathbb{R}^{n\times m}$ 的稀疏诱导, 其数学模型为

$$\begin{aligned} \min_{\boldsymbol{W}} \quad & \|\boldsymbol{X}-\boldsymbol{W}\boldsymbol{W}^T\boldsymbol{X}\|_F^2+\beta\|\boldsymbol{W}\|_{2,1} \\ \text{s.t.} \quad & \boldsymbol{W}^T\boldsymbol{W}=\boldsymbol{I}_m, \end{aligned} \tag{3.1}$$

其中, $\beta>0$ 为正则参数, 用于调节 $\boldsymbol{W}$ 稀疏度.

在稀疏主成分分析的鲁棒性拓展研究中, 鲁棒稀疏主成分分析 (robust sparse PCA, RSPCA)[45]是一类重要的改进模型. 本质上, RSPCA 将原始观测数据分解为低秩有效分量与稀疏误差分量, 以此抑制离群点与噪声带来的模型退化问题, 对应的数学模型为

$$\begin{aligned} \min_{\boldsymbol{W},\boldsymbol{S}} \quad & \|(\boldsymbol{I}-\boldsymbol{W}\boldsymbol{W}^T)(\boldsymbol{X}-\boldsymbol{S})\|_F^2+\alpha\|\boldsymbol{S}\|_1 \\ \text{s.t.} \quad & \boldsymbol{W}^T\boldsymbol{W}=\boldsymbol{I}_m, \end{aligned} \tag{3.2}$$

其中, $\boldsymbol{S}$ 为稀疏矩阵, 用于捕获离群点或噪声, 保障低维子空间 $\boldsymbol{W}$ 能够精准获得原始数据的结构信息. 此外, $\alpha>0$ 为正则参数, 用于控制噪声的强度.

3.2.2 联邦主成分分析

在物联网领域, 基于主成分分析的异常检测通过将高维数据投影到低秩子空间来识别恶意流量. 对任意样本 $\boldsymbol{x}_j\in\mathbb{R}^n$, 定义其重构误差为

$$\boldsymbol{S}(\boldsymbol{x}_j)=\|(\boldsymbol{I}-\boldsymbol{W}\boldsymbol{W}^T)\boldsymbol{x}_j\|^2. \tag{3.3}$$

当样本重构误差超出预设判别阈值时, 即可判定该样本为异常数据[46]. 然而, 传统主成分分析难以处理恶意异常值以及集中式数据聚合带来的隐私风险. 为此, 联邦主成分分析[38] 应运而生, 旨在 d 个客户端之间寻求全局共识. 记第 i 个本地客户端的观测数据为 $\boldsymbol{X}_i\in\mathbb{R}^{n\times p}$, 对应局部投影矩阵为 $\boldsymbol{W}_i\in\mathbb{R}^{n\times m}$, 则 FedPG 可表示为

$$\begin{aligned} \min_{\{\boldsymbol{W}_i\},\boldsymbol{V}} \quad & \sum_{i=1}^{d}\|\boldsymbol{X}_i-\boldsymbol{W}_i\boldsymbol{W}_i^T\boldsymbol{X}_i\|_F^2 \\ \text{s.t.} \quad & \boldsymbol{W}_i=\boldsymbol{V},\ \boldsymbol{W}_i^T\boldsymbol{W}_i=\boldsymbol{I}_m,\ \forall i\in[d], \end{aligned} \tag{3.4}$$

其中, $[d]=\{1,2,\ldots,d\}$ 为索引集合, $\boldsymbol{V}$ 为全局共识变量, 强制各局部客户端子空间保持全局一致性. 与主成分分析相比, FedPG 在分布式物联网异常检测中具备更优的适配性与泛化性能.

3.2.3 构建模型

结合分布式数据非独立同分布特性、局部噪声干扰与隐私保护的多重现实约束, 本章构建了如下 FedEP 模型

$$\begin{aligned} \min_{\{\boldsymbol{W}_i\},\{\boldsymbol{S}_i\},\boldsymbol{V}} \quad & \sum_{i=1}^{d}(\|(\boldsymbol{I}-\boldsymbol{W}_i\boldsymbol{W}_i^T)(\boldsymbol{X}_i-\boldsymbol{S}_i)\|_F^2+\alpha\|\boldsymbol{S}_i\|_1+\beta\|\boldsymbol{W}_i\|_{2,1}) \\ \text{s.t.} \quad & \boldsymbol{W}_i=\boldsymbol{V},\ \boldsymbol{W}_i^T\boldsymbol{W}_i=\boldsymbol{I},\ \forall i\in[d]. \end{aligned} \tag{3.5}$$

其中, $\boldsymbol{S}_i$ 为第 i 个本地客户端的噪声, $\|\boldsymbol{W}_i\|_{2,1}$ 表示矩阵 $\boldsymbol{W}_i$ 的 $\ell_{2,1}$ 范数, $\alpha, \beta > 0$ 为正则参数.

与 FedPG[38] 等现有框架相比, 所提 FedEP 具有自适应个性化和结构鲁棒性的优势. 具体而言, 稀疏分量 $\boldsymbol{S}_i$ 可捕获局部特定的噪声和异常值, 使模型能够适应物联网数据的非独立同分布特性. 投影矩阵 $\boldsymbol{W}_i$ 引入 $\ell_{2,1}$ 范数, 一方面实现冗余特征筛选, 提升子空间学习的稳定性, 另一方面通过特征维度压缩, 降低迭代的计算开销.

3.3 优化算法

由于模型涉及多个变量, 本节采用交替方向乘子法 (alternating direction method of multipliers, ADMM) 对其进行迭代求解. 为此, 首先引入辅助变量 $\boldsymbol{U}_i$, 将式 (3.5) 等价转化为

$$\begin{aligned} \min_{\{\boldsymbol{W}_i\},\{\boldsymbol{S}_i\},\{\boldsymbol{U}_i\},\boldsymbol{V}} \quad & \sum_{i=1}^{d} (\|(\boldsymbol{I} - \boldsymbol{W}_i\boldsymbol{W}_i^T)\boldsymbol{U}_i\|_F^2 + \alpha\|\boldsymbol{S}_i\|_1 + \beta\|\boldsymbol{W}_i\|_{2,1} \\ \text{s.t.} \quad & \boldsymbol{X}_i - \boldsymbol{S}_i = \boldsymbol{U}_i, \ \forall i \in [d], \\ & \boldsymbol{W}_i - \boldsymbol{V} = 0, \ \forall i \in [d], \\ & \boldsymbol{W}_i^T W_i = \boldsymbol{I}, \ \forall i \in [d]. \end{aligned} \tag{3.6}$$

根据 ADMM 更新规则, 全局增广拉格朗日函数 L 可分解为 d 个局部函数之和, 实现去中心化优化, 其具体形式为

$$L(\{\boldsymbol{W}_i\}, \{\boldsymbol{S}_i\}, \{\boldsymbol{U}_i\}, \boldsymbol{V}, \{\boldsymbol{\Lambda}_i\}, \{\boldsymbol{\Pi}_i\}) = \sum_{i=1}^{d} L_i(\boldsymbol{W}_i, \boldsymbol{S}_i, \boldsymbol{U}_i, \boldsymbol{V}, \boldsymbol{\Lambda}_i, \boldsymbol{\Pi}_i), \tag{3.7}$$

其中局部拉格朗日函数 L_i 定义为

$$\begin{aligned} & L_i(\boldsymbol{W}_i, \boldsymbol{S}_i, \boldsymbol{U}_i, \boldsymbol{V}, \boldsymbol{\Lambda}_i, \boldsymbol{\Pi}_i) \\ & = \|(\boldsymbol{I} - \boldsymbol{W}_i\boldsymbol{W}_i^T)\boldsymbol{U}_i\|_F^2 + \alpha\|\boldsymbol{S}_i\|_1 + \beta\|\boldsymbol{W}_i\|_{2,1} \\ & + \langle \boldsymbol{\Lambda}_i, \boldsymbol{X}_i - \boldsymbol{S}_i - \boldsymbol{U}_i \rangle + \frac{\mu}{2}\|\boldsymbol{X}_i - \boldsymbol{S}_i - \boldsymbol{U}_i\|_F^2 \\ & + \langle \boldsymbol{\Pi}_i, \boldsymbol{W}_i - \boldsymbol{V} \rangle + \frac{\nu}{2}\|\boldsymbol{W}_i - \boldsymbol{V}\|_F^2, \end{aligned} \tag{3.8}$$

其中, $\boldsymbol{\Lambda}_i$ 和 $\boldsymbol{\Pi}_i$ 为拉格朗日乘子, $\mu, \nu > 0$ 为惩罚参数. 该去中心化分解结构的优势在于每个客户端 i 可独立更新其局部变量 $\boldsymbol{W}_i$, 仅需在全局层面同步共识变量 $\boldsymbol{V}$, 既能有效保护本地数据隐私, 又能显著降低客户端间的通信开销.

3.3.1 更新 $\boldsymbol{W}_i$

在每次迭代过程中, $\boldsymbol{W}_i$ 的更新需满足 Stiefel 流形约束, 即 $\boldsymbol{W}_i \in \mathrm{St}(n, m)$. 经过整理, 变量 $\boldsymbol{W}_i$ 的子问题可表示为

$$\min_{\boldsymbol{W}_i\in\mathrm{St}(n,m)} \quad \|(\boldsymbol{I}_n-\boldsymbol{W}_i\boldsymbol{W}_i^T)\boldsymbol{U}_i^k\|_F^2+\beta\|\boldsymbol{W}_i\|_{2,1}+\frac{\nu}{2}\|\boldsymbol{W}_i-\boldsymbol{Z}_i^k\|_F^2, \tag{3.9}$$

其中, $\boldsymbol{Z}_i^k=\boldsymbol{V}^k-\boldsymbol{\Pi}_i^k/\nu$ 为迭代过程中的中间变量. 需要注意的是, 式 (3.9) 的目标函数包含两个光滑正则和一个非光滑正则, 且受非凸 Stiefel 流形约束, 直接求解存在较大的计算挑战. 为简化描述, 将该子问题重新表述为光滑部分与非光滑部分, 即

$$\min_{\boldsymbol{W}_i\in\mathrm{St}(n,m)} \quad h(\boldsymbol{W}_i)+g(\boldsymbol{W}_i), \tag{3.10}$$

其中

$$\begin{aligned} h(\boldsymbol{W}_i)&=\|\boldsymbol{U}_i^k-\boldsymbol{W}_i\boldsymbol{W}_i^T\boldsymbol{U}_i^k\|_F^2+\frac{\nu}{2}\|\boldsymbol{W}_i-\boldsymbol{Z}_i^k\|_F^2,\\ g(\boldsymbol{W}_i)&=\beta\|\boldsymbol{W}_i\|_{2,1}. \end{aligned} \tag{3.11}$$

显然, $h(\boldsymbol{W}_i)$ 为光滑函数, $g(\boldsymbol{W}_i)$ 为非光滑函数. 通过展开 Frobenius 范数项, 并利用 Stiefel 流形的正交性, 可推导出当前迭代点 $\boldsymbol{W}_i^k$ 处的梯度表达式为

$$\nabla h(\boldsymbol{W}_i^k)=-2\boldsymbol{U}_i^k(\boldsymbol{U}_i^k)^T\boldsymbol{W}_i^k+\nu(\boldsymbol{W}_i^k-\boldsymbol{Z}_i^k). \tag{3.12}$$

为在严格满足 Stiefel 流形约束的同时处理 $g(\boldsymbol{W}_i)$, 采用交替流形近端梯度法 (alternating manifold proximal gradient method, AManPG)[47] 求解该子问题. 其中, 下降方向 $\boldsymbol{D}_i$ 通过求解以下带约束的子问题确定

$$\begin{aligned} \min_{\boldsymbol{D}_i}\quad & \langle\nabla h(\boldsymbol{W}_i^k),\boldsymbol{D}_i\rangle+\frac{1}{2t}\|\boldsymbol{D}_i\|_F^2+\beta\|\boldsymbol{W}_i^k+\boldsymbol{D}_i\|_{2,1}\\ \text{s.t.}\quad & \boldsymbol{D}^T\boldsymbol{W}_i^k+(\boldsymbol{W}_i^k)^T\boldsymbol{D}=0, \end{aligned} \tag{3.13}$$

其中, $t>0$ 为步长参数。当求得最优下降方向 $\boldsymbol{D}_i^*$ 后, 下一迭代点通过黎曼收缩获得, 即 $\boldsymbol{W}_i^{k+1}=\mathrm{Retr}_{\boldsymbol{W}_i^k}(\alpha\boldsymbol{D}_i^*)$. 这里, α 通过回溯线搜索确定, 以保证目标函数的充分下降.

设 $\boldsymbol{\Lambda}_i\in\mathbb{R}^{m\times m}$ 为与切空间约束相关的对称拉格朗日乘子. 根据一阶最优性条件, 可推导出 $\boldsymbol{D}_i$ 关于 $\boldsymbol{\Lambda}_i$ 的解析形式为

$$\boldsymbol{D}_i(\boldsymbol{\Lambda}_i)=\mathrm{prox}_{2,1}(B(\boldsymbol{\Lambda}_i),t\beta)-\boldsymbol{W}_i^k, \tag{3.14}$$

其中, $B(\boldsymbol{\Lambda}_i)=\boldsymbol{W}_i^k-t(\nabla H(\boldsymbol{W}_i^k)-\boldsymbol{W}_i^k\boldsymbol{\Lambda}_i)$, 以及 $\mathrm{prox}_{2,1}$ 是与 $\ell_{2,1}$ 范数相关的近端算子. 将 $\boldsymbol{D}_i(\boldsymbol{\Lambda}_i)$ 代入切空间约束条件, 式 (3.13) 可简化为求解如下方程组的根

$$Q(\boldsymbol{\Lambda}_i)=\boldsymbol{D}_i(\boldsymbol{\Lambda}_i)^T\boldsymbol{W}_i^k+(\boldsymbol{W}_i^k)^T\boldsymbol{D}_i(\boldsymbol{\Lambda}_i)=0. \tag{3.15}$$

受文献[48] 启发, 采用半光滑牛顿 (semi-smooth Newton, SSN) 方法求解上述方程组, 以保证寻找最优下降方向的快速收敛性.

3.3.2 更新 S_i

固定其他变量, S_i 的子问题可简化为

$$\min_{S_i} \quad \alpha\|S_i\|_1 + \frac{\mu}{2}\|S_i - M_i^k\|_F^2, \tag{3.16}$$

其中, $M_i^k = X_i - U_i^k + \Lambda_i^k/\mu$ 为迭代中间变量. 该子问题可通过软阈值操作求解, 其解析形式为

$$S_i^{k+1} = \mathrm{sgn}(M_i^k) \odot \max(|M_i^k| - \alpha/\mu, 0), \tag{3.17}$$

其中, $\odot$ 表示哈达玛积 (Hadamard product), 即逐元素乘积, $\mathrm{sgn}(\cdot)$ 为逐元素符号函数.

3.3.3 更新 U_i

当 W_i 与 S_i 更新后, 计算

$$\min_{U_i} \quad \|(I - W_i^{k+1}(W_i^{k+1})^T)U_i\|_F^2 + \frac{\mu}{2}\|U_i - (X_i - S_i^{k+1} + \Lambda_i^k/\mu)\|_F^2. \tag{3.18}$$

对目标函数关于 U_i 求导并令导数为零, 可得如下线性方程组

$$2(I - W_i^{k+1}(W_i^{k+1})^T)U_i + \mu(U_i - (X_i - S_i^{k+1} + \Lambda_i^k/\mu)) = 0. \tag{3.19}$$

令 $Y_i^k = X_i - S_i^{k+1} + \Lambda_i^k/\mu$, 为加速矩阵求逆计算, 利用 Sherman-Morrison-Woodbury 公式[49] 对上述线性方程组求解, 最终得到 U_i 的解析形式为

$$U_i^{k+1} = \left(\frac{\mu}{\mu+2}I + \frac{2}{\mu+2}W_i^{k+1}(W_i^{k+1})^T\right)Y_i^k. \tag{3.20}$$

3.3.4 更新 V

作为全局共识变量, V 的子问题为

$$\min_{V} \quad \sum_{i=1}^{d}\frac{\nu}{2}\|W_i^{k+1} - V + \Pi_i^k/\nu\|_F^2, \tag{3.21}$$

其解析形式为

$$V^{k+1} = \frac{1}{d}\sum_{i=1}^{d}(W_i^{k+1} + \Pi_i^k/\nu). \tag{3.22}$$

结合文献[50], 可进一步化简为

$$V^{k+1} = \frac{1}{d}\sum_{i=1}^{d}W_i^{k+1}. \tag{3.23}$$

算法 1 求解式 (3.5) 的交替方向乘子法

输入: 数据 $\{\boldsymbol{X}_i\}$, 参数 α, β, μ, ν, γ, ε, m, t, 最大迭代次数 $K_{\max}$, $T_{\max}$

初始化: 令 $k=0$, 取 $(\{\boldsymbol{W}_i^0\},\{\boldsymbol{S}_i^0\},\{\boldsymbol{U}_i^0\},\boldsymbol{V}^0,\{\boldsymbol{\Lambda}_i^0\},\{\boldsymbol{\Pi}_i^0\})$, $i\in[d]$

当 $k<K_{\max}$ **时**

for all $i\in[d]$ **do**
  令 $j=0$, 初始化 $\boldsymbol{W}_i^0=\boldsymbol{W}_i^k$
  while $j<T_{\max}$ **do**
    根据式 (3.12) 计算欧氏梯度 $\nabla h(\boldsymbol{W}_i^j)$
    通过 SSN 方法求解式 (3.15) 得到下降方向 $\boldsymbol{D}_i^j$
    令 $\alpha=1$, 通过回溯线搜索确定步长 α
    更新 $\boldsymbol{W}_i^{j+1}=\mathrm{Retr}_{\boldsymbol{W}_i^j}(\alpha\boldsymbol{D}_i^j)$
    令 $j=j+1$
  end while
  令 $\boldsymbol{W}_i^{k+1}=\boldsymbol{W}_i^j$
  根据式 (3.17) 更新 $\boldsymbol{S}_i^{k+1}$
  根据式 (3.20) 更新 $\boldsymbol{U}_i^{k+1}$
end for
根据式 (3.23) 更新全局变量 $\boldsymbol{V}^{k+1}$
for all $i\in[d]$ **do**
  根据式 (3.24) 更新对偶变量 $\boldsymbol{\Lambda}_i^{k+1}$ 与 $\boldsymbol{\Pi}_i^{k+1}$
end for
令 $k=k+1$

结束循环

输出: 全局变量 $\boldsymbol{V}$

3.3.5 更新 $\boldsymbol{\Lambda}_i$ 和 $\boldsymbol{\Pi}_i$

拉格朗日乘子 $\boldsymbol{\Lambda}_i$ 和 $\boldsymbol{\Pi}_i$ 的更新形式为

$$\begin{aligned}\boldsymbol{\Lambda}_i^{k+1}&=\boldsymbol{\Lambda}_i^k+\mu(\boldsymbol{X}_i-\boldsymbol{S}_i^{k+1}-\boldsymbol{U}_i^{k+1}),\\ \boldsymbol{\Pi}_i^{k+1}&=\boldsymbol{\Pi}_i^k+\nu(\boldsymbol{W}_i^{k+1}-\boldsymbol{V}^{k+1}).\end{aligned}\tag{3.24}$$

综合上述各变量的更新步骤, 完整的迭代流程如算法 1 所示.

3.3.6 收敛性分析

定理 3.1 设 $(\{\boldsymbol{W}_i^k\},\{\boldsymbol{S}_i^k\},\{\boldsymbol{U}_i^k\},\boldsymbol{V}^k,\{\boldsymbol{\Lambda}_i^k\},\{\boldsymbol{\Pi}_i^k\})$ 为算法 1 生成的迭代序列, 则其增广拉格朗日函数 $L(\{\boldsymbol{W}_i^k\},\{\boldsymbol{S}_i^k\},\{\boldsymbol{U}_i^k\},\boldsymbol{V}^k,\{\boldsymbol{\Lambda}_i^k\},\{\boldsymbol{\Pi}_i^k\})$ 是单调递减的.

证明 根据文献[47], 式 (3.9) 对应的目标值是单调不增的, 因此

$$L_i(\boldsymbol{W}_i^{k+1}, \boldsymbol{S}_i^k, \boldsymbol{U}_i^k, \boldsymbol{V}^k, \boldsymbol{\Lambda}_i^k, \boldsymbol{\Pi}_i^k) \leq L_i(\boldsymbol{W}_i^k, \boldsymbol{S}_i^k, \boldsymbol{U}_i^k, \boldsymbol{V}^k, \boldsymbol{\Lambda}_i^k, \boldsymbol{\Pi}_i^k),\ \forall i \in [d]. \tag{3.25}$$

由于变量 $\boldsymbol{S}_i$ 的子问题均为凸函数, 最小化步骤满足

$$L_i(\boldsymbol{W}_i^{k+1}, \boldsymbol{S}_i^{k+1}, \boldsymbol{U}_i^k, \boldsymbol{V}^k, \boldsymbol{\Lambda}_i^k, \boldsymbol{\Pi}_i^k) \leq L_i(\boldsymbol{W}_i^{k+1}, \boldsymbol{S}_i^k, \boldsymbol{U}_i^k, \boldsymbol{V}^k, \boldsymbol{\Lambda}_i^k, \boldsymbol{\Pi}_i^k),\ \forall i \in [d]. \tag{3.26}$$

同理, 变量 $\boldsymbol{U}_i$ 的子问题均为强凸函数, 有

$$L_i(\boldsymbol{W}_i^{k+1}, \boldsymbol{S}_i^{k+1}, \boldsymbol{U}_i^{k+1}, \boldsymbol{V}^k, \boldsymbol{\Lambda}_i^k, \boldsymbol{\Pi}_i^k) < L_i(\boldsymbol{W}_i^{k+1}, \boldsymbol{S}_i^{k+1}, \boldsymbol{U}_i^k, \boldsymbol{V}^k, \boldsymbol{\Lambda}_i^k, \boldsymbol{\Pi}_i^k),\ \forall i \in [d]. \tag{3.27}$$

此外, 关于变量 $\boldsymbol{V}$ 有

$$\begin{aligned} &L(\{\boldsymbol{W}_i^{k+1}\}, \{\boldsymbol{S}_i^{k+1}\}, \{\boldsymbol{U}_i^{k+1}\}, \boldsymbol{V}^{k+1}, \{\boldsymbol{\Lambda}_i^k\}, \{\boldsymbol{\Pi}_i^k\}) \\ &< L(\{\boldsymbol{W}_i^{k+1}\}, \{\boldsymbol{S}_i^{k+1}\}, \{\boldsymbol{U}_i^{k+1}\}, \boldsymbol{V}^k, \{\boldsymbol{\Lambda}_i^k\}, \{\boldsymbol{\Pi}_i^k\}). \end{aligned} \tag{3.28}$$

关于拉格朗日乘子 $\boldsymbol{\Lambda}_i, \boldsymbol{\Pi}_i$ 有

$$\begin{aligned} &L_i(\boldsymbol{W}_i^{k+1}, \boldsymbol{S}_i^{k+1}, \boldsymbol{U}_i^{k+1}, \boldsymbol{V}^{k+1}, \boldsymbol{\Lambda}_i^{k+1}, \boldsymbol{\Pi}_i^{k+1}) \\ &\leq L_i(\boldsymbol{W}_i^{k+1}, \boldsymbol{S}_i^{k+1}, \boldsymbol{U}_i^{k+1}, \boldsymbol{V}^{k+1}, \boldsymbol{\Lambda}_i^{k+1}, \boldsymbol{\Pi}_i^k) \\ &\leq L_i(\boldsymbol{W}_i^{k+1}, \boldsymbol{S}_i^{k+1}, \boldsymbol{U}_i^{k+1}, \boldsymbol{V}^{k+1}, \boldsymbol{\Lambda}_i^k, \boldsymbol{\Pi}_i^k),\ \forall i \in [d]. \end{aligned} \tag{3.29}$$

综合上述不等式 (3.25)-(3.29), 得到

$$\begin{aligned} &L(\{\boldsymbol{W}_i^{k+1}\}, \{\boldsymbol{S}_i^{k+1}\}, \{\boldsymbol{U}_i^{k+1}\}, \boldsymbol{V}^{k+1}, \{\boldsymbol{\Lambda}_i^{k+1}\}, \{\boldsymbol{\Pi}_i^{k+1}\}) \\ &< L(\{\boldsymbol{W}_i^k\}, \{\boldsymbol{S}_i^k\}, \{\boldsymbol{U}_i^k\}, \boldsymbol{V}^k, \{\boldsymbol{\Lambda}_i^k\}, \{\boldsymbol{\Pi}_i^k\}), \end{aligned} \tag{3.30}$$

这表明生成的增广拉格朗日函数序列是单调递减的, 定理得证.

文献[38] 虽借助 Kurdyka-Lojasiewicz (KL) 函数证明了算法收敛至稳定点, 但其目标函数为光滑函数, 而式 (3.5) 包含非光滑项. 此外, 在迭代更新 $\boldsymbol{W}_i$ 的过程中, 最优解的求解需要满足额外约束条件, 这使得收敛至稳定点的理论证明具有一定的难度. 文献[51] 虽提出了一种具备收敛保障的黎曼交替方向乘子法, 但其理论假设仍不适用于算法 1. 因此, 本章将更为严谨的收敛性分析留作后续研究工作, 下一节将基于真实数据集开展实验, 进而验证算法的实际性能.

3.4 数值实验

本节在入侵检测系统 (intrusion detection system, IDS) 部署场景下, 将所提 FedEP 与多种主流联邦学习方法进行对比, 包括 FedAvg[52]、FedProx[53]、Ditto[54] 及 FedPG[38]. 为保证公平, 实验均在配备 Intel Ultra 9 Processor 285K 处理器、Ubuntu 22.04.4 LTS 操作系统、64GB 内存及 NVIDIA RTX 5090 GPU 的服务器上完成. 此外, 所提方法开源代码见链接 https://github.com/xianchaoxiu/FedEP.

3.4.1 实验设置

(1) 数据集

实验部分选择三个真实网络入侵检测数据集进行评估, 分别为 TON-IoT[1], UNSW-NB15[2]及 NSL-KDD[3]. 其中, TON-IoT 包含 49 个特征, 涵盖 DDoS、DoS 及扫描攻击等多种常见网络攻击类型, 经标准化处理后, 得到 114-956 个训练样本与 66,557 个测试样本. UNSW-NB15 由澳大利亚国防学院网络安全研究中心开发, 包含 39 个特征, 融合了真实网络正常流量与合成的现代攻击场景, 其训练集与测试集分别包含 65,000 个和 65,332 个样本. NSL-KDD 数据集包含 34 个特征, 涵盖五类攻击类型 (DoS、Probe、R2L、U2R 及 Normal), 共包含 125,973 个训练样本与 22,544 个测试样本. 各数据集的详细信息如表 3.1 所示.

表 3.1: 所选数据集信息

数据集	特征数	训练样本数	测试样本数	类别数
TON-IoT	49	114,956	66,557	10
UNSW-NB15	39	65,000	65,332	10
NSL-KDD	34	125,973	22,544	5

(2) 评估指标

为评估对比方法的检测性能, 本章采用五种通用的指标. 定义真正例 (true positive, TP)、真负例 (true negative, TN)、假正例 (false positive, FP) 及假负例 (false negative, FN).

- 准确率 (accuracy, ACC): 正确分类的比例, 定义为

$$\mathrm{ACC} = \frac{\mathrm{TP} + \mathrm{TN}}{\mathrm{TP} + \mathrm{TN} + \mathrm{FP} + \mathrm{FN}}. \tag{3.31}$$

- 精确率 (precision, PRE): 正确识别的攻击占所有预测攻击的比例, 定义为

$$\mathrm{PRE} = \frac{\mathrm{TP}}{\mathrm{TP} + \mathrm{FP}}. \tag{3.32}$$

- 召回率 (recall, REC): 正确检测的攻击占所有实际攻击的比例, 定义为

$$\mathrm{REC} = \frac{\mathrm{TP}}{\mathrm{TP} + \mathrm{FN}}. \tag{3.33}$$

- 假负率 (false negative rate, FNR): 异常被错误分类为正常的比例, 定义为

$$\mathrm{FNR} = \frac{\mathrm{FN}}{\mathrm{TP} + \mathrm{FN}}. \tag{3.34}$$

- F1 分数 (F1): 精确率与召回率的调和平均值, 定义为

[1]https://research.unsw.edu.au/projects/toniot-datasets

[2]https://research.unsw.edu.au/projects/unsw-nb15-dataset

[3]https://www.unb.ca/cic/datasets/nsl.html

$$\mathrm{F1} = \frac{2\mathrm{TP}}{2\mathrm{TP} + \mathrm{FP} + \mathrm{FN}}. \tag{3.35}$$

需要指出, ACC、PRE、REC 及 F1 的值越高, FNR 的值越低, 表明方法的异常检测效果越好. 为直观区分各指标优劣, 在对应指标后分别标注 ↑ 或 ↓.

(3) 预处理

在实际 IDS 部署中, 本地物联网设备通过网关连接, 以实现数据收集与异常检测的本地化处理. 每个数据集的训练样本均基于 dst_bytes 特征划分为 20 个非独立同分布子集. 所有数据集的特征均采用 z 分数归一化方法进行预处理, 以消除特征量纲差异对模型训练的影响. 所有联邦学习方法均在基于主成分分析的异常检测框架下实现. 每个客户端学习正常流量的低维子空间, 并根据重构误差识别异常数据. FedPG 采用官方源代码复现, FedAvg、FedProx 和 Ditto 采用完全相同的实验配置与网络结构, 仅在优化策略上存在差异. 训练过程中, 所有方法采用相同的局部迭代次数与客户端采样比例. 异常检测阈值通过验证集校准, 以确保公平对比. 此外, 所有超参数均通过网格搜索方法进行优化。

3.4.2 性能比较

本节评估了所提 FedEP 与其他对比方法的异常检测性能, 如表 3.2、表 3.3 和表 3.4 所示. 可以看出, FedEP 在三个数据集上均取得了极具竞争力的检测结果. 具体而言, 在 TON-IoT 数据集上, FedEP 的准确率达到 90.48%, F1 分数达到 94.52%, 综合表现优于其他对比方法. FedProx 在 UNSW-NB15 与 NSL-KDD 数据集上取得了最低的假负率, 体现了其在稳定本地训练方面的优势. 与此同时, 相较于 Ditto 和 FedPG, 所提 FedEP 能够在各项评价指标间实现有效平衡, 展现了该方法的鲁棒性.

表 3.2: TON-IoT 数据集上的检测性能对比 (%)

指标	FedAvg	FedProx	Ditto	FedPG	FedEP
ACC ↑	88.88	88.06	89.07	88.89	**90.48**
PRE ↑	91.93	91.44	91.03	90.84	**92.48**
REC ↑	95.28	94.83	96.66	**96.67**	96.65
FNR ↓	4.72	5.17	3.34	**3.33**	3.35
F1 ↑	93.57	93.10	93.76	93.66	**94.52**

图 3.2 给出了对比方法的接收者操作特征 (receiver operating characteristic, ROC) 曲线及对应的曲线下面积 (area under the curve, AUC) 值. 结果显示, FedEP 在三个数据集上均取得了更高的 AUC 值, 即 TON-IoT 数据集上 AUC 值为 0.8305、UNSW-NB15 数据集上为 0.8632、NSL-KDD 数据集上为 0.8899, 分别高于 FedPG 对应的 0.8132、0.8483 和 0.8851. 这表明 FedEP 在不同网络环境下, 能够更好地权衡真正例率与假正例率. 此外, 图 3.3 展示了对比方法准确率随全局通信轮次的演化过程. 结果表明, 在三个数据集上, FedEP 均能在前 10 个通信轮次内达到

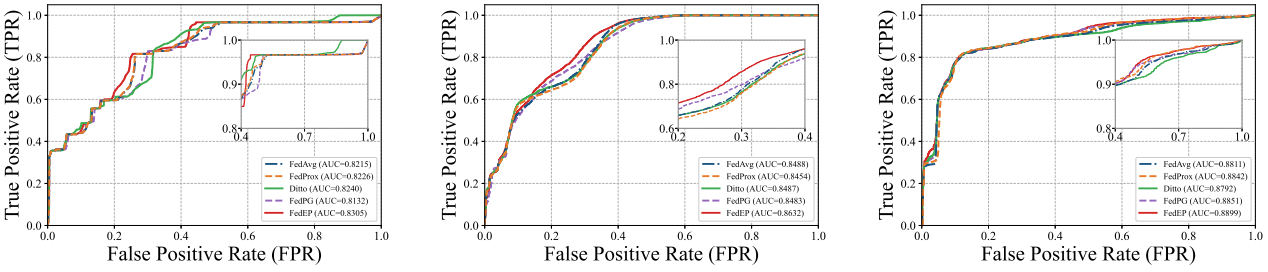
表 3.3: UNSW-NB15 数据集上的检测性能对比 (%)

指标	FedAvg	FedProx	Ditto	FedPG	FedEP
ACC ↑	82.84	82.55	82.56	80.93	**83.31**
PRE ↑	**82.89**	80.38	82.95	81.58	82.17
REC ↑	94.84	**99.02**	94.25	93.66	97.00
FNR ↓	5.16	**0.98**	5.75	6.34	3.00
F1 ↑	88.46	88.73	88.24	87.20	**88.97**

表 3.4: NSL-KDD 数据集上的检测性能对比 (%)

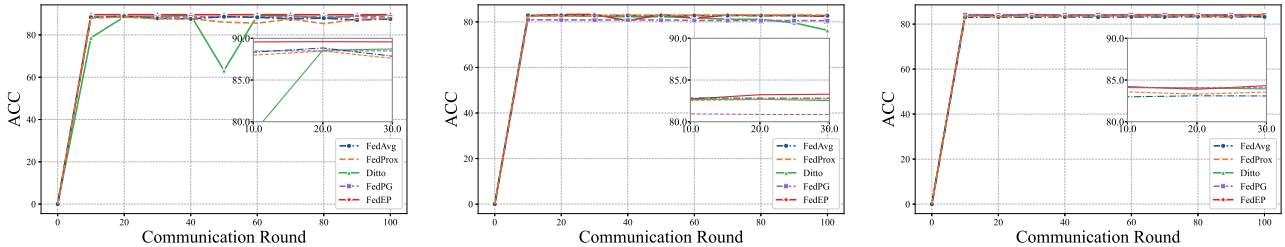
指标	FedAvg	FedProx	Ditto	FedPG	FedEP
ACC ↑	82.99	83.30	83.50	84.09	**84.24**
PRE ↑	86.07	86.55	86.98	**89.78**	89.66
REC ↑	83.66	**83.67**	83.53	81.30	81.75
FNR ↓	16.34	**16.33**	16.47	18.70	18.25
F1 ↑	84.85	85.09	85.22	85.33	**85.52**

峰值准确率. 这证实了 FedEP 中个性化的有效性, 使全局模型能够更好地针对局部分布进行微调, 从而缩减联邦物联网系统中达到最优性能所需的轮次数.



(a) TON-IoT **(b)** UNSW-NB15 **(c)** NSL-KDD

图 3.2: 各数据集上的 ROC 曲线



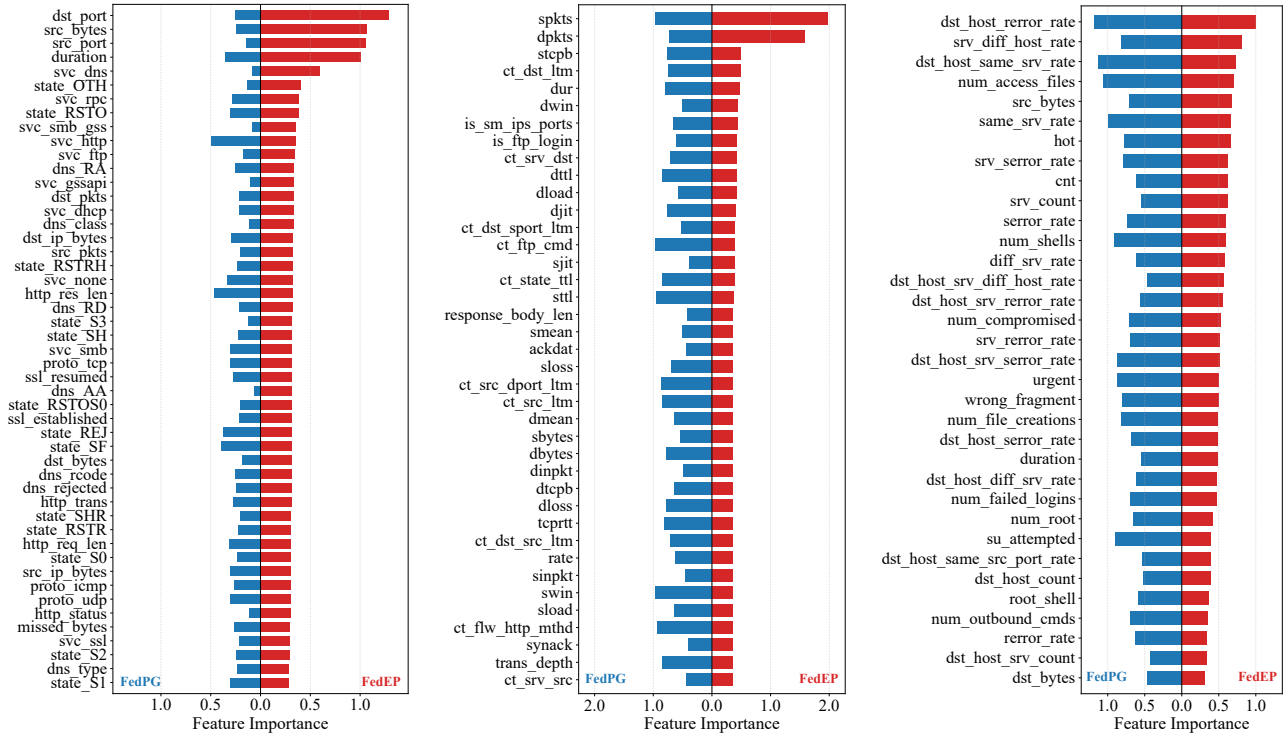
(a) TON-IoT **(b)** UNSW-NB15 **(c)** NSL-KDD

图 3.3: 各数据集上的通信轮次的准确率

3.4.3 个性化分析

(1) 特征重要性

为探究所提稀疏正则对模型可解释性的影响, 图 3.4 可视化了特征重要性. 如红色分布结果所示, FedEP 具备高度集中化的权重分配特性, 仅少量关键特征被赋予显著权重系数. 反观蓝色曲线对应的 FedPG, 其对大量次要冗余特征均分配了不可忽略的权重. 该结果说明, FedEP 能够有效抑制噪声影响, 这对于防止异构物联网环境中的过拟合至关重要.



(a) TON-IoT **(b)** UNSW-NB15 **(c)** NSL-KDD

图 3.4: 各数据集上的特征重要性可视化结果

(2) 消融实验

本节开展如下消融实验, 包括 (I) 同时去除个性化项 $\boldsymbol{S}_i$ 和特征提取项 $\|\boldsymbol{W}_i\|_{2,1}$; (II) 去除特征提取项 $\|\boldsymbol{W}_i\|_{2,1}$; (III) 去除个性化项 $\boldsymbol{S}_i$; (IV) 将 $\|\boldsymbol{W}_i\|_{2,1}$ 范数替换为 $\|\boldsymbol{W}_i\|_1$ 范数; (V) 所提 FedEP, 即式 (3.5).

不同数据集上的消融实验结果分别如表 4.3、表 3.6 和表 3.7 所示. 由表 4.3 可知, 在 TON-IoT 数据集上, 情形 V 的准确率达到 90.48%, F1 分数为 94.52%, 性能全面优于其余对比情形. 尽管在该数据集上性能提升幅度有限, 但一致的性能增益验证了误差分解与结构稀疏约束的有效性. 由表 3.6 可知, 在 UNSW-NB15 数据集上, 尽管情形 I 与情形 III 的召回率高达 100.00%, 但存在显著的过度预测问题, 导致其准确率与精确率表现不佳. 采用 ℓ_1 范数的情形 IV 虽能在一定程度上改善该问题, 但综合表现仍弱于情形 V. 这表明具有行稀疏性质的 $\ell_{2,1}$ 范数能够更有效地筛选冗余特征并抑制噪声干扰. 由表 3.7 可知, 在 NSL-KDD 数据集上, 情形 V 依旧保持

最高的准确率与 F1 分数. 综上所述，所提 FedEP 能够很好地适用于异构物联网场景下的联邦异常检测任务。

表 3.5: TON-IoT 数据集上的消融实验 (%)

指标	I	II	III	IV	V
ACC ↑	89.63	89.63	89.71	89.58	**90.48**
PRE ↑	91.60	91.60	91.68	91.55	**92.48**
REC ↑	96.66	96.66	96.66	96.66	**96.67**
FNR ↓	3.34	3.34	3.34	3.34	**3.33**
F1 ↑	94.06	94.06	94.11	94.04	**94.52**

表 3.6: UNSW-NB15 数据集上的消融实验 (%)

指标	I	II	III	IV	V
ACC ↑	77.84	81.61	75.74	82.77	**83.31**
PRE ↑	75.79	79.34	74.09	80.65	**82.17**
REC ↑	**100.00**	99.37	**100.00**	98.91	97.00
FNR ↓	**0.00**	0.63	**0.00**	1.09	3.00
F1 ↑	86.23	88.23	85.12	88.85	**88.97**

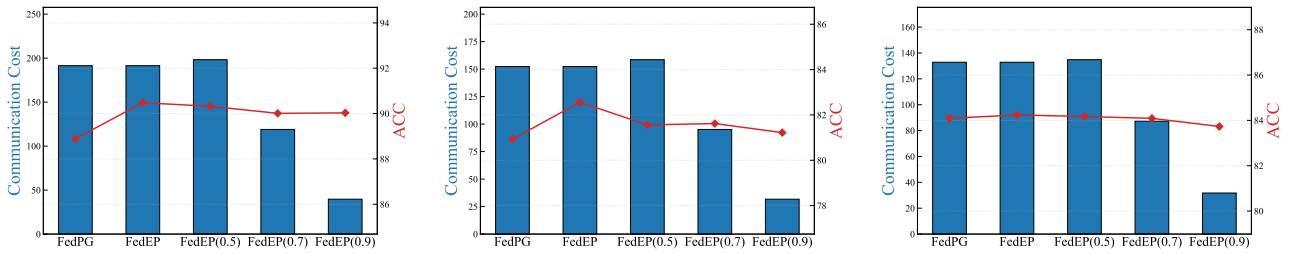
表 3.7: NSL-KDD 数据集上的消融实验 (%)

指标	I	II	III	IV	V
ACC ↑	84.09	84.20	84.12	84.15	**84.24**
PRE ↑	89.56	89.62	**89.66**	89.54	**89.66**
REC ↑	81.56	81.72	81.50	81.70	**81.75**
FNR ↓	18.44	18.28	18.50	18.30	**18.25**
F1 ↑	85.38	85.49	85.39	85.44	**85.52**

3.4.4 效率分析

(1) 通信开销

对于联邦学习方法来说, 通信开销是一个重要的问题. 图 3.5 对比了各模型的通信开销与检测准确率. 其中, FedEP(0.5)、FedEP(0.7) 与 FedEP(0.9) 分别表示将占比为 0.5、0.7、0.9 的特征以零向量形式进行传输. 由图中蓝色柱状结果可以看出, 相较于 FedPG, 所提 FedEP 能够显著降低通信开销. 具体而言, FedEP(0.9) 在三个数据集上均可将通信开销降低 80% 以上. 同时, 红色折线结果表明, 即使在高稀疏度条件下, 模型准确率仍能保持稳定. 因此, 借助 $\ell_{2,1}$ 范数正则, FedEP 实现了通信效率与模型精度之间的权衡.

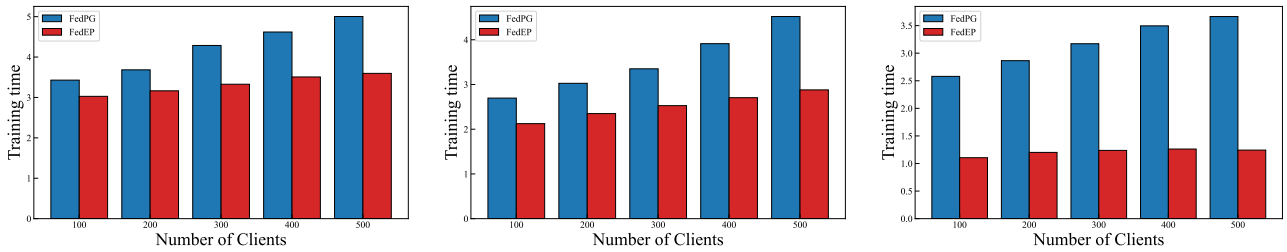


(a) TON-IoT 数据集　**(b)** UNSW-NB15 数据集　**(c)** NSL-KDD 数据集

图 3.5: 各数据集上的通信开销对比

(2) 训练时间

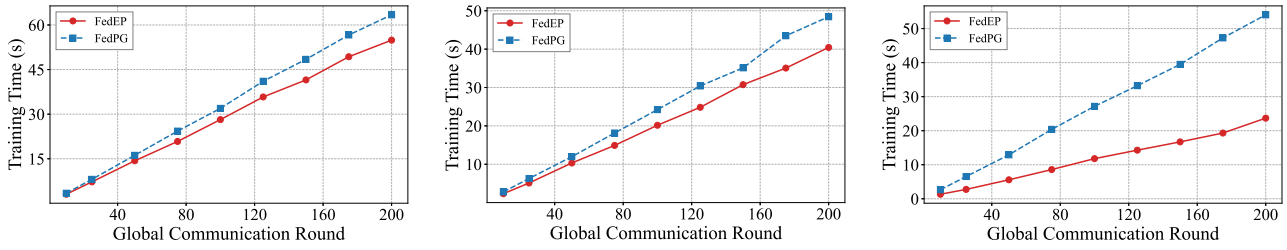
图 3.6 刻画了客户端数量由 100 递增至 500 过程中, 模型单轮训练耗时的变化规律, 以秒为单位. 可以看出, 所提 FedEP 的计算效率显著优于 FedPG, 尤其在 NSL-KDD 数据集上, 训练效率实现近 2 倍的提升.



(a) TON-IoT　**(b)** UNSW-NB15　**(c)** NSL-KDD

图 3.6: 各数据集上的每轮训练时间

图 3.7 给出了累计训练时间的综合比较. 尽管两种方法相对于全局通信轮次均呈线性增长, 但 FedEP 在三个数据集上均表现出更优的效率. 这些结果表明, 所提 FedEP 有效降低了不同网络环境下的计算开销, 使其更适合资源受限的物联网环境.



(a) TON-IoT　**(b)** UNSW-NB15　**(c)** NSL-KDD

图 3.7: 各数据集上的累计训练时间

(3) 秩选择的影响

表 3.8、表 3.9 及表 3.10 展示了秩 $r \in \{5, 10, \ldots, 30\}$ 对异常检测性能的影响. 在所有数据集上, FedEP 均在 $r = 5$ 时取得最优性能, 表明物联网流量的内在结构可通过紧凑的子空间有效捕获. 随着秩参数 r 不断增大, 两种方法的性能均有所下降. 究其原因, 高维子空间易过拟合

噪声和异常值, 从而模糊正常样本与异常样本之间的特征. 然而, FedEP 相比 FedPG 表现出更优的结构稳定性. 以 NSL-KDD 数据集为例, 当 $r = 30$ 时, FedPG 的准确率大幅跌落至 75.62%, 而 FedEP 仍可稳定维持 83.34%. 即便选取次优秩参数, FedEP 依旧能够保持更高的 F1 得分. 原因在于稀疏分量 $\boldsymbol{S}_i$ 可自适应处理数据残余噪声, 防止其扭曲估计的子空间. 综上, FedEP 表现出更强的鲁棒性, 验证了其在建模复杂联邦数据方面的有效性. 关于秩的选择, 未来可设计一种自适应选择策略, 以适配各类不同数据集.

表 3.8: TON-IoT 数据集上的秩选择结果 (%)

秩	ACC ↑		F1 ↑	
r	FedPG	FedEP	FedPG	FedEP
5	88.82	**90.48**	93.63	**94.52**
10	87.72	**89.69**	93.05	**94.09**
15	79.84	**89.74**	87.75	**94.12**
20	87.98	**89.92**	93.18	**94.22**
25	89.10	**89.38**	93.78	**93.93**
30	86.74	**88.42**	92.56	**93.39**

表 3.9: UNSW-NB15 数据集上的秩选择结果 (%)

秩	ACC ↑		F1 ↑	
r	FedPG	FedEP	FedPG	FedEP
5	80.97	**83.22**	87.24	**89.01**
10	**81.19**	77.24	**88.05**	85.91
15	**79.85**	72.88	**86.90**	83.65
20	**82.04**	81.77	87.15	**88.14**
25	**77.81**	76.73	84.54	**85.64**
30	**72.94**	70.07	80.39	**82.26**

表 3.10: NSL-KDD 数据集上的秩选择结果 (%)

秩	ACC ↑		F1 ↑	
r	FedPG	FedEP	FedPG	FedEP
5	84.09	**84.24**	85.33	**85.52**
10	83.42	**83.95**	84.87	**85.23**
15	**83.88**	81.23	**85.13**	83.80
20	**83.48**	83.15	84.80	**85.05**
25	80.99	**84.53**	81.99	**85.97**
30	75.62	**83.34**	75.43	**84.44**

(4) 训练损失

为进一步探究所提 FedEP 的收敛特性，本节在图 3.8 中可视化了三个数据集上多轮通信过程的训练损失曲线. 结果表明, FedEP 在所有数据集上均表现出收敛速度快、收敛过程稳定的特点. 具体来看, 训练损失在初始几轮通信中急剧下降, 并快速收敛至低位平稳状态, 说明模型在训练前期即可学习到有效的特征表征.

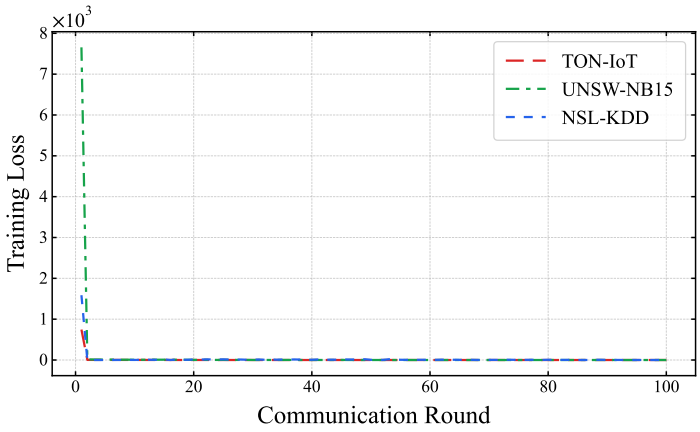


图 3.8: 各数据集上的训练损失

3.5 本章小结

本章针对物联网异常检测问题, 提出了一种高效的个性化联邦主成分分析框架. 与现有施加严格共识约束的联邦主成分分析方法不同, 所提方法不仅允许本地客户端在充分利用全局知识共享优势, 还支持自主维护个性化模型参数, 有效兼顾了全局协同性与本地适配性. 此外, 开发了一种基于交替方向乘子法的高效流形优化算法, 其子问题可通过半光滑牛顿高效求解. 数值实验结果表明, 所提方法相较于 FedAvg、FedProx、Ditto 及 FedPG 等代表性方法, 在检测准确率上实现了显著提升, 同时具备更高的计算效率.

第 4 章 基于稀疏正交非负矩阵分解的故障检测

随着现代工业过程的日益复杂, 数据驱动的故障检测技术已成为保障系统稳定运行的关键手段. 为进一步提升非负矩阵分解在故障检测任务中的性能, 本章提出了结构化联合稀疏正交非负矩阵分解 (structured joint sparse orthogonal NMF, SJSONMF). 该方法将图正则、稀疏约束与正交约束有机融入经典非负矩阵分解框架, 不仅增强了模型的判别能力, 还能有效剔除基向量间的冗余, 从而提升了故障检测结果的可解释性. 更为关键的是, 本章还开发了一种基于近端交替非负最小二乘的优化算法, 并提供了严格的收敛性分析. 最后, 通过田纳西-伊斯曼与实际轴承故障的数值仿真, 验证了所提 SJSONMF 的有效性与优越性.

4.1 引言

故障检测 (fault detection, FD) 是工业过程中的关键环节, 能够保障生产安全, 从而在一定程度上避免经济损失. 与基于模型的故障检测方法相比, 数据驱动的故障检测仅依赖于过程数据, 无需构建复杂的机理模型, 因此更适用于全厂级复杂工业过程. 作为一类主流的数据驱动方法, 多元统计分析已被广泛应用于故障检测领域, 包括主成分分析 (principal component analysis, PCA)、独立成分分析 (independent component analysis, ICA)、偏最小二乘 (partial least squares, PLS)、Fisher 判别分析 (Fisher discriminant analysis, FDA)、典型相关分析 (canonical correlation analysis, CCA) 以及非负矩阵分解 (nonnegative matrix factorization, NMF). 需要指出的是, 主成分分析要求过程数据服从高斯分布, 独立成分分析则要求数据遵循非高斯分布, 而非负矩阵分解除对数据满足非负性要求外无其他严格假设, 因此能够有效处理高斯与非高斯两类过程数据. 目前, 非负矩阵分解已在工业界和学术界引起广泛关注.

非负矩阵分解作为一种强大的降维技术, 旨在为原始数据寻求低维子空间表示[55]. 将正则项或约束条件与非负矩阵分解相结合, 能够显著提升矩阵分解的性能, 并增强局部特征的识别能力. 例如, 图非负矩阵分解 (graph NMF, GNMF)[56] 通过引入图拉普拉斯矩阵, 进而考虑了数据的局部不变性与内部几何特征. 稀疏非负矩阵分解 (sparse NMF, SNMF)[57] 有助于节省大量存储空间并增强局部特征的提取, 以更好地表示原始数据的特征. 正交非负矩阵分解 (orthogonal NMF, ONMF)[58] 能够有效避免成分重叠问题. 这些非负矩阵分解变体已在机器学习、模式识别等领域得到充分验证, 展现出良好的应用前景. 此外, 已有研究提出稀疏正交正则化联合非负矩阵分解 (sparse orthogonality-regularized joint NMF, SOJNMF)[59]. 通过引入稀疏性与正交性, 不仅能够识别多维分子调控模块, 还能在保证系数矩阵稀疏性的同时, 降低多维模块间的特征重叠率.

尽管非负矩阵分解及其变体在多个领域取得了显著进展, 但它们在故障检测领域的应用

尚未得到充分研究. Li 等[60] 首次将非负矩阵分解扩展至非高斯过程的故障检测领域, 在基准田纳西伊斯曼过程 (Tennessee Eastman Process, TEP) 上的数值实验表明, 与主成分分析、独立成分分析相比, 基于非负矩阵分解的方法具有更优的故障检测性能与更广泛的适用性. 此后, 各类基于非负矩阵分解的故障检测方法不断涌现. Li 等[61] 通过将正约束投影与非负矩阵分解相结合, 构建了广义非负矩阵投影 (generalized nonnegative matrix projection, GNMP). Zhai 等 citezhai2018nonlinear 将低维原始数据投影至更高维特征空间, 提出核非负矩阵分解 (kernel NMF, KNMF), 实现了对非线性工业过程的有效处理. Wang 等[62] 利用数据块内的局部信息与块间的全局信息, 提出自适应分区非负矩阵分解 (adaptive partition NMF, APNMF). Ren 等[63] 引入深度自编码器, 构建深度非负矩阵分解 (deep NMF, DNMF), 实现了输入过程数据的自动非线性映射. 最近, Xiu 等[64] 提出基于图拉普拉斯矩阵与联合稀疏性的结构化联合稀疏非负矩阵分解 (structured joint sparse NMF, SJSNMF), 不仅保留了内部几何结构, 还确定了潜在变量的行稀疏性.

基于非负矩阵分解的故障检测方法已展现出良好的应用成效, 但仍可通过以下两方面进一步提升其建模性能. 一方面, 正交性在主成分分析中发挥着关键作用, 不仅能够有效去除数据中的冗余信息, 还能生成更具判别性的结果, 因此将正交性约束引入非负矩阵分解模型有望改善故障检测效果. 另一方面, 现有基于非负矩阵分解求解算法的收敛性尚未得到充分讨论, 而在数值优化领域, 算法的收敛性直接影响数值结果的稳定性, 因此构建具有收敛性保证的算法是提升模型实用性的关键.

受上述分析的启发, 本章基于非负矩阵分解提出了一种有效的数据驱动故障检测模型, 称为结构化联合稀疏正交非负矩阵分解 (structured joint sparse orthogonal NMF, SJSONMF). 本章的主要贡献为

(1) 构建了新的数据驱动的故障检测模型, 通过基矩阵上的正交约束减少基向量之间的相关性, 并通过系数矩阵上的行稀疏约束丢弃系数矩阵行中的不重要信息.

(2) 设计了近端交替非负最小二乘 (proximal alternating nonnegative least squares, PANLS) 算法, 不仅能够保证收敛性, 还能显著提升收敛速度.

(3) 在复杂化工生产过程基准数据集与实际轴承故障数据集上开展数值实验, 验证了所提 SJSONMF 的性能. 尽管两类故障存在许多差异, 但所提方法均表现出优异的检测效果.

4.2 数学模型

4.2.1 非负矩阵分解

设 $\boldsymbol{X} \in \mathbb{R}^{m\times n}$ 为包含 m 个过程变量和 n 个样本的非负过程数据矩阵. 非负矩阵分解旨在寻找非负矩阵 $\boldsymbol{W}$ 和 $\boldsymbol{H}$, 使得 $\boldsymbol{X} \approx \boldsymbol{W}\boldsymbol{H}$, 其数学描述为

$$\min_{\boldsymbol{W},\boldsymbol{H}} \quad \frac{1}{2}\|\boldsymbol{X}-\boldsymbol{W}\boldsymbol{H}\|_F^2, \tag{4.1}$$

其中, $\boldsymbol{W}\in\mathbb{R}_+^{m\times r}$ 为基矩阵, $\boldsymbol{H}\in\mathbb{R}_+^{r\times n}$ 为系数矩阵, 参数 r 需满足 $r\ll\min\{m,n\}$, 用于控制分解的维度与复杂度.

基于经典非负矩阵分解方法, SJSNMF 将数据的几何信息嵌入图拉普拉斯矩阵, 从而有效捕获过程变量间的内在关联, 同时通过引入联合稀疏约束, 实现对过程数据全局结构的学习. 具体地, 数学模型为

$$\begin{aligned} \min_{\boldsymbol{W},\boldsymbol{H}} \quad & \frac{1}{2}\|\boldsymbol{X}-\boldsymbol{W}\boldsymbol{H}\|_F^2+\lambda\mathrm{tr}(\boldsymbol{H}\boldsymbol{L}\boldsymbol{H}^T) \\ \text{s.t.} \quad & \boldsymbol{W}\geq 0,\ \boldsymbol{H}\geq 0,\ \|\boldsymbol{H}\|_{2,0}\leq s, \end{aligned} \tag{4.2}$$

其中, $\lambda>0$ 为正则化参数, $\boldsymbol{L}$ 表示图拉普拉斯矩阵, $s>0$ 为稀疏度参数, 用于控制系数矩阵 $\boldsymbol{H}$ 的非零行数, 进而调整故障变量的数量.

4.2.2 构建模型

现有基于非负矩阵分解的故障检测模型普遍存在一个关键缺陷, 即未考虑基矩阵中不同基向量之间的冗余性. 已有研究表明, 引入正交约束不仅能够得到由不相交分量叠加而成的潜在有效数据表示, 还可显著提升模型的聚类性能[65]. 基于此, 本章构造了 SJSONMF 模型

$$\begin{aligned} \min_{\boldsymbol{W},\boldsymbol{H}} \quad & \frac{1}{2}\|\boldsymbol{X}-\boldsymbol{W}\boldsymbol{H}\|_F^2+\lambda\mathrm{tr}(\boldsymbol{H}\boldsymbol{L}\boldsymbol{H}^T) \\ \text{s.t.} \quad & \boldsymbol{W}\geq 0,\ \boldsymbol{H}\geq 0,\ \boldsymbol{W}^T\boldsymbol{W}=\boldsymbol{I},\ \|\boldsymbol{H}\|_{2,0}\leq s, \end{aligned} \tag{4.3}$$

其中, $\boldsymbol{W}^T\boldsymbol{W}=\boldsymbol{I}$ 是正交约束, $\boldsymbol{I}$ 为单位矩阵. 当式 (4.3) 中移除正交约束 $\boldsymbol{W}^T\boldsymbol{W}=\boldsymbol{I}$ 时, SJSONMF 模型退化为式 (4.2). 若进一步移除联合稀疏约束 $\|\boldsymbol{H}\|_{2,0}\leq s$, 则退化为 GNMF.

正如文献[66] 所述, 非负约束 $\boldsymbol{W}\geq 0$ 和正交约束 $\boldsymbol{W}^T\boldsymbol{W}=\boldsymbol{I}$ 的组合等价于 $\boldsymbol{W}$ 的每一行最多有一个正元素, 且 $\boldsymbol{W}$ 的每一列均满足单位范数约束. 该性质从理论上保证了基矩阵 W 的稀疏性, 同时确保了各基向量之间的非重叠性. 此外, 文献[64] 仅设计了基于交替最小化的优化算法, 未对算法的收敛性进行严格的理论分析. 与之不同, 本章将构建具有收敛性保证的近端交替非负最小二乘算法, 并完成详细的收敛性证明.

4.3 算法设计

由于式 (4.3) 包含两个非凸约束, 即 $\boldsymbol{W}^T\boldsymbol{W}=\boldsymbol{I}$ 与 $\|\boldsymbol{H}\|_{2,0}\leq s$, 且目标函数在变量 $\boldsymbol{W}$ 与 $\boldsymbol{H}$ 上不可分离, 目前尚无快速求解器可直接对其进行求解[67]. 为简化表述, 将目标函数记为

$$f(\boldsymbol{W},\boldsymbol{H})=\frac{1}{2}\|\boldsymbol{X}-\boldsymbol{W}\boldsymbol{H}\|_F^2+\lambda\mathrm{tr}(\boldsymbol{H}\boldsymbol{L}\boldsymbol{H}^T), \tag{4.4}$$

算法 1 求解式 (4.3) 的近端交替非负最小二乘法

输入: 数据 $\boldsymbol{X} \in \mathbb{R}^{m\times n}$, 参数 $\lambda,\ s,\ \tau_1,\ \tau_2$

初始化: 令 $k = 0$, 取 $(\boldsymbol{W}^0, \boldsymbol{H}^0)$

当 未收敛 **时**

1: 更新 $\boldsymbol{W}^{k+1}$, 即求解如下优化问题

$$\begin{aligned} \min_{\boldsymbol{W}} \quad & l(\boldsymbol{W}) = f(\boldsymbol{W}, \boldsymbol{H}^k) + \frac{\tau_1}{2}\|\boldsymbol{W} - \boldsymbol{W}^k\|_F^2 \\ \text{s.t.} \quad & \boldsymbol{W} \geq 0,\ \boldsymbol{W}^T\boldsymbol{W} = \boldsymbol{I} \end{aligned} \tag{4.7}$$

2: 更新 $\boldsymbol{H}^{k+1}$, 即求解如下优化问题

$$\begin{aligned} \min_{\boldsymbol{H}} \quad & q(\boldsymbol{H}) = f(\boldsymbol{W}^{k+1}, \boldsymbol{H}) + \frac{\tau_2}{2}\|\boldsymbol{H} - \boldsymbol{H}^k\|_F^2 \\ \text{s.t.} \quad & \boldsymbol{H} \geq 0,\ \|\boldsymbol{H}\|_{2,0} \leq s \end{aligned} \tag{4.8}$$

结束循环

输出: $(\boldsymbol{W}^{k+1}, \boldsymbol{H}^{k+1})$

分别定义变量 W 和 H 的可行域如下

$$\mathbb{O} = \{\boldsymbol{W} \in \mathbb{R}^{m\times r} \mid \boldsymbol{W} \geq 0,\ \boldsymbol{W}^T\boldsymbol{W} = \boldsymbol{I}\} \tag{4.5}$$

以及 $\mathbb{S}_+ = \mathbb{S} \bigcap \mathbb{R}_+^{r\times n}$, 其中

$$\mathbb{S} = \{\boldsymbol{H} \in \mathbb{R}^{r\times n} \mid \|\boldsymbol{H}\|_{2,0} \leq s\}. \tag{4.6}$$

算法 1 给出了求解式 (4.3) 的算法框架, 其中 $\tau_1, \tau_2 > 0$ 为近端项对应的近端参数. 需要特别说明的是, 近端项的引入不仅能保证算法收敛, 还可有效加快收敛.

4.3.1 更新 W^{k+1}

对于同时包含非负约束与正交约束的优化问题, 即便目标函数连续可微, 现有方法仍相对匮乏. 最近, Jiang 等[66] 通过等价转化优化模型、保留部分约束, 并利用精确罚函数思想简化投影计算, 提出了实用精确罚函数 (practical exact penalty, PEP), 数值表现突出. 式 (4.7) 可转化为如下形式

$$\min_{\boldsymbol{W}\in O\mathcal{B}_+^{m,r}} \quad l(\boldsymbol{W}) \quad \text{s.t.} \quad \|\boldsymbol{W}\boldsymbol{v}\| = 1, \tag{4.9}$$

其中, 函数 $l(\cdot)$ 为算法 1 中式 (4.7) 的目标函数, $O\mathcal{B}_+^{m,r} = O\mathcal{B}^{m,r} \cap \mathbb{R}_+^{m\times r}$, 且

$$O\mathcal{B}^{m,r} = \{\boldsymbol{W} \in \mathbb{R}^{m\times r} \mid \|\boldsymbol{w}_j\| = 1,\ j = 1, \ldots, r\}. \tag{4.10}$$

算法 2 求解式 (4.7) 的实用精确罚函数法

输入: 数据 $\boldsymbol{W}^k, \boldsymbol{H}^k$, 参数 $\gamma,\ \sigma,\ \epsilon_1,\ \epsilon_2$

初始化: 令 $t=0$, 取 $\boldsymbol{W}^0=\boldsymbol{W}^k$

当 未收敛 **时**

1: 采用投影梯度法求解 $\boldsymbol{W}^{t+1}$, 满足

$$\|\min(\boldsymbol{W}^{t+1},\ \mathrm{grad}P_\sigma(\boldsymbol{W}^{t+1}))\|_F \le \epsilon_1,\ P_\sigma(\boldsymbol{W}^{t+1}) \le P_\sigma(\boldsymbol{W}^t) \tag{4.12}$$

2: 检查收敛性: $\|\boldsymbol{W}^{t+1}\boldsymbol{v}\|^2 - 1 \le \epsilon_2$

结束循环

输出: $\boldsymbol{W}^{k+1}=\boldsymbol{W}^{t+1}$

此处 $\boldsymbol{v}$ 可简单取为 $\boldsymbol{e}/\sqrt{r}$, 其中 $\boldsymbol{e}\in\mathbb{R}^r$ 为所有元素均为 1 的列向量. 进一步, 根据文献[66], 通过保留约束 $O\mathcal{B}_+^{m,r}$ 并对约束 $\|\boldsymbol{W}\boldsymbol{v}\|=1$ 施加惩罚, 可得式 (4.9) 的精确罚函数模型

$$\min_{\boldsymbol{W}\in O\mathcal{B}_+^{m,r}} \quad P_\sigma(\boldsymbol{W}) = l(\boldsymbol{W}) + \sigma(\|\boldsymbol{W}\boldsymbol{v}\|^2 - 1). \tag{4.11}$$

式 (4.9) 可通过算法 2 求解. 这里, σ 为罚参数, γ 用于迭代过程中增大 σ, ϵ_1 和 ϵ_2 为容差参数. 此外, $\mathrm{grad}P_\sigma(\boldsymbol{W})$ 为 $P_\sigma(\cdot)$ 在流形 $O\mathcal{B}^{m,r}$ 上点 $\boldsymbol{W}$ 处的黎曼梯度, 即

$$\mathrm{grad}P_\sigma(\boldsymbol{W}) = \nabla P_\sigma(\boldsymbol{W}) - \boldsymbol{W}\mathrm{Diag}(\boldsymbol{W}^T\nabla P_\sigma(\boldsymbol{W})). \tag{4.13}$$

4.3.2 更新 $\boldsymbol{H}^{k+1}$

对于含 ℓ_0 范数稀疏约束与非负约束的向量型非负优化问题, 改进迭代硬阈值 (improved iterative hard thresholding, IIHT)[68] 是一种有效求解方法. 为此, 将 IIHT 算法推广至含 $\ell_{2,0}$ 范数约束与非负约束的矩阵型优化情况, 具体实现如算法 3 所示. 其中,

$$\mathrm{supp}(\boldsymbol{H}^{t+1}) = \{(i,j) \mid H_{ij}^{t+1} \ne 0\}, \tag{4.14}$$

且 $\nabla_{\mathrm{supp}(\boldsymbol{H}^{t+1})}q(\boldsymbol{H}^{t+1})\in\mathbb{R}^{r\times n}$, 如果 $(i,j)\in\mathrm{supp}(\boldsymbol{H}^{t+1})$, 其 (i,j) 元素等于 $\nabla q(\boldsymbol{H}^{t+1})$ 的 (i,j) 元素, 否则等于 0.

4.3.3 收敛性分析

对目标函数 f 关于变量 $\boldsymbol{W}$ 和 $\boldsymbol{H}$ 分别求导, 可得其梯度表达式为

$$\begin{aligned}\nabla_{\boldsymbol{W}} f(\boldsymbol{W},\boldsymbol{H}) &= -(\boldsymbol{X}-\boldsymbol{W}\boldsymbol{H})\boldsymbol{H}^T,\\ \nabla_{\boldsymbol{H}} f(\boldsymbol{W},\boldsymbol{H}) &= -\boldsymbol{W}^T(\boldsymbol{X}-\boldsymbol{W}\boldsymbol{H}) + 2\lambda\boldsymbol{H}\boldsymbol{L}.\end{aligned} \tag{4.16}$$

算法 3 求解式 (4.8) 的改进迭代硬阈值法

输入: 数据 $\boldsymbol{W}^{k+1}, \boldsymbol{H}^k$, 参数 ε, $\beta \in (0,1)$, ρ_0

初始化: 令 $t = 0$, 取 $\boldsymbol{H}^0 = \boldsymbol{H}^k$

当 未收敛 **时**

1: 计算 $\boldsymbol{H}^{t+1} \in \mathcal{P}_{\mathbb{S}_+}(\boldsymbol{H}^t - \rho_t \nabla q(\boldsymbol{H}^t))$, 其中步长 $\rho_t = \rho_0 \beta^{c_t}$, 且 c_t 是满足下式的最小非负整数 c

$$q(\boldsymbol{H}^t(\rho_0\beta^c)) \le q(\boldsymbol{H}^t) - \frac{\rho_0\beta^c}{2}\|\boldsymbol{H}^t(\rho_0\beta^c) - \boldsymbol{H}^t\|_F^2 \tag{4.15}$$

以及 $\boldsymbol{H}^t(\rho) \in \mathcal{P}_{\mathbb{S}_+}(\boldsymbol{H}^t - \rho\nabla q(\boldsymbol{H}^t))$

2: 检查收敛性: $\|\nabla_{\mathrm{supp}(\boldsymbol{H}^{t+1})} q(\boldsymbol{H}^{t+1})\|_F \le \varepsilon$

结束循环

输出: $\boldsymbol{H}^{k+1} = \boldsymbol{H}^{t+1}$

为后续收敛性分析, 设 $\mathcal{N}(\boldsymbol{W}, \mathbb{O})$ 和 $\mathcal{N}(\boldsymbol{H}, \mathbb{S}_+)$ 分别表示 $\mathbb{O}$ 在 $\boldsymbol{W}$ 处的法锥和 $\mathbb{S}_+$ 在 $\boldsymbol{H}$ 处的法锥, $\mathcal{L}(\boldsymbol{W}, \mathbb{O})$ 是 $\mathbb{O}$ 在 $\boldsymbol{W}$ 处的线性化锥, $\mathcal{T}(\boldsymbol{W}, \mathbb{O})$ 是 $\mathbb{O}$ 在 $\boldsymbol{W}$ 处的切锥. 此外, 给定一个锥 $\mathcal{K}$, 用 $\mathcal{K}^o$ 表示 $\mathcal{K}$ 的极锥.

定义 4.1 对于式 (4.3), 若可行点 $(\boldsymbol{W}, \boldsymbol{H})$ 满足

$$-\nabla_{\boldsymbol{W}} f(\boldsymbol{W}, \boldsymbol{H}) \in \mathcal{N}(\boldsymbol{W}, \mathbb{O}),\ \nabla_{\boldsymbol{H}} f(\boldsymbol{W}, \boldsymbol{H}) \in \mathcal{N}(\boldsymbol{H}, \mathbb{S}_+), \tag{4.17}$$

则称 $(\boldsymbol{W}, \boldsymbol{H})$ 为式 (4.3) 的驻点.

定理 4.1 设 $\{(\boldsymbol{W}^k, \boldsymbol{H}^k)\}$ 是由算法 1 生成的序列, 则 $\{f(\boldsymbol{W}^k, \boldsymbol{H}^k)\}$ 非增.

证明 根据算法 1 中式 (4.7) 与 (4.8)的更新规则, 有

$$\begin{aligned} f(\boldsymbol{W}^{k+1}, \boldsymbol{H}^{k+1}) &\le f(\boldsymbol{W}^{k+1}, \boldsymbol{H}^{k+1}) + \frac{\tau_2}{2}\|\boldsymbol{H}^{k+1} - \boldsymbol{H}^k\|_F^2 \\ &\le f(\boldsymbol{W}^{k+1}, \boldsymbol{H}^k) \\ &\le f(\boldsymbol{W}^{k+1}, \boldsymbol{H}^k) + \frac{\tau_1}{2}\|\boldsymbol{W}^{k+1} - \boldsymbol{W}^k\|_F^2 \\ &\le f(\boldsymbol{W}^k, \boldsymbol{H}^k). \end{aligned} \tag{4.18}$$

因此 $\{f(\boldsymbol{W}^k, \boldsymbol{H}^k)\}$ 非增. 若 $(\boldsymbol{W}^{k+1}, \boldsymbol{H}^{k+1}) \ne (\boldsymbol{W}^k, \boldsymbol{H}^k)$, 则严格递减, 即

$$f(\boldsymbol{W}^{k+1}, \boldsymbol{H}^{k+1}) < f(\boldsymbol{W}^k, \boldsymbol{H}^k). \tag{4.19}$$

定理 4.2 设 $\{(\boldsymbol{W}^k, \boldsymbol{H}^k)\}$ 是由算法 1 生成的序列, 则 $\{(\boldsymbol{W}^k, \boldsymbol{H}^k)\}$ 至少存在一个聚点.

证明 注意到式 (4.5) 中 $\boldsymbol{W}$ 的可行域 $\mathbb{O}$ 为紧集, 故 $\{\boldsymbol{W}^k\} \subseteq \mathbb{O}$ 有界, 且 $\boldsymbol{W}^k$ 均为正交矩阵. 以下证明 $\{\boldsymbol{H}^k\}$ 同样有界. 反设 $\{\boldsymbol{H}^k\}$ 无界, 则存在无穷子列 $K \subseteq \{1, 2, \dots\}$, 使得

$$\lim_{k\to\infty,\ k\in K} \|\boldsymbol{H}^k\|_F = +\infty. \tag{4.20}$$

由 $\{\boldsymbol{W}^k\}$ 的有界性, 存在子列 $K_1 \subseteq K$ 满足

$$\lim_{k\to\infty,\ k\in K_1} \boldsymbol{W}^k = \boldsymbol{W}^*, \tag{4.21}$$

其中, $(\boldsymbol{W}^*)^T\boldsymbol{W}^* = \boldsymbol{I}$. 又 $\{\frac{\boldsymbol{H}^k}{\|\boldsymbol{H}^k\|_F}\}$ 有界, 故存在进一步子列 $K_2 \subseteq K_1$, 使得

$$\lim_{k\to\infty,\ k\in K_2} \frac{\boldsymbol{H}^k}{\|\boldsymbol{H}^k\|_F} = \overline{\boldsymbol{H}}, \tag{4.22}$$

其中, $\|\overline{\boldsymbol{H}}\|_F = 1$. 结合式 (4.4) 中 $f(\boldsymbol{W},\boldsymbol{H})$ 的定义及 $\boldsymbol{L}$ 的半正定性, 有

$$\begin{aligned} f(\boldsymbol{W}^k,\boldsymbol{H}^k) &= \frac{1}{2}\|\boldsymbol{X}-\boldsymbol{W}^k\boldsymbol{H}^k\|_F^2 + \lambda\mathrm{tr}(\boldsymbol{H}^k\boldsymbol{L}(\boldsymbol{H}^k)^T) \\ &\geq \frac{1}{2}\|\boldsymbol{X}-\boldsymbol{W}^k\boldsymbol{H}^k\|_F^2. \end{aligned} \tag{4.23}$$

从而

$$\begin{aligned} \lim_{k\to\infty,\ k\in K_2} \frac{f(\boldsymbol{W}^k,\boldsymbol{H}^k)}{\|\boldsymbol{H}^k\|_F^2} &\geq \lim_{k\to\infty,\ k\in K_2} \frac{1}{2}\left\|\frac{\boldsymbol{X}}{\|\boldsymbol{H}^k\|_F} - \boldsymbol{W}^k\frac{\boldsymbol{H}^k}{\|\boldsymbol{H}^k\|_F}\right\|_F^2 \\ &= \frac{1}{2}\|\boldsymbol{W}^*\overline{\boldsymbol{H}}\|_F^2 = \frac{1}{2}\|\overline{\boldsymbol{H}}\|_F^2 = \frac{1}{2}. \end{aligned} \tag{4.24}$$

这意味着

$$\lim_{k\to\infty,\ k\in K_2} f(\boldsymbol{W}^k,\boldsymbol{H}^k) = +\infty, \tag{4.25}$$

这与 $\{f(\boldsymbol{W}^k,\boldsymbol{H}^k)\}$ 有界矛盾. 因此, $\{(\boldsymbol{W}^k,\boldsymbol{H}^k)\}$ 有界, 必至少存在一个聚点.

定理 4.3 设 $\{(\boldsymbol{W}^k,\boldsymbol{H}^k)\}$ 是由算法 1 生成的序列, 则 $\{(\boldsymbol{W}^k,\boldsymbol{H}^k)\}$ 的任意聚点 $(\boldsymbol{W}^*,\boldsymbol{H}^*)$ 均为式 (4.3) 的驻点.

证明 设 $(\boldsymbol{W}^*,\boldsymbol{H}^*)$ 为 $\{(\boldsymbol{W}^k,\boldsymbol{H}^k)\}$ 的任一聚点, 则存在无穷子列 $K \subseteq \{1,2,\dots\}$ 使得

$$\lim_{k\to\infty,\ k\in K} (\boldsymbol{W}^k,\boldsymbol{H}^k) = (\boldsymbol{W}^*,\boldsymbol{H}^*), \tag{4.26}$$

由 $f(\boldsymbol{W},\boldsymbol{H})$ 的连续性, 则

$$\lim_{k\to\infty,\ k\in K} f(\boldsymbol{W}^k,\boldsymbol{H}^k) = f(\boldsymbol{W}^*,\boldsymbol{H}^*). \tag{4.27}$$

结合式 (4.18) 和 $\boldsymbol{L}$ 是半正定矩阵, 得到 $\{f(\boldsymbol{W}^k,\boldsymbol{H}^k)\}$ 非增且有下界, 故收敛. 对式 (4.18) 取极限 $k\to\infty$, 可得

$$\lim_{k\to\infty} \|\boldsymbol{H}^{k+1}-\boldsymbol{H}^k\|_F = 0,\ \lim_{k\to\infty} \|\boldsymbol{W}^{k+1}-\boldsymbol{W}^k\|_F = 0. \tag{4.28}$$

进而

$$\lim_{k\to\infty,\ k\in K} \boldsymbol{H}^{k+1} = \boldsymbol{H}^*,\ \lim_{k\to\infty,\ k\in K} \boldsymbol{W}^{k+1} = \boldsymbol{W}^*. \tag{4.29}$$

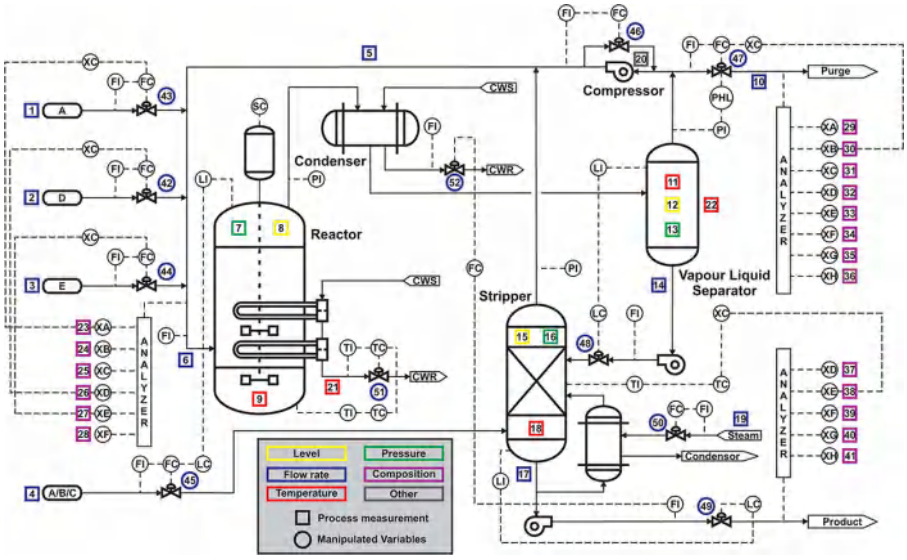


图 4.1: TEP 流程布局图

由于 $\boldsymbol{W}^{k+1}$ 是式 (4.7) 的全局极小点, 从而

$$\begin{aligned} -\nabla l(\boldsymbol{W}^{k+1}) &= -\nabla_{\boldsymbol{W}} f(\boldsymbol{W}^{k+1}, \boldsymbol{H}^k) - \tau_1(\boldsymbol{W}^{k+1} - \boldsymbol{W}^k) \\ &\in \mathcal{L}^o(\boldsymbol{W}^{k+1}, \mathbb{O}). \end{aligned}$$

根据文献[66], 可行集 $\mathbb{O}$ 满足 Guignard 约束规范, 即 $\mathcal{T}^o(\boldsymbol{W}^{k+1}, \mathbb{O}) = \mathcal{L}^o(\boldsymbol{W}^{k+1}, \mathbb{O})$. 因此,

$$-\nabla l(\boldsymbol{W}^{k+1}) \in \mathcal{N}(\boldsymbol{W}^{k+1}, \mathbb{O}). \tag{4.30}$$

另一方面, 式 (4.8) 中的目标函数 $q(\boldsymbol{H})$ 强凸且强光滑. 因 $\boldsymbol{H}^{k+1}$ 为其全局极小点, 故满足最优性条件

$$\mathbf{0} \in \mathcal{P}_{\mathcal{T}(\boldsymbol{H}^{k+1}, \mathbb{S}_+)}(-\nabla q(\boldsymbol{H}^{k+1})). \tag{4.31}$$

该条件等价于

$$\begin{aligned} -\nabla q(\boldsymbol{H}^{k+1}) &= -\nabla_{\boldsymbol{H}} f(\boldsymbol{W}^{k+1}, \boldsymbol{H}^{k+1}) - \tau_2(\boldsymbol{H}^{k+1} - \boldsymbol{H}^k) \\ &\in \mathcal{N}(\boldsymbol{H}^{k+1}, \mathbb{S}_+). \end{aligned} \tag{4.32}$$

对上述最优性条件沿子列取极限, 即知 $(\boldsymbol{W}^*, \boldsymbol{H}^*)$ 为 (4.3) 的驻点.

值得注意的是, 算法 1 中式 (4.7) 与式 (4.8) 引入的近端项在收敛性分析中起到关键作用, 保证了聚点 $(\boldsymbol{W}^*, \boldsymbol{H}^*)$ 为式 (4.3) 的驻点. 同时, 传统交替非负最小二乘算法因可行集的非凸性, 难以直接应用于式 (4.3). 尽管近端技术已在文献[67] 中采用, 但其所考虑的模型未同时包含正交约束与稀疏约束, 结构相对简单.

表 4.1: TEP 数据集上 T^2 统计量对比 (%)

故障	PCA	NMF	GNMF	SNMF	ONMF	KNMF	SJSNMF	SJSONMF
IDV(1)	99.12	99.00	99.38	99.25	99.50	99.12	99.25	**99.88**
IDV(2)	99.38	98.00	97.25	98.62	98.50	98.50	98.00	**98.62**
IDV(3)	0.88	1.88	3.25	2.50	1.12	3.38	1.88	**3.75**
IDV(4)	20.88	78.12	70.25	81.75	78.62	83.00	79.00	**92.12**
IDV(5)	24.12	21.25	23.12	22.38	22.12	23.88	18.00	**24.12**
IDV(6)	99.12	99.38	99.62	99.00	99.88	99.00	99.50	**100**
IDV(7)	**100**	**100**	**100**	**100**	**100**	**100**	**100**	**100**
IDV(8)	**96.88**	89.75	93.12	92.50	89.12	91.12	92.62	93.00
IDV(9)	1.75	2.50	1.88	2.12	1.12	0.50	1.62	**2.75**
IDV(10)	29.62	31.62	**36.38**	23.00	32.38	33.62	36.88	25.62
IDV(11)	20.62	56.12	57.12	51.62	53.12	54.37	51.38	**57.88**
IDV(12)	92.38	92.38	86.75	90.62	90.50	90.38	**93.50**	91.75
IDV(13)	93.62	92.50	93.75	91.12	79.12	**94.12**	93.62	**94.12**
IDV(14)	89.25	98.00	94.38	96.00	**99.88**	99.75	**99.88**	**99.88**
IDV(15)	1.38	0.75	4.25	**6.12**	2.00	2.62	2.62	2.88
IDV(16)	13.50	41.12	71.75	43.50	47.88	42.12	57.50	**74.25**
IDV(17)	46.25	49.50	47.12	85.12	81.12	70.38	**88.62**	84.88
IDV(18)	**89.25**	87.38	84.25	88.12	88.00	87.88	88.88	89.12
IDV(19)	1.88	3.00	4.50	**16.00**	7.75	1.38	10.88	12.88
IDV(20)	21.75	35.88	40.12	39.12	41.50	**45.75**	37.62	39.25
IDV(21)	39.25	24.88	34.12	33.12	37.12	42.15	35.00	**47.50**
平均	51.47	57.29	59.16	60.08	59.54	60.14	61.25	**63.54**

4.4 数值实验

本节通过数值实验探讨所提 SJSONMF 在基准数据集上的有效性和优越性, 对比方法包括 NMF[60]、GNMF[56]、SNMF[57]、ONMF[58]、KNMF[69] 及 SJSNMF[64]. 同时引入 PCA 作为对比, 进一步凸显基于非负矩阵分解的有效性. 此外, 所提方法开源代码见链接 https://github.com/xianchaoxiu/SJSONMF.

4.4.1 实验设置

(1) 诊断策略

对于新采集的样本 $\boldsymbol{X}_{\text{new}} \in \mathbb{R}^{m\times n}$, 矩阵 $\boldsymbol{H}$ 的重构量 $\hat{\boldsymbol{H}}$ 可表示为

$$\hat{\boldsymbol{H}} = (\boldsymbol{W}^T\boldsymbol{W})^{-1}\mathbf{W}^T\boldsymbol{X}_{\text{new}} = \boldsymbol{W}^T\boldsymbol{X}_{\text{new}}. \tag{4.33}$$

表 4.2: TEP 数据集上 SPE 统计量对比 (%)

故障	PCA	NMF	GNMF	SNMF	ONMF	KNMF	SJSNMF	SJSONMF
IDV(1)	99.25	99.38	98.75	99.12	99.38	99.25	99.38	**99.62**
IDV(2)	95.75	95.12	97.62	98.38	98.25	98.50	**98.62**	98.38
IDV(3)	2.62	1.38	1.25	1.62	1.62	**4.12**	2.12	2.25
IDV(4)	**100**	90.75	94.88	92.12	66.00	84.12	87.88	86.88
IDV(5)	20.88	22.75	22.25	21.50	26.25	**31.00**	21.50	29.50
IDV(6)	**100**	**100**	**100**	**100**	99.75	98.88	**100**	**100**
IDV(7)	**100**	**100**	**100**	99.88	**100**	**100**	**100**	**100**
IDV(8)	83.62	96.62	**97.12**	95.12	94.50	95.88	93.38	96.75
IDV(9)	**1.75**	1.25	1.25	**1.75**	0.88	0.88	**1.75**	1.38
IDV(10)	25.75	39.25	35.00	37.00	38.38	**50.12**	48.62	44.75
IDV(11)	**74.88**	64.62	60.25	63.25	44.25	60.12	57.88	59.75
IDV(12)	**98.50**	89.62	92.88	94.75	93.88	89.38	92.88	90.00
IDV(13)	**95.25**	91.75	94.12	89.75	90.88	93.12	91.75	94.25
IDV(14)	98.88	99.88	94.88	99.88	99.88	97.75	99.88	**100**
IDV(15)	3.00	1.62	1.38	2.88	3.25	**6.50**	1.50	5.50
IDV(16)	27.38	38.25	61.88	40.50	**62.25**	39.00	52.00	57.00
IDV(17)	55.38	64.12	43.88	57.38	81.62	75.50	**88.50**	82.75
IDV(18)	90.12	89.62	85.75	**90.50**	88.75	87.50	89.88	89.25
IDV(19)	3.52	10.88	6.50	7.00	9.75	0.50	10.25	**11.88**
IDV(20)	29.75	36.00	37.38	44.88	**62.00**	47.50	43.25	50.25
IDV(21)	47.25	26.38	37.12	37.00	29.38	42.25	46.12	**57.25**
平均	59.55	59.96	60.20	60.68	61.47	61.99	63.20	**64.64**

基于非负矩阵分解的检测模型, 上式可表述为 $\hat{\boldsymbol{X}} = \boldsymbol{W}\hat{\boldsymbol{H}}$. 引入 T^2 统计量与平方预测误差 (squared prediction error, SPE) 统计量, 具体构造如下

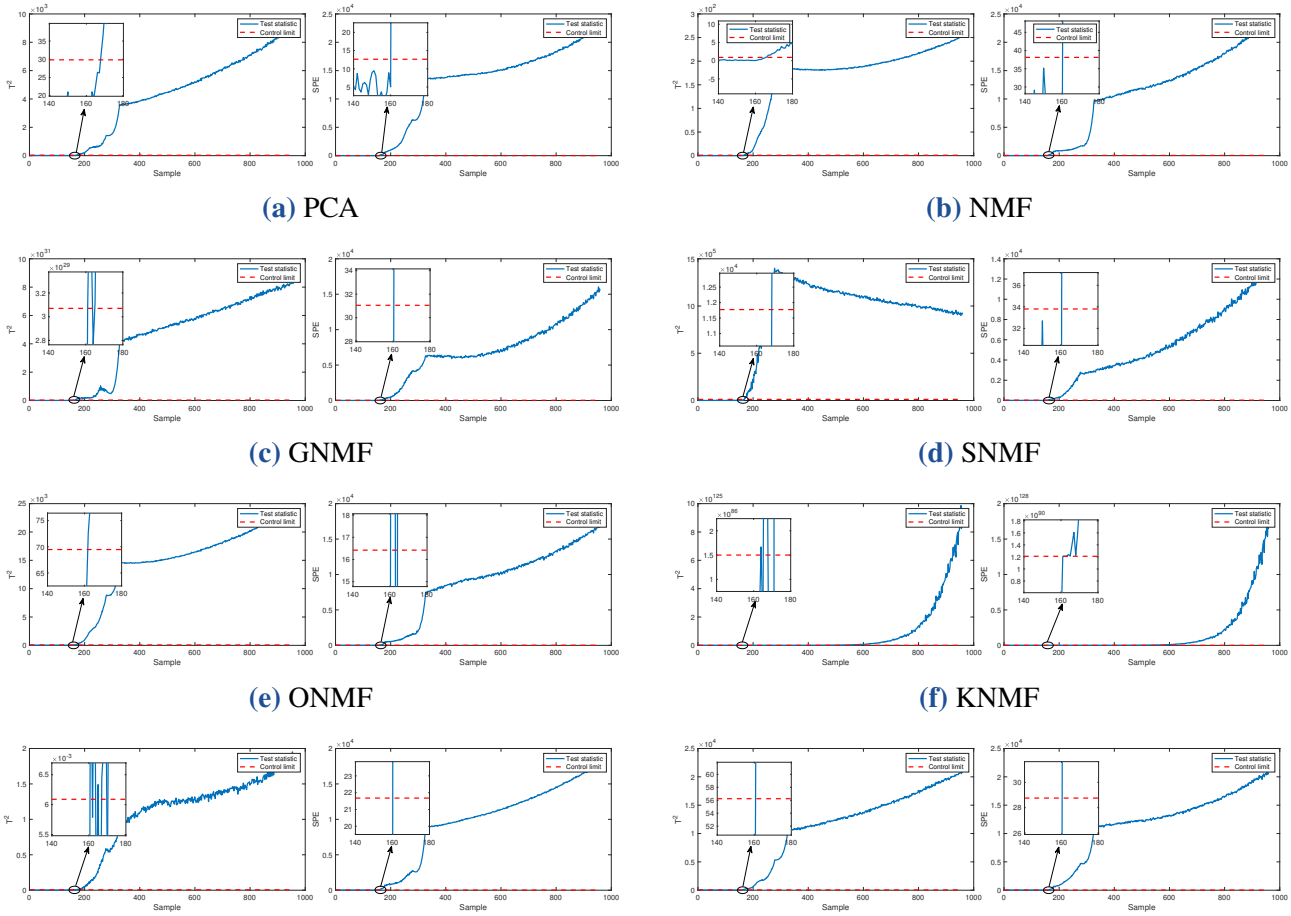
$$\mathrm{T}^2 = \hat{\boldsymbol{H}}^T \hat{\boldsymbol{H}}, \ \mathrm{SPE} = (\boldsymbol{X}_{\text{new}} - \hat{\boldsymbol{X}})^T (\boldsymbol{X}_{\text{new}} - \hat{\boldsymbol{X}}). \tag{4.34}$$

需要说明的是, T^2 统计量与 SPE 统计量的控制限无法通过特定的近似分布直接确定. 因此, 本章采用核密度估计 (kernel density estimation, KDE) 方法估计其概率密度函数，并据此计算两类统计量的控制. 对于给定的置信水平 α, 控制限 J_{th} 满足

$$P(J < J_{\text{th}}) = \int_{-\infty}^{\text{th}} p(J)dJ = \alpha. \tag{4.35}$$

对应于 T^2 统计量与 SPE 统计量的控制限分别记为 $J_{\text{th},\mathrm{T}^2}$ 和 $J_{\text{th,SPE}}$, 其故障检测逻辑为

$$\begin{cases} T^2 < J_{\text{th},\mathrm{T}^2} \text{ 且 } \mathrm{SPE} < J_{\text{th,SPE}} & \Rightarrow \text{ 无故障}, \\ T^2 \geq J_{\text{th},\mathrm{T}^2} \text{ 或 } \mathrm{SPE} \geq J_{\text{th,SPE}} & \Rightarrow \text{ 故障}. \end{cases} \tag{4.36}$$



(a) PCA **(b)** NMF

(c) GNMF **(d)** SNMF

(e) ONMF **(f)** KNMF

(g) SJSNMF **(h)** SJSONMF

图 4.2: 故障 IDV(6) 的检测性能对比

(2) 参数设置

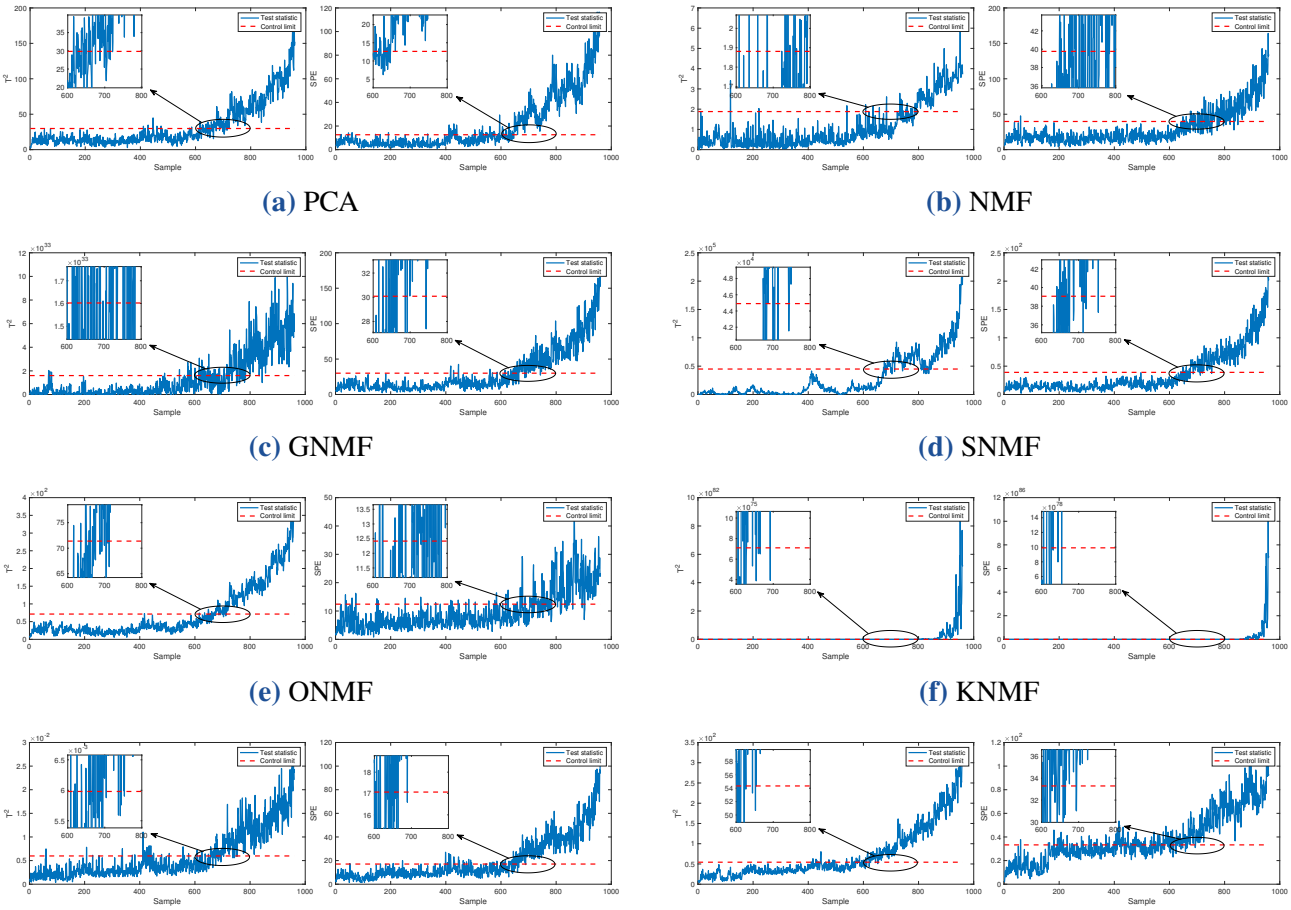
对于所提 SJSONMF, 置信限 α 设置为 0.99, 图拉普拉斯矩阵按照文献[70] 的方法构造, 参数 λ、τ_1 和 τ_2 通过五折交叉验证技术, 在候选参数集 $\{10^{-5}, 10^{-4}, \ldots, 10^{3}\}$ 内基于无故障数据进行寻优确定. 稀疏度水平 s 的取值从较小值开始逐步增大, 直至达到最优故障检测性能. 同时, 算法 1、算法 2 和算法 3 的最大迭代次数均设置为 500, 算法 2 和算法 3 的收敛阈值 ϵ_1、ϵ_2 和 ε 均取 10^{-4}, 算法 1 的停止准则采用相对误差, 具体表达式为

$$\max\left\{\frac{\|\boldsymbol{W}^{k+1}-\boldsymbol{W}^{k}\|_F}{\|\boldsymbol{W}^{k}\|_F}, \frac{\|\boldsymbol{H}^{k+1}-\boldsymbol{H}^{k}\|_F}{\|\boldsymbol{H}^{k}\|_F}\right\} \leq 10^{-5}. \tag{4.37}$$

为保证实验的公平性, 对于其他所有方法, 若涉及可调参数, 均采用相同的五折交叉验证技术进行参数寻优, 最大迭代次数统一设置为 500, 并采用式 (4.37) 作为算法停止准则, 确保所有对比算法的实验条件一致.

(3) 评估指标

本次实验采用故障检测率 (fault detection rate, FDR) 作为评估故障检测性能的指标[71], 以 T^2 统计量为例, 定义如下



(a) PCA (b) NMF

(c) GNMF (d) SNMF

(e) ONMF (f) KNMF

(g) SJSNMF (h) SJSONMF

图 4.3: IDV(21) 的检测性能对比

$$\frac{\text{样本数}\ (\mathrm{T}^2 \geq J_{\mathrm{th},T^2} \mid f \neq 0)}{\text{总样本数}\ (f \neq 0)} \times 100\%. \tag{4.38}$$

显然, 在故障检测任务中, FDR 值越高，表明故障检测性能越好.

4.4.2 TEP 数据集应用

(1) 数据集

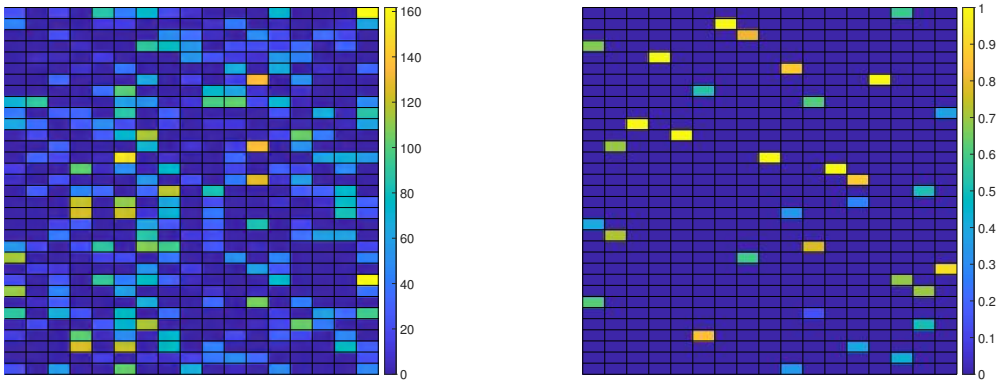
田纳西-伊斯曼过程是一个典型的化工过程, 目前已被广泛应用于各类故障检测方法的性能测试中. 该过程的具体布局详见图 4.1, 详情可参考文献[72]. 在实验中, 采用正常操作条件下的数据进行离线训练, 选取 21 种故障数据开展在线测试.

(2) 实验结果

各对比方法的 T^2 统计量与 SPE 统计量分别见表 4.1 和表 4.2, 其中各指标的最佳性能结果以粗体标注. 由实验结果可知, 基于非负矩阵分解的各类方法, 其统计量平均值均高于 PCA, 这表明非负矩阵分解在故障检测任务中更具有优势. 具体而言, 从表 4.1 中故障 IDV(16) 的测试结果来看, NMF 的 T^2 统计量为 41.12%, 而 SJSONMF 的 T^2 统计量达到 74.25%, 性能提升幅度

达 33.13%. 这说明, 在非负矩阵分解模型中加入合理的约束条件或正则项, 能够有效提升故障检测的性能. 此外, 在 SPE 统计量指标上, ONMF 的表现优于 GNMF 与 SNMF, 验证了正交约束在故障检测中具有更强的潜力. 更为重要的是, SJSNMF 与 SJSONMF 的统计量平均值均高于 KNMF, 且 SJSONMF 在多数故障场景下均取得了最优性能, 进一步表明将图拉普拉斯矩阵、联合稀疏性与正交约束融合到非负矩阵分解框架中, 极具应用前景.

图 4.2 和图 4.3 展示了故障 IDV(6) 与 IDV(21) 的检测结果, 其中红线代表控制限, 蓝线代表 T^2 或 SPE 统计量的实时数值. 由图 4.2 可见, IDV(6) 故障对应的 T^2 与 SPE 统计量的检测率均超过 99%, 表明基于非负矩阵分解的方法能够成功完成该类故障的检测任务. 故障 IDV(21) 涉及一个恒定位置故障, 具体表现为流 4 的阀门固定于稳态位置, 属于一类检测难度较高的故障类型. 从图 4.3 的检测结果可以看出, SJSONMF 在 600-800 区间内能够检测到更多的故障样本, 显著优于其他基于非负矩阵分解的对比方法, 进一步验证了所提方法的优越性。



(a) SJSNMF **(b)** SJSONMF

图 4.4: 正交矩阵可视化对比

(3) 正交性分析

本节旨在从实验计算结果出发, 阐释将正交约束引入到 SJSNMF 的有效性. 图 4.4 对比了 SJSNMF 与 SJSONMF 在 TEP 数据集上学习到的基矩阵. 可以清晰发现, SJSONMF 所获得的基矩阵具有更强的判别性, 其每一行最多仅包含一个正元素, 该特性能够大幅消除不同基向量之间的冗余信息, 从而提升模型的故障检测性能.

4.4.3 轴承数据集应用

(1) 数据集

XJTU-SY 轴承数据集[73] 为轴承全生命周期监测数据集, 涵盖了轴承从正常运行状态到完全失效的全过程, 其对应的实验平台如图 4.5 所示. 该实验测试台专门设计用于滚动轴承在不同操作工况下的加速退化测试, 其中径向载荷通过液压加载系统产生并作用于被测轴承的壳体, 转速则由交流感应电机的速度控制器进行精准设定与稳定维持. 本研究共选取 15 种典型

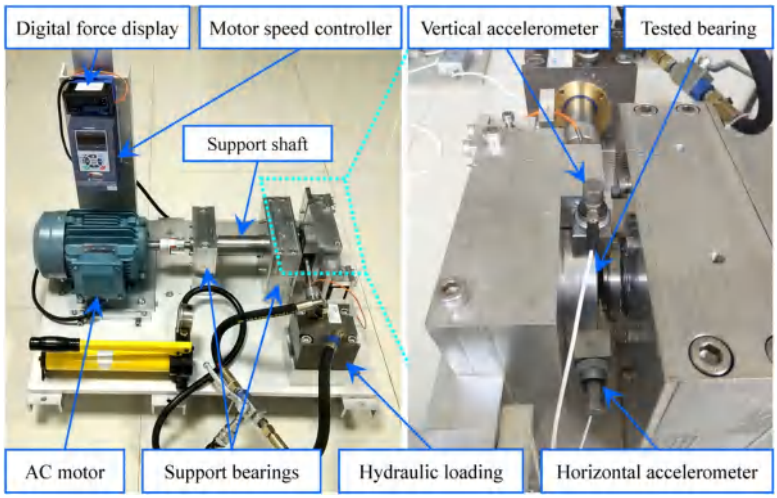


图 4.5: XJTU-SY 轴承数据集的实验平台

故障类型, 如表 4.3 所示, 所有故障均在三种不同操作条件下完成测试. 对于每种故障场景, 训练样本包含 1,000 组正常状态数据, 测试样本包含 200 组正常状态数据与 800 组故障状态数据.

表 4.3: 轴承运行工况

工况	径向载荷 (kN)	转速 (转/分钟)	数据集
1	12	2,100	1_1, 1_2, 1_3, 1_4, 1_5
2	11	2,250	2_1, 2_2, 2_3, 2_4, 2_5
3	10	2,400	3_1, 3_2, 3_3, 3_4, 3_5

(2) 实验结果

各对比方法的 T^2 统计量如表 4.4 所示. 实验结果表明, 对比模型的故障检测率在绝大多数工况下均超过 90%, 所提 SJSONMF 在多数故障场景下取得了最高的故障检测率, 且在所有工况下的平均检测性能最优, 进一步验证了 SJSONMF 的有效性.

4.5 本章小结

本章针对数据驱动故障诊断检测效果不稳定的问题, 提出了基于结构化联合稀疏正交非负矩阵分解的数据驱动故障检测方法. 通过正交约束减少相关性, 利用稀疏约束剔除冗余信息, 进而提升了故障检测的可解释性与可靠性. 同时, 设计了具有严格收敛性证明的优化算法, 为实际应用提供了理论保障. 通过在基准 TEP 数据集与实际 XJTU-SY 轴承数据集上的实验, 将所提方法与现有基于非负矩阵分解的故障检测方法进行对比分析, 验证了所提方法显著提升故障检测性能. 特别地, 该优化技术还可拓展应用于其他各类数据驱动的故障检测方法.

表 4.4: XJTU-SY 轴承上的 T^2 统计量对比 (%)

轴承	PCA	NMF	GNMF	SNMF	ONMF	KNMF	SJSNMF	JSONMF
1_1	92.50	85.88	93.00	78.00	92.38	92.75	91.75	**95.75**
1_2	98.25	94.25	93.88	97.62	92.25	94.75	99.00	**99.25**
1_3	99.33	95.25	95.67	97.88	98.12	97.75	99.12	**99.75**
1_4	39.67	28.75	33.12	**45.62**	22.12	22.75	23.12	44.25
1_5	**100**	92.50	91.50	97.88	94.25	96.25	98.75	99.38
2_1	**100**	89.62	93.38	94.88	99.62	97.25	97.62	97.25
2_2	99.00	98.33	97.50	96.88	96.12	97.00	99.38	**99.62**
2_3	**99.75**	95.25	91.88	96.88	91.88	96.62	97.62	99.50
2_4	**100**	92.25	93.75	89.62	92.50	93.25	95.00	97.88
2_5	**99.88**	88.88	89.00	87.75	88.25	87.75	93.38	96.25
3_1	95.88	80.38	87.33	85.50	77.12	86.62	90.00	**96.50**
3_2	90.88	74.25	77.12	71.50	75.25	83.75	87.38	**94.25**
3_3	90.88	92.75	95.33	93.88	92.88	93.12	94.62	**98.75**
3_4	95.88	91.50	93.38	95.25	93.50	97.12	**98.50**	98.25
3_5	**100**	92.60	89.62	83.62	84.75	87.00	90.38	98.25
平均	93.62	86.16	87.70	87.50	86.07	88.25	90.37	**94.33**

表 4.5: XJTU-SY 轴承上的 SPE 统计量对比 (%)

轴承	PCA	NMF	GNMF	SNMF	ONMF	KNMF	SJSNMF	SJSONMF
1_1	88.00	**100**	**100**	**100**	**100**	**100**	**100**	**100**
1_2	92.00	**100**	**100**	**100**	**100**	**100**	**100**	**100**
1_3	92.00	**100**	**100**	**100**	**100**	**100**	**100**	**100**
1_4	55.67	71.38	76.00	68.62	79.25	75.25	75.50	**79.50**
1_5	99.75	**100**	**100**	**100**	**100**	**100**	**100**	**100**
2_1	99.00	**100**	**100**	**100**	**100**	**100**	**100**	**100**
2_2	92.50	**100**	**100**	**100**	**100**	**100**	**100**	**100**
2_3	95.00	**100**	**100**	**100**	**100**	**100**	**100**	**100**
2_4	99.75	**100**	**100**	**100**	**100**	**100**	**100**	**100**
2_5	99.25	**100**	**100**	**100**	**100**	**100**	**100**	**100**
3_1	88.75	**100**	**100**	**100**	**100**	**100**	**100**	**100**
3_2	94.12	**100**	**100**	**100**	**100**	**100**	**100**	**100**
3_3	94.12	**100**	**100**	**100**	**100**	**100**	**100**	**100**
3_4	91.75	**100**	**100**	**100**	**100**	**100**	**100**	**100**
3_5	99.88	**100**	**100**	**100**	**100**	**100**	**100**	**100**
平均	92.10	98.09	98.40	97.91	98.62	98.35	98.37	**98.63**

第 5 章 基于稀疏张量相关分析的多视角学习

张量典型相关分析凭借其在多视角学习中有效捕获高阶相关结构的特性, 得到学术界的广泛关注. 然而, 现有张量典型相关分析方法普遍忽视个体结构的刻画, 且缺乏算法收敛性保证. 针对上述问题, 本章提出了带多阶图拉普拉斯正则的稀疏张量典型相关分析 (sparse tensor canonical correlation analysis with Laplacian, STCCA-L). 针对该非凸优化模型, 开发了高效的流形近端梯度算法, 并利用半光滑牛顿法求解子问题. 同时, 严格证明了算法的收敛性, 并对其计算复杂度进行了分析. 实验结果表明, 与现有多种代表性多视角学习方法相比, 所提 STCCA-L 具有更优的分类精度和鲁棒性.

5.1 引言

多视角学习旨在应对由不同模态观测同一对象所带来的数据异质性问题[74]. 现有多视角学习方法大致可分为三类: 基于协同训练的方法、多核学习方法以及子空间学习方法. 其中, 基于协同训练的方法通过视角间一致性约束迭代优化分类器, 多核学习方法通过组合核函数实现异构信息融合, 子空间学习方法旨在挖掘多视角数据共享的潜在低维表示. 相较其余两类方法, 多视角子空间学习因能够在捕捉跨视角共识结构的同时保留视角特有信息, 有效提升分类、聚类等下游任务的鲁棒性.

典型相关分析 (canonical correlation analysis, CCA) 是多视角子空间学习中的基础方法[75]. 通过寻找一组视角投影, 使得投影后的低维表示相关性最大化, 典型相关分析能够有效提取跨模态共享的显著判别特征. 矩阵形式的典型相关分析包括惩罚典型相关分析[76]、稀疏典型相关分析 (sparse CCA, SCCA)[77] 以及结构化广义典型相关分析 (structured generalized CCA, SGCCA)[78] 等. 此外, 深度典型相关分析[79] 借助深度神经网络挖掘复杂的数据关系, 在大数据场景下展现出优异的特征学习能力. 然而, 深度模型普遍存在可解释性弱、对大规模标注数据高度依赖等问题, 在一定程度上限制了其应用范围. 与之相对, 张量典型相关分析 (tensor CCA, TCCA)[80] 利用高阶协方差结构, 能够更精细地刻画多视角之间的复杂关系. Luo 等[81] 首次提出张量典型相关分析, 成功捕捉到许多矩阵类方法难以刻画的高阶依赖关系, 在多视角学习任务中取得了更优的性能. 随后, 一系列变体相继被提出, 如图正则张量两视角和多视角典型相关分析 (two-view and multi-view CCA, TMCCA)[82], 并在生物医学等领域得到广泛应用. 但上述张量典型相关分析未对典型向量施加正交约束, 容易导致典型分量冗余或高度相关. 为解决这个问题, Sun 等[83] 提出了正交张量典型相关分析 (tensor CCA with orthogonality, TCCA-O), 保证典型分量之间的无关性.

尽管张量典型相关分析在捕捉高阶相关性方面表现出色, 但其学习到的典型投影矩阵往

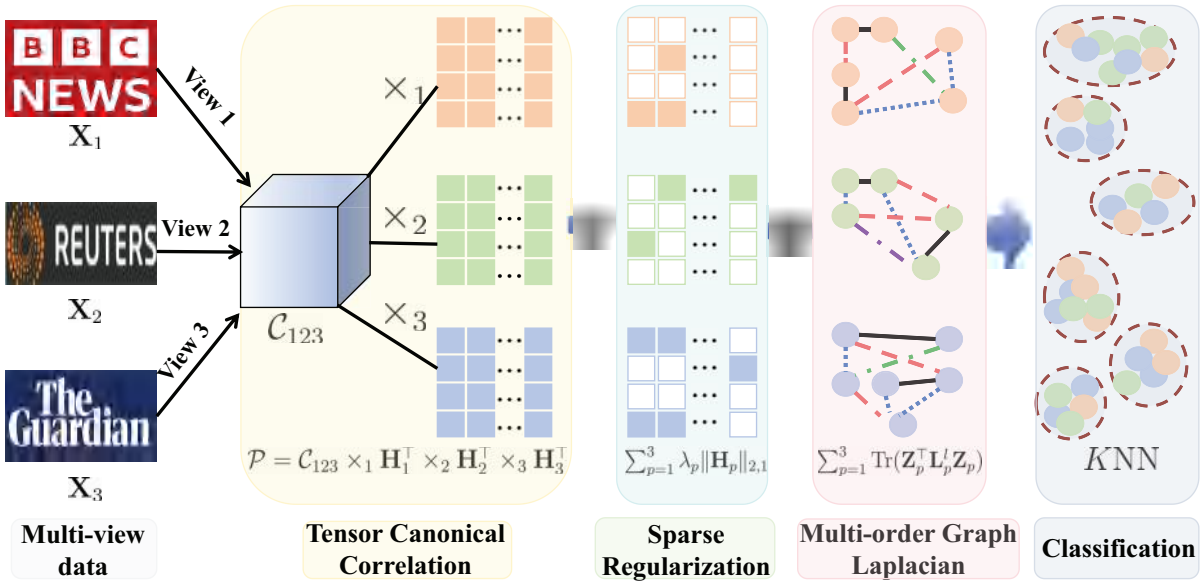


图 5.1: 所提 STCCA-L 的总体框架

往包含大量贡献微弱、解释性差的分量[84]. 稀疏学习是实现紧凑且可解释表示的重要手段. 为此, Du 等[85] 将稀疏正则项引入张量典型相关分析的目标函数, 在分析多模态脑影像数据高阶关联的同时实现特征选择. 然而, 该类稀疏结构使特征选择的精确控制变得复杂, 限制了其在特征空间中的可操作性. 此外, 现有多数张量典型相关分析更关注视角间结构关系的挖掘, 而对单个视角内部的固有几何结构重视不足. 图学习通过将数据表示为图来提供互补解决方案, 其中节点对应于数据点、边编码成对关系. 最近的研究表明, 高阶图能够捕捉多点交互信息, 提供更丰富的数据表示[86]. 多阶图学习通过自适应加权融合不同阶数的图结构, 进一步提升模型灵活性, 在多视角任务中展现出良好的潜力[87]. 值得注意的是, 现有张量典型相关分析普遍缺乏严格的算法收敛性理论分析. 仅有少数工作如 Du 等[85] 证明了目标函数的单调性, 而绝大多数方法未给出优化过程的收敛保证, 导致实际应用中学习结果的稳定性与可靠性难以保障.

受上述启发, 本章提出了带多阶图拉普拉斯正则化的稀疏张量典型相关分析 (sparse tensor CCA with Laplacian, STCCA-L). 一方面, 通过对投影矩阵施加结构稀疏正则, 实现重要特征的自动选取与冗余剔除. 另一方面, 利用多阶图拉普拉斯正则, 充分挖掘每个视角的固有几何信息. 具体如图 5.1 所示. 显然, 正则项的引入增加了计算负担. 为此, 开发了基于 Stiefel 流形优化的交替流形近端梯度算法, 在保证精度的同时维持可接受的计算效率. 本章的主要贡献为

(1) 将结构化稀疏正则与多阶图拉普拉斯正则融入张量典型相关分析, 构建了新的多视角子空间学习模型, 在缓解特征冗余的同时增强对视角内局部结构的刻画.

(2) 设计了流形近端梯度优化算法, 其子问题可通过半光滑牛顿法 (semi-smooth Newton, SSN) 高效计算. 在数学上, 严格证明了算法能够收敛至驻点.

(3) 验证了所提 STCCA-L 的有效性、鲁棒性以及稳定性, 并通过消融实验探讨了正则项、图结构、初始化等对分类结果的影响.

5.2 数学模型

5.2.1 预备知识

对于任意张量 $\mathcal{A}, \mathcal{B} \in \mathbb{R}^{I_1 \times \cdots \times I_N}$, 二者的内积定义为

$$\langle \mathcal{A}, \mathcal{B} \rangle = \sum_{i_1=1}^{I_1} \cdots \sum_{i_N=1}^{I_N} a_{i_1,\ldots,i_N} b_{i_1,\ldots,i_N}, \tag{5.1}$$

其外积为 $\mathcal{A} \circ \mathcal{B} \in \mathbb{R}^{I_1 \times \cdots \times I_N \times I_1 \times \cdots \times I_N}$, 且元素满足

$$(\mathcal{A} \circ \mathcal{B})_{i_1,\ldots,i_N,i_1,\ldots,i_N} = a_{i_1,\ldots,i_N} b_{i_1,\ldots,i_N}. \tag{5.2}$$

对于张量 $\mathcal{A}$ 与矩阵 $\boldsymbol{V} \in \mathbb{R}^{r_n \times I_n}$, 记二者 n 模乘积为

$$\mathcal{A} \times_n \boldsymbol{V} \in \mathbb{R}^{I_1 \times \cdots \times I_{n-1} \times r_n \times I_{n+1} \times \cdots \times I_N}. \tag{5.3}$$

若 $\boldsymbol{v} \in \mathbb{R}^{I_n}$ 为向量, 则 n 模乘积为

$$\mathcal{A} \times_n \boldsymbol{v} \in \mathbb{R}^{I_1 \times \cdots \times I_{n-1} \times I_{n+1} \times \cdots \times I_N}.$$

为书写简洁, 下文将集合 $\{1, 2, \ldots, N\}$ 简记为 $[N]$, 并使用 $p \in [N]$. 给定一组矩阵 $\{\boldsymbol{V}_p\}$, 其中 $\boldsymbol{V}_p \in \mathbb{R}^{r_p \times I_p}$ 且 $p \in [N]$, 张量 $\mathcal{A}$ 关于该组矩阵的收缩张量乘积定义为

$$\mathcal{B} = \mathcal{A} \times_1 \boldsymbol{V}_1 \times_2 \cdots \times_N V_N \in \mathbb{R}^{r_1 \times \cdots \times r_N}. \tag{5.4}$$

相应地, 张量 $\mathcal{B}$ 的 p 模展开矩阵可表示为

$$\mathcal{B}_{(p)} = \boldsymbol{V}_p \mathcal{A}_{(p)} (\boldsymbol{V}_{N-1} \otimes \cdots \otimes \boldsymbol{V}_{p+1} \otimes \boldsymbol{V}_{p-1} \otimes \cdots \otimes \boldsymbol{V}_1)^T, \tag{5.5}$$

其中 $\otimes$ 表示克罗内克积 (Kronecker) 积.

5.2.2 张量典型相关分析

考虑多视角数据 $\boldsymbol{X} = [\boldsymbol{X}_1, \ldots, \boldsymbol{X}_m]$, 其中每个视角 $\boldsymbol{X}_p \in \mathbb{R}^{d_p \times N}$ 对应同一组样本在不同特征空间下的表示. 张量典型相关分析的数学模型为

$$\begin{aligned} \max_{\{\boldsymbol{h}_p\}} \quad & \boldsymbol{C}_{1\ldots m} \times_1 \boldsymbol{h}_1^T \times_2 \cdots \times_m \boldsymbol{h}_m^T \\ \text{s.t.} \quad & \boldsymbol{h}_p^T C_{pp} \boldsymbol{h}_p = 1, \ p \in [m], \end{aligned} \tag{5.6}$$

其中

$$\boldsymbol{C}_{12\ldots m} = \frac{1}{N} \sum_{n=1}^{N} x_{1n} \circ \cdots \circ x_{mn} \in \mathbb{R}^{d_1 \times \cdots \times d_m} \tag{5.7}$$

为协方差张量, $\{\boldsymbol{h}_p\}$, $p \in [m]$ 为投影向量, $\boldsymbol{C}_{pp} = \boldsymbol{X}_p\boldsymbol{X}_p^T$ 表示第 p 个视角的协方差矩阵. 式 (5.6) 提供了一种探索潜在公共方向的紧凑方式. 然而, 该式仅能为每个视角学习一维投影, 且无法保证典型向量之间的不相关性, 往往导致学习到的表示存在冗余.

为克服上述局限, TCCA-O[83] 通过学习一组正交投影矩阵, 将张量典型相关分析推广至多维投影情形, 即考虑

$$\boldsymbol{H}_p = [\boldsymbol{h}_{p1}, \ldots, \boldsymbol{h}_{pr}] \in \mathbb{R}^{d_p \times r}, \ p \in [m] \tag{5.8}$$

同时捕获多个相关方向. 具体地, 数学模型可表述为

$$\begin{aligned} \max_{\{\boldsymbol{H}_p\}} \quad & \frac{1}{2}\|\boldsymbol{C}_{1\ldots m} \times_1 \boldsymbol{H}_1^T \times_2 \cdots \times_m \boldsymbol{H}_m^T\|_{\mathrm{F}}^2 \\ \text{s.t.} \quad & \boldsymbol{H}_p^T\boldsymbol{C}_{pp}\boldsymbol{H}_p = \boldsymbol{I}, \ p \in [m]. \end{aligned} \tag{5.9}$$

实验结果表明, TCCA-O 不仅保留了通过张量建模捕获高阶相关性的能力, 还能使学习到的投影在每个视角内形成正交基, 从而避免冗余并增强表示能力.

5.2.3 构建模型

设第 p 个视角对应加权无向图 $G_p = (V_p, \boldsymbol{W}_p)$, 其中 V_p 为节点集合, $\boldsymbol{W}_p \in \mathbb{R}^{N\times N}$ 为一阶邻接权重矩阵. 高阶图结构, 基于邻居的邻居也是邻居的思想, 能够挖掘一阶图中难以体现的深层拓扑信息. 给定一阶图 $\boldsymbol{W}_p$, 其第 h 阶图定义为

$$\boldsymbol{W}_p^h = \begin{cases} \boldsymbol{W}_p, & \text{若 } h = 1, \\ \boldsymbol{W}_p^{h-1}\boldsymbol{W}_p, & \text{若 } h > 1. \end{cases} \tag{5.10}$$

为避免阶数人工选择带来的困难, 可通过加权融合方式构造多阶图, 具体形式为

$$\boldsymbol{W}_p^l = \sum_{i=1}^{l} q^i \boldsymbol{W}_p^i, \tag{5.11}$$

其中, $q^i \in [0, 1)$ 满足 $\sum\limits_{i=1}^{l} q^i = 1$, 且 l 为最大阶数.

记相关性张量为

$$\boldsymbol{\mathcal{P}} = \boldsymbol{C}_{1\ldots m} \times_1 \boldsymbol{H}_1^T \times_2 \cdots \times_m \boldsymbol{H}_m^T \in \mathbb{R}^{r\times\cdots\times r}. \tag{5.12}$$

在此基础上, 为实现多视角独有特征与共享表示的联合学习, 构造如下模型

$$\begin{aligned} \min_{\{\boldsymbol{H}_p\}} \quad & -\frac{1}{2}\|\boldsymbol{\mathcal{P}}\|_F^2 + \sum_{p=1}^{m}\lambda_p\|\boldsymbol{H}_p\|_{2,1} + \sum_{p=1}^{m}\mathrm{tr}(\boldsymbol{Z}_p^T\boldsymbol{L}_p^l\boldsymbol{Z}_p) \\ \text{s.t.} \quad & \boldsymbol{H}_p^T\boldsymbol{X}_p\boldsymbol{X}_p^T\boldsymbol{H}_p = \boldsymbol{I}, \ p \in [m], \end{aligned} \tag{5.13}$$

其中, $\boldsymbol{L}_p^l = \boldsymbol{S}_p^l - \boldsymbol{W}_p^l$ 为多阶图对应的拉普拉斯矩阵, $\boldsymbol{S}_p^l$ 为度矩阵, 及 $\boldsymbol{Z}_p = \boldsymbol{X}_p^T \boldsymbol{H}_p$.

本章将式 (5.13) 简记为 STCCA-L. 与式 (5.9) 中的 TCCA-O 相比, STCCA-L 直接对投影矩阵 $\boldsymbol{H}_p$ 施加 $\ell_{2,1}$ 范数结构化正则, 不仅减少了学习子空间的冗余, 还促进了特征选择, 从而产生更具可解释性和更紧凑的表示. 此外, STCCA-L 引入图拉普拉斯正则项 $\mathrm{tr}(\boldsymbol{Z}_p^T \boldsymbol{L}_p^l \boldsymbol{Z}_p)$, 使得特征提取过程能够保持各视角数据固有的局部几何结构.

5.3 优化算法

依托 Stiefel 流形的定义, 可将式 (5.13) 等价改写为如下形式

$$\begin{aligned} \min_{\{\boldsymbol{H}_p\}} \quad & -\frac{1}{2}\|\boldsymbol{\mathcal{P}}\|_F^2 + \sum_{p=1}^{m} \lambda_p \|\boldsymbol{H}_p\|_{2,1} + \sum_{p=1}^{m} \mathrm{tr}(\boldsymbol{Z}_p^T \boldsymbol{L}_p^l \boldsymbol{Z}_p) \\ \text{s.t.} \quad & \boldsymbol{X}_p^T \boldsymbol{H}_p \in \mathrm{St}(r, N),\ p \in [m]. \end{aligned} \tag{5.14}$$

由于非光滑 $\ell_{2,1}$ 范数与非凸 Stiefel 流形的存在, 上述优化问题的求解并不容易. 为此, 受文献[88] 的启发, 本节设计了一种高效的流形近端梯度算法.

5.3.1 流形近端梯度法

记式 (5.14) 目标函数中的光滑部分为

$$f(\{\boldsymbol{H}_p\}) = -\frac{1}{2}\|\boldsymbol{\mathcal{P}}\|_F^2 + \sum_{p=1}^{m} \mathrm{tr}(\boldsymbol{Z}_p^T \boldsymbol{L}_p^l \boldsymbol{Z}_p). \tag{5.15}$$

在 Stiefel 流形上进行梯度迭代时, 下降方向必须限制在当前点的切空间内. 为表述简洁, 仅关注第 p 个视角对应变量 $\boldsymbol{H}_p$ 的子问题. 在第 k 次迭代中, 下降方向 $\boldsymbol{D}_p^k$ 可通过求解如下切空间上的近端子问题得到

$$\begin{aligned} \min_{\boldsymbol{D}_p} \quad & \langle \mathrm{grad} f(\boldsymbol{H}_p^k), \boldsymbol{D}_p \rangle + \frac{1}{2t}\|\boldsymbol{D}_p\|_F^2 + \lambda_p \|\boldsymbol{H}_p^k + \boldsymbol{D}_p\|_{2,1} \\ \text{s.t.} \quad & \boldsymbol{D}_p \in \mathrm{T}_{\boldsymbol{H}_p^k} \mathrm{St}(r, N), \end{aligned} \tag{5.16}$$

其中, $\mathrm{T}_{\boldsymbol{H}_p^k} \mathrm{St}(r, N) = \{\boldsymbol{D}_p \mid \boldsymbol{D}_p^T \boldsymbol{X}_p \boldsymbol{X}_p^T \boldsymbol{H}_p + \boldsymbol{H}_p^T \boldsymbol{X}_p \boldsymbol{X}_p^T \boldsymbol{D}_p = \boldsymbol{0}\}$ 为 Stiefel 流形 $\mathrm{St}(r, N)$ 在点 $\boldsymbol{H}_p^k$ 处的切空间. 根据黎曼梯度的定义, 对任意 $\boldsymbol{D}_p \in \mathrm{T}_{\boldsymbol{H}_p^k} \mathrm{St}(r, N)$, 有

$$\langle \mathrm{grad} f(\boldsymbol{H}_p^k), \boldsymbol{D}_p \rangle = \langle \nabla f(\boldsymbol{H}_p^k), \boldsymbol{D}_p \rangle. \tag{5.17}$$

注意到 f 由张量 Frobenius 范数和迹正则组成, 则 $\nabla f(\boldsymbol{H}_p^k)$ 可按结构分别计算. 对于张量部分, 固定除第 p 个投影矩阵外的其余所有变量, 并沿 $-p$ 计算导数. 结合迹正则项的梯度, 得到

$$\nabla f(\boldsymbol{H}_p) = \sum_{i=1}^{m-1} \boldsymbol{C}_{p_i} + \boldsymbol{X}_p \boldsymbol{L}_p^l \boldsymbol{Z}_p^k. \tag{5.18}$$

定义线性算子 $A^k(\boldsymbol{D}_p) = \boldsymbol{D}_p^T \boldsymbol{X}_p \boldsymbol{X}_p^T \boldsymbol{H}_p + \boldsymbol{H}_p^T \boldsymbol{X}_p \boldsymbol{X}_p^T \boldsymbol{D}_p$, 则式 (5.16) 可重写为

$$\begin{aligned} \min_{\boldsymbol{D}_p} \quad & \langle \nabla f, \boldsymbol{D}_p \rangle + \frac{1}{2t} \|\boldsymbol{D}_p\|_F^2 + \lambda_p \|\boldsymbol{H}_p^k + \boldsymbol{D}_p\|_{2,1} \\ \text{s.t.} \quad & A^k(\boldsymbol{D}_p) = \boldsymbol{0}, \end{aligned} \tag{5.19}$$

上式本质是将近端梯度迭代约束在 Stiefel 流形的切空间内完成. 在通过求解式 (5.19) 得到下降方向 $\boldsymbol{D}_p^k$ 后, 采用 Armijo 线搜索确定合适步长 α^k, 并通过黎曼收缩映射将切空间中的更新结果投影回流形, 保证迭代点始终满足流形可行性, 即

$$\boldsymbol{H}_p^{k+1} = \mathrm{Retr}_{\boldsymbol{H}_p^k}(\alpha^k \boldsymbol{D}_p^k). \tag{5.20}$$

5.3.2 半光滑牛顿法

一个关键问题是如何高效求解式 (5.19). 近年来, 半光滑牛顿法因在求解结构化凸优化问题上兼具高精度与高收敛速率, 得到了广泛的应用. 首先定义拉格朗日函数

$$\begin{aligned} L(\boldsymbol{D}_p, \boldsymbol{\Lambda}_p) = & \langle \nabla f(\boldsymbol{H}_p^k), \boldsymbol{D}_p \rangle + \lambda_p \|\boldsymbol{H}_p^k + \boldsymbol{D}_p\|_{2,1} \\ & + \frac{1}{2t} \|\boldsymbol{D}_p\|_F^2 - \langle A^k(\boldsymbol{D}_p), \boldsymbol{\Lambda}_p \rangle, \end{aligned} \tag{5.21}$$

其中, $\boldsymbol{\Lambda}_p$ 为拉格朗日乘子. 下面分四步设计高效的半光滑牛顿法求解方案.

(1) 最优性条件

式 (5.19) 的 Karush-Kuhn-Tucker (KKT) 条件为

$$\boldsymbol{0} \in \partial_{\boldsymbol{D}_p} L(\boldsymbol{D}_p, \boldsymbol{\Lambda}_p), \ A^k(\boldsymbol{D}_p) = \boldsymbol{0}. \tag{5.22}$$

由次微分条件可导出

$$\boldsymbol{D}_p = \mathrm{prox}_{2,1}(B(\boldsymbol{\Lambda}_p), t) - \boldsymbol{H}_p^k, \tag{5.23}$$

其中, $B(\boldsymbol{\Lambda}_p) = \boldsymbol{H}_p^k - t(\nabla f(\boldsymbol{H}_p^k) - 2\boldsymbol{X}_p \boldsymbol{X}_p^T \boldsymbol{H}_p^k \boldsymbol{\Lambda}_p)$. 将式 (5.23) 代入式 (5.22), 得到关于乘子 $\boldsymbol{\Lambda}_p$ 的非线性方程

$$Q(\boldsymbol{\Lambda}_p) = \boldsymbol{D}_p^T \boldsymbol{X}_p \boldsymbol{X}_p^T \boldsymbol{H}_p^k + (\boldsymbol{H}_p^k)^T \boldsymbol{X}_p \boldsymbol{X}_p^T \boldsymbol{D}_p = \boldsymbol{0}. \tag{5.24}$$

(2) 计算广义雅可比矩阵

文献[88] 已证明, 算子 Q 单调且 Lipschitz 连续, 满足半光滑牛顿法的适用条件. 为构造牛顿迭代, 需要计算 Q 的广义雅可比矩阵. 对 $Q(\boldsymbol{\Lambda}_p)$ 向量化可表示为

$$\begin{aligned}\mathrm{vec}(Q(\boldsymbol{\Lambda}_p)) = (\boldsymbol{K}_{rr} + \boldsymbol{I}_{r^2})((\boldsymbol{H}_p^k)^T \boldsymbol{X}_p \boldsymbol{X}_p^T \otimes \boldsymbol{I}_r) \\ [\mathrm{prox}_{2,1}(\mathrm{vec}((\boldsymbol{H}_p^k)^T \boldsymbol{X}_p \boldsymbol{X}_p^T) - t\nabla f(\boldsymbol{H}_p^k), t)] \\ + 2t(\boldsymbol{X}_p \boldsymbol{X}_p^T \boldsymbol{H}_p^k \otimes \boldsymbol{I}_r)\,\mathrm{vec}(\boldsymbol{\Lambda}_p) - \mathrm{vec}((\boldsymbol{H}_p^k)^T),\end{aligned} \tag{5.25}$$

其中, $\boldsymbol{K}_{rd_p}$ 与 $\boldsymbol{K}_{rr}$ 为交换矩阵. 定义

$$\Xi_{p_j} = \begin{cases} \boldsymbol{I}_r - \frac{\tau_1 t}{\|\boldsymbol{b}_j\|}\boldsymbol{R}, & \text{若 } \|\boldsymbol{b}_j\| > t\tau_1, \\ \gamma \frac{\boldsymbol{b}_j \boldsymbol{b}_j^T}{(t\tau_1)^2}, & \text{若 } \|\boldsymbol{b}_j\| = t\tau_1, \\ 0, & \text{其他}, \end{cases} \tag{5.26}$$

其中, $p \in [m]$, $j \in [d_p]$, $\boldsymbol{R} = (\boldsymbol{I}_r - \frac{\boldsymbol{b}_j \boldsymbol{b}_j^T}{\|\boldsymbol{b}_j\|^2})$, $\gamma \in [0,1]$, 且 $\boldsymbol{b}_j$ 是 $B(\boldsymbol{\Lambda}_p)^T$ 的第 j 列. 令

$$\boldsymbol{J}(\mathrm{y})|_{\mathrm{y}=\mathrm{vec}(B(\boldsymbol{\Lambda}_p)^T)} = \mathrm{Diag}(\Xi_{p_{d_p}}, \ldots, \Xi_{p_{d_p}}), \tag{5.27}$$

则 $\mathrm{vec}(Q(\boldsymbol{\Lambda}_p))$ 的广义雅可比矩阵为

$$\boldsymbol{V} = 2t(\boldsymbol{K}_{rr} + \boldsymbol{I}_{r^2})((\boldsymbol{H}_p^k)^T \boldsymbol{X}_p \boldsymbol{X}_p^T \otimes \boldsymbol{I}_r)\boldsymbol{J}(\boldsymbol{y})(\boldsymbol{X}_p \boldsymbol{X}_p^T \boldsymbol{H}_p^k \otimes \boldsymbol{I}_r). \tag{5.28}$$

由 Q 的单调性可知, $\boldsymbol{V}$ 是半正定的, 则对于任意 $\boldsymbol{\sigma} \in \mathbb{R}^{r^2}$, 有

$$\boldsymbol{V}\boldsymbol{\sigma} = \nabla(\mathrm{vec}(Q(\mathrm{vec}(\boldsymbol{\Lambda}_p))))\boldsymbol{\sigma}. \tag{5.29}$$

(3) 加速处理

由于 $\boldsymbol{\Lambda}_p$ 为对称矩阵, 可用 $\overline{\mathrm{vec}}(\boldsymbol{\Lambda}_p)$ 表示仅保留下三角元素的 $\frac{1}{2}r(r+1)$ 维压缩向量. 引入矩阵 $\boldsymbol{U}_p \in \mathbb{R}^{r^2 \times \frac{1}{2}r(r+1)}$ 及其 Moore-Penrose 逆 $\boldsymbol{U}_p^+$, 满足 $\boldsymbol{U}_p \overline{\mathrm{vec}}(\boldsymbol{\Lambda}_p) = \mathrm{vec}(\boldsymbol{\Lambda}_p)$, 则广义雅可比矩阵可约简为

$$\boldsymbol{V}(\overline{\mathrm{vec}}(\boldsymbol{\Lambda}_p)) = t\boldsymbol{U}_p^+ \boldsymbol{V} \boldsymbol{U}_p. \tag{5.30}$$

(4) 半光滑牛顿更新

半光滑牛顿方向 $\boldsymbol{d}_k$ 通过如下带正则的线性系统求解

$$(\boldsymbol{V} + \eta \boldsymbol{I}_{r^2})\boldsymbol{d} = -\overline{\mathrm{vec}}(Q(\overline{\mathrm{vec}}(\boldsymbol{\Lambda}_p^k))), \tag{5.31}$$

其中, $\eta > 0$ 为正则化参数. 乘子迭代更新为

$$\overline{\mathrm{vec}}(\boldsymbol{\Lambda}_p^{k+1}) = \overline{\mathrm{vec}}(\boldsymbol{\Lambda}_p^k) + \boldsymbol{d}_k. \tag{5.32}$$

综上, 完整的流形近端梯度算法框架如算法 3 所示.

算法 3 求解式 (5.14) 的流形近端梯度算法

输入: 数据 $\boldsymbol{X} = [\boldsymbol{X}_1, \ldots, \boldsymbol{X}_m]$, 步长 t, 最大迭代次数 T, $\gamma \in (0,1)$, 计算张量 $\boldsymbol{C}_{1\ldots m}$

初始化: $\boldsymbol{H}_p^0 \in \mathrm{St}(n, r)$

for $p \in [m]$ **do**
  if $k < T$ **then**
    采用半光滑牛顿法求解式 (5.19), 得到下降方向 $\boldsymbol{D}_p^k$
    while $f(\mathrm{Retr}_{\boldsymbol{H}_p^k}(\alpha \boldsymbol{D}_p^k)) \geq f(\boldsymbol{H}_p^k) - \frac{\alpha \|\boldsymbol{D}_p^k\|_F^2}{2t}$ **do**
      $\alpha = \gamma \alpha$
    end while
    令 $\boldsymbol{H}_p^{k+1} = \mathrm{Retr}_{\alpha \boldsymbol{H}_p^k}(\alpha \boldsymbol{D}_p^k)$
  end if
end for

输出: $\{\boldsymbol{H}_p^k\}$

5.3.3 收敛性分析

现有多数张量典型相关分析算法虽在数值实验上表现良好, 但缺少严格的收敛性理论. 接下来, 本节将对所提算法 3 建立完整的收敛性分析. 为便于分析, 记

$$\boldsymbol{H}_{(p)}^k(\alpha) = \{\boldsymbol{H}_1^k, \ldots, \boldsymbol{H}_{p-1}^k, \boldsymbol{H}_p^k + \alpha \boldsymbol{D}_p^k, \boldsymbol{H}_{p+1}^k, \ldots, \boldsymbol{H}_m^k\}. \tag{5.33}$$

并将式 (5.19) 的目标函数简记为

$$g(\boldsymbol{D}_p) = \langle \nabla f, \boldsymbol{D}_p \rangle + \frac{1}{2t} \|\boldsymbol{D}_p\|_F^2 + \lambda_p \|\boldsymbol{H}_p^k + \boldsymbol{D}_p\|_{2,1}. \tag{5.34}$$

引理 5.1 设 $t \leq 1/L_p$, 其中 L_p 为 $\nabla_{\boldsymbol{H}_p} f$ 的 Lipschitz 常数, 则对任意 $\alpha \in [0,1]$, 有

$$f(\boldsymbol{H}_{(p)}^k(\alpha)) + \|\boldsymbol{H}_p^k + \alpha \boldsymbol{D}_p^k\|_{2,1} \leq f(\boldsymbol{H}_{(p)}^k(0)) + \|\boldsymbol{H}_p^k\|_{2,1}. \tag{5.35}$$

证明 由于目标函数 $g(\boldsymbol{D}_p)$ 为 $\frac{1}{t}$-强凸函数, 因此对任意 $\widehat{\boldsymbol{D}}_p$, $\boldsymbol{D}_p$, 满足

$$g(\widehat{\boldsymbol{D}}_p) \geq g(\boldsymbol{D}_p) + \langle \partial g(\boldsymbol{D}_p), \widehat{\boldsymbol{D}}_p - \boldsymbol{D}_p \rangle + \frac{\alpha}{2} \|\widehat{\boldsymbol{D}}_p - \boldsymbol{D}_p\|_F^2. \tag{5.36}$$

考虑 $\widehat{\boldsymbol{D}}_p$, $\boldsymbol{D}_p \in \mathrm{T}_{\boldsymbol{H}_p^k} \mathrm{St}(r, N)$, 则有

$$\langle \partial g(\boldsymbol{D}_p), \widehat{\boldsymbol{D}}_p - \boldsymbol{D}_p \rangle = \langle \mathrm{proj}_{\mathrm{T}_{\boldsymbol{H}_p^k}}(\partial g(\boldsymbol{D}_p)), \widehat{\boldsymbol{D}}_p - \boldsymbol{D}_p \rangle. \tag{5.37}$$

由黎曼最优性条件, 可得

$$0 \in \mathrm{proj}_{\mathrm{T}_{\boldsymbol{H}_p^k}}(\partial g(\boldsymbol{D}_p^k)). \tag{5.38}$$

在式 (5.36) 中令 $\boldsymbol{D}_p = \boldsymbol{D}_p^k$, $\widehat{\boldsymbol{D}}_p = \alpha \boldsymbol{D}_p^k$, 且 $\alpha \in [0,1]$, 可得

$$g(\alpha \boldsymbol{D}_p^k) - g(\boldsymbol{D}_p^k) \geq \frac{(1-\alpha)^2}{2t}\|\boldsymbol{D}_p^k\|_F^2. \tag{5.39}$$

结合 g 的定义与 $\ell_{2,1}$ 范数的凸性, 进一步得到

$$\begin{aligned}&(1-\alpha)\langle \nabla f(\boldsymbol{H}_{(p)}^k(0)), \boldsymbol{D}_p^k\rangle + \frac{1-\alpha}{t}\|\boldsymbol{D}_p^k\|_F^2 \\ &+ (1-\alpha)(\|\boldsymbol{H}_p^k + \boldsymbol{D}_p^k\|_{2,1} - \|\boldsymbol{H}_p^k\|_{2,1}) \leq 0.\end{aligned} \tag{5.40}$$

再利用 f 的 Lipschitz 连续性, 有

$$\begin{aligned}&f(\boldsymbol{H}_{(p)}^k(\alpha)) - f(\boldsymbol{H}_{(p)}^k(0)) + \|\boldsymbol{H}_p^k + \alpha \boldsymbol{D}_p^k\|_{2,1} - \|\boldsymbol{H}_p^k\|_{2,1} \\ &\leq \alpha\langle \nabla f(\boldsymbol{H}_{(p)}^k(0)), \boldsymbol{D}_p^k\rangle + \frac{\alpha^2}{2t}\|\boldsymbol{D}_p^k\|_F^2 + \alpha(\|\boldsymbol{H}_p^k + \boldsymbol{D}_p^k\|_{2,1} - \|\boldsymbol{H}_p^k\|_{2,1}) \\ &\leq -\frac{\alpha}{2t}\|\boldsymbol{D}_p^k\|_F^2.\end{aligned} \tag{5.41}$$

上述引理表明, $\boldsymbol{D}_p^k$ 是切空间中的下降方向. 进一步, 当 $\boldsymbol{D}_p^k = \boldsymbol{0}$, $p \in [m]$ 时, 当前迭代点即为原问题的驻点.

引理 5.2 若对所有 $p \in [m]$ 均有 $\boldsymbol{D}_p^k = \boldsymbol{0}$, 则序列 $\{\boldsymbol{H}_p^k\}$ 是式 (5.14) 的驻点.

证明 对任意 $p \in [m]$, 式 (5.19) 的最优性条件为

$$\boldsymbol{0} \in \frac{1}{t}\boldsymbol{D}_p^k + \nabla f(\boldsymbol{H}_p^k) + \mathrm{proj}_{\mathrm{T}_{\boldsymbol{H}_p^k}} \partial\|\boldsymbol{H}_p^k + \boldsymbol{D}_p^k\|_{2,1}, \tag{5.42}$$

其中 $\boldsymbol{D}_p^k \in \mathrm{T}_{\boldsymbol{H}_p^k}\mathrm{St}(r,N)$. 若 $\boldsymbol{D}_p^k = \boldsymbol{0}$, 则上式退化为

$$\boldsymbol{0} \in \nabla f(\boldsymbol{H}_p^k) + \mathrm{proj}_{\mathrm{T}_{\boldsymbol{H}_p^k}} \partial\|\boldsymbol{H}_p^k + \boldsymbol{D}_p^k\|_{2,1}. \tag{5.43}$$

此即为式 (5.14) 在 Stiefel 流形上的一阶必要性条件, 故序列 $\{\boldsymbol{H}_p^k\}$ 为驻点.

定义式 (5.14) 的目标函数为

$$\phi(\boldsymbol{H}_p) = f(\boldsymbol{H}_p) + \|\boldsymbol{H}_p + \boldsymbol{D}_p\|_{2,1}. \tag{5.44}$$

引理 5.3 设序列 $\{\boldsymbol{H}_p^k\}$ 由算法 3 生成, 则 $\{\phi(\boldsymbol{H}_p^k)\}$ 单调递减, 且满足

$$\phi(\mathrm{Retr}_{\boldsymbol{H}_p^k}(\alpha \boldsymbol{D}_p^k)) - \phi(\boldsymbol{H}_p^k) \leq -\frac{\alpha}{2t}\|\boldsymbol{D}_p^k\|_F^2, \ p \in [m]. \tag{5.45}$$

证明 令 $\boldsymbol{H}_p^{k+} = \boldsymbol{H}_p^k + \alpha \boldsymbol{D}_p^k$. 根据 $\nabla\phi$ 的 L-Lipschitz 连续性, 对于任意 $\alpha > 0$, 有

$$\begin{aligned}&\phi(\mathrm{Retr}_{\boldsymbol{H}_p^k}(\alpha \boldsymbol{D}_p^k)) - \phi(\boldsymbol{H}_p^k) \\ &\leq \langle \nabla\phi(\boldsymbol{H}_p^k), \mathrm{Retr}_{\boldsymbol{H}_p^k}(\alpha \boldsymbol{D}_p^k) - \boldsymbol{H}_p^{k+} + \boldsymbol{H}_p^{k+} - \boldsymbol{H}_p^k\rangle + \frac{L}{2}\|\mathrm{Retr}_{\boldsymbol{H}_p^k}(\alpha \boldsymbol{D}_p^k) - \boldsymbol{H}_p^k\|_F^2 \\ &\leq \zeta_2\|\nabla\phi(\boldsymbol{H}_p^k)\|_F\|\alpha \boldsymbol{D}_p^k\|_F^2 + \alpha\langle \nabla\phi(\boldsymbol{H}_p^k), \boldsymbol{D}_p^k\rangle + \frac{L\zeta_1^2}{2}\|\alpha \boldsymbol{D}_p^k\|_F^2.\end{aligned} \tag{5.46}$$

其中, $\zeta_1, \zeta_2 > 0$ 为常数. 由于 $\nabla\phi$ 在紧流形 $\mathrm{St}(r,N)$ 上连续, 存在 $\mu > 0$ 使得对任意 $\boldsymbol{H}_p \in \mathrm{St}(r,N)$

满足 $\|\nabla\phi(\boldsymbol{H}_p^k)\|_F \le \mu$. 记 $c_0 = \zeta_2\mu + L\zeta_1^2/2$, 则

$$\phi(\mathrm{Retr}_{\boldsymbol{H}_p^k}(\alpha\boldsymbol{D}_p^k)) - \phi(\boldsymbol{H}_p^k) \le \alpha\langle\nabla\phi(\boldsymbol{H}_p^k), \boldsymbol{D}_p^k\rangle + c_0\alpha^2\|\boldsymbol{D}_p^k\|_F^2 \tag{5.47}$$

由 $\alpha\langle\nabla\phi(\boldsymbol{H}_p^k), \boldsymbol{D}_p^k\rangle \le -\frac{1}{t}\|\alpha\boldsymbol{D}_p^k\|_F^2$, 代入可得

$$\phi(\mathrm{Retr}_{\boldsymbol{H}_p^k}(\alpha\boldsymbol{D}_p^k)) - \phi(\boldsymbol{H}_p^k) \le (c_0 + \delta\zeta_2 - \frac{1}{\alpha t})\|\alpha\boldsymbol{D}_p^k\|_F^2. \tag{5.48}$$

取 $\bar{\alpha} = 1/(2(c_0 + \delta\zeta_2)t)$, 对于任意 $0 < \alpha \le \min\{\bar{\alpha}, 1\}$, 满足

$$\phi(\mathrm{Retr}_{\boldsymbol{H}_p^k}(\alpha\boldsymbol{D}_p^k)) - \phi(\boldsymbol{H}_p^k) \le -\frac{\alpha}{2t}\|\boldsymbol{D}_p^k\|_F^2. \tag{5.49}$$

即经收缩映射后仍保持目标函数下降, 于是引理得证.

定理 5.1 算法 3 生成的迭代序列 $\{\boldsymbol{H}_p^k\}$ 收敛到式 (5.14) 的驻点.

证明 目标函数 ϕ 在 $\mathrm{St}(r, N)$ 上有下界, 结合引理 5.3, 可得

$$\lim_{k\to\infty} \|\boldsymbol{D}_p^k\|_F^2 = 0. \tag{5.50}$$

再由引理 5.2, 可知 $\{\boldsymbol{H}_p^k\}$ 的任一极限点均满足一阶最优性条件, 即为驻点.

5.3.4 复杂度分析

算法 3 的总体计算复杂度为 $O(Tm(r^m + d_aN + d_a^2r))$, 其中 T 为外层迭代次数, m 为视角数量, $d_a = \max_p\{d_p\}$ 为各视角维度的最大值. 可以看出, 主要计算成本来自构造协方差张量、求解半光滑牛顿子问题、评估目标函数以及 Stiefel 流形收缩映射. 后续数值实验部分将给出不同算法的实际运行时间对比, 以验证本章算法的计算效率.

5.4 数值实验

本节将所提 STCCA-L 与主流多视角学习方法进行对比, 包括典型相关分析方法 CCA[1] (2009)、SCCA[2] (2014)、SGCCA[3] (2024), 张量典型相关分析方法 TCCA[4] (2015)、TCCA-O[5] (2023)、TCCA-OS[5] (2023) 以及 TMCCA (2023), 以及 k 近邻 (k-nearest neighbors, KNN)、RTSL[6] (2024)、CDPML[7] (2025). 此外, 所提方法开源代码见链接 https://github.com/xianchaoxiu/STCCA-L.

[1] https://github.com/tmarino2/scca

[2] https://github.com/htpusa/scanoncorr

[3] https://github.com/kelenlv/SGCCA2023

[4] https://github.com/yluopku/TCCA

[5] https://github.com/xianchaoxiu/TCCA

[6] https://github.com/suxiao1824308603/Multi-view-Learning

[7] https://github.com/zzf495/CDPMVL

5.4.1 实验设置

(1) 数据集

为验证所提方法的有效性, 选择八个多视角数据集, 包括 3Sources[8], MSRC[9], BBCsport[10], Reusters[9], Caltech101[11], Handwritten[12], MNIST[13] 以及 Animal[9]. 根据样本数规模的大小, 将这些数据集划分为三组, 如表 5.1 所示.

表 5.1: 数据集信息

尺寸	数据集	类别数	视角数	样本数
小	3Sources	6	3	169
	MSRC	7	5	210
	BBCsport	5	2	544
中	Reuters	6	5	1,200
	Caltech101	7	4	1,474
	Handwritten	10	5	2,000
大	MNIST	10	2	10,000
	Animal	20	4	11,673

(2) 参数设置

实验中, 对所有数据集均先采用主成分分析进行预处理, 以间隔 2 将特征维度从 2 调整至 20. 对于第 p 个视角, 其投影矩阵计算为 $\boldsymbol{Z}_p = \boldsymbol{X}_p^T \boldsymbol{H}_p$, 进一步得到 $\boldsymbol{Z} = [\boldsymbol{Z}_1, \ldots, \boldsymbol{Z}_m] \in \mathbb{R}^{N \times mr}$. 最后, 使用 KNN 来测量分类性能. 为保证对比的公平性, 将根据各数据集的特征自适应调整邻居数 k, 且所有方法在同一数据集上使用相同的 k 值. 权重矩阵 $\boldsymbol{W}_p$ 采用自适应邻居图进行初始化, 有效性将在消融实验中验证. 所有惩罚参数均通过交叉验证确定, 测试集比例设置为 0.3. 为降低随机因素对实验结果的影响, 每次实验均随机重复 10 次, 最终记录实验的平均结果.

(3) 评估指标

本章采用分类准确率 (Accuracy) 和 F1 分数 (F1-score) 作为评估分类性能的指标. 其中, 准确率用于衡量分类结果的整体正确性, 其定义为

$$\text{Accuracy} = \frac{\sum_{i=1}^{C} (\text{TP}_i + \text{TN}_i)}{\sum_{i=1}^{C} (\text{TP}_i + \text{FP}_i + \text{TN}_i + \text{FN}_i)}, \tag{5.51}$$

[8] http://mlg.ucd.ie/datasets/3sources.html

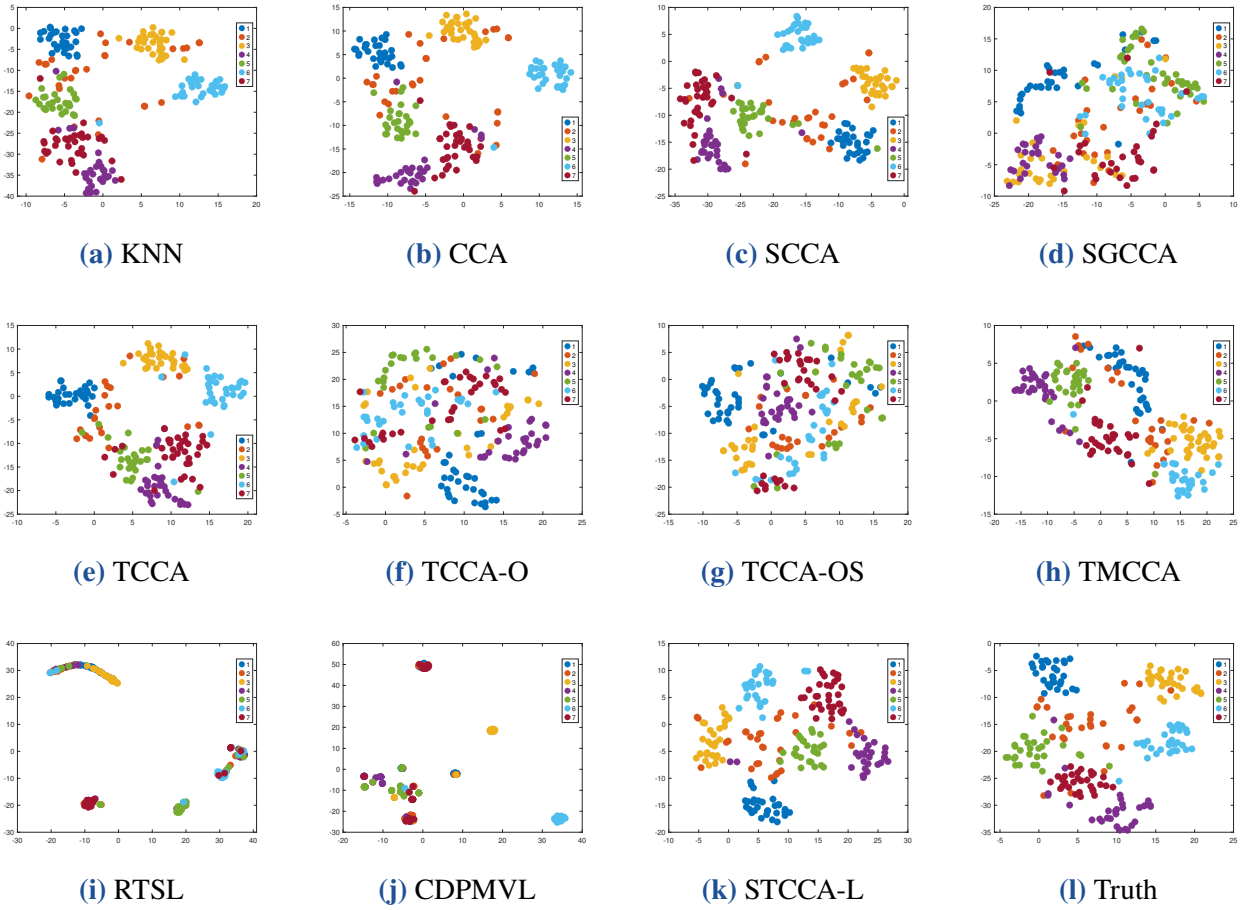
[9] https://github.com/zhudafa/Multi-view-datasets

[10] http://mlg.ucd.ie/datasets/bbc.html

[11] https://data.caltech.edu/records/mzrjq-6wc02

[12] https://github.com/cvdfoundation/mnist

[13] https://tensorflow.google.cn/datasets/catalog/mnist



(a) KNN (b) CCA (c) SCCA (d) SGCCA

(e) TCCA (f) TCCA-O (g) TCCA-OS (h) TMCCA

(i) RTSL (j) CDPMVL (k) STCCA-L (l) Truth

图 5.2: MSRC 数据集上 t-SNE 的可视化对比

其中 C 表示类别总数, TP_i、TN_i、FP_i、FN_i 分别为第 i 类样本的真阳性数量、真阴性数量、假阳性数量、假阴性数量.

F1 分数 (F1-score) 考虑了精确率和召回率, 适用于类别分布不平衡情形, 其定义为

$$\text{F1-score} = \frac{1}{C}\sum_{i=1}^{C}\frac{2\mathrm{TP}_i}{2\mathrm{TP}_i + \mathrm{FP}_i + \mathrm{FN}_i}. \tag{5.52}$$

准确率和 F1 分数的数值越高, 表明对比方法的分类性能越优.

5.4.2 实验结果

表 5.2 和表 5.3 分别列出了所提 STCCA-L 与其他方法在分类准确率和 F1 分数上的对比结果. 其中, 最佳结果以粗体标注, "-" 表示内存不足. 由实验结果可见, 所提 STCCA-L 在所有数据集上的分类准确率与 F1 分数均优于其他方法, 验证了其在多视角分类任务中的有效性. 具体而言, 在 Animals、MSRC 及 3Sources 数据集上, STCCA-L 相较于次佳方法, 分类准确率分别提升了 5.29%、4.76% 和 4.50%, F1 分数分别提升了 3.41%、5.37% 和 6.77%, 提升效果显著. 进一步, 图 5.2 展示了所有对比方法在 MSRC 数据集上的 t-SNE 可视化结果. 显然, 所提

表 5.2: 对比方法在最佳维度下的分类准确率 (%)

方法	3Sources	MSRC	BBCsport	Reuters	Caltech101	Handwritten	MNIST	Animal
KNN	82.00±6.46	71.74±0.36	93.25±1.37	70.83±1.57	85.47±1.38	85.40±4.99	92.87±0.31	27.23±0.50
CCA	86.20±5.99	73.17±0.54	95.71±1.45	71.67±5.11	87.73±0.22	87.43±1.18	92.40±0.55	27.38±0.36
SCCA	63.50±8.99	63.65±8.84	61.76±9.23	41.25±12.37	82.86±9.26	71.37±3.54	48.39±3.20	17.40±2.34
SGCCA	63.50±8.99	63.65±8.84	61.76±9.23	41.25±12.37	82.86±9.26	71.37±3.54	48.39±3.20	17.40±2.34
TCCA	83.00±6.23	85.24±4.30	91.00±1.49	52.08±1.76	89.98±1.51	94.52±1.46	83.53±0.94	27.77±0.64
TCCA-O	90.50±1.91	73.02±6.03	96.32±1.07	72.91±1.37	89.37±1.33	80.37±2.68	93.13±0.13	21.89±0.86
TCCA-OS	83.50±5.97	74.13±4.97	95.19±1.66	72.67±1.37	90.61±1.24	78.10±1.92	92.49±0.28	22.31±0.63
TMCCA	64.60±4.34	53.97±7.18	94.58±1.75	56.67±1.17	84.73±3.10	87.45±4.73	59.50±6.83	18.22±4.85
RTSL	68.00±5.06	63.49±2.47	90.49±1.18	71.94±1.37	85.75±0.96	94.50±1.65	–	–
CDPML	79.20±5.67	66.03±7.63	92.14±2.27	58.58±4.44	90.27±2.05	93.07±1.16	87.77±0.66	19.70±1.21
STCCA-L	**95.00±4.24**	**93.65±3.42**	**98.01±0.90**	**76.11±0.23**	**94.29±0.39**	**98.45±0.63**	**94.48±0.40**	**33.06±0.77**

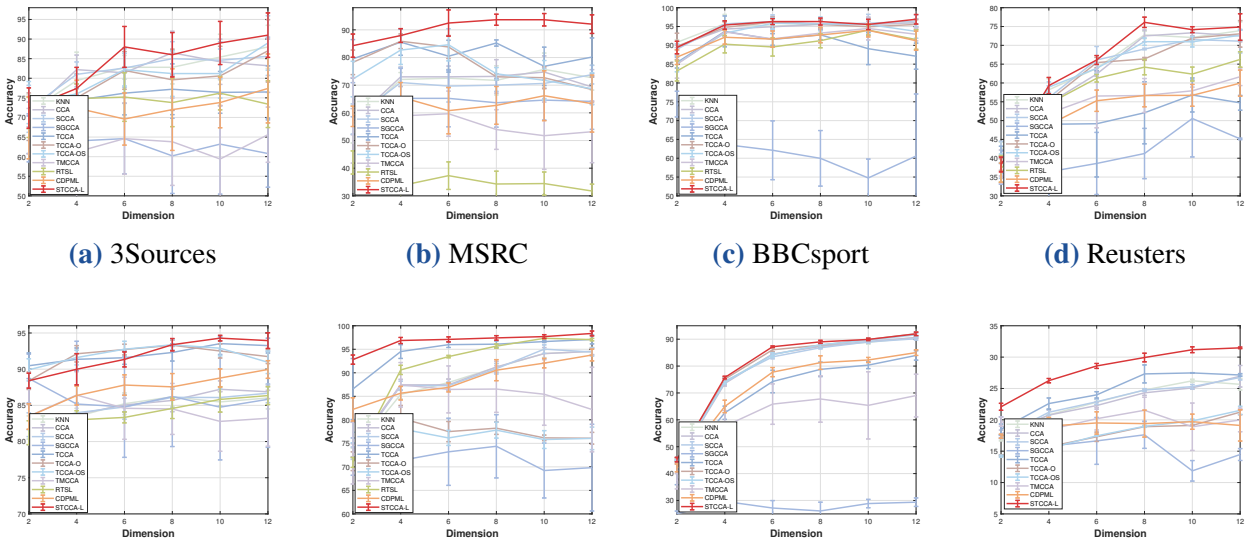
表 5.3: 对比方法在最佳维度下的 F1 分数 (%)

方法	3Sources	MSRC	BBCsport	Reuters	Caltech101	Handwritten	MNIST	Animal
KNN	75.61±4.91	84.82±3.54	93.54±1.45	68.52±3.02	58.82±3.41	77.82±1.06	92.36±0.71	23.53±0.32
CCA	83.75±3.60	74.01±1.57	95.82±1.98	71.55±2.84	56.39±4.48	78.07±0.44	92.37±0.39	23.26±0.45
SCCA	80.13±6.38	88.17±1.51	96.62±0.79	68.20±1.01	57.44±3.57	79.09±0.21	92.19±0.22	23.95±0.58
SGCCA	51.45±12.01	57.30±1.93	54.66±10.31	46.68±5.43	45.39±6.46	74.28±4.26	25.05±1.77	15.24±2.01
TCCA	83.41±6.75	85.72±2.18	90.23±6.07	61.83±1.00	58.69±2.13	96.49±0.82	77.12±2.33	23.65±0.61
TCCA-O	87.51±4.66	72.44±2.25	97.20±2.32	71.89±3.01	71.34±1.43	92.05±0.82	91.13±0.24	18.10±1.02
TCCA-OS	80.24±3.92	68.72±4.89	95.64±1.44	67.07±0.22	71.13±1.92	91.39±1.26	91.78±0.33	18.22±0.98
TMCCA	87.38±4.95	79.69±3.27	94.08±0.51	62.77±2.28	52.23±6.93	86.52±5.78	59.75±4.95	16.22±3.57
RTSL	37.75±4.90	61.42±3.06	88.26±1.69	72.06±1.46	48.83±3.69	94.55±1.65	–	–
CDPML	75.06±7.84	65.15±7.21	92.65±2.10	58.76±4.27	70.97±5.59	93.09±1.20	87.55±0.71	15.81±1.04
STCCA-L	**95.16±4.55**	**93.54±2.08**	**98.63±0.47**	**75.05±2.67**	**77.25±1.56**	**98.05±0.70**	**94.44±0.36**	**27.36±0.44**

STCCA-L 对应的样本点中, 相同类别 (同颜色) 的聚集度最高, 分类边界最清晰, 进一步佐证了其优异的分类性能.

与 KNN 相比, 各类多视角子空间学习方法的分类性能均有提升. 此外, 与 RTSL、CDPMVL 等方法相比, 基于典型相关分析的方法整体表现出更稳健的分类性能. 值得注意的是, RTSL 在大型数据集上出现内存不足错误, 说明该方法不适用于大规模多视角数据. 在 CCA、SCCA、SGCCA 等矩阵型典型相关分析方法中, SCCA 取得了最优性能, 这主要归功于其引入的稀疏正则. 相较于矩阵型典型相关分析方法, 所提 STCCA-L 展现出显著的性能优势, 例如在 Handwritten 数据集上, STCCA-L 相较于最优矩阵型典型相关分析方法, 分类准确率提升 11.02%, F1 分数提升 18.96%. 同时也注意到, 张量典型相关分析方法在分类性能方面普遍优于矩阵方法. 但 TCCA 在 BBCSport、Reuters、MNIST 等数据集上未能取得令人满意的结果, 相比之下, 所提 STCCA-L 既能够通过稀疏正则化筛选关键特征, 通过正交正则化保证特征的独立性, 又能够通过拉普拉斯正则化利用数据的图结构信息, 从而在全部多视角数据集上实现了优异的分类性能. 值得强调的是, 在包含 5 个视角的 MSRC 数据集上, STCCA-L 相较于性能最优的其他张量型 CCA 方法, 分类准确率提升 8.41%, F1 分数提升 7.82%.

图 5.3 为带有误差棒的准确率折线图, 反映了不同方法在不同维度的分类准确率变化. 从图中可看出, STCCA-L 在八个数据集上的分类准确率均高于其他对比方法. 就趋势而言, 随着提取特征数量的增加, STCCA-L 的分类准确率呈现出更稳定的变化趋势. 以 Caltech101 数据集为例, 当提取的特征数量大于 8 时, TCCA-O 与 TCCA-OS 的分类准确率呈现明显下降趋势, 推测其原因是特征数量过多导致的特征冗余问题, 而所提 STCCA-L 由于能够充分利用数据的图结构信息, 有效抑制了特征冗余, 从而维持了性能的稳定性. 此外, 从误差棒的大小分析可知, STCCA-L 显著小于其他对比方法, 进一步验证了该方法的可靠性.



(a) 3Sources **(b)** MSRC **(c)** BBCsport **(d)** Reusters

(e) Caltech101 **(f)** Handwritten **(g)** MNIST **(h)** Animal

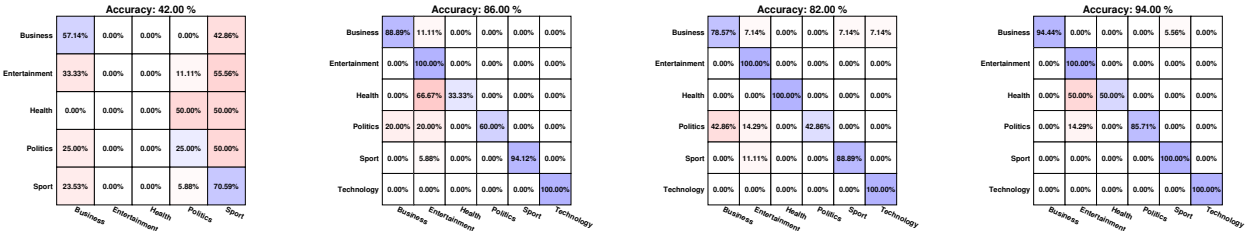
图 5.3: 不同维度下的分类准确率

5.4.3 消融研究

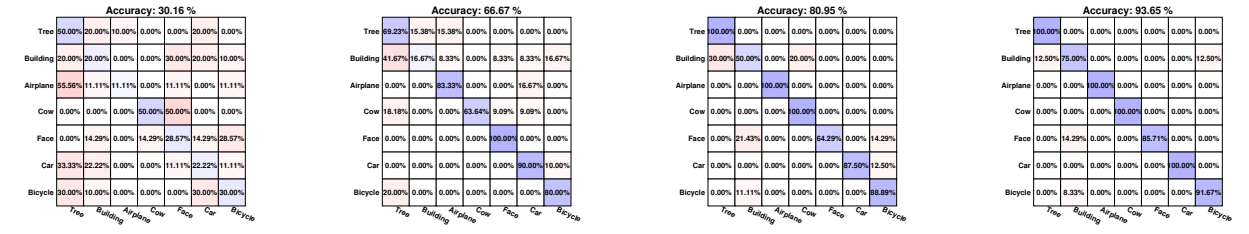
本节在 3Sources、MSRC 及 Caltech101 三个数据集上开展消融研究, 设置四组对照实验: (I) 移除正交约束, (II) 移除稀疏正则化, (III) 移除图正则化, (IV) 完整模型. 表 5.4 详细列出了各组对照实验的分类性能指标. 可清晰看出, 移除稀疏正则或图正则后, STCCA-L 的分类性能均出现下降. 移除正交约束后, 模型性能与完整 STCCA-L 存在显著差异, 这表明正交约束对模型具有决定性作用. 图 5.4 可视化了 STCCA-L 及其三个退化模型的分类混淆矩阵, 进一步验证了各组件的有效性.

表 5.4: 消融研究的结果对比 (%)

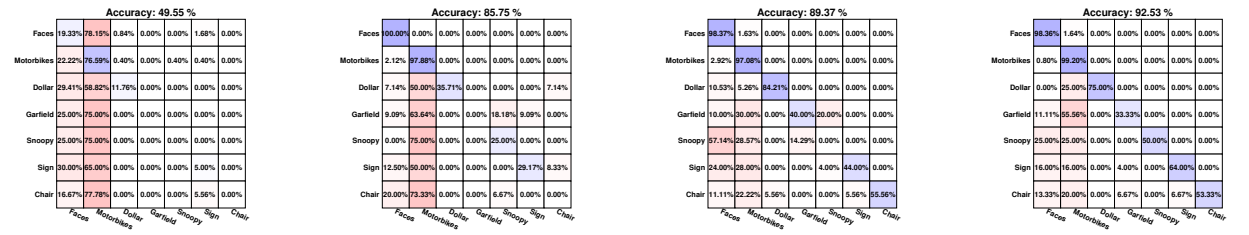
方案	3Sources		MSRC		Caltech101	
	Accuracy	F1-sorce	Accuracy	F1-score	Accuracy	F1-score
I	40.50±2.52	33.90±5.54	25.00±7.14	24.18±6.25	52.26±1.15	23.18±3.92
II	85.50±7.00	78.95±8.71	68.25±3.43	66.00±3.63	87.40±1.23	55.05±3.54
III	82.00±5.03	77.00±6.57	84.92±4.76	83.85±4.89	90.72±1.09	73.25±3.26
IV	**91.50±4.12**	**89.67±4.94**	**89.92±2.71**	**83.84±2.82**	**92.08±1.23**	**78.42±3.23**



(a) 3Sources (I) **(b)** 3Sources (II) **(c)** 3Sources (III) **(d)** 3Sources (IV)



(e) MSRC (I) **(f)** MSRC (II) **(g)** MSRC (III) **(h)** MSRC (IV)



(i) Caltech101 (I) **(j)** Caltech101 (II) **(k)** Caltech101 (III) **(l)** Caltech101 (IV)

图 5.4: 混淆矩阵的可视化对比

为验证所提图构造策略的有效性, 选取四种经典的方法作为对照, 包括高斯核 (Gaussian)、KNN、余弦相似度 (Cosine) 及稀疏表示 (Sparse), 同时本章方法记为 Adaptive. 表 5.5 的实验结果显示, 所提自适应邻居图在不同图构造方法下均能保持稳定的优异性能, 表明其对初始权重矩阵选择的鲁棒性.

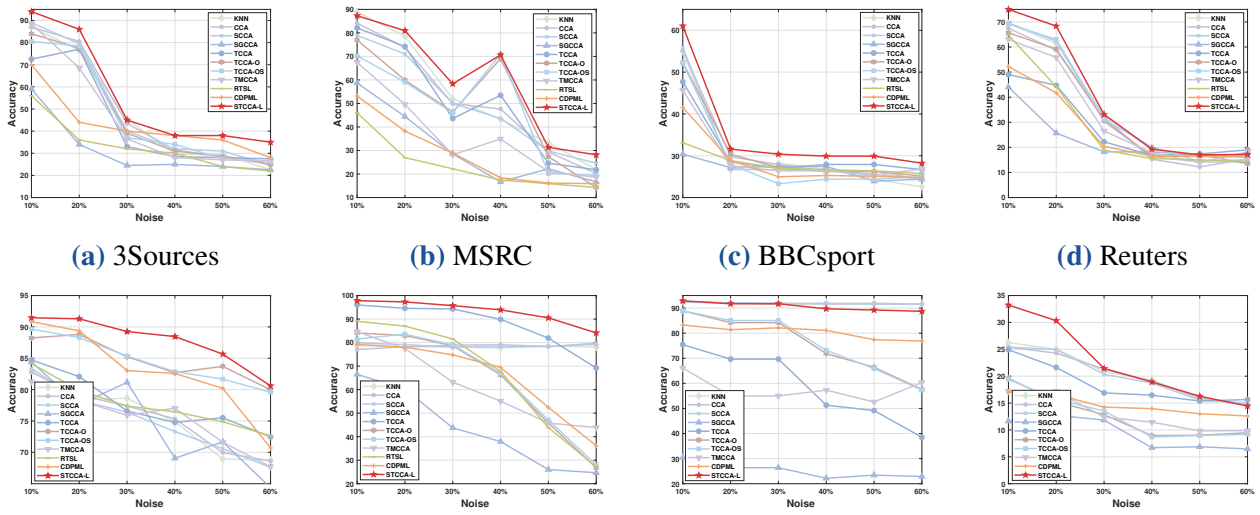
表 5.5: 图构造策略的消融研究 (%)

方法	3Sources		MSRC		Caltech101	
	Accuracy	F1-score	Accuracy	F1-score	Accuracy	F1-score
Gaussian	78.60±7.12	69.74±9.01	88.09±4.38	87.25±5.15	93.25±1.21	77.15±4.11
KNN	77.80±4.47	69.75±7.28	90.47±4.85	89.51±5.83	92.89±1.35	74.95±3.06
Cosine	78.40±6.98	68.88±8.97	89.36±2.70	88.84±3.09	93.34±0.89	76.81±2.37
Sparse	75.40±5.96	64.98±6.54	87.94±5.03)	87.21±5.89	93.19±0.87	76.63±2.93
Adaptive	**91.50±4.12**	**89.67±4.94**	**91.27±2.72**	**91.02±3.16**	**93.62±0.99**	**78.62±4.41**

5.4.4 讨论

(1) 鲁棒性分析

本节在八个多视角数据集上分别添加 10%-60% 比例的高斯噪声, 相应的分类准确率结果如图 5.5 所示. 显然, 所有对比方法在含噪数据集上的分类性能均出现不同程度的下降. 尽管所提 STCCA-L 的性能也有小幅下滑, 但相较于其他对比方法, 其性能下降幅度更小. 例如, 在 MNIST 数据集上, STCCA-L 的分类性能几乎不受噪声干扰.



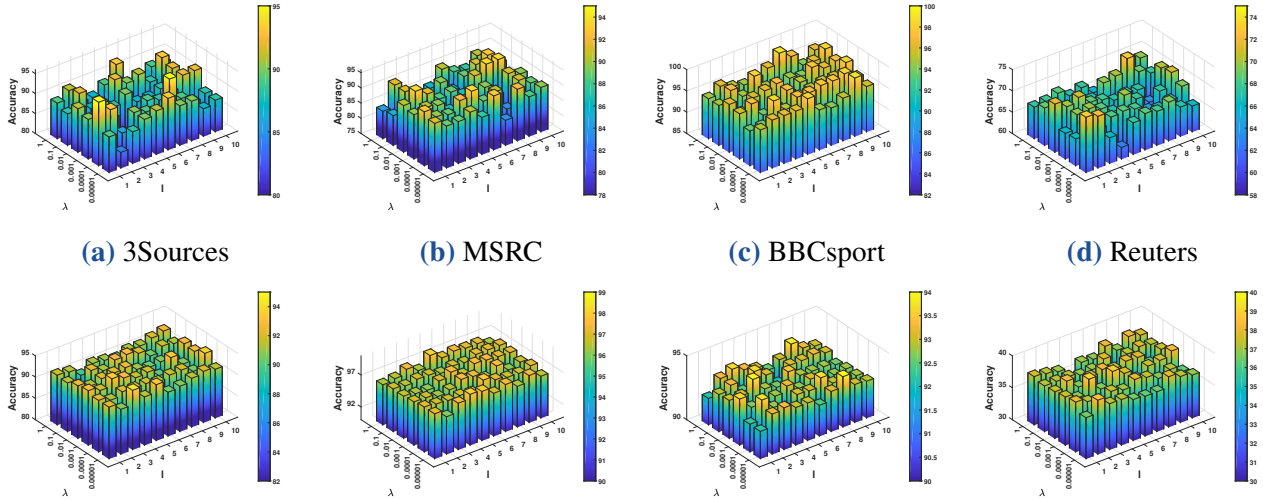
(a) 3Sources **(b)** MSRC **(c)** BBCsport **(d)** Reuters

(e) Caltech101 **(f)** Handwritten **(g)** MNIST **(h)** Animal

图 5.5: 随高斯噪声比例变化的分类准确率

(2) 参数分析

所提 STCCA-L 包含两个关键可调参数, 即多阶图的最大阶数 l 和稀疏度 λ. 图 5.6 展示了不同参数下的分类准确率结果, 其中 $\lambda = \{0.00001, 0.0001, \ldots, 1\}$ 以及 $l = \{1, 2, \ldots, 10\}$. 可以看出, 降低 λ 值有助于提升分类准确率. 例如, 在 BBCSport 数据集上, 当 $l = 3$ 且 $\lambda = 0.0001$ 时, 准确率达到 98.20%, 显著优于 $\lambda = 1$ 时获得的 94.50%. 参数 l 对性能的影响未呈现一致规律, 但实验发现, 当 l 在 3 到 7 的范围内时, 往往能取得最优性能.



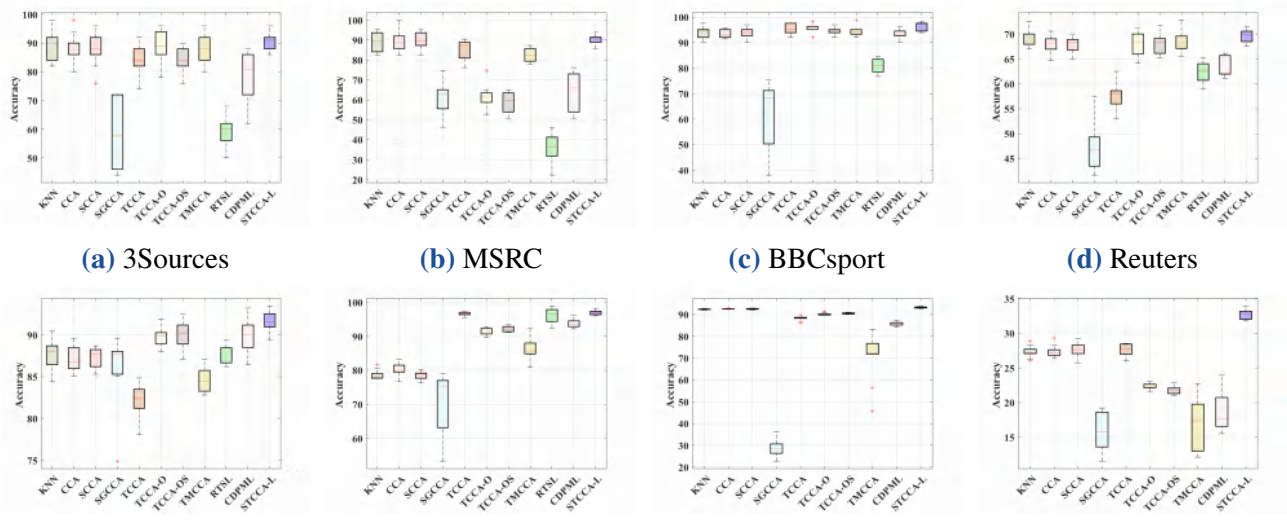
(a) 3Sources **(b)** MSRC **(c)** BBCsport **(d)** Reuters

(e) Caltech101 **(f)** Handwritten **(g)** MNIST **(h)** Animal

图 5.6: 不同参数对分类性能的影响

(3) 稳定性分析

图 5.7 给出了对比方法在八个数据集上的箱线图可视化结果. 从稳定性角度来看, 张量典型相关分析方法的表现显著优于其他对比方法. 与 TCCA-O 相比, 所提 STCCA-L 不仅稳定性更优, 还能获得更高的分类准确率.



(a) 3Sources **(b)** MSRC **(c)** BBCsport **(d)** Reuters

(e) Caltech101 **(f)** Handwritten **(g)** MNIST **(h)** Animal

图 5.7: 稳定性分析的可视化对比

(4) 时间分析

表 5.6 列出了所有张量典型相关分析方法在八个数据集上的平均时间消耗. 可以看出, 所提 STCCA-L 在大多数数据集上实现了具有竞争力的时间成本. 尽管 TCCA 和 TCCA-O 的运行时间通常更短, 但其分类性能较差. 值得注意的是, TMCCA 在 MNIST 和 Animal 数据集上的运行速度最慢, 表明其难以适配大规模数据集的实际应用需求. 总之, STCCA-L 在保证优异分类性能的同时, 具备良好的计算效率。

表 5.6: 张量典型相关分析方法的时间比较 (秒)

方法	3Sources	MSRC	BBCsport	Reuters	Caltech101	Handwritten	MNIST	Animal
TCCA	0.12	**0.89**	0.18	7.56	1.90	6.58	**3.18**	10.75
TCCA-O	0.11	1.01	0.18	7.77	1.35	6.79	3.31	10.95
TCCA-OS	0.69	5.71	0.53	10.95	4.03	8.42	3.82	25.83
TMCCA	0.34	15.98	1.32	45.64	2.41	26.53	54.39	97.90
STCCA-L	**0.10**	1.08	**0.17**	**7.47**	**1.27**	**6.43**	3.59	**10.60**

5.5 本章小结

本章针对现有典型相关分析方法鲁棒性差的问题, 提出了基于张量表示的多视角子空间学习方法. 将稀疏正则与多阶图拉普拉斯正则有机融合, 既有效抑制了特征冗余现象, 又充分挖掘并利用了不同视角数据的内在结构信息. 为求解由此产生的优化问题, 设计了一种流形近端梯度算法, 并引入子空间牛顿法对其进行加速. 从理论层面, 严格证明了该算法所生成的迭代序列能够收敛至问题的驻点. 实验结果表明, 所提方法优于现有主流的典型相关分析方法与其他多视角学习方法.

第 6 章 基于稀疏低秩对比学习的特征选择

传统基于稀疏主成分分析的方法普遍依赖欧氏距离构造损失函数, 难以有效刻画样本间的复杂关系. 为此, 本章提出了融合对比学习的双稀疏约束的无监督特征选择 (double sparsity constrained optimization feature selection with contrastive learning, DSCOFS-CL). 该模型利用对比学习构造损失函数, 并引入稀疏低秩约束增强投影空间中对样本关系的表征, 从而提高特征选择的性能. 针对该非凸非连续问题, 设计了基于梯度下降和硬阈值的近端交替最小化算法. 实验结果表明, 所提 DSCOFS-CL 在真实数据集上的平均聚类准确率提升了至少 1.70%.

6.1 引言

基于主成分分析 (principal component analysis, PCA)[89] 和稀疏主成分分析 (sparse principal component analysis, SPCA)[44] 的无监督特征选择方法受到学术界的广泛关注. 正如第 2 章所述, 作为经典的统计分析方法, 主成分分析的核心思想是通过正交的线性变换矩阵将原始高维数据投影到新的低维空间, 使投影后的数据方差最大, 从而在较低的维度下尽可能地保留数据的主要信息. 给定数据 $\boldsymbol{A} = [\boldsymbol{a}_1, \boldsymbol{a}_2, \ldots, \boldsymbol{a}_n] \in \mathbb{R}^{d\times n}$, 若将数据降维至 m 维, 可以记投影矩阵为 $\boldsymbol{X} = [\boldsymbol{x}_1, \boldsymbol{x}_2, \ldots, \boldsymbol{x}_m] \in \mathbb{R}^{d\times m}$, 其中 $\boldsymbol{x}_i \in \mathbb{R}^d$ 是投影矩阵 $\boldsymbol{X}$ 的第 i 个列向量, 也表示第 i 个投影方向. 同时, 每个投影方向还应满足 $\boldsymbol{x}_i^T\boldsymbol{x}_i = 1$ 和 $\boldsymbol{x}_i^T\boldsymbol{x}_j = 0\ (i \neq j)$. 从重构误差的角度看, 主成分分析的数学模型可表示为

$$\begin{aligned} \min_{\boldsymbol{X}} \quad & \|\boldsymbol{A} - \boldsymbol{X}\boldsymbol{X}^T\boldsymbol{A}\|_F^2 \\ \text{s.t.} \quad & \boldsymbol{X}^T\boldsymbol{X} = \boldsymbol{I}_m. \end{aligned} \tag{6.1}$$

最小化重构误差模型通常采用 Frobenius 范数作为度量标准, 而该范数隐含地假设各特征之间具有相同的尺度和重要性. 但是在实际数据中, 特征之间的尺度和分布往往存在差异性. 这意味着, 即使仅有少数噪声和离群值, 其平方差也可能显著增加, 甚至可能导致误差评估失真. 虽然稀疏主成分分析一定程度缓解了数据噪声和冗余的情况, 但是使用何种范数度量重构误差仍是一个值得深入研究的问题.

为了提高稳定性, Ke 等[90] 采用 ℓ_1 范数度量重构误差, 其数学模型为

$$\begin{aligned} \min_{\boldsymbol{X}} \quad & \|\boldsymbol{A} - \boldsymbol{X}\boldsymbol{X}^T\boldsymbol{A}\|_1 \\ \text{s.t.} \quad & \boldsymbol{X}^T\boldsymbol{X} = \boldsymbol{I}_m. \end{aligned} \tag{6.2}$$

与 Frobenius 范数不同, ℓ_1 范数刻画的是所有元素的绝对值之和, 因此对少数噪声和离群值的影响相对较小. 当数据中存在较多不相关或冗余特征时, ℓ_1 范数倾向于找到较少的非零元素.

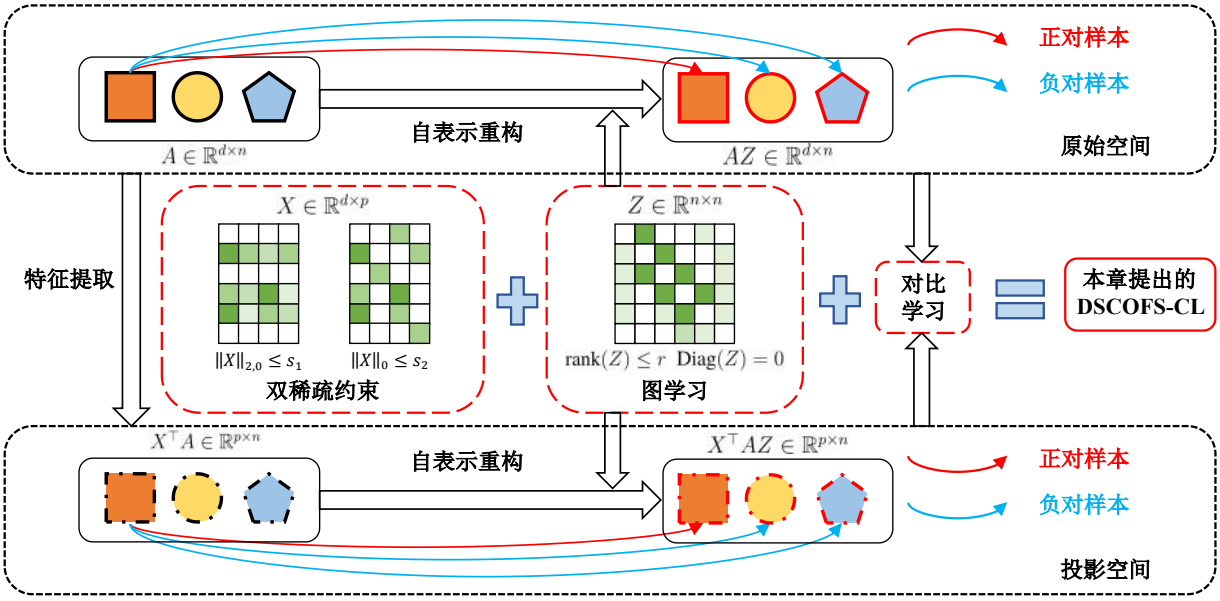


图 6.1: 所提 DSCOFS-CL 特征选择的流程图

值得注意的是,ℓ_1 范数不具有旋转不变性[91], 且依赖数据本身结构的方向性. 此外, ℓ_1 范数不能正确计算重构数据和原始数据之间的欧氏距离. 为了解决上述问题, Wang 等[92] 使用 $\ell_{2,p}$ $(0 < p < 2)$ 范数度量重构误差, 其数学模型为

$$\begin{aligned} \min_{\boldsymbol{X}} \quad & \|\boldsymbol{A} - \boldsymbol{X}\boldsymbol{X}^T\boldsymbol{A}\|_{2,p}^p \\ \text{s.t.} \quad & \boldsymbol{X}^T\boldsymbol{X} = \boldsymbol{I}_m. \end{aligned} \tag{6.3}$$

这里, $\ell_{2,p}$ 范数与 Frobenius 范数相比, 降低了少数噪声和离群值的影响. 因此, $\ell_{2,p}$ 范数在保留原本旋转不变性的同时又具有一定的鲁棒性. 在实际的应用中, p 常取值为 $(0, 1)$, 而当 p 取一个非常小的值时, $\ell_{2,p}$ 范数将会失去区分正确样本的能力. 综上所述, 现有基于不同损失函数度量重构误差的方法, 依然无法充分反映样本间复杂关系, 这可能导致重构误差评估不合理, 从而降低模型的判别性能, 难以满足下游任务的需求.

对比学习[93] 作为一种新兴的人工智能技术, 其核心在于通过缩小相似样本 (正样本对) 的距离, 同时扩大不同类别样本 (负样本对) 之间的差异, 从而在无监督条件下挖掘数据中的潜在类别结构. 最近, Zhang 等[94] 首先建立用于定义正负样本的对比学习图, 然后通过最小化对比损失函数来确定投影矩阵. 此外, Zhou 等[95] 引入对比学习损失[96] 来度量主成分分析的重构误差, 同时对投影矩阵和自表示矩阵施加 $\ell_{1,2}$ 范数约束, 实现了更好的无监督特征选择性能. 随后, Zhou 等[97] 通过在原始空间和投影空间中联合学习自表示矩阵, 进一步提出基于联合自表示图学习的无监督特征选择方法.

受此启发, 为最大限度保留图的全局结构, 克服现有方法中稀疏结构表示不充分、局部特征辨别不准确等问题, 本章提出了融合对比学习的双稀疏约束的无监督特征选择方法, 其流程如图 6.1 所示. 本章的主要贡献为

(1) 通过同时引入对比学习、双稀疏约束及低秩约束, 并采用自表示重构框架, 构建了新的基于主成分分析的特征选择模型.

(2) 设计了有效的近端交替最小化算法, 其所有子问题具有良好的解析表达式, 或可被高效求解器快速计算.

(3) 数值实验评估了所提方法的性能, 验证了对比学习框架的作用, 并表明了稀疏约束与低秩约束的有效性.

6.2 数学模型

6.2.1 对比学习

给定重构数据 $\hat{\boldsymbol{A}} = [\hat{\boldsymbol{a}}_1, \hat{\boldsymbol{a}}_2, \ldots, \hat{\boldsymbol{a}}_n] \in \mathbb{R}^{d\times n}$. 基于负样本共享采样策略[98], 对于样本 $\boldsymbol{a}_i$, 其正样本为对应的重构样本 $\hat{\boldsymbol{a}}_i$, 令剩余的 $2(n-1)$ 为负样本. 为了度量对比学习损失误差, 可以使用归一化温度缩放交叉熵损失[99]. 具体地, 对于原始数据 $\boldsymbol{a}_i$, 其交叉熵损失为

$$L_c(\boldsymbol{a}_i) = -\log \frac{\exp(s(\boldsymbol{a}_i, \hat{\boldsymbol{a}}_i)/\tau)}{\sum\limits_{j=1, j\neq i}^{n} \exp(s(\boldsymbol{a}_i, \boldsymbol{a}_j)/\tau) + \sum\limits_{j=1}^{n} \exp(s(\boldsymbol{a}_i, \hat{\boldsymbol{a}}_j)/\tau)}, \tag{6.4}$$

其中, τ 是用来控制相似度计算的温度参数, $s(\boldsymbol{a}_i, \hat{\boldsymbol{a}}_j)$ 为相似度度量, 这里取

$$s(\boldsymbol{a}_i, \hat{\boldsymbol{a}}_j) = \boldsymbol{a}_i^T \hat{\boldsymbol{a}}_j. \tag{6.5}$$

类似地, 对于重构数据 $\hat{\boldsymbol{a}}_i$, 其交叉熵损失为

$$L_c(\hat{\boldsymbol{a}}_i) = -\log \frac{\exp(s(\hat{\boldsymbol{a}}_i, \boldsymbol{a}_i)/\tau)}{\sum\limits_{j=1, j\neq i}^{n} \exp(s(\hat{\boldsymbol{a}}_i, \hat{\boldsymbol{a}}_j)/\tau) + \sum\limits_{j=1}^{n} \exp(s(\hat{\boldsymbol{a}}_i, \boldsymbol{a}_j)/\tau)}. \tag{6.6}$$

对于交叉熵损失, 分母越大表示样本相似度越高, 相反地, 分母越小表示样本相似度越低, 即负样本对的相似度越低. 对比学习的本质是最大化正对之间的相似性, 同时最小化负对之间的相似性, 即交叉熵损失越小. 因此, 对比学习损失函数可表示为

$$L_c(\boldsymbol{A}, \hat{\boldsymbol{A}}) = \frac{1}{2n} \sum_{i=1}^{n} (L_c(\boldsymbol{a}_i) + L_c(\hat{\boldsymbol{a}}_i)). \tag{6.7}$$

图 6.2 展示了分别利用欧氏距离度量和对比学习损失得到的投影空间结果. 可以看出, 利用欧氏距离评估相似度可能会导致不同类别样本之间的相似度较高, 而同类别的相似度较低, 这使得投影空间中不同类别样本之间的划分不清晰. 相比之下, 采用对比学习损失设定正负样本对后, 投影空间中的同类样本由于包含了大量相似信息而更加紧凑, 且不同类别样本之间的边界更加明显, 从而提升了判别性能.

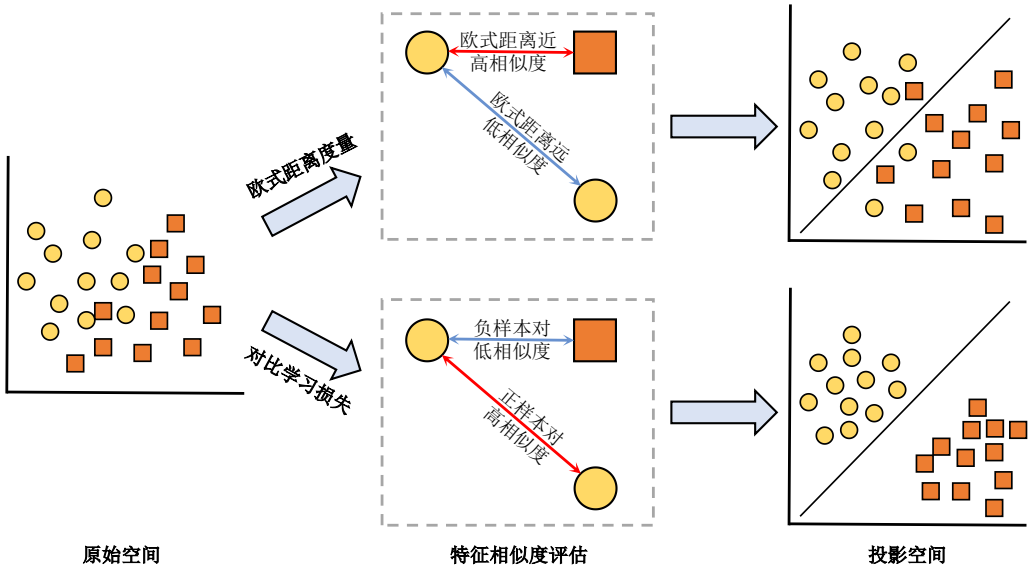


图 6.2: 不同度量得到的投影空间

6.2.2 稀疏主成分分析

通过引入稀疏约束, Nie 等[20] 提出特征稀疏约束主成分分析 (feature-sparsity constrained PCA, FSPCA) , 即

$$\begin{aligned} \min_{\boldsymbol{X}} \quad & \|\boldsymbol{A}-\boldsymbol{X}\boldsymbol{X}^T\boldsymbol{A}\|_F^2 \\ \text{s.t.} \quad & \boldsymbol{X}^T\boldsymbol{X}=\boldsymbol{I}_m,\ \|\boldsymbol{X}\|_{2,0}\le s, \end{aligned} \tag{6.8}$$

其中, $s>0$ 表示非零行的个数, 对应所选择特征的数量. 注意, Li 等[9] 考虑了上述模型非凸松弛的情形, 即 $\ell_{2,p}(0<p\le 1)$ 范数正则, 提出了用于特征选择的稀疏主成分分析 (sparse PCA for feature selection, SPCAFS) 模型.

为了探究数据的复杂稀疏结构, Xiu 等[100] 提出了双稀疏约束 (即行稀疏和元素稀疏) 优化特征选择 (double sparsity constrained optimization feature selection, DSCOFS), 具体形式为

$$\begin{aligned} \min_{\boldsymbol{X}} \quad & \|\boldsymbol{A}-\boldsymbol{X}\boldsymbol{X}^T\boldsymbol{A}\|_F^2 \\ \text{s.t.} \quad & \boldsymbol{X}^T\boldsymbol{X}=\boldsymbol{I}_m,\ \|\boldsymbol{X}\|_{2,0}\le s_1,\ \|\boldsymbol{X}\|_0\le s_2, \end{aligned} \tag{6.9}$$

其中, $s_1>0$ 表示非零行的个数, $s_2>0$ 表示非零元素的个数.

6.2.3 构建模型

正如前文所述, 以 Frobenius 范数度量损失的无监督学习方法难以有效挖掘数据中潜在的类别结构. 为此, 本章在式 (6.9) 的基础上, 引入如下对比学习机制构建新型损失函数. 这种改进不仅保留了原始模型对数据稀疏结构的捕捉能力, 更重要的是通过对比学习策略可以显式

地建模样本间的相似性和差异性, 进而提升无监督特征选择的性能. 基于对比学习的稀疏低秩无监督特征选择 (DSCOFS with contrastive learning, DSCOFS-CL) 的数学模型为

$$\begin{aligned} \min_{\boldsymbol{X},\boldsymbol{Z}} \quad & \lambda L_c(\boldsymbol{A}, \boldsymbol{AZ}) + (1-\lambda) L_c(\boldsymbol{X}^T\boldsymbol{A}, \boldsymbol{X}^T\boldsymbol{AZ}) \\ \text{s.t.} \quad & \boldsymbol{X}^T\boldsymbol{X} = \boldsymbol{I}_m,\ \|\boldsymbol{X}\|_{2,0} \le s_1,\ \|\boldsymbol{X}\|_0 \le s_2, \\ & \operatorname{rank}(\boldsymbol{Z}) \le r,\ \operatorname{diag}(\boldsymbol{Z}) = 0, \end{aligned} \tag{6.10}$$

其中, $\boldsymbol{Z} \in \mathbb{R}^{n\times n}$ 为自表示矩阵, $\boldsymbol{AZ}$ 和 $\boldsymbol{X}^T\boldsymbol{AZ}$ 分别表示原始空间数据 $\boldsymbol{A}$ 和投影空间数据 $\boldsymbol{X}^T\boldsymbol{A}$ 通过自表示学习的重构数据. $0 < \lambda < 1$ 用于调节原始空间和投影空间参与自表示图学习的比重. $\operatorname{diag}(\boldsymbol{Z}) = 0$ 表示矩阵 $\boldsymbol{Z}$ 的对角线元素为零, 用于避免一般解 $\boldsymbol{Z} = \boldsymbol{I}_n$, 其中 $\boldsymbol{I}_n$ 为 $n \times n$ 单位矩阵. $r > 0$ 用于控制矩阵 $\boldsymbol{Z}$ 的秩, 可根据实际需求进行设置.

与现有无监督特征选择方法相比, DSCOFS-CL 具有以下特点: (1) 对投影矩阵嵌入约束 $\|\boldsymbol{X}\|_{2,0} \le s_1$ 和 $\|\boldsymbol{X}\|_0 \le s_2$, 可以去除无关特征和噪音, 即双稀疏约束; (2) 对自表示矩阵引入低秩约束 $\operatorname{rank}(\boldsymbol{Z}) \le r$ 和 $\operatorname{diag}(\boldsymbol{Z}) = 0$, 可以保留数据的全局结构, 即图学习; (3) 利用交叉熵衡量正负样本之间的相似性, 即对比学习.

6.3 优化算法

由于式 (6.10) 是非凸非连续的优化问题, 且对比学习损失的计算较为繁琐, 于是引入辅助变量 $\boldsymbol{P} = \boldsymbol{X}$、$\boldsymbol{Q} = \boldsymbol{X}$ 和 $\boldsymbol{Y} = \boldsymbol{Z}$, 将式 (6.10) 等价转化为

$$\begin{aligned} \min_{\boldsymbol{X},\boldsymbol{Z},\boldsymbol{Y},\boldsymbol{P},\boldsymbol{Q}} \quad & \lambda L_c(\boldsymbol{A}, \boldsymbol{AZ}) + (1-\lambda) L_c(\boldsymbol{X}^T\boldsymbol{A}, \boldsymbol{X}^T\boldsymbol{AZ}) \\ \text{s.t.} \quad & \boldsymbol{X}^T\boldsymbol{X} = \boldsymbol{I}_m,\ \boldsymbol{P} = \boldsymbol{X},\ \boldsymbol{Q} = \boldsymbol{X},\ \boldsymbol{Y} = \boldsymbol{Z}, \\ & \boldsymbol{P} \in \mathcal{S}_1,\ \boldsymbol{Q} \in \mathcal{S}_2,\ \boldsymbol{Y} \in \mathcal{R},\ \boldsymbol{Z} \in \mathcal{D}, \end{aligned} \tag{6.11}$$

其中

$$\begin{aligned} &\mathcal{S}_1 = \{\boldsymbol{P} \in \mathbb{R}^{d\times m} \mid \|\boldsymbol{P}\|_{2,0} \le s_1\},\ \mathcal{S}_2 = \{\boldsymbol{Q} \in \mathbb{R}^{d\times m} \mid \|\boldsymbol{Q}\|_0 \le s_2\}, \\ &\mathcal{R} = \{\boldsymbol{Y} \in \mathbb{R}^{n\times n} \mid \operatorname{rank}(\boldsymbol{Y}) \le r\},\ \mathcal{D} = \{\boldsymbol{Z} \in \mathbb{R}^{n\times n} \mid \operatorname{diag}(\boldsymbol{Z}) = 0\}. \end{aligned} \tag{6.12}$$

利用惩罚函数方法, 式 (6.11) 可转化为

$$\begin{aligned} \min_{\boldsymbol{X},\boldsymbol{Z},\boldsymbol{Y},\boldsymbol{P},\boldsymbol{Q}} \quad & \lambda L_c(\boldsymbol{A}, \boldsymbol{AZ}) + (1-\lambda) L_c(\boldsymbol{X}^T\boldsymbol{A}, \boldsymbol{X}^T\boldsymbol{AZ}) + \mu\|\boldsymbol{X}^T\boldsymbol{X} - \boldsymbol{I}_m\|_F^2 \\ & + \alpha\|\boldsymbol{Z} - \boldsymbol{Y}\|_F^2 + \beta\|\boldsymbol{X} - \boldsymbol{P}\|_F^2 + \gamma\|\boldsymbol{X} - \boldsymbol{Q}\|_F^2 \\ \text{s.t.} \quad & \boldsymbol{Z} \in \mathcal{D},\ \boldsymbol{Y} \in \mathcal{R},\ \boldsymbol{P} \in \mathcal{S}_1,\ \boldsymbol{Q} \in \mathcal{S}_2, \end{aligned} \tag{6.13}$$

其中, $\mu, \alpha, \beta, \gamma > 0$ 是惩罚参数. 设 $\boldsymbol{X}^k$、$\boldsymbol{Z}^k$、$\boldsymbol{Y}^k$、$\boldsymbol{P}^k$ 和 $\boldsymbol{Q}^k$ 是第 k 次更新的变量, 同时在迭代过程中, 引入近端参数 $0 < \tau_i < \infty$ $(i = 1, 2, 3, 4, 5)$.

6.3.1 更新 X

经过简化, 得到 $\boldsymbol{X}$ 的子问题为

$$\begin{aligned}\min_{\boldsymbol{X}} \quad & (1-\lambda)L_c(\boldsymbol{X}^T\boldsymbol{A}, \boldsymbol{X}^T\boldsymbol{A}\boldsymbol{Z}^k) + \mu\|\boldsymbol{X}^T\boldsymbol{X} - \boldsymbol{I}_m\|_F^2 \\ & + \beta\|\boldsymbol{X} - \boldsymbol{P}^k\|_F^2 + \gamma\|\boldsymbol{X} - \boldsymbol{Q}^k\|_F^2 + \tau_1\|\boldsymbol{X} - \boldsymbol{X}^k\|_F^2.\end{aligned} \tag{6.14}$$

式 (6.14) 是一个无约束的优化问题, 可以通过梯度下降法求解. 设目标函数为 $f(X)$, 则 $\boldsymbol{X}^{k+1}$ 的更新可以根据下式计算

$$\boldsymbol{X}^{k+1} = \boldsymbol{X}^k - \eta_1 \frac{\partial f(\boldsymbol{X})}{\partial \boldsymbol{X}}, \tag{6.15}$$

其中, $\eta_1 > 0$ 表示更新的步长. 由梯度定义, 对比学习损失函数式 (6.7) 的梯度为

$$\frac{\partial L_c(\boldsymbol{A}, \hat{\boldsymbol{A}})}{\partial X_{pq}} = \frac{1}{2n}\sum_{i=1}^{n}\left(\frac{\partial L_c(\boldsymbol{a}_i)}{\partial X_{pq}} + \frac{\partial L_c(\hat{\boldsymbol{a}}_i)}{\partial X_{pq}}\right). \tag{6.16}$$

根据文献[95], 令 $\boldsymbol{g} = [\boldsymbol{a}_i^T\boldsymbol{X}\boldsymbol{X}^T\boldsymbol{a}_1, \boldsymbol{a}_i^T\boldsymbol{X}\boldsymbol{X}^T\boldsymbol{a}_2, \ldots, \boldsymbol{a}_i^T\boldsymbol{X}\boldsymbol{X}^T\boldsymbol{a}_n] \in \mathbb{R}^n$, 则上式可表示为

$$\begin{aligned}\frac{\partial L_c(\boldsymbol{A}, \hat{\boldsymbol{A}})}{\partial X_{pq}} = & -\frac{t + \sum_{j=1}^{n}\exp(\boldsymbol{g}_j/\tau)}{\exp(\boldsymbol{g}_i/\tau)} \times \left(\frac{\partial(\exp(\boldsymbol{g}_i/\tau))}{\partial X_{pq}} + \frac{\partial(1/(t + \sum_{j=1}^{n}\exp(\boldsymbol{g}_j/\tau)))}{\partial X_{pq}}\right) \\ & - \frac{\sum_{j=1:j\neq i}^{n}\exp(\boldsymbol{g}_i/\tau) + \sum_{j=1}^{n}\exp(\boldsymbol{g}_j/\tau)}{\exp(\boldsymbol{g}_i/\tau)} \\ & \times \left(\frac{\partial(\exp(\boldsymbol{g}_i/\tau))}{\partial X_{pq}} + \frac{\partial(1/(\sum_{j=1:j\neq i}^{n}\exp(\boldsymbol{g}_i/\tau) + \sum_{j=1}^{n}\exp(\boldsymbol{g}_j/\tau)))}{\partial X_{pq}}\right),\end{aligned} \tag{6.17}$$

其中 X_{pq} 表示矩阵 $\boldsymbol{X}$ 的第 p 行 q 列元素, $t = \sum_{j=1:j\neq i}^{n}\exp(\boldsymbol{a}_i^T\boldsymbol{X}\boldsymbol{X}^T\boldsymbol{a}_j/\tau)$, 以及

$$\frac{\partial \boldsymbol{g}_j}{\partial X_{pq}} = \frac{\partial(\boldsymbol{a}_i^T\boldsymbol{X}\boldsymbol{X}^T\boldsymbol{a}_j)}{\partial X_{pq}} = \sum_{l=1}^{d} X_{lq}(A_{li}A_{pj} + A_{pi}A_{lj}). \tag{6.18}$$

因此, 要计算目标函数的梯度只需将式 (6.16) 中的 $\boldsymbol{A}$ 和 $\hat{\boldsymbol{A}}$ 分别替换为 $\boldsymbol{X}^T\boldsymbol{A}$ 和 $\boldsymbol{X}^T\boldsymbol{A}\boldsymbol{Z}^k$. 不过由于上述计算较为繁琐, 本章在迭代过程中直接使用 Pytorch 的自动推导机制[95].

6.3.2 更新 Z

关于 $\boldsymbol{Z}$ 的子问题可整理为

$$\begin{aligned}\min_{\boldsymbol{Z}} \quad & \lambda L_c(\boldsymbol{A}, \boldsymbol{AZ}) + (1-\lambda) L_c(\boldsymbol{X}^{k+1,T}\boldsymbol{A}, \boldsymbol{X}^{k+1,T}\boldsymbol{AZ}) \\ & + \alpha\|\boldsymbol{Z} - \boldsymbol{Y}^k\|_F^2 + \tau_2\|\boldsymbol{Z} - \boldsymbol{Z}^k\|_F^2 \\ \text{s.t.} \quad & \operatorname{diag}(\boldsymbol{Z}) = \boldsymbol{0}.\end{aligned} \tag{6.19}$$

为了使 $\boldsymbol{Z}$ 的对角元素为零, 令 $\boldsymbol{Z} = \boldsymbol{M} - \operatorname{Diag}(\boldsymbol{M})$, 其中 $\operatorname{Diag}(\boldsymbol{M})$ 是由 $\boldsymbol{M}$ 对角元素组成的对角矩阵. 于是, 式 (6.19) 可以改写为

$$\begin{aligned}\min_{\boldsymbol{M}} \quad & \lambda L_c(\boldsymbol{A}, \boldsymbol{A}(\boldsymbol{M} - \operatorname{Diag}(\boldsymbol{M}))) + (1-\lambda) L_c(\boldsymbol{X}^{k+1,T}\boldsymbol{A}, \boldsymbol{X}^{k+1,T}\boldsymbol{A}(\boldsymbol{M} - \operatorname{Diag}(\boldsymbol{M}))) \\ & + \alpha\|\boldsymbol{M} - \operatorname{Diag}(\boldsymbol{M}) - \boldsymbol{Y}^k\|_F^2 + \tau_2\|\boldsymbol{M} - \operatorname{Diag}(\boldsymbol{M}) - \boldsymbol{M}^k + \operatorname{Diag}(\boldsymbol{M}^k)\|_F^2.\end{aligned} \tag{6.20}$$

设式 (6.20) 的目标函数为 $h(\boldsymbol{M})$, 则 $\boldsymbol{M}^{k+1}$ 可以通过梯度下降计算

$$\boldsymbol{M}^{k+1} = \boldsymbol{M}^k - \eta_2 \frac{\partial h(\boldsymbol{M})}{\partial \boldsymbol{M}}, \tag{6.21}$$

其中, $\eta_2 > 0$ 表示更新的步长. 从而可以得到

$$\boldsymbol{Z}^{k+1} = \boldsymbol{M}^{k+1} - \operatorname{Diag}(\boldsymbol{M}^{k+1}). \tag{6.22}$$

注意, 此处迭代过程依旧直接使用 Pytorch 的自动推导机制.

6.3.3 更新 $\boldsymbol{Y}$

更新完 $\boldsymbol{X}, \boldsymbol{Z}$, 接下来计算

$$\begin{aligned}\min_{\boldsymbol{Y}} \quad & \|\boldsymbol{Z}^{k+1} - \boldsymbol{Y}\|_F^2 + \tau_3\|\boldsymbol{Y} - \boldsymbol{Y}^k\|_F^2 \\ \text{s.t.} \quad & \operatorname{rank}(\boldsymbol{Y}) \le r.\end{aligned} \tag{6.23}$$

将式 (6.23) 的两项 Frobenius 范数合并, 得到

$$\begin{aligned}\min_{\boldsymbol{Y}} \quad & \|\frac{\boldsymbol{Z}^{k+1} + \tau_3\boldsymbol{Y}^k}{1+\tau_3} - \boldsymbol{Y}\|_F^2 \\ \text{s.t.} \quad & \operatorname{rank}(\boldsymbol{Y}) \le r.\end{aligned} \tag{6.24}$$

式 (6.24) 可以通过奇异值分解求解. 设 $\boldsymbol{B}^{k+1} = \frac{\boldsymbol{Z}^{k+1}+\tau_3\boldsymbol{Y}^k}{1+\tau_3}$, 则有

$$\boldsymbol{B}^{k+1} = \boldsymbol{U}\boldsymbol{\Sigma}\boldsymbol{V}^T, \tag{6.25}$$

其中 $\boldsymbol{U}, \boldsymbol{V}, \boldsymbol{\Sigma} \in \mathbb{R}^{n\times n}$, $\boldsymbol{\Sigma}$ 的对角元素是矩阵 $\boldsymbol{B}^{k+1}$ 按降序排列的奇异值. 取 $\boldsymbol{U}$ 和 $\boldsymbol{V}^T$ 的前 r 列和前 r 行分别记为 $\boldsymbol{U}_r \in \mathbb{R}^{n\times r}$ 和 $\boldsymbol{V}_r^T \in \mathbb{R}^{r\times n}$, 取 $\boldsymbol{\Sigma}$ 的前 r 个奇异值得到 $\boldsymbol{\Sigma}_r \in \mathbb{R}^{r\times r}$. 根据 Eckart-Young 逼近引理[101] 得, $\boldsymbol{Y}^{k+1}$ 的解析形式为

$$\boldsymbol{Y}^{k+1} = \boldsymbol{U}_r\boldsymbol{\Sigma}_r\boldsymbol{V}_r^T. \tag{6.26}$$

6.3.4 更新 P

关于 $\boldsymbol{P}$ 子问题, 可简化为

$$\begin{aligned} \min_{\boldsymbol{P}} \quad & \|\boldsymbol{X}^{k+1}-\boldsymbol{P}\|_F^2+\tau_4\|\boldsymbol{P}-\boldsymbol{P}^k\|_F^2 \\ \text{s.t.} \quad & \|\boldsymbol{P}\|_{2,0}\leq s_1, \end{aligned} \tag{6.27}$$

等价于

$$\begin{aligned} \min_{\boldsymbol{P}} \quad & \|\frac{\boldsymbol{X}^{k+1}+\tau_4\boldsymbol{P}^k}{1+\tau_4}-\boldsymbol{P}\|_F^2 \\ \text{s.t.} \quad & \|\boldsymbol{P}\|_{2,0}\leq s_1. \end{aligned} \tag{6.28}$$

设 $\boldsymbol{C}^{k+1}=\frac{\boldsymbol{X}^{k+1}+\tau_4\boldsymbol{P}^k}{1+\tau_4}$, 计算 $\boldsymbol{C}^{k+1}$ 每一行的 ℓ_2 范数 $\|(\boldsymbol{c}^i)^{k+1}\|_2$, 并将其中第 s_1 大的值记为 $t_{s_1}^{k+1}$. 考虑到 $\ell_{2,0}$ 范数的行稀疏性[102], $\boldsymbol{P}^{k+1}$ 的解析形式为

$$(\boldsymbol{p}^i)^{k+1}=\begin{cases} (\boldsymbol{c}^i)^{k+1}, & \|(\boldsymbol{c}^i)^{k+1}\|_2\geq t_{s_1}^{k+1}, \\ \boldsymbol{0}, & \|(\boldsymbol{c}^i)^{k+1}\|_2< t_{s_1}^{k+1}. \end{cases} \tag{6.29}$$

6.3.5 更新 Q

类似地, 计算

$$\begin{aligned} \min_{\boldsymbol{Q}} \quad & \|\boldsymbol{X}^{k+1}-\boldsymbol{Q}\|_F^2+\tau_5\|\boldsymbol{Q}-\boldsymbol{Q}^k\|_F^2 \\ \text{s.t.} \quad & \|\boldsymbol{Q}\|_0\leq s_2, \end{aligned} \tag{6.30}$$

等价于

$$\begin{aligned} \min_{\boldsymbol{Q}} \quad & \|\frac{\boldsymbol{X}^{k+1}+\tau_5\boldsymbol{Q}^k}{1+\tau_5}-\boldsymbol{Q}\|_F^2 \\ \text{s.t.} \quad & \|\boldsymbol{Q}\|_0\leq s_2. \end{aligned} \tag{6.31}$$

记 $\boldsymbol{D}^{k+1}=\frac{\boldsymbol{X}^{k+1}+\tau_5\boldsymbol{Q}^k}{1+\tau_5}$, 对 $\boldsymbol{D}^{k+1}$ 取绝对值, 并将其中第 s_2 大的绝对值记为 $t_{s_2}^{k+1}$. 通过硬阈值算子[103] 可以直接得到 $\boldsymbol{Q}^{k+1}$ 的解析形式为

$$Q_{ij}^{k+1}=\begin{cases} D_{ij}^{k+1}, & |D_{ij}^{k+1}|\geq t_{s_2}^{k+1}, \\ 0, & |D_{ij}^{k+1}|< t_{s_2}^{k+1}. \end{cases} \tag{6.32}$$

综上所述, 求解式 (6.13) 的算法框架可以概括为算法 3.

算法 3 求解式 (6.13) 的近端交替最小化算法

输入: 数据 $\boldsymbol{A}$, 参数 $\lambda, \mu, \alpha, \beta, \gamma, s_1, s_2, r, \tau_i$ $(i = 1, 2, 3, 4, 5)$ 以及 m

初始化: 令 $k = 0$, 根据初始化策略得到 $(\boldsymbol{X}^0, \boldsymbol{Z}^0, \boldsymbol{Y}^0, \boldsymbol{P}^0, \boldsymbol{Q}^0)$ 以及 $\boldsymbol{M}^0$

当 未收敛 **时**

1: 通过式 (6.15) 得到 $\boldsymbol{X}^{k+1}$
2: 通过式 (6.22) 得到 $\boldsymbol{Z}^{k+1}$
3: 通过式 (6.26) 得到 $\boldsymbol{Y}^{k+1}$
4: 通过式 (6.29) 得到 $\boldsymbol{P}^{k+1}$
5: 通过式 (6.32) 得到 $\boldsymbol{Q}^{k+1}$

结束循环

输出: $(\boldsymbol{X}^{k+1}, \boldsymbol{Z}^{k+1}, \boldsymbol{Y}^{k+1}, \boldsymbol{P}^{k+1}, \boldsymbol{Q}^{k+1})$

6.3.6 复杂度分析

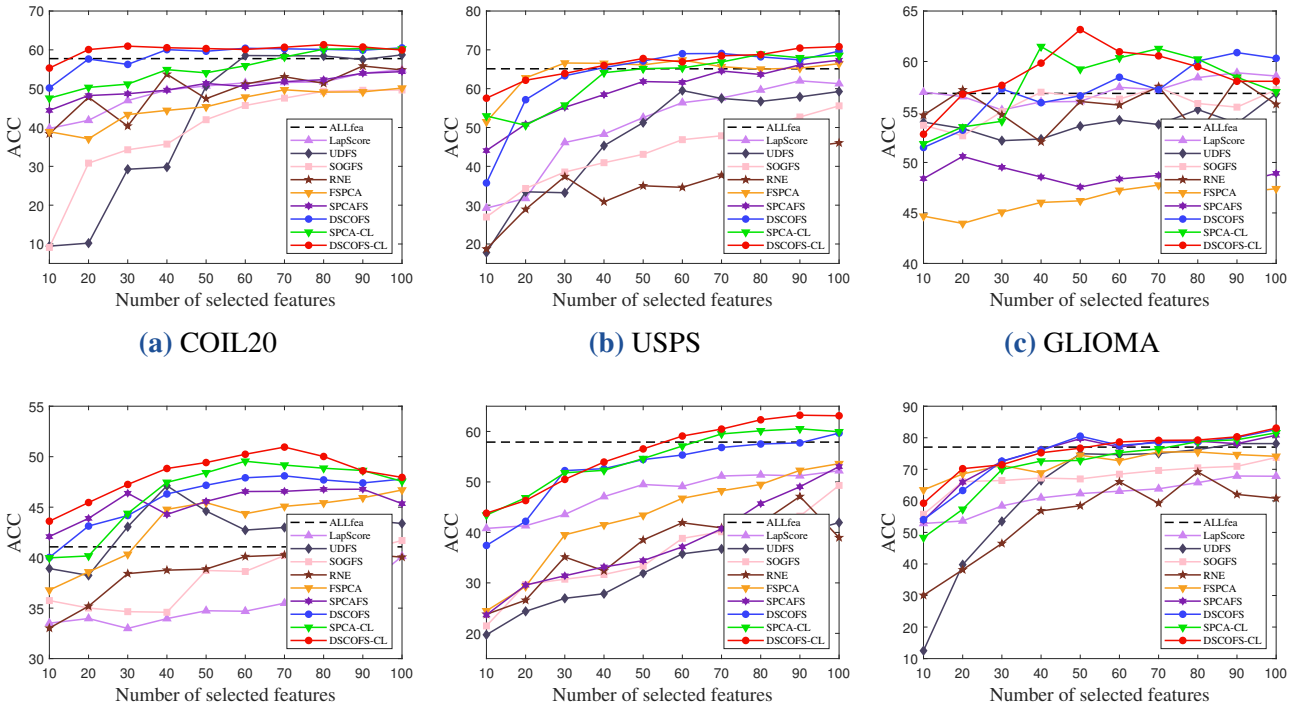
对于算法 3, 计算主要体现在五个子问题的迭代过程. 在更新 $\boldsymbol{Y}$ 子问题时, 需要进行奇异值分解, 其标准算法复杂度通常为 $O(n^3)$, 这主要归因于特征值和特征向量的计算过程. 在更新 $\boldsymbol{P}$ 子问题时, 需要先计算 $\boldsymbol{C}^{k+1}$ 再将其投影到行稀疏集合上, 因此计算量为 $O(dm)$ 和 $O(d\log s_1)$. 类似地, 在更新 $\boldsymbol{Q}$ 子问题时, 需要先计算 $\boldsymbol{D}^{k+1}$ 再将其投影到稀疏集合上, 因此计算量为 $O(dm)$ 和 $O(dm\log s_2)$.

6.4 数值实验

本节将通过数值实验验证 DSCOFS-CL 的有效性和优越性, 对比方法共 8 种, 包括 LapScore[14]、UDFS[15]、SOGFS[17]、RNE[16]、FSPCA[20]、SPCAFS[9]、DSCOFS[100] 和 SPCA-CL[97]. 数据集信息如表 6.1 所示. 本章使用 Python 3.12, PyTorch 2.3.0, 并辅以 NumPy 等科学计算库. 硬件环境为 Intel Core i9-13900K CPU, 64GB 内存, NVIDIA RTX A4000 显卡. 此外, 所提方法开源代码见链接 https://github.com/xianchaoxiu/DSCOFS-CL.

表 6.1: 所选数据集的信息

数据集	特征数	样本数	类别数
COIL20	1,024	1,440	20
USPS	256	1,000	10
GLIOMA	4,434	50	4
UMIST	644	575	20
ISOLET	617	1,560	26
MSTARSC	1,024	2,425	10



(a) COIL20 **(b)** USPS **(c)** GLIOMA

(d) UMIST **(e)** ISOLET **(f)** MSTARSC

图 6.3: 对比方法的 ACC 曲线

6.4.1 实验设置

(1) 参数设置

对于 DSCOFS-CL, 投影维度固定为数据的类别数. 元素稀疏度参数设置为 $s_2 = cdm$, 其中 c 是稀疏度百分比, 表示保留元素个数的百分比, 而 dm 是变换矩阵 X 中的元素总数. c 从 $\{0.1, 0.2, \ldots, 0.5\}$ 中选择. 设定低秩约束 $r = 0.1n$, 保证自表示矩阵的低秩性. 根据文献[9] 中的参数设定, 正则化参数和惩罚参数通过网格搜索策略从 $\{10^{-6}, 10^{-4}, \ldots, 10^{6}\}$ 中调优. 对于所有数据集, 特征数量从 $\{10, 20, \ldots, 100\}$ 中选取. 设定平衡参数 $\lambda = 0.5$, 保证原始空间和投影空间的同等重要性.

(2) 初始化

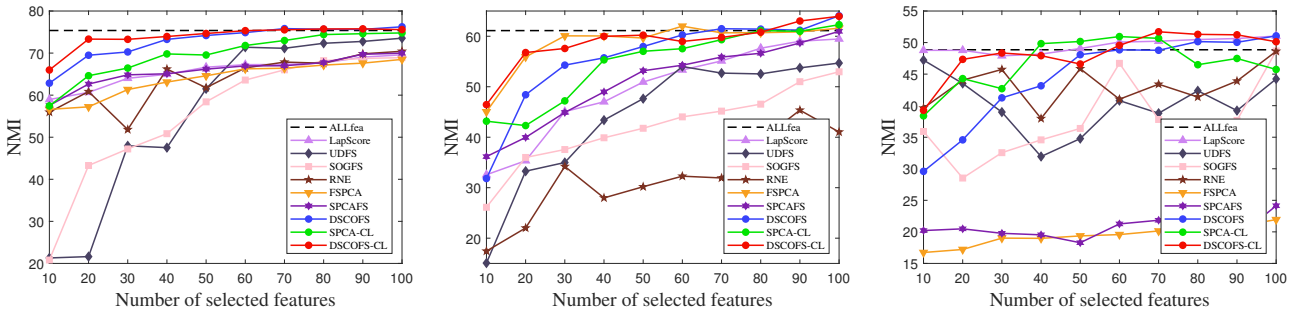
为了取得一般的非正交初始解, 实验中采用 Xavier 均匀初始化. 对于变量 $\boldsymbol{X} \in \mathbb{R}^{d\times m}$ 和 $\boldsymbol{M} \in \mathbb{R}^{n\times n}$, 初始化范围为

$$\boldsymbol{X}^0 \sim U(-\sqrt{\frac{6}{d+m}}, \sqrt{\frac{6}{d+m}}),\ \boldsymbol{M}^0 \sim U(-\sqrt{\frac{3}{n}}, \sqrt{\frac{3}{n}}). \tag{6.33}$$

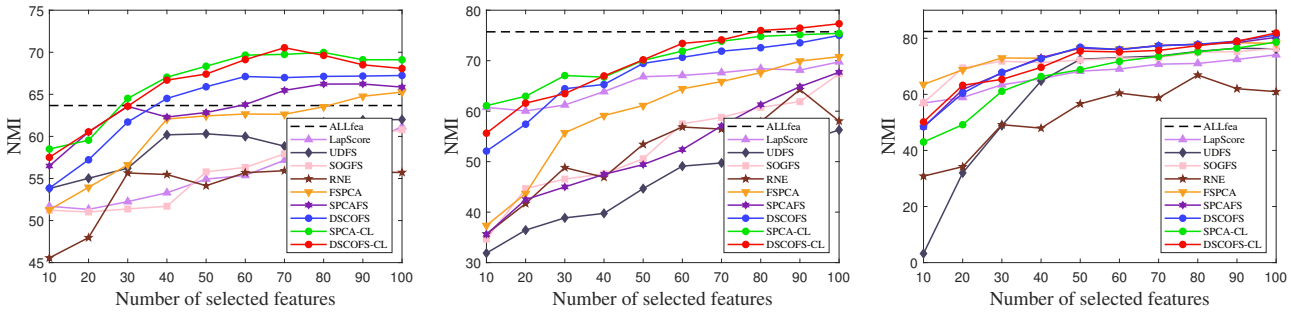
设 $\boldsymbol{B}^0 = \boldsymbol{Z}^0/(1+\tau_3)$、$\boldsymbol{C}^0 = \boldsymbol{X}^0/(1+\tau_4)$ 和 $\boldsymbol{D}^0 = \boldsymbol{X}^0/(1+\tau_5)$, 则 $\boldsymbol{Y}^0$、$\boldsymbol{P}^0$ 和 $\boldsymbol{Q}^0$ 可以根据式 (6.26)、(6.29) 和 (6.32) 式得到. 此外, 算法 3 的迭代次数满足 500 步时停止 (与文献[97] 保持一致).

(3) 评估指标

评估指标选择第 2 章中的准确率 (accuracy, ACC) 和归一化互信息 (normalized mutual in-



(a) COIL20 **(b)** USPS **(c)** GLIOMA



(d) UMIST **(e)** ISOLET **(f)** MSTARSC

图 6.4: 对比方法的 NMI 曲线

formation, NMI). 需要注意的是, k 均值聚类算法受初始点的影响较大, 所以选择执行 50 次 k 均值聚类并计算出平均值和标准差, 同时记录下最优参数下的聚类结果.

6.4.2 实验结果

图 6.3 和图 6.4 展示了不同特征数量 ACC 和 NMI 的均值曲线, 其中 ALLfea 作为参考基准, 最优实验结果采用黑体标注. 表 6.2 和表 6.3 给出了在 100 个特征范围内最佳 ACC 和 NMI 的平均值, 标准差以及相应的特征数量.

从图 6.3 可以看出, DSCOFS-CL 在所有数据集上都表现出卓越的性能. 在 DSCOFS 有着优越性能的前提下, DSCOFS-CL 融合了对比学习后相比 DSCOFS 有了进一步的提升, 同时相比 SPCA-CL 也有一定的提升, 这在 UMIST、GLIOMA 和 ISOLET 数据集上的表现较为明显. 从表 6.2 可以观察到, DSCOFS-CL 在所有数据集上均表现最好, 同时第二好的结果均由 DSCOFS 和 SPCA-CL 得到. 特别地, DSCOFS-CL 在 GLIOMA、UMIST 和 ISOLET 数据集上有不错的提升, 相较于第二好的结果分别有 1.68%、1.40% 和 2.69% 的增长. 在六个数据集上的平均 ACC 结果, DSCOFS 和 SPCA-CL 仅相差 0.15%, 而 DSCOFS-CL 相较于 DSCOFS 和 SPCA-CL 分别提升了 1.85% 和 1.7%.

从图 6.4 也可以观察到, DSCOFS-CL 展现出卓越的性能. DSCOFS-CL 在 ISOLET 数据集上有着一定的提升, 且是唯一超过基线的方法. 值得注意的是, 在 GLIOMA 数据集上, 当 DSCOFS-CL 所选特征为 50 时, NMI 结果较低. 这与 ACC 曲线上的表现相反, 进一步说明了

表 6.2: 对比方法的 ACC (平均值 ± 标准差) 结果 (%)

数据集	ALLfea	LapScore	UDFS	SOGFS	RNE	FSPCA	SPCAFS	DSCOFS	SPCA-CL	DSCOFS-CL
COIL20	57.74±4.93	54.82±3.91 (100)	58.71±3.47 (100)	49.66±4.81 (100)	55.84±4.41 (90)	50.15±4.70 (100)	54.39±3.67 (100)	60.51±4.63 (100)	60.31±3.49 (90)	**61.32±5.18 (80)**
USPS	65.12±4.95	62.02±4.09 (90)	59.52±2.97 (60)	55.58±3.07 (100)	46.04±2.69 (100)	67.38±4.36 (60)	67.34±4.49 (100)	69.67±4.97 (100)	68.88±4.05 (80)	**70.82±4.77 (100)**
GLIOMA	56.84±5.24	58.88±3.96 (90)	56.80±4.85 (100)	57.44±6.16 (70)	58.32±7.31 (90)	47.92±4.61 (80)	50.60±5.02 (20)	60.88±6.31 (90)	61.48±6.20 (40)	**63.16±7.46 (50)**
UMIST	41.07±2.38	40.13±2.79 (100)	47.12±2.49 (40)	41.70±3.17 (100)	40.35±2.26 (90)	46.70±2.29 (100)	46.78±2.51 (90)	48.10±3.01 (70)	49.55±3.00 (60)	**50.95±3.15 (70)**
ISOLET	57.89±3.82	52.21±2.76 (100)	41.95±2.07 (100)	49.31±2.32 (100)	47.12±2.06 (90)	53.62±2.36 (100)	53.04±2.33 (100)	59.67±3.46 (100)	60.53±3.75 (90)	**63.22±3.50 (90)**
MSTARSC	77.04±7.98	67.87±3.49 (90)	78.15±5.80 (90)	73.74±5.89 (100)	69.16±6.03 (80)	75.52±6.22 (70)	80.80±5.95 (100)	82.59±7.41 (100)	81.57±6.28 (100)	**83.06±6.31 (100)**
平均	59.28±4.88	55.99±3.50	57.04±3.69	54.57±4.24	52.81±4.13	56.88±4.09	58.83±4.00	63.57±4.96	63.72±4.46	**65.42±5.06**

表 6.3: 对比方法的 NMI (平均值 ± 标准差) 结果 (%)

数据集	ALLfea	LapScore	UDFS	SOGFS	RNE	FSPCA	SPCAFS	DSCOFS	SPCA-CL	SCOFS-CL
COIL20	75.37±1.96	69.59±1.48 (100)	73.54±1.76 (100)	68.92±1.84 (100)	70.43±1.92 (100)	68.50±1.56 (100)	69.98±1.45 (100)	**76.25±1.71 (100)**	74.79±1.48 (100)	75.76±1.76 (90)
USPS	61.12±2.01	59.46±1.80 (100)	54.69±2.11 (100)	52.96±1.54 (100)	45.36±1.93 (90)	62.00±1.87 (60)	60.98±2.37 (100)	**64.06±2.58 (100)**	62.29±2.40 (100)	63.95±2.67 (100)
GLIOMA	48.86±5.72	51.03±2.48 (100)	47.22±3.53 (10)	48.67±10.98 (100)	48.62±6.32 (100)	21.94±5.28 (100)	24.14±6.97 (100)	51.06±6.19 (100)	50.95±4.10 (60)	**51.71±5.03 (70)**
UMIST	63.67±1.85	61.16±1.71 (100)	62.00±1.58 (100)	60.79±1.54 (100)	55.92±1.57 (70)	65.27±1.58 (100)	66.23±1.60 (90)	67.24±1.85 (100)	69.98±1.84 (80)	**70.54±1.70 (70)**
ISOLET	75.72±1.70	69.77±1.20 (100)	56.29±1.11 (100)	67.40±1.44 (100)	64.27±0.95 (90)	70.79±1.12 (100)	67.71±1.33 (100)	75.01±1.35 (100)	75.41±1.51 (100)	**77.32±1.37 (100)**
MSTARSC	82.42±3.31	74.10±1.76 (100)	76.45±2.47 (90)	76.39±1.70 (100)	66.87±1.99 (80)	78.39±2.17 (90)	80.33±2.50 (100)	81.14±3.13 (100)	78.63±2.50 (100)	**81.88±2.03 (100)**
平均	67.86±2.76	64.19±1.74	61.70±2.09	62.52±3.17	58.58±2.45	61.15±2.26	61.56±2.70	69.13±2.80	68.68±2.31	**70.19±2.43**

ACC 和 NMI 在同组参数下不一定是完全正相关的. 从表 6.3 可以发现, DSCOFS-CL 均取到了最好或者第二好的结果. 与 ACC 结果类似,DSCOFS-CL 在 ISOLET 数据集上依然有不错的提升, 相比于第二好的 SPCA-CL 有 1.91% 的增加. 在六个数据集上的平均 NMI 结果, DSCOFS 取得第二好的表现, 并且相较于 SPCA-CL 提升了 0.45%, 而 DSCOFS-CL 相较于 DSCOFS 和 SPCA-CL 分别提升了 1.06% 和 1.51%.

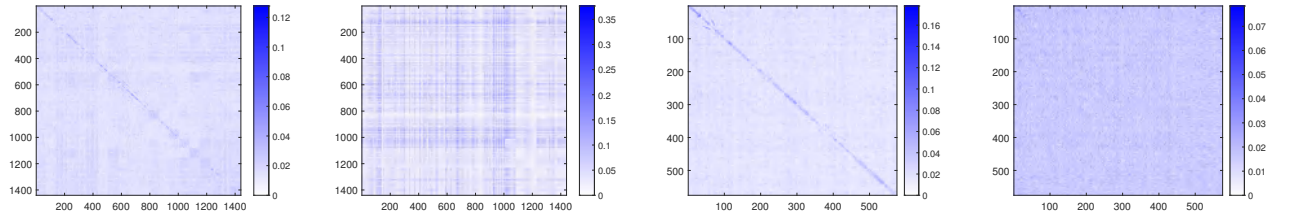
由此可以得出结论, 对比学习损失作为重构误差的度量能够进一步挖掘数据的有效信息, 这也展现了 DSCOFS-CL 的优越性和未来潜力.

6.4.3 消融实验

关于对比学习损失项, 在实验中选择了平衡参数 $\lambda = 0.5$, 从而保证原始空间和投影空间的同等重要性. 下面考虑当 $\lambda = 0$ 的情形, 即仅通过投影空间学习自表示矩阵. 需要注意的是, DSCOFS-CL 的特征选择是根据投影矩阵 $\boldsymbol{X}$ 确定的, 因此当 $\lambda = 1$ 时无法获得投影矩阵, 于是在本实验中不考虑 $\lambda = 1$ 的情形. 为了便于描述, 记式 (6.10) 为 Case1, 而当 $\lambda = 0$ 时模型为 Case2. 在 COIL20 和 UMIST 数据集上进行特征选择实验, 记录自表示矩阵 $\boldsymbol{Z}$ 以及 ACC 和 NMI 结果. 利用 $\boldsymbol{Z}$ 计算数据之间的相似度矩阵, 即

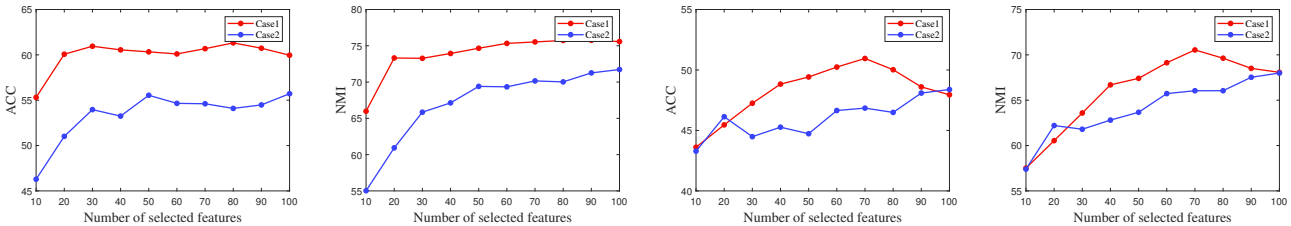
$$\boldsymbol{S} = \frac{|\boldsymbol{Z}| + |\boldsymbol{Z}|^T}{2}. \tag{6.34}$$

相似度矩阵 $\boldsymbol{S}$ 的可视化结果如图 6.5 所示, ACC 和 NMI 的结果如图 6.6 所示.



(a) COIL20 (Case1) **(b)** COIL20 (Case2) **(c)** UMIST (Case1) **(d)** UMIST (Case2)

图 6.5: 相似度矩阵的可视化结果对比



(a) COIL20 (ACC) **(b)** COIL20 (NMI) **(c)** UMIST (ACC) **(d)** UMIST (NMI)

图 6.6: 消融实验结果对比

从图 6.5 可见, Case1 的相似度矩阵 $\boldsymbol{S}$ 显示出明显的簇结构, 反映了数据在自表示过程中的内在关联性. 同时, Case1 还呈现低秩结构, 表明低秩约束在模型中得到了有效体现. 与之相

比, Case2 虽然揭示了低秩结构, 但未能有效学习到簇结构. 因此, 仅通过投影空间学习最优图的效果并不理想, 其原因在于数据经过投影降维后, 可能会破坏原有的结构特征. 从图 6.6 可以看到, Case2 的性能与 Case1 的相比存在一定的差距. 结合图 6.5 中的相似度矩阵 $\boldsymbol{S}$, 可以得出结论, 学习原始空间中的数据结构对提升无监督特征选择性能是有效的. 综上, DSCOFS-CL 通过在原始空间和投影空间中联合学习自表示矩阵, 能够获得更优的特征选择效果.

6.4.4 统计检验

Friedman 检验是一种基于排名的统计方法, 常用于比较多种方法整体的平均性能是否存在显著差异. 而后验 Nemenyi 检验可以通过临界差异 (critical difference, CD) 值来衡量两种方法之间是否差异. 在本实验中, Friedman 检验的原假设 $\mathcal{H}_0$ 表示所有对比方法的性能没有显著差异. 在显著性水平设定为 $\alpha = 0.05$ 的情况下对 DSCOFS-CL 进行 Friedman 检验和后验 Nemenyi 检验, 其结果分别如表 6.4 和图 6.7 所示.

表 6.4: Friedman 检验结果

方法	平均排名	p 值	假设
LapScore	6.67		
UDFS	6.00		
SOGFS	7.33		
RNE	7.17		
FSPCA	6.17	2.28×10^{-5}	拒绝
SPCAFS	5.67		
DSCOFS	2.50		
SPCA-CL	2.50		
DSCOFS-CL	1.00		

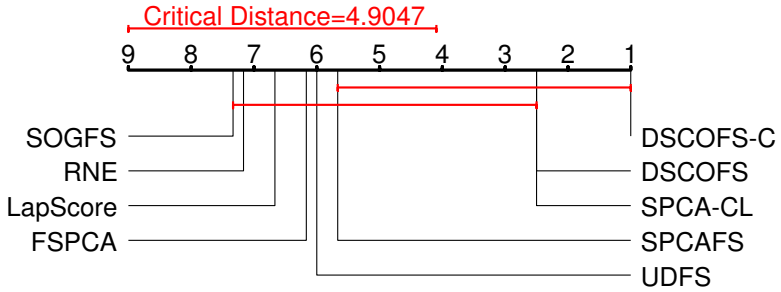


图 6.7: 后验 Nemenyi 检验结果

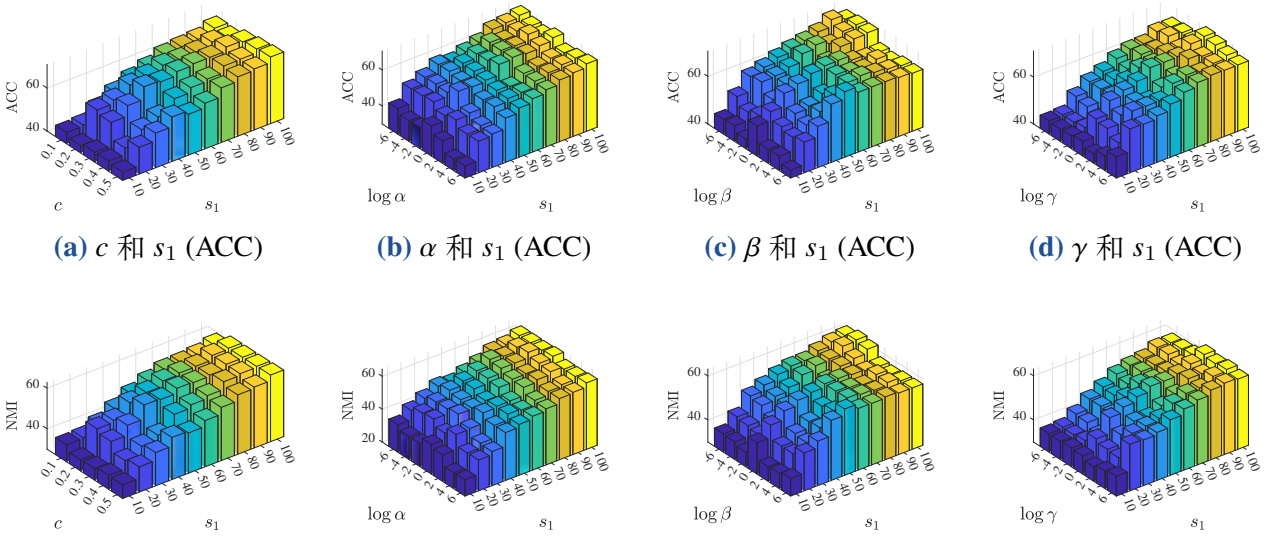
从表 6.4 可以看到 $p = 2.28 \times 10^{-5}$, 这意味着结果拒绝原假设 $\mathcal{H}_0$, 即所有对比方法之间确实存在显著差异. 从图 6.7 则可以看到, DSCOFS、SPCA-CL 和 SPCAFS 与其他的方法都在同一个 CD 值内, 这说明引入双稀疏和对比学习的无监督特征选择方法并没有与其他方法产生

明显差异, 仍然有提升空间. 而通过融入对比学习, DSCOFS-CL 与 LapScore、UDFS、SOGFS、RNE 和 FSPCA 之间产生了明显的差异, 这表明对比学习可以进一步挖掘数据中的有效信息, 从而实现更好的无监督特征选择性能.

6.4.5 讨论

(1) 参数敏感度分析

对于 DSCOFS-CL, 双稀疏约束参数 s_1 和 $s_2 = cdm$ 是控制模型学习稀疏结构的关键, 而 r 是控制自表示矩阵 Z 学习数据结构的关键. 此外, 惩罚参数 α、β 和 γ 在式 (6.13) 也会影响自表示矩阵和双稀疏约束. 在实验中, 将低秩结构固定为 $r = 0.1n$, 选择 s_1、c、α、β 和 γ 分析这些参数对特征选择性能的影响. 在 USPS 数据集上的参数敏感度结果如图 6.8 所示.



(a) c 和 s_1 (ACC) **(b)** α 和 s_1 (ACC) **(c)** β 和 s_1 (ACC) **(d)** γ 和 s_1 (ACC)

(e) c 和 s_1 (NMI) **(f)** α 和 s_1 (NMI) **(g)** β 和 s_1 (NMI) **(h)** γ 和 s_1 (NMI)

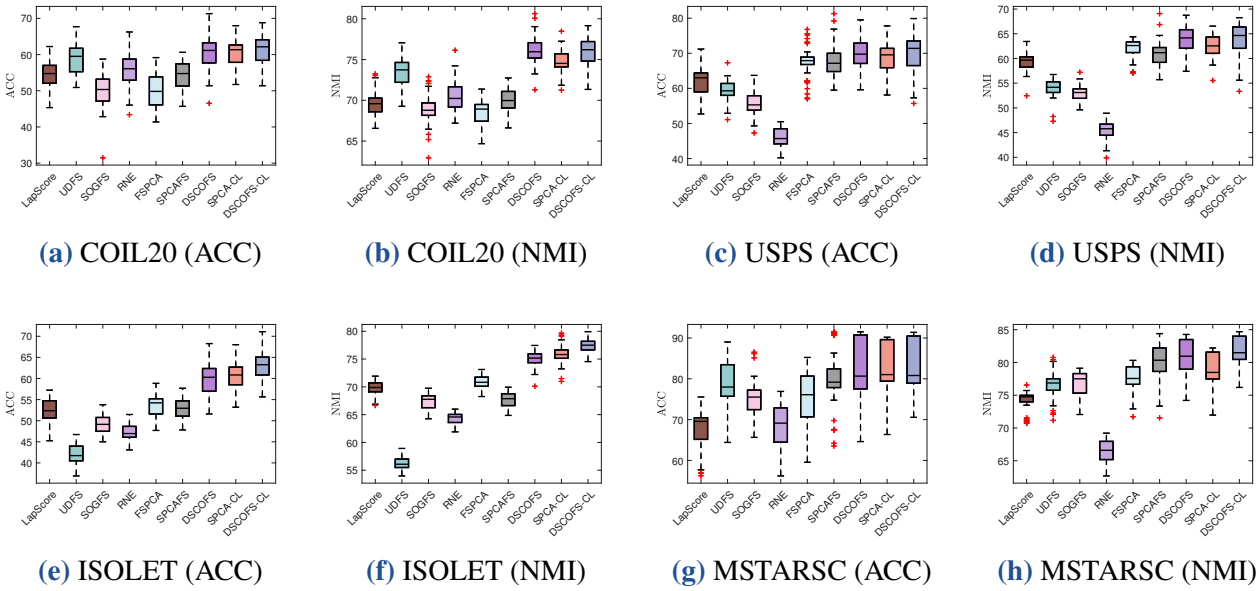
图 6.8: USPS 数据集上的参数敏感度分析结果

从图 6.8 中的 (a) 和 (e) 可以看出, 元素稀疏度 s_2 对性能有较明显的影响. 尤其在稀疏度百分比 $c = 0.3$ 时, 性能有一定的提升, 而在 $c = 0.1$ 时, 性能明显下降, 这说明稀疏度并不是越低越好, 过于低的稀疏度可能造成有效数据的丢失而影响性能. 从图 6.8中的 (b) - (h) 可以观察到, 惩罚参数 α、β 和 γ 对性能都有一定的影响, 其中 β 和 γ 的影响相较于 α 更大. 这些参数都是影响模型求解的关键, 因此在实验中要谨慎选择. 未来, 可以尝试使用自适应方法进行参数调节, 例如深度展开网络, 以进一步优化模型的性能.

(2) 稳定性分析

图 6.9 显示了最佳聚类结果的 50 次聚类分布. 可以看到 DSCOFS-CL 聚类结果有一定的波动, 但 DSCOFS-CL 的整体结果优于其他对比方法. 特别是在 ISOLET 数据集上, 相比于 DSCOFS 以及对比的 SPCA-CL, 本章提出的 DSCOFS-CL 有着较为稳定的聚类结果和更高的

性能表现. 同时也注意到, DSCOFS-CL 在 USPS 数据集上的最大值和最小值波动较大, 这可能是由于模型得到的特征区分度仍然不够所带来的影响. 综合考虑所有数据集的性能表现, DSCOFS-CL 展现出了良好的稳定性.

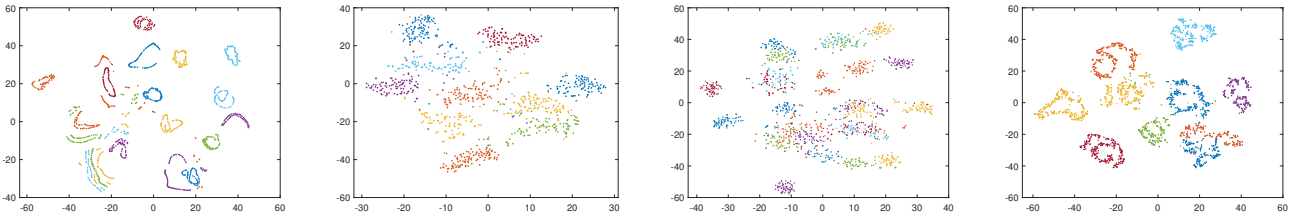


(a) COIL20 (ACC) **(b)** COIL20 (NMI) **(c)** USPS (ACC) **(d)** USPS (NMI)

(e) ISOLET (ACC) **(f)** ISOLET (NMI) **(g)** MSTARSC (ACC) **(h)** MSTARSC (NMI)

图 6.9: 模型稳定性分析结果

为了直观地观察数据的分布以及理解聚类的结果, 本实验采用 t-随机邻近嵌入 (t-distributed stochastic neighbor embedding, t-SNE)[104] 技术展示特征子空间的数据分布. 首先通过特征选择得到数据子集, 然后利用 t-SNE 技术将数据降维至二维, 最后通过散点图呈现数据的低维分布. 从图 6.10 可以观察到, 在低维空间中数据呈现明显的聚类特征, 因此使用聚类来检验特征选择的性能是合理的.



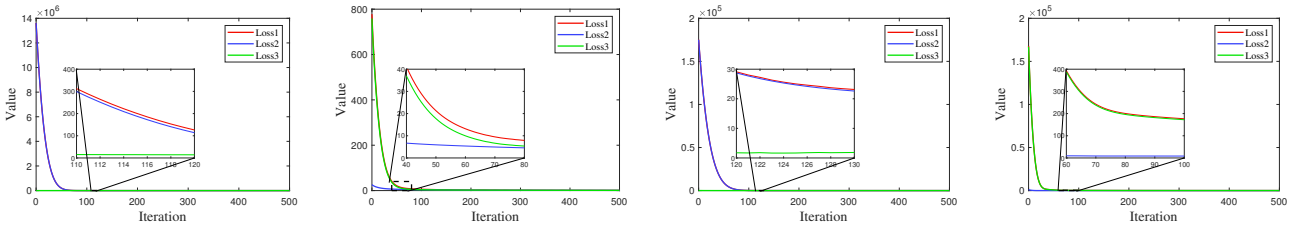
(a) COIL20 **(b)** USPS **(c)** ISOLET **(d)** MSTARSC

图 6.10: t-SNE 可视化结果

(3) 收敛性分析

为了验证算法 3 的收敛性, 记式 (6.13) 的目标函数为 Loss1, 式 (6.14) 和 (6.20) 的目标函数为 Loss2 和 Loss3. 损失函数在最优参数下迭代过程的变化如图 6.11 所示. 从图中可以看出, Loss1、Loss2 和 Loss3 在迭代过程中呈现一致的下降趋势, 同时在 100 次迭代内完成了快速的收敛, 并最终缓慢平稳. 值得注意的是, Loss2 在 COIL20 和 ISOLET 数据集上接近 Loss1, 而在另外两个数据集上, Loss1 的变化趋势与 Loss3 相反. 这表明, 原始空间和投影空间在不同数据

集上对自表示矩阵学习的影响程度存在差异. 最终, 收敛曲线的变化从数值上验证了算法的收敛性.



(a) COIL20 **(b)** USPS **(c)** ISOLET **(d)** MSTARSC

图 6.11: 收敛曲线

6.5 本章小结

本章针对无监督特征选择对异常值敏感的问题, 提出了基于对比学习的稀疏低秩方法. 首先, 相较于传统基于欧氏距离作为损失函数的方法, 该方法融合对比学习策略, 能够更充分地挖掘样本之间的关系. 通过在原始空间与投影空间联合学习最优图结构, 实现了数据全局与局部分布的自适应表达. 同时, 利用双稀疏约束表示特征的稀疏结构, 使得图学习可以在低维空间中更有效地捕捉数据特性. 此外, 还通过对自表示矩阵施加低秩约束, 使得模型能够保留图的全局结构. 针对模型非凸非连续的特点, 设计了基于梯度下降和硬阈值的一阶优化算法. 实验结果表明了对比学习构建损失函数对无监督特征选择的有效性.

第二部分

进阶篇

第 7 章 基于深度张量低秩表示的图像去噪

尽管遥感图像去噪方法已取得了令人瞩目的效果, 但当前大多数基于深度学习的方法以黑盒模式运行, 缺乏与物理信息模型的融合, 导致其可解释性受限. 此外, 许多方法难以捕捉遥感图像中固有的非局部自相似性特征, 且需通过繁琐的调参才能达到最优性能. 针对上述问题, 本章首先提出了稀疏张量辅助表示网络 (sparse tensor-aided representation network, STAR-Net). 该方法借助低秩先验信息, 可有效捕获遥感图像内部的非局部自相似性. 在此基础上, 进一步将 STAR-Net 扩展为稀疏变体 STAR-Net-S, 以应对原始遥感图像中非高斯噪声带来的干扰. 在算法方面, 设计了由交替方向乘子法引导的深度展开网络, 其所有正则参数均可通过网络训练自动学习, 从而兼具基于模型方法的可解释性与基于深度学习方法的高效性. 实验结果表明, 所提 STAR-Net 与 STAR-Net-S 的去噪性能均优于当前主流的遥感图像去噪方法.

7.1 引言

随着遥感技术的快速发展, 遥感图像在异常检测、去噪、分类及解混等领域得到了广泛应用. 通过捕获精细的光谱信息, 遥感图像能够精准分析地面目标特征, 为环境监测、资源勘探等实际场景提供数据支撑. 然而, 在图像采集过程中, 受传感器精度、外界干扰等因素影响, 遥感图像不可避免地受到噪声污染, 严重制约了后续数据分析的准确性与可靠性. 因此, 从含噪遥感图像中恢复干净的目标图像, 已成为遥感图像领域亟待解决的关键挑战. 目前, 主流的遥感图像去噪方法可分为两大类: 基于模型的方法与基于深度学习的方法.

基于模型的方法, 其核心思路是利用与自然图像统计特性或图像形成机制相关的物理先验, 建立干净图像与含噪图像之间的关联, 进而实现噪声去除. 其中, 块匹配三维滤波 (block matching 3D, BM3D)[105] 是应用最为广泛的方法之一. 它通过挖掘图像中非局部块的相似性, 实现高效去噪. 为充分利用遥感图像的跨波段信息, Maggioni 等[106] 提出了块匹配四维滤波 (block matching 4D, BM4D). 低秩矩阵分解是另一类极具代表性的方法, Zhang 等[107] 首次将其引入遥感图像去噪领域, 为后续相关研究奠定了基础. 随后, Xu 等[108] 基于该方法构建鲁棒主成分分析模型, 有效加速了迭代过程. 由于遥感图像本质上属于三维数据, 采用张量表示更契合其数据特性. 近年来, 低秩张量分解技术取得了显著进展. Chang 等[109] 提出超拉普拉斯正则化的单向低秩张量恢复 (hyper-laplacian low-rank tensor recovery, LLRT), 验证了其相较于基于矩阵方法的性能优势. Wang 等[110] 将全变分先验融入低秩张量分解 (low-rank tensor decomposition with total variation, LRTDTV), 实现了优异的遥感图像去噪效果. He 等[111] 引入非局部自相似性, 充分挖掘图像的全局与局部几何结构, 提出了非局部与全局相遇 (non-local meets global, NGMeet). Zha 等[112] 利用全局光谱特征与非局部结构化稀疏先验, 构造了非局部

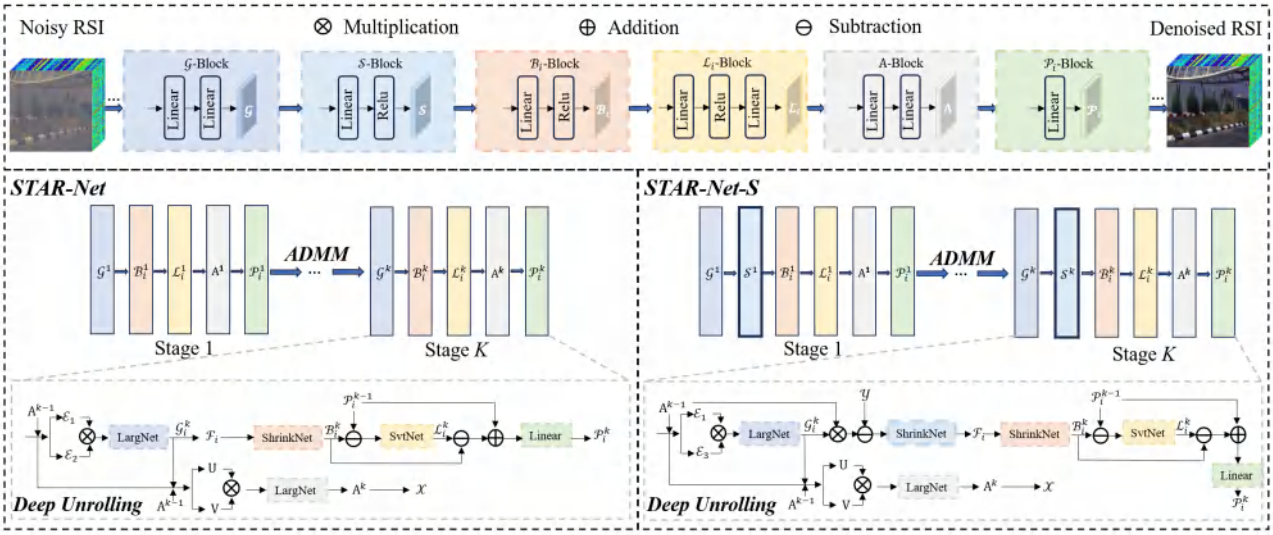


图 7.1: 所提 STAR-Net 和 STAR-Net-S 的网络示意图

结构化稀疏正则化 (non-local structured sparsity regularization, NLSSR). 值得注意的是, 尽管上述基于模型的方法提供了强大的可解释性和理论保证, 但它们通常需要繁琐的正则参数调优才能达到最佳性能.

基于深度学习的方法凭借强大的非线性拟合能力, 在遥感图像去噪领域取得了令人瞩目的效果. 例如, Yuan 等[113] 设计空间-光谱卷积神经网络 (convolutional neural network, CNN), 用于挖掘含噪图像与干净图像之间的非线性映射关系. Wei 等[114] 采用三维卷积构建三维准循环神经网络, 实现遥感图像特征的同步提取. Maffei 等[115] 引入可逆下采样算子, 提出高光谱图像单去噪卷积神经网络 (hyperspectral image single denoising CNN, HSI-SDeCNN). Zhuang 等[116] 通过分解遥感图像特征, 并将其与卷积神经网络结合, 进一步改善了遥感图像去噪性能. 事实上, 在训练数据充足的情况下, 基于深度学习的方法通常优于传统基于模型的方法, 但由于神经网络的黑盒特性, 此类方法的可解释性较差, 难以清晰揭示底层去噪机制.

近年来, 研究人员致力于融合深度学习的灵活性与数学模型的可解释性, 涌现出一系列优秀成果. Zhuang 等[117] 将快速灵活去噪网络[118] 集成到子空间表示框架中, 提出快速无参数高光谱图像混合噪声去除方法, 简称为 FastHyMix. 针对遥感图像的多维结构特性, Xiong 等[119] 构建基于子空间的多维稀疏张量模型, 并将其展开为神经网络, 称为 SMDS-Net. 非局部自相似性先验能够捕获图像内远距离区域的纹理与结构重复模式, 是保留图像边缘和细节的关键, 但 SMDS-Net 未能充分利用该重要先验. 最近, Peng 等[120] 提出基于代表性系数图像与光谱低秩张量分解的去噪框架 (learnable deep denoisier for representative coefficient images, RCILD), 但在可解释性方面存在不足. 综上, 这些方法要么未能将物理模型的完整迭代过程完全展开为网络结构, 要么忽视了非局部自相似性, 导致学习到的图像几何结构不够准确.

受上述研究现状的启发, 本章首先引入低秩先验以刻画遥感图像的非局部自相似性, 弥补现有方法对该关键先验的忽视. 其次, 针对基于矩阵表示的方法易破坏图像固有结构的缺陷,

采用张量表示形式有效地保留遥感图像的完整空间-光谱结构, 进而捕获其非局部自相似性. 最后, 利用深度展开策略[121], 将迭代优化过程中的正则参数视为神经网络的可学习参数, 避免繁琐的手动调参. 与 SMDS-Net 和 RCILD 不同, 本章利用交替方向乘子法对所提模型进行迭代优化, 并将完整的迭代过程展开为端到端网络. 为便于表述, 将该方法命名为稀疏张量辅助表示网络 (sparse tensor-aided representation network, STAR-Net). 此外, 针对原始遥感图像中常见的稀疏噪声, 通过引入额外稀疏先验对 STAR-Net 改进, 得到其稀疏变体 STAR-Net-S. 图 7.1 展示了所提网络的整体框架, 本章的主要贡献为

- 基于张量低秩表示模型, 融合非局部相似性先验和正交约束, 提出了一种有效的高光谱图像去噪新模型, 即 STAR-Net.
- 将 STAR-Net 扩展为 STAR-Net-S, 从而增强对真实遥感图像中非高斯噪声的鲁棒性, 同时更好地保留图像关键的空间与光谱信息.
- 开发了一种交替方向乘子法引导的深度展开网络, 能够端到端学习所有正则化参数, 将模型方法的可解释性与深度学习方法的灵活性相结合.

7.2 相关工作

7.2.1 张量表示

对于遥感图像而言, 张量表示能够在有效去除噪声的前提下, 保留图像的光谱一致性与空间结构完整性. 然而, 高维张量的直接处理往往面临计算复杂度高的问题, 而子空间表示方法提供了有效的解决方案. 通过将高维遥感图像数据投影至低维子空间, 既能保留数据的光谱信息, 又能显著提升计算效率.

设遥感图像张量为 $\boldsymbol{\mathcal{Y}} \in \mathbb{R}^{n_1 \times n_2 \times n_3}$, 可以将其分解为

$$\boldsymbol{\mathcal{Y}} = \boldsymbol{\mathcal{X}} + \boldsymbol{\mathcal{N}}, \tag{7.1}$$

其中, $\boldsymbol{\mathcal{X}}$ 代表干净的遥感图像张量, $\boldsymbol{\mathcal{N}}$ 代表图像中存在的噪声张量. 鉴于遥感图像的光谱低秩特性, 干净遥感图像的子空间表示能够有效捕获其固有的光谱冗余信息[122]. 因此, 干净遥感图像 $\boldsymbol{\mathcal{X}}$ 可通过以下张量分解形式进行近似表示

$$\boldsymbol{\mathcal{X}} = \boldsymbol{\mathcal{G}} \times_3 \boldsymbol{A}, \tag{7.2}$$

其中, $\boldsymbol{\mathcal{G}} \in \mathbb{R}^{n_1 \times n_2 \times n_4}$ 为代表性系数图像, 且满足 $n_4 \ll n_3$, $\boldsymbol{A}$ 为正交基矩阵. 于是, 结合 $\boldsymbol{\mathcal{G}}$ 的先验信息, 文献[119] 中的 SMDS-Net 表述为

$$\begin{aligned} &\min_{\boldsymbol{\mathcal{G}}, \boldsymbol{\mathcal{B}}_i, \boldsymbol{A}} \frac{1}{2}\|\boldsymbol{\mathcal{Y}} - \boldsymbol{\mathcal{G}} \times_3 \boldsymbol{A}\|_F^2 + \lambda \sum_i (\phi(\boldsymbol{\mathcal{G}}, \boldsymbol{\mathcal{B}}_i) + \gamma_1 \|\boldsymbol{\mathcal{B}}_i\|_1) \\ &\text{s.t.} \quad \boldsymbol{A}^T \boldsymbol{A} = \boldsymbol{I}, \end{aligned} \tag{7.3}$$

其中, $\|\cdot\|_F$ 表示 Frobenius 范数, $\|\cdot\|_1$ 表示 ℓ_1 范数, $\lambda, \gamma_1 > 0$ 为权衡参数. 这里, $\phi(\mathcal{G}, \mathcal{B}_i)$ 的具体形式定义为

$$\phi(\mathcal{G}, \mathcal{B}_i) = \frac{1}{2}\|\mathcal{R}_i\mathcal{G} - \mathcal{B}_i \times_1 \boldsymbol{D}_1 \times_2 \boldsymbol{D}_2 \times_3 \boldsymbol{D}_3\|_F^2, \tag{7.4}$$

其中, $\boldsymbol{D}_j$ $(j = 1, 2, 3)$ 分别为沿第 j 个模张量展开的字典矩阵, 其尺寸大小直接影响模型的拟合性能与去噪效果. 需要注意的是, $\mathcal{R}_i$ 表示从 $\mathcal{G}$ 中提取子张量 $\mathcal{G}_i$ 的算子, 其中 i 代表提取的子张量数量.

7.2.2 深度展开

深度展开是一种将传统迭代优化算法与深度学习技术融合的新兴方法, 它将每个迭代步骤等价为神经网络的一个层, 实现优化算法与深度学习的有机结合. 经典的迭代优化算法依赖多轮迭代逐步逼近最优解, 这类算法通常采用固定的迭代策略与参数设置, 难以自适应复杂多变的数据分布. 深度展开技术通过将这类迭代优化步骤逐层展开, 引入可学习参数替代传统算法中需要人工调试的正则参数, 使模型能够通过深度学习的训练过程自动优化参数配置. 例如, Yang 等[123] 提出了一种基于交替方向乘子法的压缩感知神经网络, 有效提升了压缩感知重建的精度与效率. 深度展开技术的优势在于, 既保留了传统优化方法的可解释性, 又充分利用了深度学习的强大表达能力, 避免了传统优化算法中繁琐的人工调参过程. 受此启发, 本章将基于深度展开思想提出一种高效的遥感图像去噪方法.

7.3 模型与算法

7.3.1 STAR-Net

现有 SMDS-Net 未充分挖掘遥感图像固有的非局部自相似先验, 同时忽略了实际遥感数据中普遍存在的非高斯噪声干扰, 而这两类结构信息对实现高精度遥感图像去噪都至关重要. 为此, 本节将非局部相似性先验融入张量表示, 提出如下数学模型

$$\begin{aligned} \min_{\mathcal{G}, \mathcal{B}_i, \boldsymbol{A}} \quad & \frac{1}{2}\|\mathcal{Y} - \mathcal{G} \times_3 \boldsymbol{A}\|_F^2 + \lambda \sum_i (\phi(\mathcal{G}, \mathcal{B}_i) + \gamma_1\|\mathcal{B}_i\|_1 + \gamma_2\|\mathcal{B}_i\|_*) \\ \text{s.t.} \quad & \boldsymbol{A}^T\boldsymbol{A} = \boldsymbol{I}, \end{aligned} \tag{7.5}$$

其中, $\|\cdot\|_*$ 为张量核范数, $\gamma_2 > 0$ 为正则化参数. 显然, 如果 $\gamma_2 \to 0$, 式 (7.5) 将退化为式 (7.3).

为了便于计算, 引入辅助变量 $\mathcal{L}_i = \mathcal{B}_i$, 并将式 (7.5) 重写为

$$\begin{aligned} \min_{\mathcal{G}, \mathcal{B}_i, \mathcal{L}_i, \boldsymbol{A}} \quad & \frac{1}{2}\|\mathcal{Y} - \mathcal{G} \times_3 \boldsymbol{A}\|_F^2 + \lambda \sum_i (\phi(\mathcal{G}, \mathcal{B}_i) + \gamma_1\|\mathcal{B}_i\|_1 + \gamma_2\|\mathcal{L}_i\|_*) \\ \text{s.t.} \quad & \boldsymbol{A}^T\boldsymbol{A} = \boldsymbol{I}, \ \mathcal{L}_i = \mathcal{B}_i. \end{aligned} \tag{7.6}$$

对应增广拉格朗日函数为

$$\begin{aligned} &L_\beta(\mathcal{G}, \mathcal{B}_i, \mathcal{L}_i, A, \mathcal{P}_i) \\ &= \frac{1}{2}\|\mathcal{Y} - \mathcal{G} \times_3 A\|_F^2 + \lambda \sum_i (\phi(\mathcal{G}, \mathcal{B}_i) + \gamma_1 \|\mathcal{B}_i\|_1 + \gamma_2 \|\mathcal{L}_i\|_*) \\ &+ \langle \mathcal{P}_i, \mathcal{L}_i - \mathcal{B}_i \rangle + \frac{\beta}{2}\|\mathcal{L}_i - \mathcal{B}_i\|_F^2, \end{aligned} \tag{7.7}$$

其中, $\mathcal{P}_i$ 为拉格朗日乘子, $\beta > 0$ 为惩罚参数. 采用交替方向乘子法 (alternating direction method of multipliers, ADMM), 迭代更新规则如下

$$\begin{cases} \mathcal{G}^{k+1} = \underset{\mathcal{G}}{\operatorname{argmin}}\ L_\beta(\mathcal{G}, \mathcal{B}_i^k, \mathcal{L}_i^k, \mathcal{A}^k, \mathcal{P}_i^k), & \text{(7.8a)} \\ \mathcal{B}_i^{k+1} = \underset{\mathcal{B}_i}{\operatorname{argmin}}\ L_\beta(\mathcal{G}^{k+1}, \mathcal{B}_i, \mathcal{L}_i^k, \mathcal{A}^k, \mathcal{P}_i^k), & \text{(7.8b)} \\ \mathcal{L}_i^{k+1} = \underset{\mathcal{L}_i}{\operatorname{argmin}}\ L_\beta(\mathcal{G}^{k+1}, \mathcal{B}_i^{k+1}, \mathcal{L}_i, \mathcal{A}^k, \mathcal{P}_i^k), & \text{(7.8c)} \\ A^{k+1} = \underset{\mathcal{A}}{\operatorname{argmin}}\ L_\beta(\mathcal{G}^{k+1}, \mathcal{B}_i^{k+1}, \mathcal{L}_i^{k+1}, \mathcal{A}, \mathcal{P}_i^k), & \text{(7.8d)} \\ \mathcal{P}_i^{k+1} = \mathcal{P}_i^k + \beta(\mathcal{L}_i^{k+1} - \mathcal{B}_i^{k+1}). & \text{(7.8e)} \end{cases}$$

下面分别对各子问题进行求解, 并通过深度展开构造对应的神经网络模块.

(1) 更新 $\mathcal{G}$

固定其余变量, 子问题 (7.8a) 可简化为

$$\min_{\mathcal{G}} \frac{1}{2}\|\mathcal{Y} - \mathcal{G} \times_3 \mathcal{A}^k\|_F^2 + \frac{\lambda}{2} \sum_i \|\mathcal{R}_i \mathcal{G} - \mathcal{B}_i^k \times_1 \mathcal{D}_1 \times_2 \mathcal{D}_2 \times_3 \mathcal{D}_3\|_F^2. \tag{7.9}$$

对目标函数关于 $\mathcal{G}$ 求导并令梯度为零, 可得解析形式

$$\mathcal{G}^{k+1} = (\mathcal{I} + \lambda \sum_i \mathcal{R}_i^T \mathcal{R}_i)^{-1} (\lambda \sum_i \mathcal{R}_i^T \mathcal{B}_i^k \times_1 \mathcal{D}_1 \times_2 \mathcal{D}_2 \times_3 \mathcal{D}_3 + \mathcal{Y} \times_3 (\mathcal{A}^k)^T). \tag{7.10}$$

记

$$\begin{aligned} \mathcal{E}_1 &= (\mathcal{I} + \lambda \sum_i \mathcal{R}_i^T \mathcal{R}_i)^{-1}, \\ \mathcal{E}_2 &= \lambda \sum_i \mathcal{R}_i^T \mathcal{B}_i^k \times_1 \mathcal{D}_1 \times_2 \mathcal{D}_2 \times_3 \mathcal{D}_3 + \mathcal{Y} \times_3 (\mathcal{A}^k)^T, \end{aligned} \tag{7.11}$$

则 $\mathcal{G}^{k+1}$ 可通过线性网络模块更新

$$\mathcal{G}^{k+1} = \text{LargNet}(\mathcal{E}_1, \mathcal{E}_2), \tag{7.12}$$

其中, LargNet 可由两层线性变换实现. 值得注意的是, $\mathcal{E}_1$ 仅需在迭代初始计算一次.

(2) 更新 $\mathcal{B}_i$

固定其余变量, $\mathcal{B}_i$ 子问题 (7.8b) 可写为

$$\begin{aligned}\min_{\boldsymbol{\mathcal{B}}_i} \quad & \frac{\lambda}{2}\|\boldsymbol{\mathcal{R}}_i\boldsymbol{\mathcal{G}}^{k+1} - \boldsymbol{\mathcal{B}}_i \times_1 \boldsymbol{D}_1 \times_2 \boldsymbol{D}_2 \times_3 \boldsymbol{D}_3\|_F^2 \\ & + \frac{\beta}{2}\|\boldsymbol{\mathcal{L}}_i^k - \boldsymbol{\mathcal{B}}_i + \boldsymbol{\mathcal{P}}_i^k/\beta\|_F^2 + \lambda\gamma_1\|\boldsymbol{\mathcal{B}}_i\|_1,\end{aligned} \tag{7.13}$$

进一步整理可得等价形式

$$\begin{aligned}\min_{\boldsymbol{\mathcal{B}}_i} \quad & \frac{1}{2}\|(\beta\boldsymbol{\mathcal{I}} + \lambda\boldsymbol{\mathcal{I}} \times_1 \boldsymbol{D}_1 \times_2 \boldsymbol{D}_2 \times_3 \boldsymbol{D}_3)\boldsymbol{\mathcal{B}}_i \\ & - (\lambda\boldsymbol{\mathcal{R}}_i\boldsymbol{\mathcal{G}}^{k+1} + \beta\boldsymbol{\mathcal{L}}_i^k + \boldsymbol{\mathcal{P}}_i^k)\|_F^2 + \lambda\gamma_1\|\boldsymbol{\mathcal{B}}_i\|_1.\end{aligned} \tag{7.14}$$

令

$$\boldsymbol{\mathcal{F}}_i = \boldsymbol{\mathcal{B}}_i + \frac{1}{l}\boldsymbol{\mathcal{H}}^T(\lambda\boldsymbol{\mathcal{R}}_i\boldsymbol{\mathcal{G}}^{k+1} + \beta\boldsymbol{\mathcal{L}}_i^k + \boldsymbol{\mathcal{P}}_i^k - \boldsymbol{\mathcal{H}}\boldsymbol{\mathcal{B}}_i), \tag{7.15}$$

其中, $\boldsymbol{\mathcal{H}} = \beta\boldsymbol{\mathcal{I}} + \lambda\boldsymbol{\mathcal{I}} \times_1 \boldsymbol{D}_1 \times_2 \boldsymbol{D}_2 \times_3 \boldsymbol{D}_3$, $l > 0$ 为 Lipschitz 常数. 借鉴迭代收缩阈值算法 (iterative shrinkage thresholding algorithm, ISTA), 式 (7.14) 的解可表示为

$$\boldsymbol{\mathcal{B}}_i^{k+1} = \mathcal{M}_{\lambda\gamma_1/l}(\boldsymbol{\mathcal{F}}_i), \tag{7.16}$$

其中, $\mathcal{M}_{\lambda\gamma_1/l}(\boldsymbol{\mathcal{F}}_i) = \mathrm{sgn} \odot (\boldsymbol{\mathcal{F}}_i)\{|\boldsymbol{\mathcal{F}}_i| - \lambda\gamma_1/l\}_+$ 是软阈值算子, sgn 为符号函数, $\{\cdot\}_+ = \max(0, x)$, $\odot$ 表示哈达玛积 (Hadamard product). 由于修正线性单元 (rectified linear unit, ReLU) 与 $\{\cdot\}_+$ 形式一致, 可将上述迭代步骤展开为可学习收缩网络

$$\begin{aligned}\boldsymbol{\mathcal{B}}_i^{k+1} &= \mathrm{ShrinkNet}(\boldsymbol{\mathcal{F}}_i, \lambda\gamma_1/l) \\ &= \mathrm{sign}(\boldsymbol{\mathcal{F}}_i) \odot \mathrm{ReLU}(|\boldsymbol{\mathcal{F}}_i| - \lambda\gamma_1/l\boldsymbol{\mathcal{I}}).\end{aligned} \tag{7.17}$$

(3) 更新 $\boldsymbol{\mathcal{L}}_i$

$\boldsymbol{\mathcal{L}}_i$ 子问题等价于

$$\min_{\boldsymbol{\mathcal{L}}_i} \quad \frac{\beta}{2}\|\boldsymbol{\mathcal{L}}_i - \boldsymbol{\mathcal{B}}_i^{k+1} + \boldsymbol{\mathcal{P}}_i^k/\beta\|_F^2 + \lambda\gamma_2\|\boldsymbol{\mathcal{L}}_i\|_*. \tag{7.18}$$

设张量奇异值分解为 $\boldsymbol{\mathcal{B}}_i^{k+1} - \boldsymbol{\mathcal{P}}_i^k/\beta = \boldsymbol{\mathcal{U}}_i * \boldsymbol{\mathcal{W}}_i * \boldsymbol{\mathcal{V}}_i^T$, 则根据奇异值阈值 (singular value thresholding, SVT)[124], 其解析形式为

$$\boldsymbol{\mathcal{L}}_i^{k+1} = \boldsymbol{\mathcal{U}}_i * \mathrm{Diag}(\{\boldsymbol{\mathcal{W}}_i - \lambda\gamma_2/\beta\boldsymbol{\mathcal{I}}\}_+) * \boldsymbol{\mathcal{V}}_i^T. \tag{7.19}$$

同理, 可构造奇异值阈值网络

$$\begin{aligned}\boldsymbol{\mathcal{L}}_i^{k+1} &= \mathrm{SvtNet}(\boldsymbol{\mathcal{B}}_i^{k+1} - \boldsymbol{\mathcal{P}}_i^k/\beta, \lambda\gamma_2/\beta) \\ &= \boldsymbol{\mathcal{U}}_i * \mathrm{ReLU}(\mathrm{Diag}(\boldsymbol{\mathcal{W}}_i - \lambda\gamma_2/\beta\boldsymbol{\mathcal{I}})) * \boldsymbol{\mathcal{V}}_i^T.\end{aligned} \tag{7.20}$$

(4) 更新 $\boldsymbol{A}$

正交约束下的 $\boldsymbol{A}$ 子问题 (7.8d) 可写为

算法 1 求解式 (7.5) 的深度展开网络

输入: 数据 $\mathcal{Y}$, 参数 λ, γ_1, γ_2, β

初始化: $(\mathcal{G}^0, \mathcal{B}_i^0, \mathcal{L}_i^0, \boldsymbol{A}^0, \mathcal{P}_i^0)$

当 未收敛 **时**

1: 通过式 (7.12) 更新 $\mathcal{G}$
2: 通过式 (7.17) 更新 $\mathcal{B}_i$
3: 通过式 (7.20) 更新 $\mathcal{L}_i$
4: 通过式 (7.23) 更新 $\boldsymbol{A}$
5: 通过式 (7.25) 更新 $\mathcal{P}_i$

结束循环

输出: 去噪后的图像 $\mathcal{X} = \mathcal{G}^{k+1} \times_3 \boldsymbol{A}^{k+1}$

$$\begin{aligned} \min_{A} \quad & \frac{1}{2}\|\mathcal{Y} - \mathcal{G}^{k+1} \times_3 \boldsymbol{A}\|_F^2 \\ \text{s.t.} \quad & \boldsymbol{A}^T\boldsymbol{A} = \boldsymbol{I}. \end{aligned} \tag{7.21}$$

该问题本质为降秩 Procrustes 旋转问题[44]. 记 $\text{unfold}(\mathcal{Y}, 3)\text{unfold}(\mathcal{G}^{k+1}, 3)^T = \boldsymbol{U}\boldsymbol{\Sigma}\boldsymbol{V}^T$, 其中 unfold 表示沿第 i 模矩阵化, 则正交投影解为

$$\boldsymbol{A}^{k+1} = \boldsymbol{U}\boldsymbol{V}^T, \tag{7.22}$$

并可通过以下线性网络实现

$$\boldsymbol{A}^{k+1} = \text{LargNet}(\boldsymbol{U}, \boldsymbol{V}^T). \tag{7.23}$$

(5) 更新 $\mathcal{P}_i$

拉格朗日乘子的更新为

$$\mathcal{P}_i^{k+1} = \mathcal{P}_i^k + \beta(\mathcal{L}_i^{k+1} - \mathcal{B}_i^{k+1}), \tag{7.24}$$

可由线性层参数化表示为

$$\mathcal{P}_i^{k+1} = \text{Linear}(\Theta_i), \tag{7.25}$$

其中, Linear 为线性变换层, Θ_i 可通过 $\mathcal{P}_i^k + \beta(\mathcal{L}_i^{k+1} + \mathcal{B}_i^{k+1})$ 计算, β 为可学习参数.

综上, STAR-Net 的完整深度展开迭代流程如算法 1 所示.

7.3.2 STAR-Net-S

实际场景中, 遥感图像除高斯噪声外, 常受到脉冲、条带等非高斯稀疏噪声污染. 为增强模型对复杂噪声的鲁棒性, 本节在 STAR-Net 基础上引入稀疏噪声分量, 构建如下数学模型

$$\begin{aligned}\min_{\mathcal{G},\mathcal{S},\mathcal{B}_i,\boldsymbol{A}} \quad & \frac{1}{2}\|\mathcal{Y}-\mathcal{G}\times_3\boldsymbol{A}-\mathcal{S}\|_F^2+\mu\|\mathcal{S}\|_1+\lambda\sum_i(\phi(\mathcal{G},\mathcal{B}_i)+\gamma_1\|\mathcal{B}_i\|_1+\gamma_2\|\mathcal{B}_i\|_*)\\ \text{s.t.} \quad & \boldsymbol{A}^T\boldsymbol{A}=\boldsymbol{I},\end{aligned} \tag{7.26}$$

其中, $\mathcal{S}$ 为稀疏噪声张量, $\mu>0$ 为稀疏正则参数. 为便于记号, 将式 (7.26) 称为 STAR-Net-S.

与 STAR-Net 类似, 引入辅助变量 $\mathcal{L}_i=\mathcal{B}_i$ 得到等价模型

$$\begin{aligned}\min_{\mathcal{G},\mathcal{S},\mathcal{B}_i,\mathcal{L}_i,\boldsymbol{A}} \quad & \frac{1}{2}\|\mathcal{Y}-\mathcal{G}\times_3\boldsymbol{A}-\mathcal{S}\|_F^2+\mu\|\mathcal{S}\|_1+\lambda\sum_i(\phi(\mathcal{G},\mathcal{B}_i)+\gamma_1\|\mathcal{B}_i\|_1+\gamma_2\|\mathcal{L}_i\|_*)\\ \text{s.t.} \quad & \boldsymbol{A}^T\boldsymbol{A}=\boldsymbol{I},\ \mathcal{L}_i=\mathcal{B}_i,\end{aligned} \tag{7.27}$$

对应增广拉格朗日函数为

$$\begin{aligned}&L_\beta(\mathcal{G},\mathcal{S},\mathcal{B}_i,\mathcal{L}_i,\boldsymbol{A},\mathcal{P}_i)\\ &=\frac{1}{2}\|\mathcal{Y}-\mathcal{G}\times_3\boldsymbol{A}-\mathcal{S}\|_F^2+\mu\|\mathcal{S}\|_1+\lambda\sum_i(\phi(\mathcal{G},\mathcal{B}_i)+\gamma_1\|\mathcal{B}_i\|_1+\gamma_2\|\mathcal{L}_i\|_*)\\ &+\langle\mathcal{P}_i,\mathcal{L}_i-\mathcal{B}_i\rangle+\frac{\beta}{2}\|\mathcal{L}_i-\mathcal{B}_i\|_F^2.\end{aligned} \tag{7.28}$$

对比式 (7.7) 可知, STAR-Net-S 仅在 $\mathcal{G}$ 更新与新增稀疏噪声 $\mathcal{S}$ 与 STAR-Net 存在差异, 其余变量更新形式保持一致. 此时 $\mathcal{G}^{k+1}$ 的子问题为

$$\min_{\mathcal{G}} \quad \frac{1}{2}\|\mathcal{Y}-\mathcal{G}\times_3\boldsymbol{A}^k-\mathcal{S}^k\|_F^2+\lambda\sum_i\frac{1}{2}\|\mathcal{R}_i\mathcal{G}-\mathcal{B}_i^k\times_1\boldsymbol{D}_1\times_2\boldsymbol{D}_2\times_3\boldsymbol{D}_3\|_F^2. \tag{7.29}$$

求导可得闭式解

$$\begin{aligned}\mathcal{G}^{k+1}=(\mathcal{I}+\lambda\sum_i\mathcal{R}_i^T\mathcal{R}_i)^{-1}(\lambda\sum_i\mathcal{R}_i^T\mathcal{B}_i^k\times_1\boldsymbol{D}_1\\ \times_2\boldsymbol{D}_2\times_3\boldsymbol{D}_3+\mathcal{Y}\times_3(\boldsymbol{A}^k)^T-\mathcal{S}^k\times_3(\boldsymbol{A}^k)^T).\end{aligned} \tag{7.30}$$

记

$$\mathcal{E}_3=\lambda\sum_i\mathcal{R}_i^T\mathcal{B}_i^k\times_1\boldsymbol{D}_1\times_2\boldsymbol{D}_2\times_3\boldsymbol{D}_3+\mathcal{Y}\times_3(\boldsymbol{A}^k)^T-\mathcal{S}^k\times_3(\boldsymbol{A}^k)^T, \tag{7.31}$$

则 $\mathcal{G}^{k+1}$ 可以通过以下方式更新

$$\mathcal{G}^{k+1}=\mathrm{LargNet}(\mathcal{E}_1,\mathcal{E}_3). \tag{7.32}$$

此外, 稀疏噪声分量 $\mathcal{S}^{k+1}$ 的子问题为

$$\min_{\mathcal{S}}\frac{1}{2}\|\mathcal{Y}-\mathcal{G}^{k+1}\times_3\boldsymbol{A}^k-\mathcal{S}\|_F^2+\mu\|\mathcal{S}\|_1. \tag{7.33}$$

其解同样为软阈值操作, 可展开为收缩网络

$$\mathcal{S}^{k+1}=\mathrm{ShrinkNet}(\mathcal{Y}-\mathcal{G}^{k+1}\times_3\boldsymbol{A}^k,\mu). \tag{7.34}$$

算法 2 求解式 (7.26) 的深度展开网络

输入: 数据 $\mathcal{Y}$, 参数 λ, μ, γ_1, γ_2, β

初始化: $(\mathcal{G}^0, \mathcal{S}^0, \mathcal{B}_i^0, \mathcal{L}_i^0, \boldsymbol{A}^0, \mathcal{P}_i^0)$

当 未收敛 **时**

1: 通过式 (7.32) 更新 $\mathcal{G}$
2: 通过式 (7.17) 更新 $\mathcal{B}_i$
3: 通过式 (7.34) 更新 $\mathcal{S}$
4: 通过式 (7.20) 更新 $\mathcal{L}_i$
5: 通过式 (7.23) 更新 $\boldsymbol{A}$
6: 通过式 (7.25) 更新 $\mathcal{P}_i$

结束循环

输出: 去噪后的图像 $\mathcal{X} = \mathcal{G}^{k+1} \times_3 \boldsymbol{A}^{k+1}$

最后, STAR-Net-S 的深度展开流程列于算法 2.

7.4 数值实验

本节将所提 STAR-Net 与 STAR-Net-S 与当前最先进的方法进行比较, 包括基于模型的方法, 即 BM4D[106]、LLRT[109]、LRTDTV[110]、NGMeet[111] 与 NLSSR[112], 以及基于深度学习的方法, 即 FastHyMix[117]、HSI-SDeCNN[115]、SMDS-Net[119]、Eigen-CNN[116] 与 RCILD[120]. 此外, 所提方法开源代码见链接 https://github.com/xianchaoxiu/STAR-Net.

7.4.1 实验设置

(1) 数据集

根据文献[114], 从 ICVL 数据集中选取 100 幅高光谱遥感图像作为训练集. 图像分辨率为 1,392 × 1,300, 包含 31 个光谱波段, 范围为 400-700 nm. 为提升模型泛化能力与训练效率, 训练前对所有图像进行数据增强, 包括随机翻转、裁剪等, 并调整尺寸至 56 × 56 × 31.

合成测试集包括 ICVL 数据集和 PaviaU 数据集, 其中 PaviaU 图像的大小为 610 × 340 × 103, 属于典型的大规模高光谱数据集[125]. 真实测试集包括北京首都机场数据集与 Indian Pines 数据集, 其中北京首都机场遥感图像的大小为 300 × 300× 155, Indian Pines 图像的大小为 145 × 145× 206, 详情参考文献[115]. 为生成含噪遥感图像, 将高斯噪声引入遥感图像的每个波段. 具体而言, 考虑四种不同场景, 标准方差 σ 分别设置为 10、30、50 和 70.

(2) 参数设置

所提 STAR-Net 和 STAR-Net-S 均基于 PyTorch 框架实现, 训练过程在 NVIDIA GeForce RTX 4090 GPU 上完成, 总训练轮次设为 300. 初始学习率设置为 5×10^{-3}, 采用阶梯式学习率衰

减策略, 每 80 次迭代将学习率衰减 0.35 倍. 优化器选用 Adam, 输入图像大小为 56×56, 批次大小设为 2, 网络展开迭代次数 K 设为 9. 此外, 字典采用维度为 [9, 9, 9] 的离散余弦变换 (discrete cosine transform, DCT) 基进行初始化, 使得 3 个模式的字典尺寸均为 9×9.

一般来说, 神经网络使用随机初始化的参数进行训练. 为加速训练过程, 采用基于交替方向乘子法得到的原始参数对网络参数进行初始化. 此外, 所有可学习参数 γ_1、γ_2、l、λ、μ、β 的初始值均设为 0.02. 对于给定的训练数据集, 损失函数指定为

$$Loss = \|\text{STAR-Net}(\boldsymbol{\mathcal{Y}}) - \boldsymbol{\mathcal{X}}\|_F^2, \tag{7.35}$$

其中, $\boldsymbol{\mathcal{Y}}$ 表示模型输入的遥感图像, $\boldsymbol{\mathcal{X}}$ 表示对应的真实值, 即没有噪声的原始遥感图像.

(3) 评价指标

为了定量评估所有对比方法的去噪性能, 选取高光谱遥感图像去噪领域常用的四个评价指标, 分别为峰值信噪比 (peak signal-to-noise ratio, PSNR)、结构相似性 (structural similarity, SSIM)、光谱角度映射器 (spectral angle mapper, SAM) 及相对全局无量纲综合误差 (erreur relative globale adimensionnelle de synthese, ERGAS). 其中, PSNR 用于评估去噪图像与真实图像的重建精度, SSIM 用于衡量图像结构信息的感知一致性, SAM 用于量化干净图像与去噪图像之间的光谱差异, ERGAS 用于评估整体的重建误差. PSNR ↑ 与 SSIM ↑ 值越高, SAM ↓ 与 ERGAS ↓ 值越低, 去噪性能越优.

7.4.2 合成数据结果

表 7.1 列出了 ICVL 数据集 30 幅测试遥感图像在不同噪声水平下的 PSNR、SSIM 、SAM 及 ERGAS 对比结果. 可以看出, 基于模型的方法性能略逊于基于深度学习的方法. 在所有测试方法中, STAR-Net 与 STAR-Net-S 在噪声方差为 10、30、50 及 70 的场景下均取得了优异的去噪效果. 此外, 从平均性能上看, STAR-Net-S 在四项评价指标上均表现最优, 其次是 STAR-Net.

为更直观地展示各方法的去噪效果, 选取 ICVL 数据集中的 gavyam_0823-0933 遥感图像进行可视化分析. 图 7.2 分别呈现了干净图像、噪声图像, 以及所有方法在噪声方差为 50 时的去噪结果. 通过对比可见, 尽管 LRTDTV 和 NLSSR 表现相对较好, 但两者均存在残留噪声及局部细节丢失的问题. 在基于深度学习的方法中, 除 HSI-SDeCNN 外, 其余方法均能有效滤除噪声, 去噪结果接近干净图像.

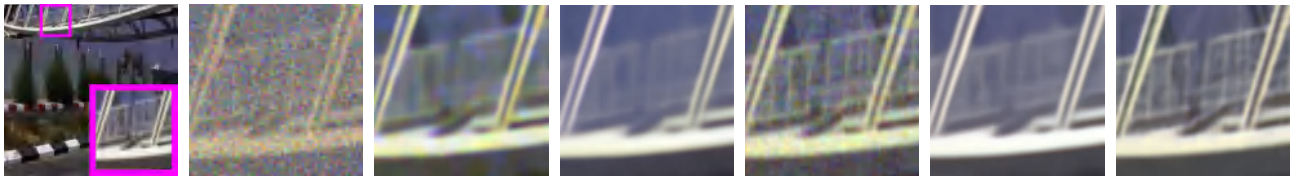
除小规模 ICVL 数据集外, 本节还在 PaviaU 数据集上开展去噪实验. 由表 7.2 可知, 即便在大规模数据集上, STAR-Net 与 STAR-Net-S 仍能保持强大的去噪能力. 进一步, 图 7.3 展示了 PaviaU 数据集 (90, 130) 像素处的光谱反射率曲线. 可以观察到, FastHyMix、Eigen-CNN、STAR-Net 及 STAR-Net-S 的光谱变化趋势均与干净图像一致, 其中 STAR-Net-S 的光谱曲线与干净图像的光谱曲线几乎完全重叠, 表明了其在各波段去噪的有效性和强大的光谱恢复能力.

表 7.1: ICVL 数据集上的性能比较

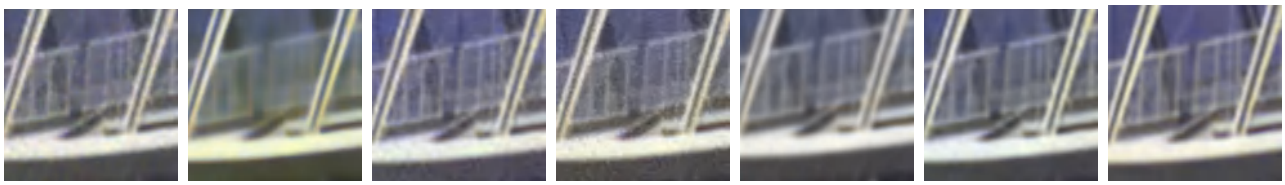
噪声	指标	Noisy	BM4D	LLRT	LRTDTV	NGMeet	NLSSR	FastHyMix	HSI-SDeCNN	SMDS-Net	Eigen-CNN	RCILD	STAR-Net	STAR-Net-S
10	PSNR ↑	29.018	42.987	39.810	43.882	42.383	45.928	43.628	41.519	46.371	47.321	42.458	47.286	**47.345**
	SSIM ↑	0.521	0.973	0.962	0.979	0.968	0.984	0.988	0.969	0.985	**0.989**	0.987	0.988	**0.989**
	SAM ↓	0.229	0.080	0.045	0.077	0.074	0.066	0.035	0.075	0.028	0.032	0.044	**0.025**	**0.025**
	ERGAS ↓	243.021	36.420	59.279	44.893	34.764	28.026	24.893	61.289	20.056	25.355	25.800	18.124	**17.951**
30	PSNR ↑	21.591	37.630	34.250	38.245	36.791	41.629	38.286	36.840	42.337	41.491	38.514	42.435	**42.500**
	SSIM ↑	0.146	0.930	0.921	0.877	0.915	0.968	0.966	0.926	**0.972**	0.963	0.971	**0.972**	**0.972**
	SAM ↓	0.535	0.142	0.084	0.149	0.139	0.084	0.068	0.124	0.040	0.060	0.067	0.039	**0.038**
	ERGAS ↓	729.026	62.662	40.725	69.464	60.101	41.388	48.675	103.084	32.393	49.458	49.362	32.056	**31.862**
50	PSNR ↑	18.402	35.242	32.067	33.659	34.399	39.713	35.397	34.342	37.481	36.579	35.838	39.853	**39.963**
	SSIM ↑	0.042	0.888	0.899	0.862	0.887	0.955	0.941	0.893	0.907	0.879	0.951	**0.956**	**0.956**
	SAM ↓	0.779	0.190	0.107	0.195	0.177	0.109	0.096	0.134	0.066	0.106	0.092	0.050	**0.047**
	ERGAS ↓	1215.105	81.133	54.869	110.351	80.585	53.811	68.681	136.304	55.786	85.426	68.945	43.362	**42.923**
70	PSNR ↑	18.126	33.586	30.746	30.565	32.389	37.450	33.377	32.794	37.197	32.194	33.980	37.342	**38.237**
	SSIM ↑	0.038	0.844	0.852	0.762	0.858	0.934	0.915	0.855	0.923	0.752	0.930	**0.943**	**0.943**
	SAM ↓	0.897	0.231	0.214	0.304	0.217	0.128	0.120	0.186	0.066	0.154	0.098	0.058	**0.055**
	ERGAS ↓	1701.060	97.267	66.690	163.788	97.309	65.893	88.790	173.430	58.223	126.017	89.340	57.341	**52.261**
平均	PSNR ↑	21.784	37.361	34.218	36.588	36.490	41.180	37.672	36.374	40.846	39.396	37.697	41.729	**42.011**
	SSIM ↑	0.187	0.909	0.909	0.870	0.907	0.960	0.953	0.911	0.947	0.896	0.960	**0.965**	**0.965**
	SAM ↓	0.610	0.161	0.113	0.181	0.152	0.097	0.080	0.130	0.050	0.088	0.075	0.043	**0.041**
	ERGAS ↓	972.053	69.370	55.391	97.124	68.190	47.280	57.760	118.527	41.614	71.564	58.362	37.721	**36.249**

表 7.2: PaviaU 数据集上的性能比较

噪声	指标	Noisy	BM4D	LLRT	LRTDTV	NGMeet	NLSSR	FastHy Mix	HSI-SDe CNN	SMDS-Net	Eigen-CNN	RCILD	STAR-Net	STAR-Net-S
10	PSNR ↑	29.181	34.044	30.671	31.796	36.493	35.419	39.646	38.509	35.882	39.660	40.509	40.881	**41.172**
	SSIM ↑	0.654	0.902	0.821	0.856	0.933	0.925	0.968	0.953	0.938	0.969	0.962	**0.971**	**0.971**
	SAM ↓	0.221	0.108	0.159	0.141	0.084	0.092	0.060	0.067	0.068	0.060	0.056	**0.049**	**0.049**
	ERGAS ↓	157.975	77.328	116.590	102.736	60.251	67.076	44.100	48.351	67.165	44.016	44.690	40.944	**39.927**
30	PSNR ↑	21.409	28.398	30.394	31.756	30.448	33.835	32.960	32.236	30.010	33.028	35.947	35.637	**35.976**
	SSIM ↑	0.237	0.746	0.809	0.855	0.818	0.901	0.919	0.849	0.929	0.922	0.899	0.935	**0.936**
	SAM ↓	0.580	0.202	0.165	0.141	0.610	0.111	0.125	0.134	0.077	0.124	0.090	0.074	**0.073**
	ERGAS ↓	473.723	147.139	120.494	103.199	119.247	80.904	96.429	98.539	73.304	95.860	69.178	68.764	**66.131**
50	PSNR ↑	18.760	26.159	27.022	31.648	27.756	31.750	31.909	29.198	31.748	32.031	31.416	33.227	**33.243**
	SSIM ↑	0.114	0.650	0.666	0.850	0.722	0.856	0.897	0.757	0.878	0.899	0.831	0.898	**0.902**
	SAM ↓	0.816	0.259	0.242	0.196	0.220	0.140	0.138	0.182	0.098	0.136	0.150	0.093	**0.092**
	ERGAS ↓	789.831	188.949	177.238	104.514	162.735	103.138	105.438	138.135	104.315	104.304	115.949	89.223	**88.743**
70	PSNR ↑	16.705	24.940	26.626	30.644	26.281	30.180	31.087	27.435	30.989	31.097	30.560	31.658	**31.694**
	SSIM ↑	0.064	0.595	0.643	0.820	0.659	0.814	**0.876**	0.694	0.859	0.872	0.779	0.868	0.868
	SAM ↓	0.971	0.298	0.254	0.160	0.265	0.166	0.150	0.215	0.113	0.150	0.157	**0.105**	**0.105**
	ERGAS ↓	1105.483	216.652	185.492	117.070	193.479	122.756	114.277	169.228	113.503	114.060	126.980	106.249	**105.060**
平均	PSNR ↑	21.514	28.385	28.678	31.461	30.245	32.796	33.901	31.844	32.157	33.954	34.608	35.351	**35.521**
	SSIM ↑	0.267	0.723	0.735	0.845	0.783	0.874	0.915	0.813	0.901	0.915	0.868	0.918	**0.919**
	SAM ↓	0.647	0.217	0.205	0.159	0.295	0.127	0.118	0.149	0.089	0.118	0.113	**0.080**	**0.080**
	ERGAS ↓	631.753	157.517	149.953	106.880	133.928	93.468	90.061	113.563	89.572	89.560	89.199	76.295	**74.965**

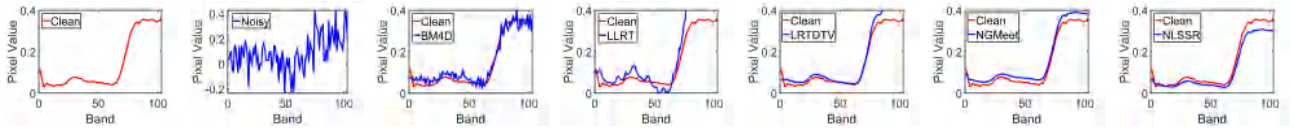


(a) Clean (b) Noisy (c) BM4D (d) LLRT (e) LRTDTV (f) NGMeet (g) NLSSR

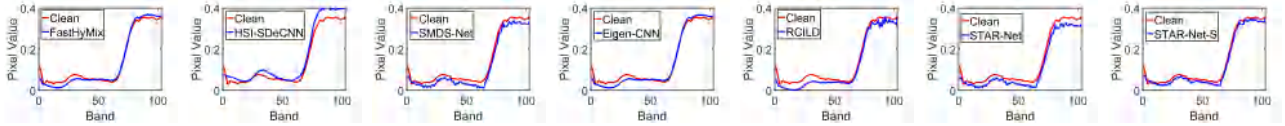


(h) FastHyMix (i) HSI-SDeCNN (j) SMDS-Net (k) Eigen-CNN (l) RCILD (m) STAR-Net (n) STAR-Net-S

图 7.2: 噪声方差为 50 时 gavyam_0823-0933 上的去噪结果



(a) Clean (b) Noisy (c) BM4D (d) LLRT (e) LRTDTV (f) NGMeet (g) NLSSR



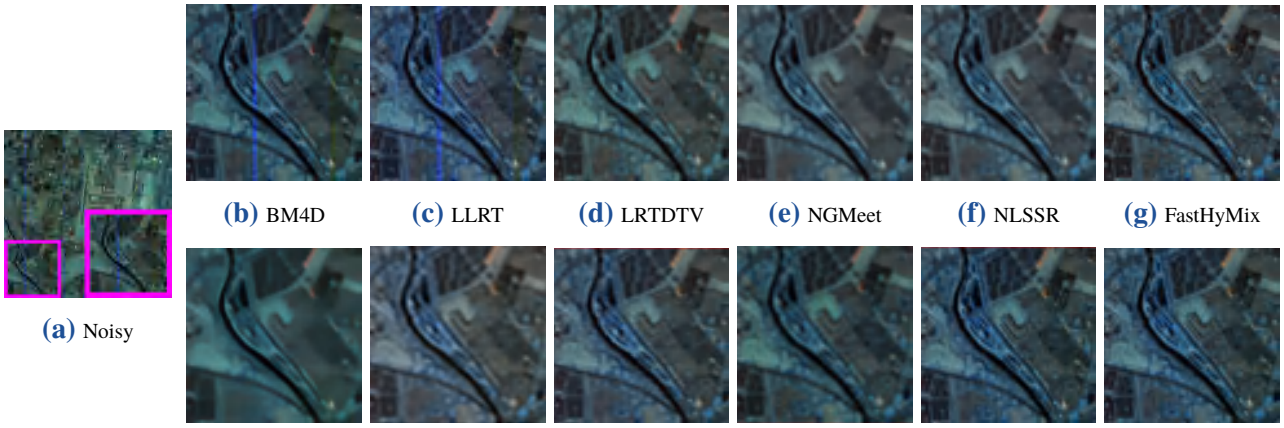
(h) FastHyMix (i) HSI-SDeCNN (j) SMDS-Net (k) Eigen-CNN (l) RCILD (m) STAR-Net (n) STAR-Net-S

图 7.3: 噪声方差为 50 时 PaviaU 上像素 (90, 130) 的去噪结果

7.4.3 真实数据结果

为进一步评估 STAR-Net 与 STAR-Net-S 的可靠性, 本节在含真实噪声的数据集上开展验证实验. 由于缺乏无噪声的干净遥感图像作为参照基准, 去噪效果的评估仅能通过分析去噪后遥感图像的视觉效果来实现. 如图 7.4 所示, BM4D 与 LLRT 去噪后仍存在显著的噪声残留, 未能有效抑制原始数据中的噪声干扰. 其余方法虽可实现一定程度的噪声去除, 但 Eigen-CNN、STAR-Net 及 STAR-Net-S 三者不仅能高效完成噪声抑制, 还能最大程度保留图像的原始信息, 避免了去噪过程中的信息失真.

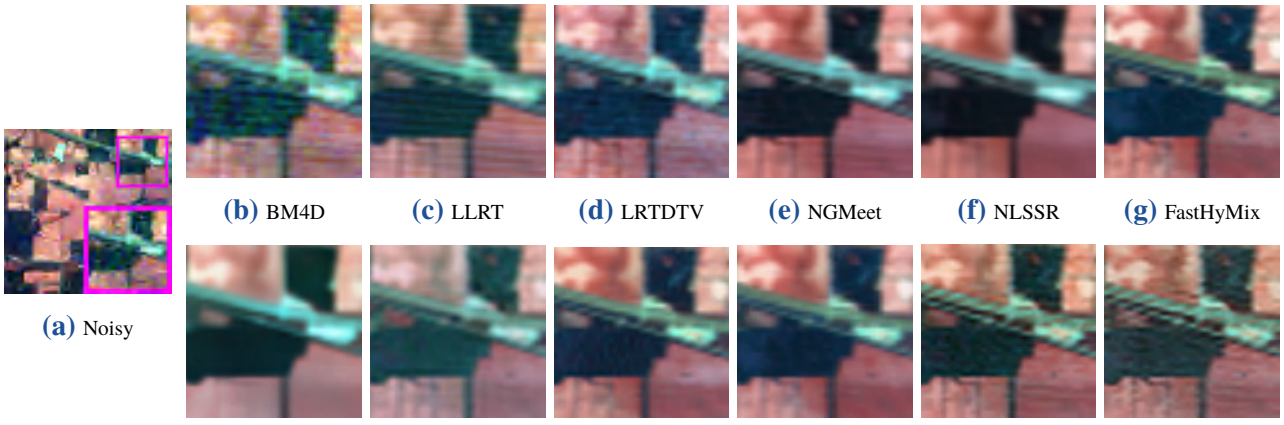
类似地, Indian Pines 数据集的去噪结果如图 7.5 所示. 结果表明, 虽然 SMDS-Net、Eigen-CNN 与 RCILD 可保留图像部分细节信息, 但其去噪性能与保真效果仍不及 STAR-Net 与 STAR-Net-S. 为量化不同去噪方法对下游任务的影响, 采用支持向量机开展分类实验, 如图 7.6 所示. 可以明显看出, STAR-Net-S 去噪后图像的分类结果与真实标签最为接近, 其次是 LLRT 与 STAR-Net. 该实验结果进一步验证了本文所提出的 STAR-Net 与 STAR-Net-S 的有效性.



(a) Noisy (b) BM4D (c) LLRT (d) LRTDTV (e) NGMeet (f) NLSSR (g) FastHyMix

(h) HSI-SDeCNN (i) SMDS-Net (j) Eigen-CNN (k) RCILD (l) STAR-Net (m) STAR-Net-S

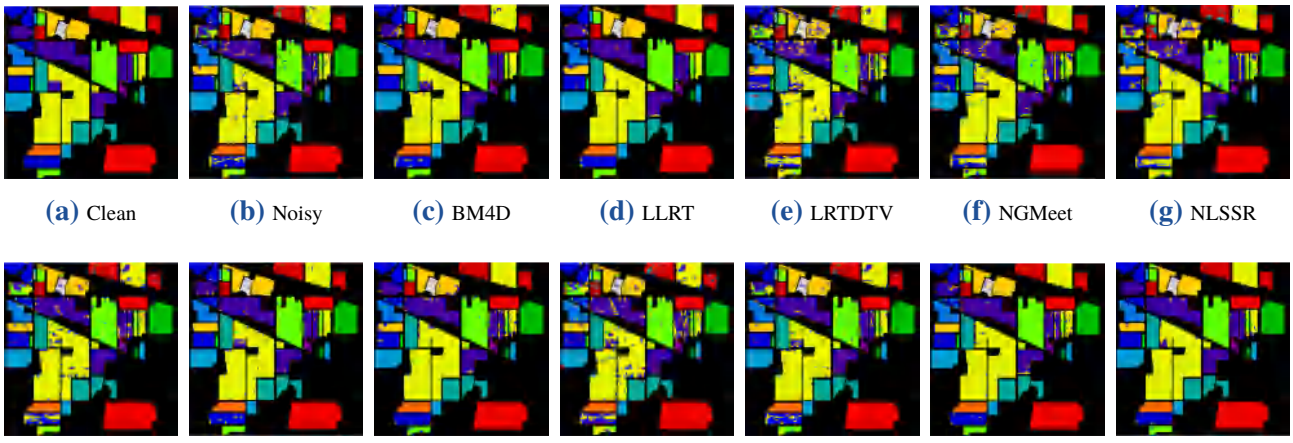
图 7.4: 北京首都机场数据集上的去噪结果



(a) Noisy (b) BM4D (c) LLRT (d) LRTDTV (e) NGMeet (f) NLSSR (g) FastHyMix

(h) HSI-SDeCNN (i) SMDS-Net (j) Eigen-CNN (k) RCILD (l) STAR-Net (m) STAR-Net-S

图 7.5: Indian Pines 数据集上的去噪结果



(a) Clean (b) Noisy (c) BM4D (d) LLRT (e) LRTDTV (f) NGMeet (g) NLSSR

(h) FastHyMix (i) HSI-SDeCNN (j) SMDS-Net (k) Eigen-CNN (l) RCILD (m) STAR-Net (n) STAR-Net-S

图 7.6: Indian Pines 数据集上的分类结果

7.4.4 讨论

(1) 参数量分析

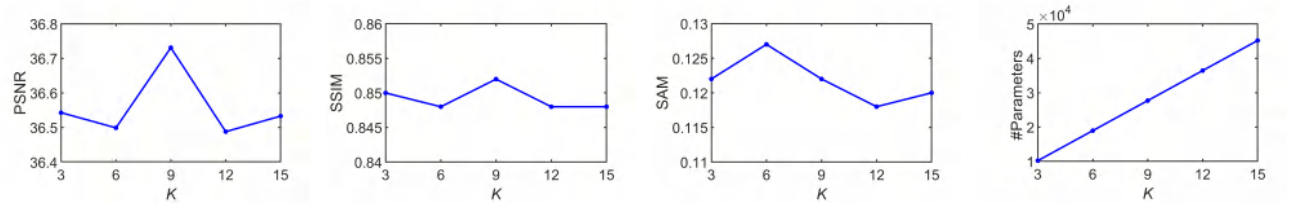
基于深度学习方法的参数量详细列于表 7.3. 由表可知, HSI-SDeCNN 与 RCILD 的参数量相对较多, SMDS-Net 的参数量最少. 相比来说, STAR-Net 与 STAR-Net-S 的参数量处于较低水平, 该差异主要源于后两者采用了模型辅助表示网络的设计思路, 有效实现了参数的精简. 后续章节将探讨采用卷机神经网络, 参数量会进一步降低.

表 7.3: 网络的参数量比较

方法	FastHyMix	HSI-SDeCNN	SMDS-Net	Eigen-CNN	RCILD	STAR-Net	STAR-Net-S
参数量	/	1,892,100	**5,103**	/	2,892,288	27,702	28,487

(2) 展开阶段数分析

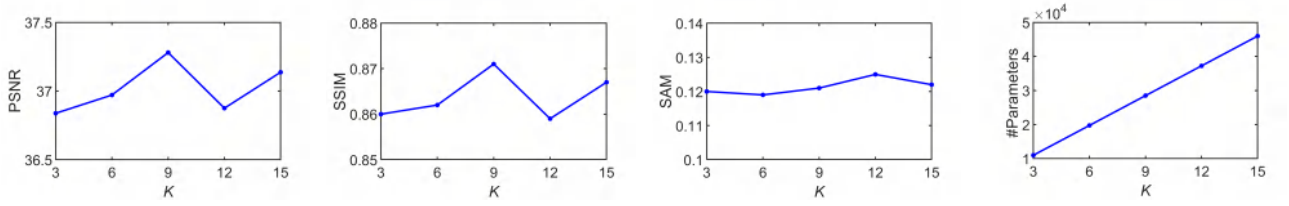
本节将分析 K 取值对 STAR-Net 与 STAR-Net-S 模型性能的影响. 由图 7.7 可知, 随着 K 值的增大, 网络参数量呈单调递增趋势. 对于 STAR-Net, 当 $K = 9$ 时, PSNR 与 SSIM 指标达到最优. 当 $K = 12$ 时, SAM 取得最佳值. 综合考虑四个指标, 将 STAR-Net 的展开阶段数确定为 9.



(a) PSNR **(b)** SSIM **(c)** SAM **(d)** #Params

图 7.7: STAR-Net 展开阶段数 K 的影响

同理, 图 7.8 的结果表明, STAR-Net-S 的最优展开阶段数同样为 9.



(a) PSNR **(b)** SSIM **(c)** SAM **(d)** #Params

图 7.8: STAR-Net-S 展开迭代次数 K 的影响

(3) 去噪可视化

图 7.9 展示了 STAR-Net-S 模型在 ICVL 与 PaviaU 数据集上的中间更新过程. 在迭代初始阶段, 模型主要聚焦于去除图像中的高频噪声, 使图像整体结构趋于清晰. 随着迭代过程的逐步推进, 模型进一步恢复图像的结构细节与纹理信息, 提升图像对比度, 同时消除残留噪声, 最终输出接近高质量干净图像的去噪结果.

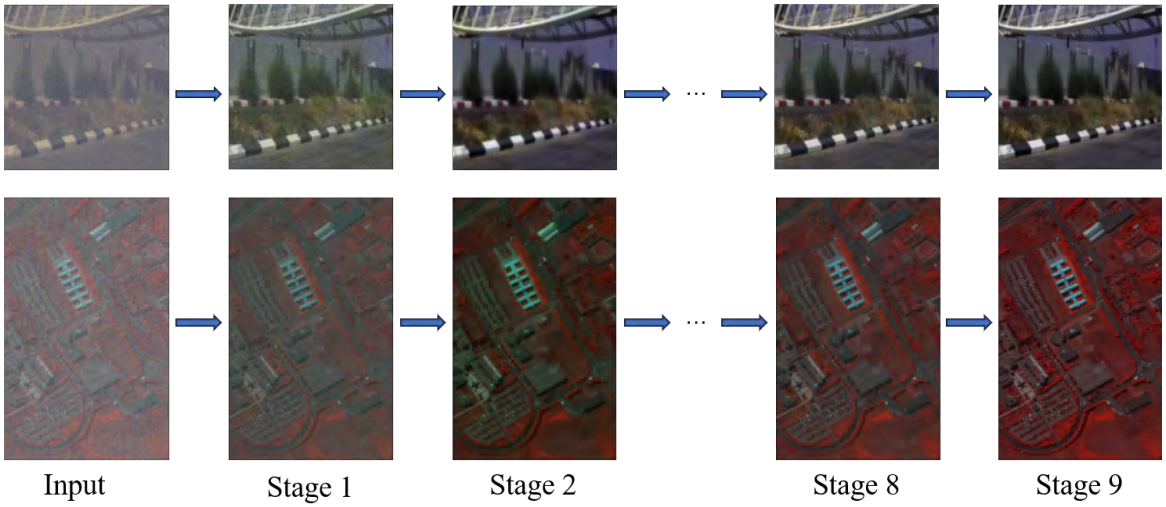


图 7.9: STAR-Net-S 的逐阶段可视化过程

(4) 初始化分析

表 7.4 给出了可学习参数 γ_1、γ_2、l、λ、μ、β 的不同初始值对模型性能的影响. 为便于对比分析, 实验中将所有可学习参数的初始值统一设置为相同数值。结果显示, 当初始值设为 0 时, 模型的各项性能指标均出现显著下降, 表明此时模型的学习能力受到严重抑制. 而在其他初始值设置下, 模型性能差异相对较小. 其中, 当初始值设置为 0.02 时, 模型在四个评价指标上均取得最优性能. 因此, 本章中所有可学习参数均初始化为 0.02.

表 7.4: 初始化的影响

指标	0	0.01	0.02	0.03	0.04
PSNR ↑	36.730	37.455	**37.548**	37.542	37.346
SSIM ↑	0.854	0.876	**0.879**	0.878	0.873
SAM ↓	0.134	0.117	**0.115**	0.116	0.117
ERGAS ↓	126.123	122.737	**120.349**	121.118	124.545

(5) 时间比较

表 7.5 中列出了对比方法在 ICVL 与 PaviaU 数据集上的计算时间. 由表可知, 传统基于模型的方法运行速度相对较慢, 其中 LLRT 的耗时最为显著. STAR-Net 与 STAR-Net-S 由于采用了将完整模型展开为网络组件的设计, 其运行时间长于 HSI-SDeCNN、Eigen-CNN、RCILD 等典型深度学习方法.

7.5 本章小结

本章针对高光谱图像易受噪声干扰问题, 提出了两种新颖的去噪方法, 即 STAR-Net 和 STAR-Net-S. 首先引入低秩先验以保留非局部自相似性, 并通过稀疏先验以提高对非高斯噪

表 7.5: 测试时间对比 (秒)

数据集	BM4D	LLRT	LRTDTV	NGMeet	NLSSR	FastHy-Mix
ICVL	561.307	888.305	136.796	335.265	206.964	8.555
PaviaU	1,123.815	2,581.424	354.911	716.722	491.568	82.029
数据集	HSI-SDeCNN	SMDS-Net	Eigen-CNN	RCILD	STAR-Net	STAR-Net-S
ICVL	11.093	105.109	**6.478**	24.743	107.552	107.958
PaviaU	60.522	349.292	**11.361**	30.706	337.133	341.220

声的鲁棒性. 随后, 将经典的交替方向乘子法框架与深度展开结合, 将迭代优化过程转换为可训练网络. 这种设计使模型能够以端到端的方式学习参数, 从而消除了传统基于模型方法通常需要的繁琐手动调参. 因此, STAR-Net 和 STAR-Net-S 继承了基于模型和基于深度学习方法的优势, 具有强大的可解释性和可学习性. 实验结果表明, STAR-Net 与 STAR-Net-S 在遥感图像去噪任务中表现出显著的优越性. 具体而言, 在 ICVL 数据集上, STAR-Net 与 STAR-Net-S 的 PSNR 较基准方法分别提升了 2.16% 和 2.85%. 此外, 还详细讨论了网络参数量、展开阶段数、可视化结果、网络初始化以及测试时间等.

第 8 章 基于深度自适应低秩稀疏的目标检测

红外小目标检测是图像处理中的关键技术之一. 尽管深度展开方法展现出良好的检测性能, 但现有方法在参数轻量化设计与复杂噪声鲁棒性方面仍面临重大挑战. 为此, 本章提出了基于鲁棒主成分分析的高度轻量化网络 (lightweight robust principal component analysis network, L-RPCANet). 在技术层面, 构建了一种层次化瓶颈结构, 用于对单通道输入红外图像进行通道维度的降维与升维, 实现逐通道特征精炼. 进一步, 嵌入了专门设计的噪声抑制模块, 增强算法对复杂场景下各类噪声的抗干扰能力. 此外, 利用压缩激励网络作为通道注意力机制, 捕捉不同特征在各通道间的重要性差异, 从而在保持轻量化和鲁棒性的同时实现优异性能. 实验表明, 相较于 RPCANet、DRPCANet 及 RPCANet++ 等当前领域内的最先进方法, 所提 L-RPCANet 在检测精度、速度及鲁棒性等方面均展现出优越性.

8.1 引言

与传统可见光成像技术不同, 红外成像通过记录物体自然发射的热辐射捕获环境信息, 使其在强电磁干扰、黑暗等复杂场景下仍能可靠检测小目标. 因此, 红外小目标检测 (infrared small target detection, ISTD) 受到学术界与工业界的广泛关注, 其应用已覆盖遥感监测、智能交通、航空航天及医学成像等多个领域. 作为一项基础性图像处理任务, 红外小目标检测面临两大主要挑战[126]. 一方面, 红外图像中的目标通常仅占据极少数像素, 且信噪比 (signal-to-noise ratio, SNR) 较低, 导致目标纹理信息极度匮乏. 另一方面, 移动检测设备的计算资源往往有限, 难以满足实时检测系统的低时延需求. 过去数十年间, 研究者们开发了大量红外小目标检测方法, 通常可分为三类: 基于模型的方法、基于数据的方法以及基于模型-数据混合的方法.

基于模型的红外小目标检测方法将检测任务融入物理模型框架. 低秩表示因其简洁的数学表达, 已被广泛应用于红外小目标检测任务. 低秩方法的核心是将缓慢变化且高度相关的背景分量建模为低秩矩阵, 将尺寸极小的目标分量建模为稀疏矩阵[127]. 基于此, Gao 等[128] 提出红外块图像 (infrared patch image, IPI), 利用经典主成分分析 (principal component analysis, PCA) 实现低秩背景矩阵与稀疏目标矩阵的同步分离, 进而提取目标纹理特征. 然而, IPI 将红外图像直接转换为二维矩阵的处理方式, 破坏了图像内部固有的空间-光谱邻域相关性. 为解决该问题, Zhang 等[129] 摒弃传统矩阵表征方式, 采用低秩张量捕获背景特征, 并通过基于张量核范数部分和 (partial sum of the tensor nuclear norm, PSTNN) 的非凸优化方法实现目标检测. 借助非局部自相似性原理, 上述两种方法均可在完成背景估计与目标提取的同时, 有效抑制大尺度纹理干扰与固定模式噪声[130]. 除低秩类方法外, Wei 等[131] 通过不同尺寸的邻域窗口逐像素扫描生成多层显著图, 再结合自适应阈值分割实现红外小目标提取, 提出多尺度基于块的对比度

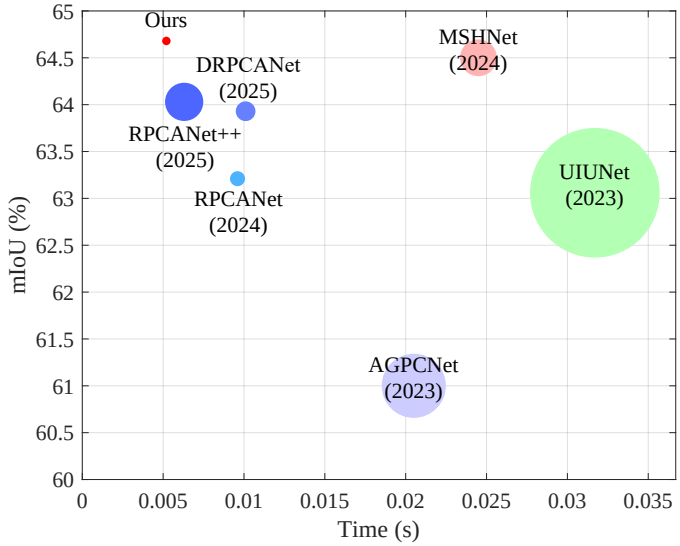


图 8.1: 现有深度网络方法的性能比较

测量 (multiscale patch-based contrast measurement, MPCM), 数值性能表现优异. 尽管基于模型的方法具有较强的可解释性, 但这类方法易受噪声与复杂背景的影响, 且建模过程中需引入大量参数, 导致鲁棒性与泛化能力受限.

基于数据的红外小目标检测方法依托各类语义分割网络实现端到端学习, 展现出良好的应用前景. 例如, Zhang 等[132] 将扩张空间金字塔池化模块集成至 ResNet-101 网络, 通过卷积与上采样操作融合低级细节特征与高级语义特征, 提出注意力引导金字塔上下文网络 (attention-guided pyramid context networks, AGPCNet), 其性能较 IPI、PSTNN 等传统方法有显著提升. Liu 等[133] 以 ResNet-50 为编码器, 在标准 U-Net 网络基础上设计简单多尺度头, 实现了轻量级多尺度语义分割, 简称为 MSHNet. 此外, Wu 等[134] 采用 U-Net 嵌套 U-Net 的双重结构, 在编解码过程中嵌入同构子网络, 通过外部网络保留目标全局轮廓、内部网络聚焦像素级热点区域的设计, 进一步提升目标检测的精准度, 称为 UIUNet. 然而, 数据驱动方法仍面临诸多挑战, 如网络融合阶段计算开销大、自训练过程需大量标注数据等, 限制了其在资源受限环境中的应用[135].

基于模型-数据混合的红外小目标检测方法既保留模型的可解释性, 又具备数据驱动方法的强拟合能力. 其中, 设计深度展开处理红外小目标检测任务是当前研究热点. 该类网络弥补了传统迭代优化算法与深度神经网络之间的鸿沟, 将机理建模与网络化求解有机结合[136]. 近年来, Wu 等[137] 将红外小目标检测任务转化为鲁棒主成分分析 (robust PCA, RPCA) 问题, 并将优化步骤展开为深度网络, 该方法称为 RPCANet. Xiong 等[138] 采用动态参数生成机制和动态残差组模块, 将稀疏目标从低秩背景中分离, 提出动态鲁棒主成分分析网络 (dynamic robust PCA network, DRPCANet). 为进一步提升检测效率, Wu 等[139] 引入记忆增强模块以保留背景特征、深度对比先验模块以加速目标提取, 在 RPCANet 基础上提出 RPCANet++. 这些研究充分表明了深度展开在平衡检测性能与可解释性方面的优势, 也激发了本章将鲁棒主成分分析框架应用于红外小目标检测任务的研究兴趣.

受上述研究启发, 本章提出了具有增强鲁棒性的轻量级深度展开网络 (lightweight robust PCA network, L-RPCANet). 与现有 RPCANet、DRPCANet 及 RPCANet++ 相比, 该方法不仅提升了红外小目标检测的性能与噪声鲁棒性, 还通过轻量化设计减少了模型参数, 更适用于实时检测场景. 如图 8.1 所示, 在 IRSTD-1k 数据集上, 所提 L-RPCANet 具有最小的参数量, 且兼具检测性能和推理时间. 本章的主要贡献为

(1) 通过对单通道红外输入图像进行维度的降维与升维转换, 构建了层次化瓶颈结构, 有效解决红外小目标特征提取困难的问题.

(2) 通过引入压缩激励网络作为通道注意力机制, 在维持轻量化架构的前提下提升模型检测性能, 有效缓解红外图像中小目标易被背景淹没的问题.

(3) 通过增加噪声处理模块, 显著提升模型应对复杂背景的鲁棒性, 解决了传统红外小目标检测方法易受噪声干扰的问题.

8.2 相关工作

8.2.1 深度展开网络

深度展开网络的研究可追溯至 Gregor 与 LeCun[140] 的开创性工作, 他们将迭代软阈值算法 (iterative soft thresholding algorithm, ISTA) 拓展为可学习的 LISTA (learned ISTA), 通过前馈神经网络架构实现了压缩感知 (compressed sensing, CS) 问题的高效求解. 在此基础上, Yang 等[123] 将交替方向乘子法 (alternating direction method of multipliers, ADMM) 展开为可学习框架, 称为 ADMM-CSNet. 最近, Sun 等[141] 引入卷积神经网络 (convolutional neural networks, CNNs) 来提升 LISTA 的性能, 提出 ISTA-Net. 进一步, You 等[142] 通过引入残差连接与更深层的网络结构, 构建改进版本 ISTA-Net++, 大幅增强了模型的特征提取能力. 此外, Han 等[143] 通过动态生成卷积权重与阈值参数, 将传统稀疏重建方法转化为自适应的动态深度学习框架, 提出动态迭代收缩阈值网络 (dynamic iterative shrinkage thresholding network, DISTANet), 进一步提升了模型对复杂场景的适配能力.

在红外小目标检测任务中, 深度展开网络与鲁棒主成分分析的结合已被证实具有良好的应用前景, 但现有相关方法仍存在一些不足. 例如, RPCANet[137] 缺乏逐通道特征优先级排序机制与有效的噪声处理模块, 限制了其在复杂噪声场景中的检测性能. DRPCANet[138] 过度依赖输入特征的固有分布, 导致在真实复杂场景中易出现误报目标. RPCANet++[139] 采用的全卷积架构设计虽提升了特征提取的完整性, 却导致模型推理速度显著慢于轻量级网络. 针对上述问题, 本章提出的方法在轻量化扩展过程中, 嵌入瓶颈层与可学习噪声降低模块, 旨在克服 RPCANet 的特征盲区与噪声敏感性、DRPCANet 对特定输入的过度依赖, 以及 RPCANet++ 计算成本过高的缺陷.

8.2.2 注意力机制

注意力机制能够引导神经网络在生成预测时, 自适应地聚焦于输入数据中的相关区域, 目前已被广泛应用于图像分类、语义分割、目标检测等领域. 在红外小目标检测任务中, 由于目标像素占比极低, 目标信息往往在深层语义特征提取过程中被背景噪声淹没. 因此, 大多数数据驱动红外小目标检测方法均引入注意力机制增强特征表示能力, 从而提升检测性能[144].

具体而言, UIUNet[134] 引入空间注意力机制, 通过生成像素级权重图, 有效突出目标区域特征并抑制背景纹理干扰. DRPCANet[138] 设计动态残差组模块, 将残差学习与动态空间注意力机制相结合, 使模型能够更精准地捕获背景区域的上下文变化, 进而实现更可靠的低秩背景估计与小目标分离. 此外, RPCANet++[139] 采用时序演化注意力机制, 通过门控机制动态分配跨阶段背景特征的权重, 这种隐式时空注意力机制不仅避免了显式计算相似矩阵带来的高额计算负担, 还能在迭代展开过程中, 自适应聚焦于低秩背景重建中最具价值的通道与空间位置信息. 综上, 现有应用于红外小目标检测任务的注意力机制虽各有特色, 但普遍面临高计算复杂度、特征选择偏差、模型训练难度大等挑战.

8.3 模型与算法

8.3.1 数学模型

给定红外图像数据矩阵 $\boldsymbol{D} \in \mathbb{R}^{m\times n}$, 结合红外成像物理先验知识, 可将其分解为背景分量 $\boldsymbol{B} \in \mathbb{R}^{m\times n}$、小目标分量 $\boldsymbol{T} \in \mathbb{R}^{m\times n}$ 与噪声分量 $\boldsymbol{N} \in \mathbb{R}^{m\times n}$ 的叠加形式, 即

$$\boldsymbol{D} = \boldsymbol{B} + \boldsymbol{T} + \boldsymbol{N}. \tag{8.1}$$

一般来说, 红外图像中的背景普遍具备低秩结构特性, 小目标呈现显著的空间稀疏分布特征, 而噪声通常近似服从高斯分布. 基于鲁棒主成分分析思想, 红外小目标检测任务可构造为如下约束优化模型

$$\begin{aligned} \min_{\boldsymbol{B},\boldsymbol{T},\boldsymbol{N}} \quad & \|\boldsymbol{B}\|_* + \lambda\|\boldsymbol{T}\|_1 + \frac{\mu}{2}\|\boldsymbol{N}\|_F^2 \\ \text{s.t.} \quad & \boldsymbol{D} = \boldsymbol{B} + \boldsymbol{T} + \boldsymbol{N}, \end{aligned} \tag{8.2}$$

其中, $\lambda, \mu > 0$ 为正则参数, $\|\boldsymbol{B}\|_*$ 表示矩阵核范数 (即所有奇异值之和), $\|\boldsymbol{T}\|_1$ 为矩阵 ℓ_1 范数 (即所有元素绝对值之和), $\|\boldsymbol{N}\|_F^2$ 为 Frobenius 范数的平方.

然而, 在实际复杂红外场景中, 背景纹理与目标形态往往非常复杂, 单一固定范数约束难以精准刻画数据真实结构, 同时实际噪声分布往往偏离理想高斯假设. 为此, 本章引入更一般的函数 $\mathcal{L}(\boldsymbol{B})$、$\mathcal{S}(\boldsymbol{T})$ 与 $\mathcal{R}(\boldsymbol{N})$, 分别自适应表征背景低秩特性、目标稀疏特性与复杂噪声分布特征. 据此, 将式 (8.2) 的经典优化模型推广为如下通用形式

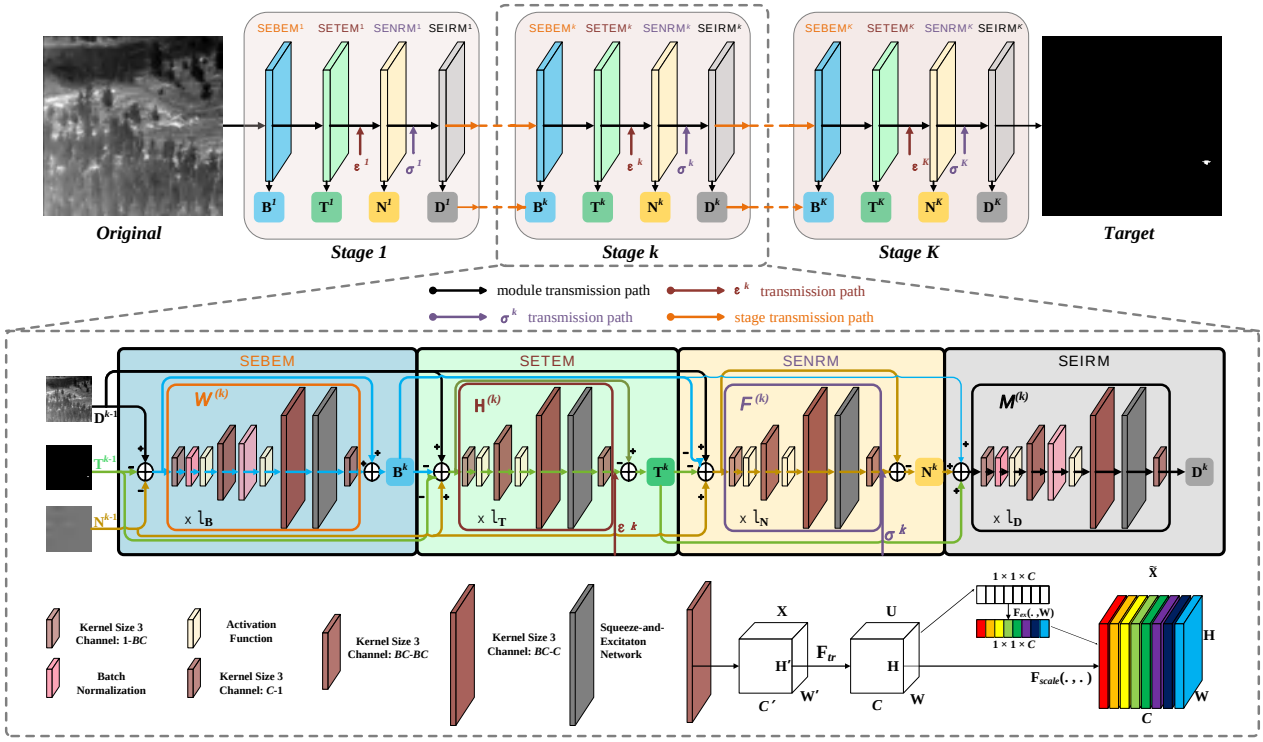


图 8.2: 所提 L-RPCANet 的整体结构

$$\begin{aligned} \min_{\boldsymbol{B},\boldsymbol{T},\boldsymbol{N}} \quad & \mathcal{L}(\boldsymbol{B}) + \lambda \mathcal{S}(\boldsymbol{T}) + \mu \mathcal{R}(\boldsymbol{N}) \\ \text{s.t.} \quad & \boldsymbol{D} = \boldsymbol{B} + \boldsymbol{T} + \boldsymbol{N}. \end{aligned} \tag{8.3}$$

相较于 RPCANet[137]、DRPCANet[138] 与 RPCANet++[139]等现有方法, 式 (8.3) 在分解约束中引入可学习的噪声项 $\mathcal{R}(\boldsymbol{N})$, 更具鲁棒性.

通过罚函数法, 可将式 (8.3) 等价转化为以下无约束优化问题

$$L(\boldsymbol{B},\boldsymbol{T},\boldsymbol{N}) = \mathcal{L}(\boldsymbol{B}) + \lambda \mathcal{S}(\boldsymbol{T}) + \mu \mathcal{R}(\boldsymbol{N}) + \frac{\alpha}{2}\|\boldsymbol{D} - \boldsymbol{B} - \boldsymbol{T} - \boldsymbol{N}\|_F^2, \tag{8.4}$$

其中, $\alpha > 0$ 为惩罚参数. 后续章节将详细阐述如何使用神经网络更新 $\boldsymbol{B}$、$\boldsymbol{T}$ 和 $\boldsymbol{N}$. 同时, 本章额外引入图像重建模块对观测矩阵 $\boldsymbol{D}$ 进行更新, 弥补了传统迭代优化方法忽略原始观测数据修正的缺陷.

8.3.2 网络架构

所提 L-RPCANet 的整体结构如图 8.2 所示. 在介绍模块设计前, 首先引入压缩激励网络 (squeeze-and-excitation networks, SENets) 作为基础特征增强单元, 其基础结构如图 8.2 右下角所示. 压缩操作依托全局平均池化完成特征图的空间维度压缩, 为每一通道生成专属通道描述符, 有效捕获全局空间范围内特征响应的分布特性. 激励操作则通过自门控机制, 为各通道自适应学习样本专属激活权重, 实现通道维度的特征重要性动态建模.

8.3.2.1 更新 $\boldsymbol{B}$

针对背景分量 $\boldsymbol{B}$, 子问题可表述为

$$\boldsymbol{B}^k = \underset{\boldsymbol{B}}{\operatorname{argmin}}\ \mathcal{L}(\boldsymbol{B}) + \frac{\alpha}{2}\|\boldsymbol{D}^{k-1} - \boldsymbol{B} - \boldsymbol{T}^{k-1} - \boldsymbol{N}^{k-1}\|_F^2. \tag{8.5}$$

当正则项取核范数约束 $\mathcal{L}(\boldsymbol{B}) = \|\boldsymbol{B}\|_*$ 时, 可直接采用奇异值阈值 (singular value thresholding, SVT) 算法求解, 对应近端算子记为 $\operatorname{prox}_{\alpha\|\cdot\|_*}(\cdot)$. 但该传统求解方案应对大规模数据时计算开销较高, 正则参数 α 的选取缺乏自适应策略. 同时, 背景分量往往具备更复杂的低秩先验, 如加权核范数、非凸重叠核范数等. 为统一拟合各类低秩近端映射算子, 并实现参数自适应学习, 本节借鉴文献[141] 的网络化优化思想, 构建残差结构 $\operatorname{proxNet}(\cdot)$ 替代传统解析算子. 由此, 优化问题 (8.5) 的解可表示为

$$\begin{aligned}\boldsymbol{B}^k &= \operatorname{proxNet}(\boldsymbol{D}^{k-1} - \boldsymbol{T}^{k-1} - \boldsymbol{N}^{k-1}) \\ &\approx \boldsymbol{D}^{k-1} - \boldsymbol{T}^{k-1} - \boldsymbol{N}^{k-1} + \mathcal{W}^k(\boldsymbol{D}^{k-1} - \boldsymbol{T}^{k-1} - \boldsymbol{N}^{k-1}),\end{aligned} \tag{8.6}$$

其中, $\mathcal{W}^k(\cdot)$ 为 3×3 卷积组.

如图 8.2 所示, SEBEM 模块完成背景分量 $\boldsymbol{B}^k$ 的估计. 具体而言, $\mathcal{W}^k(\cdot)$ 包含两次跨维度通道映射, 首先将原始通道数映射至中间瓶颈维度 BC, 再由 BC 维度映射回原始通道数 C. 该模块引入批归一化 (batch normalization, BN) 与修正线性单元 (rectified linear unit, ReLU) 构建非线性变换. 经 SENets 通道注意力机制调制后, 具有 C 通道的特征图被调整为每个通道的权重, 从而增强表示能力.

8.3.2.2 更新 $\boldsymbol{T}$

针对稀疏目标分量 $\boldsymbol{T}$, 子问题写成如下形式

$$\boldsymbol{T}^k = \underset{\boldsymbol{T}}{\operatorname{argmin}}\ \lambda\mathcal{S}(\boldsymbol{T}) + \frac{\alpha}{2}\|\boldsymbol{D}^{k-1} - \boldsymbol{B}^k - \boldsymbol{T} - \boldsymbol{N}^{k-1}\|_F^2. \tag{8.7}$$

若稀疏正则项取 $\mathcal{S}(\boldsymbol{T}) = \|\boldsymbol{T}\|_1$, 现有研究已形成一系列高效求解算法. 然而此类方法泛化性不足, 难以适配非凸稀疏、复合稀疏等复杂先验约束. 为获得更简单直观的形式, 本节利用一阶泰勒展开对稀疏正则项 $\mathcal{S}(\boldsymbol{T})$ 进行局部近似, 将原问题转化为

$$\begin{aligned}\boldsymbol{T}^k = \underset{\boldsymbol{T}}{\operatorname{argmin}}\ &\frac{\lambda L_T}{2}\|\boldsymbol{T} - \boldsymbol{T}^{k-1} - \frac{1}{L_T}\nabla\mathcal{S}(\boldsymbol{T}^{k-1})\|_F^2 \\ &+ \frac{\alpha}{2}\|\boldsymbol{D}^{k-1} - \boldsymbol{B}^k - \boldsymbol{T} - \boldsymbol{N}^{k-1}\|_F^2,\end{aligned} \tag{8.8}$$

其中, $\nabla\mathcal{S}(\boldsymbol{T}^{k-1})$ 为 $\boldsymbol{T}^{k-1}$ 处的梯度, L_T 为函数 $\mathcal{S}(\boldsymbol{T}^{k-1})$ 对应的 Lipschitz 常数. 对目标函数求导并令梯度为零, 得到

$$\begin{aligned}\boldsymbol{T}^{k} = &\frac{\lambda L_T}{\lambda L_T+\alpha}\boldsymbol{T}^{k-1} + \frac{\alpha}{\lambda L_T+\alpha}(\boldsymbol{D}^{k-1}-\boldsymbol{B}^{k}-\boldsymbol{N}^{k-1}) \\ &- \frac{\lambda}{\lambda L_T+\alpha}\nabla\mathcal{S}(\boldsymbol{T}^{k-1}).\end{aligned} \tag{8.9}$$

为简化表达式, 定义

$$\gamma = \frac{\lambda L_T}{\lambda L_T+\alpha},\ \varepsilon = \frac{\lambda}{\lambda L_T+\alpha}, \tag{8.10}$$

则式 (8.8) 可等价改写为

$$\boldsymbol{T}^{k} = \gamma\boldsymbol{T}^{k-1} + (1-\gamma)(\boldsymbol{D}^{k-1}-\boldsymbol{B}^{k}-\boldsymbol{N}^{k-1}) - \varepsilon\nabla\mathcal{S}(\boldsymbol{T}^{k-1}). \tag{8.11}$$

进一步, 固定系数 $\gamma = 0.5$, 并将 ε 松弛为各迭代阶段可学习的自适应参数 ε^k. 最终, 稀疏目标分量的迭代更新规则为

$$\boldsymbol{T}^{k} \approx \boldsymbol{T}^{k-1} + \boldsymbol{D}^{k-1} - \boldsymbol{B}^{k} - \boldsymbol{N}^{k-1} - \varepsilon^{k}\mathcal{H}^{k}(\boldsymbol{T}^{k-1} + \boldsymbol{D}^{k-1} - \boldsymbol{B}^{k} - \boldsymbol{N}^{k-1}), \tag{8.12}$$

其中, $\mathcal{H}^{k}(\cdot)$ 由浅层卷积网络构成, 用于数值拟合稀疏正则项的梯度映射 $\nabla\mathcal{S}(\cdot)$.

图 8.2 中的 SETEM 模块实现稀疏目标分量 $\boldsymbol{T}^{k}$ 的自适应提取. 该模块沿用与 SEBEM 一致的双路径通道映射策略与瓶颈特征提取结构, 区别在于, 批归一化层的缩放与偏置参数会随训练动态更新, 破坏映射变换的 Lipschitz 连续性[145]. 因此, 此处移除批归一化层以保障优化稳定性. 同时, 模块依托轻量化卷积层捕捉梯度的空间局部变化特征, 结合 ReLU 非线性激活函数强化梯度信息的表征与反向传播能力, 在无归一化约束的条件下更准确地逼近稀疏梯度映射 $\nabla\mathcal{S}(\cdot)$.

8.3.2.3 更新 N

针对噪声分量 $\boldsymbol{N}$, 沿用与式 (8.7)-(8.8) 的近似技术, 同理推导可得噪声分量迭代形式

$$\begin{aligned}\boldsymbol{N}^{k} = &\frac{\mu L_N}{\mu L_N+\alpha}\boldsymbol{N}^{k-1} + \frac{\alpha}{\mu L_N+\alpha}(\boldsymbol{D}^{k-1}-\boldsymbol{B}^{k}-\boldsymbol{T}^{k}) \\ &- \frac{\mu}{\mu L_N+\alpha}\nabla\mathcal{G}(\boldsymbol{N}^{k-1}),\end{aligned} \tag{8.13}$$

其中, $\nabla\mathcal{G}(\boldsymbol{N}^{k-1})$ 为噪声正则函数的梯度, L_N 为对应 Lipschitz 常数.

引入记号

$$\delta = \frac{\mu L_N}{\mu L_N+\alpha},\ \sigma = \frac{\mu}{\mu L_N+\alpha}, \tag{8.14}$$

固定系数 $\delta = 0.5$, 对式 (8.13) 做残差化近似, 得到

$$\begin{aligned}\boldsymbol{N}^{k} &= \delta\boldsymbol{N}^{k-1} + (1-\delta)(\boldsymbol{D}^{k-1}-\boldsymbol{B}^{k}-\boldsymbol{T}^{k}) - \sigma\nabla\mathcal{G}(\boldsymbol{N}^{k-1}) \\ &\approx \boldsymbol{N}^{k-1} + \boldsymbol{D}^{k-1} - \boldsymbol{B}^{k} - \boldsymbol{T}^{k} - \sigma^{k}\mathcal{F}^{k}(\boldsymbol{N}^{k-1} + \boldsymbol{D}^{k-1} - \boldsymbol{B}^{k} - \boldsymbol{T}^{k}).\end{aligned} \tag{8.15}$$

如图 8.2 所示, SENRM 模块以 SEBEM 更新的 $\boldsymbol{B}^k$、SETEM 更新的 $\boldsymbol{T}^k$、噪声 $\boldsymbol{N}^{k-1}$ 和重建结果 $\boldsymbol{D}^{k-1}$ 作为输入, 完成噪声分量分离. 尽管 $\mathcal{F}^k(\cdot)$ 的残差连接、网络结构设计与前述 $\mathcal{H}^k(\cdot)$ 相近, 但二者参数空间相互独立, 从而实现不同的功能.

8.3.2.4 更新 D

对于重建部分 $\boldsymbol{D}$, 按如下方式更新

$$\boldsymbol{D}^k = \boldsymbol{B}^k + \boldsymbol{T}^k + \boldsymbol{N}^k \approx \mathcal{M}^k(\boldsymbol{B}^k + \boldsymbol{T}^k + \boldsymbol{N}^k), \tag{8.16}$$

其中, $\mathcal{M}^k(\cdot)$ 为轻量化卷积神经网络, 网络结构参考文献[118], 统一卷积核配置并设置 $l_D = 3$ 层中间隐藏层, 如图 8.2 所示.

8.4 数值实验

本节将所提 L-RPCANet 与当前主流基准方法进行对比, 包括基于模型的方法: IPI[1] (2013)、MPCM[2] (2016)、PSTNN[3] (2019), 基于数动方法: AGPCNet[4] (2023)、UIUNet[5] (2023)、MSHNet[6] (2024), 以及基于模型-数据的方法: RPCANet[7](2024)、DRPCANet[8] (2025)、RPCANet++[9] (2025). 此外, 所提方法开源代码见链接 https://github.com/xianchaoxiu/L-RPCANet.

8.4.1 实验设置

(1) 数据集

实验选用红外小目标检测领域的三类典型数据集, 包括小型数据集 NUDT-SIRST[10]、IRSTD-1k[11], 以及大型数据集 SIRST-Aug[12]. 这些数据集涵盖了真实与合成两类红外成像场景, 不仅在目标特性 (大小、强度、形状) 上存在显著差异, 还包含了城市、海洋、航空、自然景观等多种复

[1] https://github.com/gaocq/IPI-for-small-target-detection

[2] https://github.com/wzy-99/MPCM

[3] https://github.com/Lanneeee/Infrared-Small-Target-Detection-based-on-PSTNN

[4] https://github.com/Tianfang-Zhang/AGPCNet

[5] https://github.com/danfenghong/IEEE_TIP_UIU-Net

[6] https://github.com/Lliu666/MSHNet

[7] https://github.com/fengyiwu98/RPCANet

[8] https://github.com/GrokCV/DRPCA-Net

[9] https://github.com/fengyiwu98/RPCANet

[10] https://github.com/YeRen123455/Infrared-Small-Target-Detection

[11] https://github.com/RuiZhang97/ISNet

[12] https://github.com/Tianfang-Zhang/AGPCNet

杂背景, 同时兼顾了不同传感器的成像特性差异. 图像分辨率方面, NUDT-SIRST 与 SIRST-Aug 数据集的图像尺寸为 256×256 像素, IRSTD-1k 数据集的图像尺寸为 512×512 像素.

(2) 参数设置

所提方法基于 PyTorch 深度学习框架实现, 训练过程依托 Nvidia GeForce 4090 GPU 完成, 在每个数据集上均训练 400 个轮次以确保模型充分收敛. 优化器采用 Adam, 初始学习率设置为 10^{-4}, 批处理大小设为 8, 其余超参数通过交叉验证确定最优值. 为保证对比的公平性, 所有对比方法均严格遵循其官方代码中的默认参数配置, 未进行额外调优.

(3) 损失函数

红外小目标检测任务本质上可分解为目标分割与红外图像重建两个子任务, 因此实验设计的总损失函数由分割损失 $L_{\text{segmentation}}$ 与保真度损失 L_{fidelity} 两部分构成. 其中, 分割损失采用 SoftIoU 指标[146] 评估目标分割的像素级精度, 保真度损失通过原始图像与重建图像之间的最小方差衡量图像重建质量. 具体地, 损失函数定义为

$$\begin{aligned} L_{\text{total}} &= L_{\text{segmentation}} + \eta \cdot L_{\text{fidelity}} \\ &= (1 - \frac{1}{M_t}\sum_{i=1}^{M_t}\frac{\text{TP}}{\text{FP} + \text{TP} + \text{FN}}) + \eta \cdot \frac{1}{M_t M}\sum_{i=1}^{M_t}\|\boldsymbol{D}^K - \boldsymbol{D}\|_F^2, \end{aligned} \tag{8.17}$$

其中, i 表示第 i 个训练样本, M_t 为训练样本总数, M 为单张图像的总像素数. TP (true positive)、FP (false positive)、FN (false negative) 分别表示真正例、假正例、假负例像素数. 此外, η 为平衡系数, 用于调节分割损失与保真度损失的贡献权重, 其最优取值将在后续章节评估.

(4) 评估指标

参考文献[137], 实验选取四个指标来评估红外小目标检测的性能. 其中, 平均交并比 (mean intersection over union, mIoU) 主要用于评估目标分割任务的性能, F_1 分数、检测概率 (probability of detection, P_d) 与虚警率 (fault alarm rate, F_a) 则主要用于评估红外图像重建任务的性能.

- 平均交并比: 作为语义分割任务的像素级评估指标, 用于衡量预测分割结果与真实标签的重叠程度. 设 M 为类别总数, IoU_c 为第 c 类别的交并比, 则 mIoU 定义为

$$\text{mIoU} = \frac{1}{M}\sum_{c=1}^{M}\text{IoU}_c. \tag{8.18}$$

- F_1 分数: 综合精确率 (precision) 与召回率 (recall), 其中精确率表示预测为正例的像素中真正例像素的比例, 召回率表示所有真正例像素中被成功检测的比例. F_1 分数定义为

$$\text{F}_1 = 2 \cdot \frac{\text{precision} \cdot \text{recall}}{\text{precision} + \text{recall}}. \tag{8.19}$$

- 检测概率: 用于评估对比方法对真实目标的检测能力, 即所有真实目标中被正确检测到的比例. P_d 定义为

表 8.1: 不同方法的性能比较

方法	参数量	NUDT-SIRST				SIRST-Aug				IRSTD-1k				时间 (s)
		mIoU ↑	F_1 ↑	P_d ↑	F_a ↓	mIoU ↑	F_1 ↑	P_d ↑	F_a ↓	mIoU ↑	F_1 ↑	P_d ↑	F_a ↓	CPU/GPU
IPI	–	34.83	51.49	92.58	7.14	21.90	35.97	80.36	**2.20**	18.67	31.48	78.54	11.11	3.0972/–
MPCM	–	25.96	40.78	78.59	7.91	19.49	33.00	93.58	3.04	14.81	25.93	69.03	6.51	0.0624/–
PSTNN	–	25.46	40.58	78.52	7.95	19.76	33.00	93.40	3.14	14.87	25.89	68.73	6.51	0.2249/–
AGPCNet	12.360M	85.31	92.45	97.90	4.77	72.36	83.83	99.03	35.56	61.00	75.75	89.35	5.34	–/0.0205
UIUNet	50.540M	88.71	94.01	91.43	1.89	71.80	83.59	98.35	28.29	63.06	77.35	**93.60**	6.57	–/0.0317
MSHNet	4.065M	89.99	93.57	96.07	2.63	71.64	84.16	90.78	23.09	64.50	77.55	91.68	4.46	–/0.0245
RPCANet	0.680M	89.31	94.35	97.14	2.87	72.54	84.08	98.21	34.14	63.21	77.45	88.31	4.39	–/0.0096
DRPCANet	1.169M	**93.12**	96.02	98.02	1.95	73.93	85.39	98.12	30.45	63.93	78.15	92.09	4.92	–/0.0101
RPCANet++	4.396M	92.46	96.05	98.05	**1.44**	73.14	84.39	97.36	32.48	64.03	77.26	89.35	**4.28**	–/0.0063
Ours	**0.216M**	92.37	**96.54**	**98.41**	1.79	**74.56**	**85.43**	**99.17**	29.78	**64.68**	**78.55**	89.39	4.66	–/**0.0052**

$$P_d = \frac{TP}{TP + FN}. \tag{8.20}$$

- 虚警率: 用于衡量对比方法的误检程度, 即错误预测为正例的像素 (P_{false}) 占图像总像素数 (P_{all}) 的比例. F_a 定义为

$$F_a = \frac{P_{false}}{P_{all}}. \tag{8.21}$$

8.4.2 实验比较

表 8.1 列出了所有对比方法的实验结果. 可以清晰地看出, 与基于数据和基于模型-数据的方法相比, 基于模型的方法 (即 IPI、MPCM 和 PSTNN) 在 mIoU 和 F_1 方面上表现欠佳, 存在显著的性能差距. 这表明, 引入神经网络对提升语义分割能力的重要作用. 此外, 基于数据的方法 (如 AGPCNet) 存在明显的过拟合现象及参数冗余问题, 导致其在不同数据集上的性能表现波动较大. 相比之下, 基于模型-数据的方法通过融合基于物理的先验知识与数据引导的精确刻画, 成功完成了红外小目标检测任务. 需要注意的是, RPCANet 由于缺乏对通道间映射关系的分析, 模型鲁棒性较弱. 而 RPCANet++ 虽性能有所提升, 但复杂的网络结构设计导致其所需网络参数大幅增加, 推理效率受到影响. 总体而言, 所提 L-RPCANet 通过构建带有注意力机制的层次化通道映射结构, 并引入噪声降低模块, 不仅实现了令人满意的检测性能, 还具备网络参数更少、GPU 推理速度更快的优势.

此外, 表 8.2 给出了各方法的受试者工作特征 (receiver operating characteristic, ROC) 曲线下面积 (area under curve, AUC) 结果, 可见所提 L-RPCANet 在几乎所有数据集上均保持了较高的 AUC 值, 展现了其优异的跨域鲁棒性.

表 8.2: 不同方法的 AUC 比较

方法	NUDT-SIRST	SIRST-Aug	IRSTD-1k
IPI	0.8746	0.8344	0.7946
MPCM	0.8645	0.8246	0.7813
PSTNN	0.8816	0.7955	0.7451
AGPCNet	0.9712	0.9646	0.9215
UIUNet	0.9547	0.9477	0.9177
MSHNet	0.9900	0.9899	0.9485
RPCANet	0.9804	0.9879	0.9346
DRPCANet	0.9931	**0.9935**	0.9616
RPCANet++	0.9954	0.9910	0.9556
Ours	**0.9987**	0.9913	**0.9699**

为直观展示各方法的检测性能, 图 8.3 绘制了 NUDT-SIRST 数据集上的检测可视化结果. 其中, “Input” 代表原始输入图像, “GT” 代表参考真值图像, 检测结果中蓝色、黄色和红色标记

分别对应真正例目标、假正例目标和假负例目标. 从可视化结果可以发现, 基于模型的方法由于对图像 a 中目标特征的提取能力有限, 检测结果中存在大量假正例和假负例, 这表明真实场景下的红外小目标检测任务难以通过简单的低秩与稀疏模型实现有效处理. 具备自动特征学习能力的基于数据的方法, 其检测结果中的假正例目标数量显著减少, 这与表 8.1 中的定量实验结果保持一致. 由于 UIUNet 所采用的双层嵌套结构保留了跳跃连接, 导致卷积核在连续平滑区域提取的特征辨识度较低. 这使得网络在反向传播过程中弱目标的梯度被淹没, 从而引发目标漏检问题. 尽管 MSHNet 能够在云背景中检测到目标, 但该方法在多尺度特征拼接阶段未引入噪声先验知识, 且缺乏抑制高频孤立尖峰的有效机制, 导致高对比度噪声在特征融合后依然较为突出. 总之, 对于图 8.3 中的所有测试图像, 基于模型-数据的方法均能成功检测出小目标, 验证了模型与数据方法相结合的有效性.

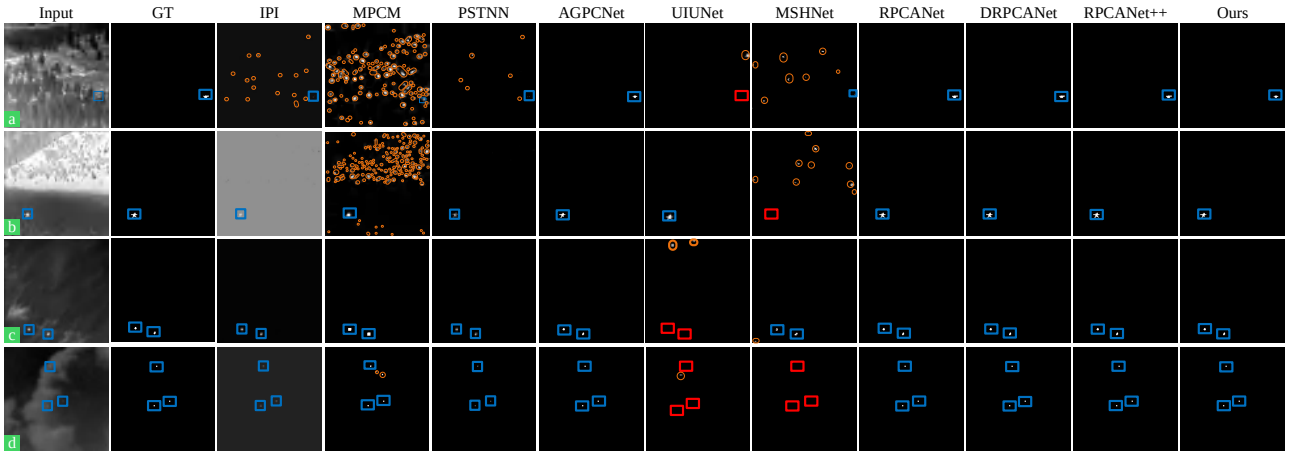


图 8.3: NUDT-SIRST 数据集上的可视化结果

图 8.4 和图 8.5 分别展示了 SIRST-Aug 合成数据集和 IRSTD-1k 真实数据集上的检测结果. 从图 8.5 可以观察到, RPCANet 由于存在稀疏权重过度收缩的问题, 检测结果中出现了部分假正例和假负例. DRPCANet 在生成动态参数时过度依赖输入特征, 导致其在部分复杂场景中存在假正例干扰. 而 RPCANet++ 由于对模型自训练过程的要求过于严苛, 易出现目标漏检现象. 所提 L-RPCANet 通过将输入图像同时投影到三个可学习子空间, 并借助注意力机制动态调整各通道的权重分配, 能够在不同数据集上实现了更稳定的小目标检测.

8.4.3 消融研究

在消融实验中, 设置以下 5 种对比: (I) 不含 SENets 模块, (II) 仅 SEBEM 引入 SENets 模块, (III) SEBEM 与 SETEM 均引入 SENets 模块, (IV) SEBEM、SETEM、SENRM 引入 SENets 模块, (V) SEBEM、SETEM、SENRM、SEIRM 全部引入 SENets 模块.

如表 8.3 所示, SENets 对各模块的检测性能均具有一定提升作用. 具体而言, SETEM 的目标提取模块与 SENRM 的噪声降低模块, 均依赖于对目标与噪声特征的通道权重优化, 这与 SENets 的操作逻辑高度契合, 因此引入 SENets 后性能得到了有效提升. 对于 SEBEM 所处理

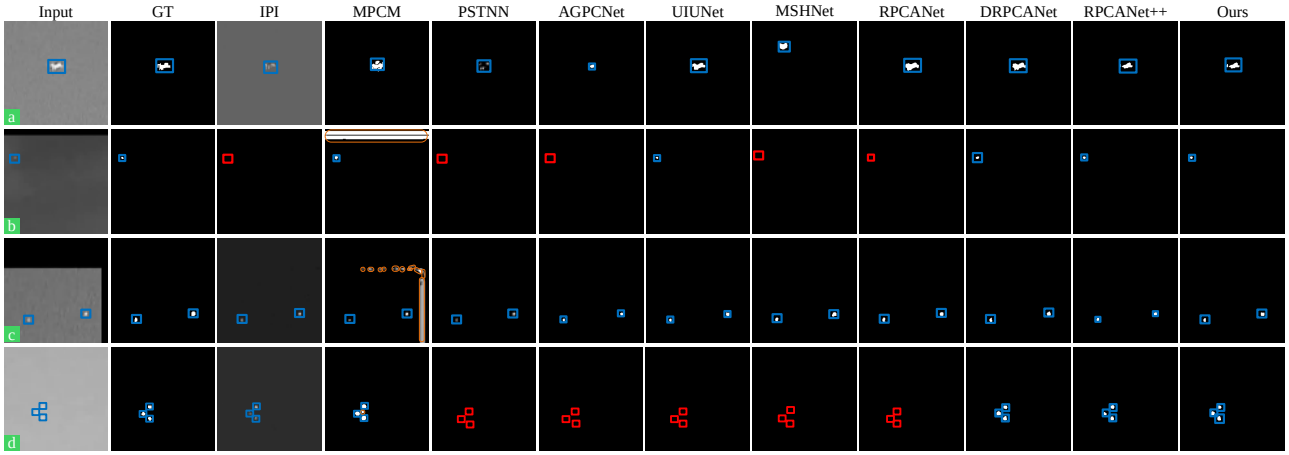


图 8.4: SIRST-Aug 数据集上的可视化结果

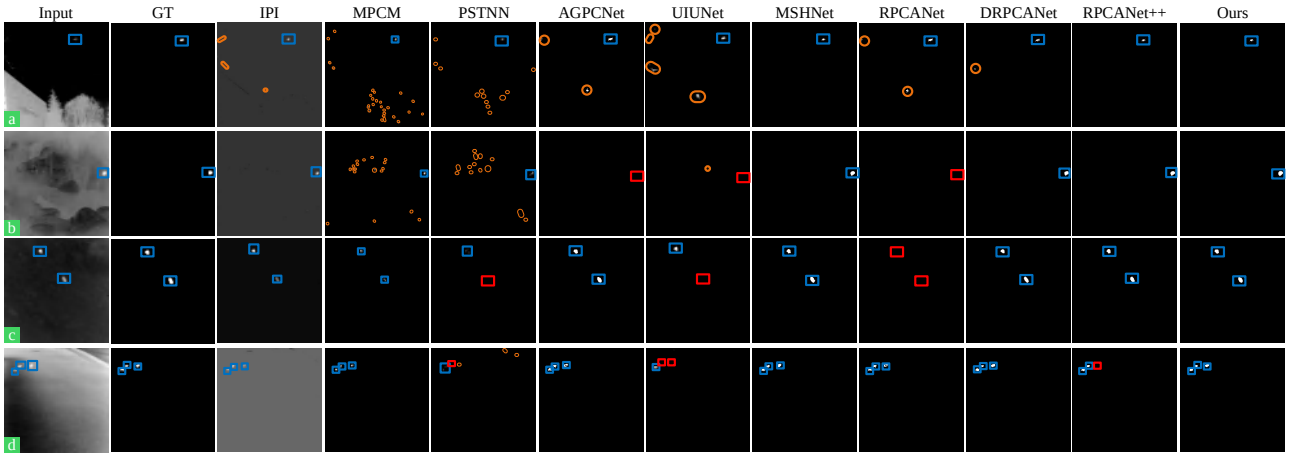


图 8.5: IRSTD-1k 数据集上的可视化结果

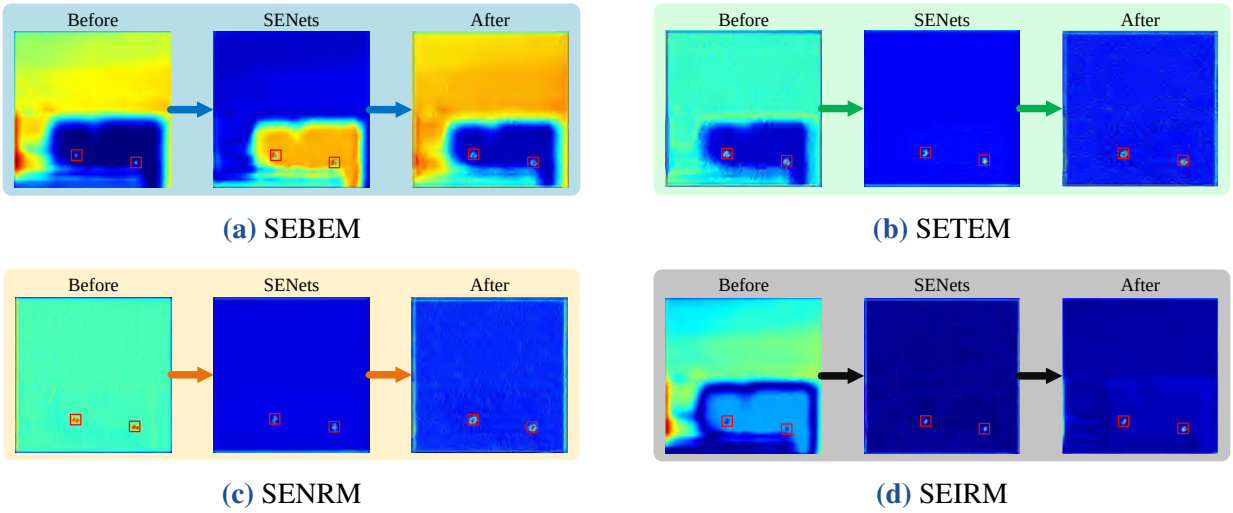
的复杂背景估计任务, 该模块的性能提升依赖于通道权重调整与空间特征优化的双重作用, 故而将 SENets 引入 SEBEM 后, 其检测性能虽获得了一定改善, 但不如 SETEM 和 SENRM 显著. 与之相比, SEIRM 的图像重建过程以空间信息的精确恢复为目标, SENets 的通道权重优化机制与该模块的需求匹配度较低, 因此其在图像重建模块中的性能提升效果相对有限.

表 8.3: SENets 的消融研究

SENets	NUDT-SIRST				SIRST-Aug				IRSTD-1k			
	mIoU ↑	F_1 ↑	P_d ↑	F_a ↓	mIoU ↑	F_1 ↑	P_d ↑	F_a ↓	mIoU ↑	F_1 ↑	P_d ↑	F_a ↓
I	73.56	78.45	79.36	8.56	60.75	70.28	81.24	43.67	50.26	61.34	70.57	15.95
II	80.57	81.39	86.35	5.68	65.96	75.37	86.99	39.82	55.84	65.26	76.36	11.12
III	88.36	89.45	90.18	4.00	70.17	80.78	93.17	34.78	60.56	72.58	83.70	8.10
IV	91.14	96.06	97.18	2.05	73.27	84.45	98.07	**27.73**	63.58	77.53	88.71	**3.95**
V	**92.37**	**96.54**	**98.41**	**1.79**	**74.56**	**85.43**	**99.17**	29.78	**64.68**	**78.55**	**89.39**	4.66

为进一步验证各模块的贡献, 图 8.6 可视化了第一阶段各模块提取的特征. SEBEM 首先对图像背景中目标的形态与结构进行初步估计, SETEM 完成潜在目标的精准提取, SENRM 分离出既不属于背景也不属于目标的综合噪声信息, SEIRM 基于前三个模块输出的特征信息, 重建

仅包含纯净目标信息的图像. 得益于各模块采用的双层通道特征连接策略, 以及噪声降低模块的有效介入, 所提 L-RPCANet 能够以更少的迭代次数, 精准识别背景结构并提取目标特征.



(a) SEBEM **(b)** SETEM

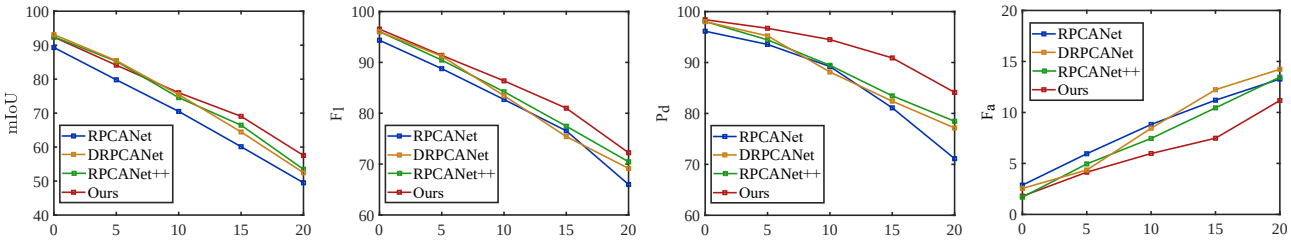
(c) SENRM **(d)** SEIRM

图 8.6: 各模块 SENets 操作的可视化特征图

8.4.4 讨论

(1) 鲁棒性分析

本节在 NUDT-SIRST 数据集施加不同强度的噪声干扰, 以验证所提方法的鲁棒性. 当向数据集加入均值为 0、方差取值为 $\{0, 5, 10, 15, 20\}$ 的高斯噪声时, 由图 8.7 可知, 所有对比方法的检测性能均随噪声强度的增加呈下降趋势, 但所提 L-RPCANet 性能下降幅度显著更小. 如图 8.8 所示, 当施加盐值为 $\{0, 0.02, 0.04, 0.06, 0.08, 0.10\}$、椒值为 0.04 的椒盐噪声时, RPCANet、DRPCANet 及 RPCANet++ 在达到某一噪声强度阈值后便完全失去检测能力, 而所提 L-RPCANet 仍能保持良好的检测精度与稳定性.

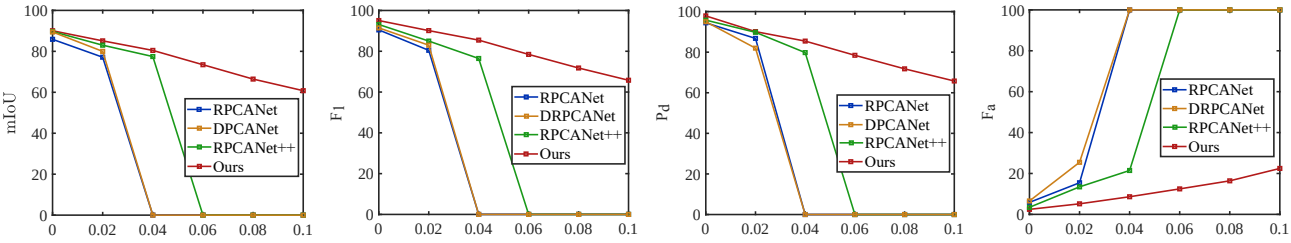


(a) mIoU (b) F_1 (c) P_d (d) F_a

图 8.7: 高斯噪声下的结果

(2) 权重参数分析

本节分析式 (8.17) 中权重参数 η 对检测性能的影响, 结果如表 8.4所示. 当参数 $\eta = 0.01$ 时, 模型在 NUDT-SIRST、SIRST-Aug 及 IRSTD-1k 三个数据集上均实现了最优检测性能. 若 η



(a) mIoU **(b)** F_1 **(c)** P_d **(d)** F_a

图 8.8: 椒盐噪声下的结果

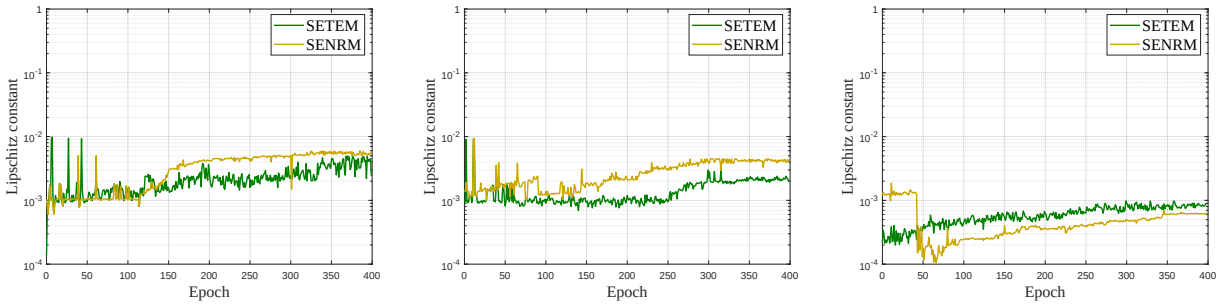
取值过大 (如 $\eta = 0.2$), 会导致模型过度侧重图像重建任务, 若 η 取值过小 (如 $\eta = 0.005$), 则会削弱正则化约束作用. 基于此, 所有实验均将 $\eta = 0.010$ 设定为默认参数.

表 8.4: 损失权重 η 的影响

η	NUDT-SIRST				SIRST-Aug				IRSTD-1k			
	mIoU ↑	F_1 ↑	P_d ↑	F_a ↓	mIoU ↑	F_1 ↑	P_d ↑	F_a ↓	mIoU ↑	F_1 ↑	P_d ↑	F_a ↓
0.005	77.56	82.19	84.25	8.78	65.47	75.58	87.39	35.73	57.45	68.19	78.43	20.54
0.010	**92.37**	**96.54**	**98.41**	**1.79**	**74.56**	**85.43**	**99.17**	**29.78**	**64.68**	**78.55**	**89.39**	**4.66**
0.015	90.36	93.45	94.18	2.90	71.17	82.78	95.17	30.78	61.56	75.58	86.70	6.10
0.200	73.27	78.15	70.35	18.05	60.28	74.45	80.07	40.73	50.36	70.37	80.37	16.78

(3) Lipschitz 条件分析

图 8.9 给出了 SETEM 与 SENRM 模块的 Lipschitz 常数随训练轮次的变化曲线. 由图可知, Lipschitz 常数在训练过程中逐步趋于稳定. 该结果不仅验证了映射 $\mathcal{S}(\boldsymbol{T})$ 与 $\mathcal{G}(\boldsymbol{N})$ 的收敛特性, 也佐证了相关模块收敛性假设的合理性.



(a) NUDT-SIRST **(b)** SIRST-Aug **(c)** IRSTD-1k

图 8.9: 随训练轮次增加的 Lipschitz 条件

8.5 本章小结

本章针对复杂背景下红外小目标检测问题, 提出了融合通道注意力机制的轻量级、高鲁棒性且可解释性强的 L-RPCANet. 该架构由四个网络模块构成, 包括用于背景估计的近端网络、

用于目标提取的稀疏约束网络、用于噪声抑制的降噪神经层, 以及用于图像精准重建的简易卷积神经网络. 对比实验表明, 与现有主流小目标检测方法相比, mIoU 至少提升 3%, F_1 提升 2%. 此外, 所提 L-RPCANet 通过有效融合通道注意力机制与各神经层，实现了对低秩背景、稀疏目标、噪声抑制及图像重建的端到端学习, 削弱了神经网络在检测任务中普遍存在的 “黑盒” 特性, 为稀疏优化在图像检测领域的可解释性研究提供了技术思路.

第 9 章 基于注意力深度支持矩阵机的图像分类

支持矩阵机作为一种新兴的分类框架, 通过直接处理矩阵结构的观测数据, 能够避免向量化过程中破坏数据固有的空间相关性. 然而, 现有大多数支持矩阵机方法高度依赖预定义的正则项, 难以准确捕捉真实世界数据中复杂的非线性结构. 同时, 这类方法在求解时, 往往需要反复执行奇异值分解, 计算开销巨大. 为克服上述局限性, 本章提出了注意力引导深度支持矩阵机 (attention-guided deep support matrix machine, AD-SMM). 该方法引入注意力机制以自适应获得数据的关键结构特征, 将传统支持矩阵机由模型驱动的迭代优化转化为数据驱动的多阶段神经网络, 实现了模型-数据双驱动的有机结合. 实验结果表明, 与现有支持矩阵机方法相比, 所提 AD-SMM 的分类准确率平均提升约 15%. 同时, 与主流深度分类方法相比, 其精度虽略有下降, 但参数量实现了显著缩减, 具有重要的工程意义.

9.1 引言

支持向量机 (support vector machine, SVM) 凭借其严谨的数学基础与完善的统计学习理论, 在众多分类方法中脱颖而出. 它的基本思想是通过最大化不同类别间的分类间隔来寻找最优判别超平面, 输入数据为向量形式. 然而, 在医学影像、人脸图像、脑电信号等实际应用中, 数据往往以矩阵形式存在. 为适配支持向量机的输入要求, 通常将矩阵数据展开为一维向量. 不过, 这种向量化操作不仅会破坏原始数据固有的空间相关性, 还会导致特征维度急剧膨胀, 进而引发所谓的 “维数灾难”.

近年来, 支持矩阵机 (support matrix machine, SMM) 被提出并受到广泛研究[147]. 该方法借助合页损失与核范数直接处理矩阵数据, 在充分利用数据结构的同时确保了分类的有效性. 此后, 为进一步提升其计算效率与泛化能力, 研究人员开发了多种支持矩阵机变体. 例如, Liang 等[148] 将合页损失替换为最小二乘损失, 构建了适用于矩阵结构脑电图分类的最小二乘支持矩阵机. Li 等[149] 引入非平行超平面, 将其应用于红外热图像故障诊断. Li 等[150] 通过自相关函数变换对输入矩阵进行扩展, 从而提高了支持矩阵机的泛化性能. 实际上, 支持矩阵机中采用的合页损失函数是 Heaviside 损失函数的凸近似, 这种近似虽然简化了计算过程, 却也使得模型对噪声更为敏感. 为解决这一局限性, Feng 等[151] 提出了一种采用弹球损失的新方法, 使支持矩阵机能够更好地捕捉底层数据分布. Xiu 等[152] 直接采用 Heaviside 损失函数取代传统的合页损失或斜坡损失, 提出了名为 HL-SMM 的低秩支持矩阵机, 并探讨了最优性条件和收敛性理论. 为进一步提升模型鲁棒性, Zheng 等[153] 将输入矩阵分解为潜在的低秩矩阵与稀疏矩阵, 结合 ℓ_1 范数对稀疏矩阵进行正则处理, 提出了鲁棒支持矩阵机 (robust SMM, RSMM). 基于类似理念, Razzak 等[154] 通过联合 $\ell_{2,1}$ 范数与核范数最小化, 构建了适用于复杂高维噪声数据

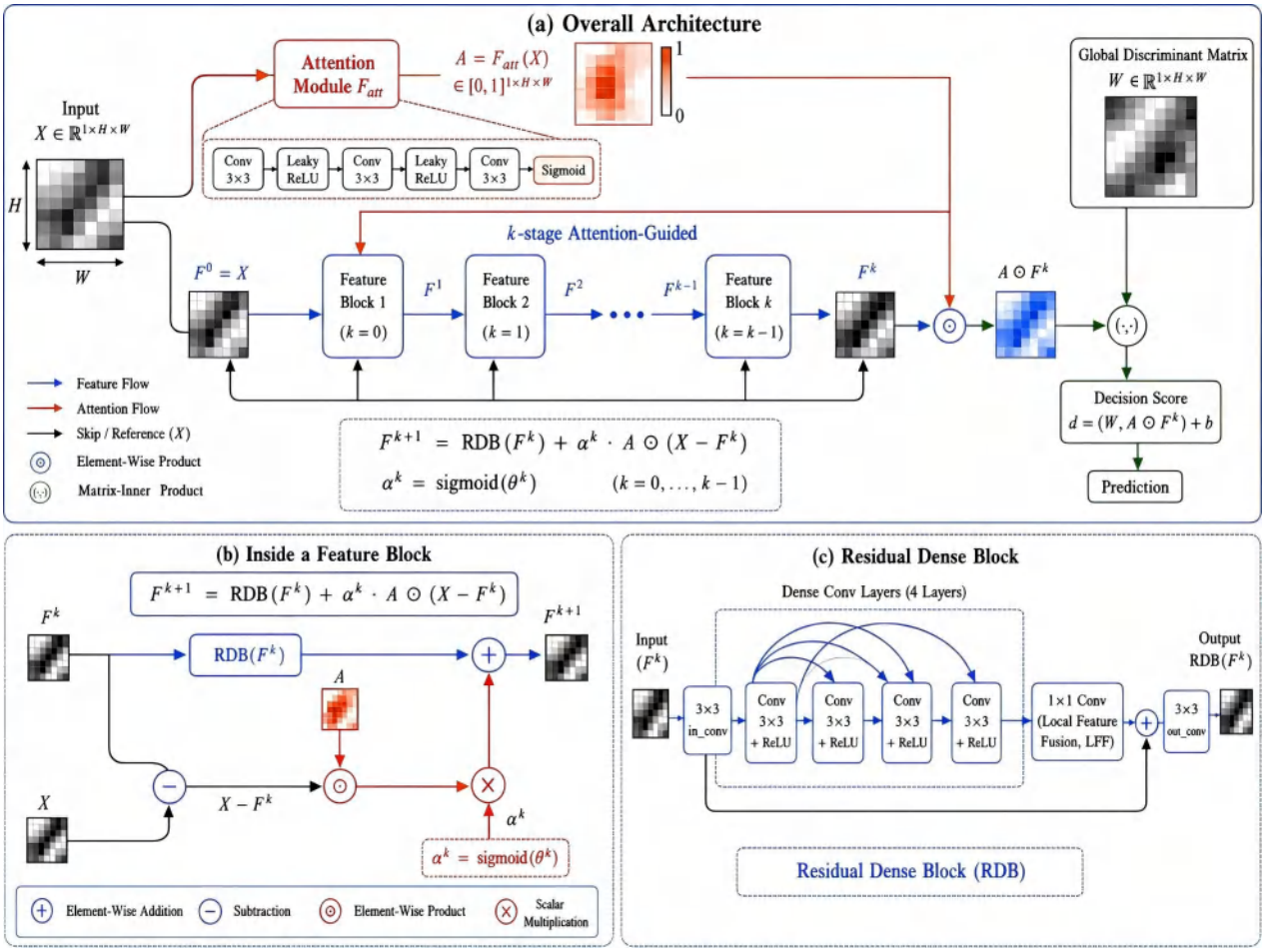


图 9.1: 所提 AD-SMM 网络结构

的 SSMRe. 然而, 这些方法均高度依赖于人工设置的正则函数, 如刻画低秩性的核范数、刻画稀疏性的 ℓ_1 范数, 这在很大程度上限制了模型的泛化能力与自适应性能. 关于支持矩阵机的研究进展, 可参考最新的综述论文[155].

在算法设计方面, 针对支持向量机相关模型的求解, 通常采用交替方向乘子法 (alternating direction method of multipliers, ADMM)、增广拉格朗日法 (augmented Lagrangian method, ALM)、近端交替极小化法 (proximal alternating minimization, PAM) 等迭代类算法. 虽然结构简洁、易于实现, 但收敛速度往往较慢, 难以满足高效求解的需求. 最近, Wu 等[156] 设计了一种基于半光滑牛顿共轭梯度法的增广拉格朗日方法 (semismooth Newton-CG based augmented Lagrangian method, ALMSNCG), 并通过严格的理论推导证明了超线性收敛特性. 然而, 该算法的子问题求解过程中涉及大量的奇异值分解 (singular value decomposition, SVD) 和求逆的计算, 难以满足大规模计算的实际需求. 由此提出一个自然的研究问题, 能否借助神经网络强大的非线性拟合能力与 GPU 的高效并行计算能力, 实现支持向量机问题的高效求解? 经文献调研发现, 目前尚未有相关研究报道.

基于上述分析, 本章提出了一种注意力引导的深度支持矩阵机网络, 其框架如图 9.1 所示. 该方法将支持矩阵机的优化求解过程转化为数据驱动的端到端学习网络, 既保留了支持矩阵

机的可解释性优势, 又有效兼顾了计算优势与泛化性能. 本章的主要贡献为

(1) 首次在支持矩阵机框架中引入空间注意力机制, 通过自适应加权聚焦数据中的判别性区域, 有效抑制了图像背景噪声的干扰.

(2) 采用残差密集块作为可学习的非线性近端算子, 隐式实现特征映射, 避免了计算奇异值分解, 更好地适应复杂的真实数据分类问题.

(3) 构造带有类别平衡权重的二分类交叉熵损失进行端到端训练, 从而设计了轻量化的深度支持矩阵机网络, 并在多个数据集上进行了仿真验证.

9.2 相关工作

9.2.1 典型损失函数

损失函数用于衡量模型预测值与真实标签之间的偏差, 是支持矩阵机目标函数的重要组成部分. 以下介绍几类支持矩阵机研究中常用的典型损失函数.

- Heaviside 损失 (0/1 loss) 定义为

$$\ell_{0/1}(u) = \begin{cases} 0, & u \geq 0, \\ 1, & u < 0, \end{cases} \tag{9.1}$$

其中, $u = yf(x)$, $y \in \{-1, +1\}$ 为样本真实标签, $f(x)$ 为分类决策函数的输出结果.

- 合页损失 (hinge loss) 定义为

$$\ell_{\text{hinge}}(u) = \begin{cases} 0, & u \geq 1, \\ 1-u, & u < 1. \end{cases} \tag{9.2}$$

合页损失是 Heaviside 损失的凸上界近似, 具备凸性和分段线性的特点.

- 弹球损失 (pinball loss) 定义为

$$\ell_{\tau}(u) = \begin{cases} 1-u, & u < 1, \\ \tau(u-1), & u \geq 1, \end{cases} \tag{9.3}$$

其中, $\tau \in [0, 1]$ 为非对称参数. 当 $\tau = 1$ 时, 弹球损失退化为经典的 ℓ_1 损失.

- 斜坡损失 (ramp loss) 定义为

$$\ell_{\text{ramp}}(u) = \max(0, 1-u) - \max(0, s-u), \tag{9.4}$$

其中, $s < 1$ 为截断点. 该损失函数对所有离群点分配恒定最大惩罚, 有效抑制异常值对模型的干扰.

9.2.2 支持矩阵机

给定训练数据 $\{(\boldsymbol{X}_i, y_i)\} \in \mathcal{D}$, 其中 $\boldsymbol{X}_i \in \mathbb{R}^{p\times q}$ 为观测数据, $y_i \in \{-1, 1\}$ 为对应样本的分类标签, $i = 1, \ldots, m$. 支持矩阵机旨在寻找一个最优超平面

$$y_i = \langle \boldsymbol{W}, \boldsymbol{X}_i \rangle + b, \tag{9.5}$$

使得不同标签的样本能够被该超平面有效分离. 基于此, 支持矩阵机的分类问题可表示为

$$\begin{aligned} \min_{\boldsymbol{W},b} \quad & \frac{1}{2}\langle \boldsymbol{W}, \boldsymbol{W} \rangle \\ \text{s.t.} \quad & y_i(\langle \boldsymbol{W}, \boldsymbol{X}_i \rangle + b) \geq 1,\ i = 1, \ldots, m. \end{aligned} \tag{9.6}$$

高维矩阵 $\boldsymbol{W}$ 往往具有低秩结构, 于是上式可进一步改写为

$$\begin{aligned} \min_{\boldsymbol{W},b} \quad & \frac{1}{2}\langle \boldsymbol{W}, \boldsymbol{W} \rangle \\ \text{s.t.} \quad & y_i(\langle \boldsymbol{W}, \boldsymbol{X}_i \rangle + b) \geq 1,\ i = 1, \ldots, m, \\ & \operatorname{rank}(\boldsymbol{W}) \leq r, \end{aligned} \tag{9.7}$$

其中, r 为满足 $r < \min\{p, q\}$ 的正整数. 式 (9.7) 建立在两类数据可被超平面完全分离的理想假设之上, 但在实际应用中, 数据往往存在噪声、重叠等问题. 同时, 秩约束本身属于非凸约束, 这使得该优化问题的求解极具挑战性. 一种更为实用的方法是引入核范数松弛的正则惩罚模型, 其具体形式为

$$\min_{\boldsymbol{W},b} \quad \frac{1}{2}\langle \boldsymbol{W}, \boldsymbol{W} \rangle + \alpha\|\boldsymbol{W}\|_* + \beta \sum_{i=1}^{m} \phi\left[1 - y_i(\langle \boldsymbol{W}, \boldsymbol{X}_i \rangle + b)\right], \tag{9.8}$$

其中, $\alpha, \beta > 0$ 为惩罚参数, $\|\boldsymbol{W}\|_*$ 为矩阵 $\boldsymbol{W}$ 的核范数, ϕ 对应上述各类损失函数.

最近, Xiu 等[152] 将 Heaviside 损失与秩约束相结合, 建立了如下 HL-SMM 模型

$$\begin{aligned} \min_{\boldsymbol{W},b} \quad & \frac{1}{2}\langle \boldsymbol{W}, \boldsymbol{W} \rangle + \beta \sum_{i=1}^{m} \ell_{0/1}\left[1 - y_i(\langle \boldsymbol{W}, \boldsymbol{X}_i \rangle + b)\right] \\ \text{s.t.} \quad & \operatorname{rank}(\boldsymbol{W}) \leq r. \end{aligned} \tag{9.9}$$

该方法既保留式 (9.7) 中的低秩约束, 又引入式 (9.8) 中的 Heaviside 损失, 在图像分类任务中取得了优异的效果.

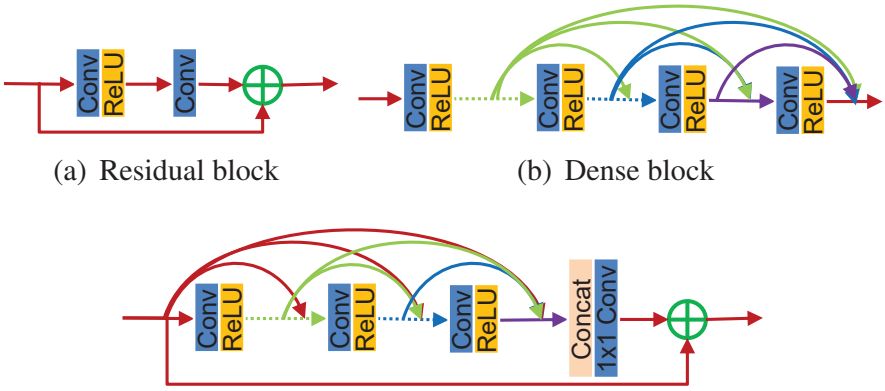
9.2.3 空间注意力机制

注意力机制的核心在于模拟人类视觉系统, 通过对输入数据的不同部分动态分配不同的权重, 从而聚焦于判别性更强的区域. 由空间注意力机制生成的注意力图的权重值直接对应空间位置的贡献度, 高权重区域为任务决策提供关键视觉线索, 低权重区域则被弱化起到噪声

抑制作用. 相较于单一的卷积特征提取, 空间注意力机制可自适应聚焦目标区域, 从而提升特征的表达效率与图像任务的适配. 当前主流的空间注意力模块包含 AHDRNet (attention-guided high dynamic range network)[157]、CBAM (convolutional block attention module)[158]、FFA (feature fusion attention)[159]、SimAM (simple attention module)[160]等.

9.2.4 残差密集块

残差密集块 (residual dense block, RDB)[161]融合了残差学习与密集连接的思想, 在深度卷积网络中实现了高效的空间特征复用与稳定的梯度传播, 如图 9.2 所示. 具体来说, 模块内部采用密集连接机制, 每一层卷积的输入均拼接此前所有层输出的特征图, 通过特征级联实现多层级空间特征的复用, 充分挖掘图像的边缘、纹理等局部空间结构信息. 其次, 通过局部特征融合操作, 采用 1×1 卷积算子对密集连接后的高维特征进行通道整合与降维, 在保留有效特征的同时控制模型参数量与计算复杂度. 最后, 引入局部残差短路连接, 将模块原始输入特征与融合后的输出特征直接相加, 使网络专注于学习残差形式的细节特征, 有效缓解深层卷积网络训练过程中易出现的梯度消失问题.



(a) Residual block (b) Dense block

(c) Residual dense block

图 9.2: 残差密集块[161]

9.3 模型与算法

9.3.1 数学模型

尽管传统支持矩阵机展现出了优异的分类性能, 但仍存在进一步提升的空间. 一方面, 直接使用原始图像 $\boldsymbol{X}$ 进行分类判别, 分类器会无差别地对待目标特征与背景噪声, 从而导致判别结果极易受到干扰. 另一方面, 传统的先验信息难以精确捕捉图像的潜在特征, 且求解过程中存在大量的矩阵运算, 无法满足大规模数据的训练需求.

为此, 本章通过深度神经网络参数化的迭代特征表示 $\boldsymbol{F}$, 并使用空间注意力图 $\boldsymbol{A} = \mathcal{F}_{att}(\boldsymbol{X})$ 对判别区域进行自适应引导, 相应的 AD-SMM 模型为

$$\min_{\boldsymbol{W},b,\Theta} \quad \mathcal{J} = \underbrace{\sum_{i=1}^{N} L_{cls}\left(y_i, \langle \boldsymbol{W}, \boldsymbol{A}_i \odot \boldsymbol{F}_i \rangle + b\right)}_{\text{经验风险}} + \underbrace{\frac{\lambda}{2}\|\boldsymbol{W}\|_F^2}_{\text{结构风险}}, \tag{9.10}$$

其中, $\boldsymbol{A}_i = \mathcal{F}_{att}(\boldsymbol{X}_i) \in [0,1]$ 为针对样本 $\boldsymbol{X}_i$ 生成的空间注意力掩码, 用于内积运算前对特征矩阵进行空间维度的显著性加权. $\boldsymbol{F}_i$ 为蕴含高阶语义信息的特征矩阵, 其内部集成的残差密集结构能够学习数据隐含的结构先验, 从而弥补了人工设计正则项的表达不足. L_{cls} 为带有类别平衡权重的二分类交叉熵损失, 用于替代支持矩阵机模型的合页损失函数, 以保证深层网络反向传播的数值稳定性[162]. 此外, $\odot$ 表示哈达玛积 (Hadamard product).

与现有支持矩阵机不同, 全局判别矩阵 $\boldsymbol{W}$、偏置参数 b, 以及所有特征提取网络的参数 Θ 均为可学习变量.

9.3.2 算法设计

网络设计的要点在于如何获取高质量的特征矩阵 $\boldsymbol{F}$. 假设对于给定的输入 $\boldsymbol{X}$ 和注意力掩码 $\boldsymbol{A}$, 理想的特征矩阵 $\boldsymbol{F}$ 应当既保留原始数据在关键区域的信息, 又具备图像先验的判别特征. 其优化目标函数可表示为

$$\min_{\boldsymbol{F}} \quad \frac{1}{2}\|\boldsymbol{A} \odot (\boldsymbol{X} - \boldsymbol{F})\|_F^2 + \mathcal{R}(\boldsymbol{F}), \tag{9.11}$$

其中, 第一项为注意力加权的数据保真项, 通过哈达玛积确保特征更新主要集中在注意力高响应区域, $\mathcal{R}(\boldsymbol{F})$ 为刻画图像的隐式正则项.

本节采用近端梯度下降法 (proximal gradient descent, PGD) 对上述优化目标进行求解. 首先对数据保真项求梯度, 在迭代步中将注意力图 $\boldsymbol{A}$ 视作常量, 可得第一项的梯度为

$$\nabla_{\boldsymbol{F}}\left(\frac{1}{2}\|\boldsymbol{A} \odot (\boldsymbol{X} - \boldsymbol{F})\|_F^2\right) = -\boldsymbol{A} \odot \boldsymbol{A} \odot (\boldsymbol{X} - \boldsymbol{F}). \tag{9.12}$$

由于注意力图 $\boldsymbol{A}$ 的元素值介于 0 到 1 之间, 为简化网络结构并加速收敛, 将 $\boldsymbol{A} \odot \boldsymbol{A}$ 的非线性缩放效应吸收到后续的可学习步长中. 由此, 第 k 步的梯度下降中间状态 $\boldsymbol{Z}^k$ 可表示为

$$\boldsymbol{Z}^k = \boldsymbol{F}^k + \alpha_k \boldsymbol{A} \odot (\boldsymbol{X} - \boldsymbol{F}^k). \tag{9.13}$$

随后, 对正则项 $\mathcal{R}(\boldsymbol{F})$ 执行近端映射, 得到第 $k+1$ 步的特征矩阵更新公式

$$\boldsymbol{F}^{k+1} = \operatorname{prox}_{\lambda\mathcal{R}}(\boldsymbol{Z}^k). \tag{9.14}$$

传统近端算子不仅计算成本高昂, 且难以适配复杂数据的分布特性, 此处采用残差密集块作为

算法 2 注意力深度支持矩阵机 (AD-SMM) 训练算法

输入: 训练集 $\mathcal{D}=\{(\boldsymbol{X}_i,y_i)\}_{i=1}^{N}$, 参数 k, η, λ, $Epochs$

输出: 最优参数集 $\Omega^*=\{\boldsymbol{W}^*,b^*,\Theta_{att}^*,\Theta_{rdb}^*,\alpha^*\}$

初始化: Ω, $\boldsymbol{W}\sim\mathcal{N}(0,0.01)$, $b=0$, $\Theta_{att},\Theta_{rdb}$, $\alpha^k=0.5$ $(k=0,\ldots,k-1)$

for $epoch=1$ **to** $Epochs$ **do**

for 每个微批次 $(\boldsymbol{X},y)\in\mathcal{D}$ **do**
  // 空间注意力引导
  计算 $\boldsymbol{A}=\mathcal{F}_{att}(\boldsymbol{X},\Theta_{att})$
  // 近端梯度迭代
  初始化 $\boldsymbol{F}^0=\boldsymbol{X}$
  for $k=0$ **to** $k-1$ **do**
    更新 $\boldsymbol{F}^{k+1}=\mathrm{RDB}(\boldsymbol{F}^k)+\sigma(\alpha^k)\boldsymbol{A}\odot(\boldsymbol{X}-\boldsymbol{F}^k)$
  end for
  // 全局超平面判别
  计算 $\hat{y}=\langle\boldsymbol{W},\boldsymbol{A}\odot\boldsymbol{F}^k\rangle+b$
  // Adam 优化与参数更新
  计算 $L_{cls}(\hat{y},y)$ 与 $\nabla_{\Omega}L_{cls}$
  更新 Ω
end for
验证集评估
end for

可学习的非线性近端算子. 将上述数学迭代过程映射为神经网络的前向传播层, 具体公式为

$$\boldsymbol{F}^{k+1}=\mathrm{RDB}_{\Theta_k}(\boldsymbol{F}^k)+\sigma(\alpha_k)\cdot\boldsymbol{A}\odot(\boldsymbol{X}-\boldsymbol{F}^k). \tag{9.15}$$

这里, $\mathrm{RDB}_{\Theta_k}(\cdot)$ 替代了传统的奇异值收缩算子, 借助深度卷积网络隐式完成特征提取. $\sigma(\alpha_k)$ 则将传统算法中的固定步长转化为各层独立的可学习参数, 并通过 Sigmoid 函数将其约束在 $(0,1)$ 区间内, 有效保障了网络的数值稳定性. 整个 AD-SMM 的框架如算法 2 所示.

9.4 数值实验

为验证所提 AD-SMM 的性能, 首先选取五种支持矩阵机方法进行对比, 包括 SMM[147]、RSMM[153]、SSMRe[154]、ALMSNCG[156]、HL-SMM[152]. 然后, 引入两种神经网络分类方法进一步分析, 即 FasterNet[163] 与 MobileNetV4-S[164]. 本实验基于 RTX 4090D (24GB) 显卡搭建硬件环境, 采用 CUDA 12.8 与稳定版 PyTorch 作为软件支撑, 所有对比方法均统一使用原文开源代码的默认参数配置. 此外, 所提方法开源代码见链接 https://github.com/xianchaoxiu/AD-SMM.

9.4.1 实验设置

(1) 数据集

本研究选取 6 个具有代表性的公开数据集, 包括通用物体识别数据集 CIFAR10[1]、脑肿瘤磁共振成像数据集 Brain[2]、乳腺超声图像数据集 BUSI[3]、胃肠道内窥镜数据集 Kvasir[4]、混凝土裂缝检测数据集 Concrete[5]以及疟疾细胞图像数据集 Malaria[6]. 所有数据集均按照 70% 训练集、15% 验证集、15% 测试集的比例划分, 各数据集的样本数量统计如表 9.1 所示.

表 9.1: 各数据集样本的数量统计

数据集	总数量	训练集	验证集	测试集
CIFAR10	60, 000	42, 000	9, 000	9, 000
Brain	7, 200	5, 040	1, 080	1, 080
BUSI	647	452	97	98
Kvasir	8, 000	5, 600	1, 200	1, 200
Concrete	40, 000	28, 000	6, 000	6, 000
Malaria	27, 558	19, 290	4, 134	4, 134

为保证模型输入的一致性, 对所有原始数据进行标准化预处理. 首先, 将所有输入图像统一缩放至 64×64 或 32×32 像素. 其次, 将所有 RGB 图像转换为灰度图, 同时将图像像素值从 $[0, 255]$ 区间线性映射至 $[0, 1]$ 区间, 并利用各数据集自身的均值与标准差完成标准化处理. 最后, 将预处理后的图像矩阵转换为浮点型张量, 训练过程中将图像标签映射为 $\{0, 1\}$, 适配二分类交叉熵损失函数的计算需求.

(2) 参数设置

在网络参数方面, AD-SMM 的特征提取部分由 k 个串行的残差密集块组成. 为平衡模型的表征能力与计算效率, 对不同数据集采用差异化的迭代层数设置, CIFAR10 数据集设置 $k = 8$, Kvasir 数据集设置 $k = 4$, 其余数据集统一设置 $k = 6$. 每个残差密集块内部包含 4 个卷积层, 其初始增长率设置为 32, 后续各层增长率设为 16, 以逐步提升特征表达能力. 在优化器与正则策略方面, 权重衰减系数统一设为 $\lambda = 1 \times 10^{-4}$, 全局梯度范数裁剪阈值为 5.0. 初始学习率根据各数据集的特性进行微调, 采用余弦退火策略动态调整学习率, 确保训练过程的稳定性, 最低学习率固定为 1×10^{-6}. 同时引入早停机制, 若验证集准确率在连续 15 个轮数内未提升, 则自动终止训练. 部分参数如表 9.2 所示.

[1]https://tensorflow.google.cn/datasets/catalog/cifar10

[2]https://docs.ultralytics.com/zh/datasets/detect/brain-tumor/

[3]https://www.kaggle.com/datasets/sabahesaraki/breast-ultrasound-images-dataset

[4]https://datasets.simula.no/kvasir/

[5]https://archive.ics.uci.edu/dataset/165/concrete+compressive+strength

[6]https://data.unicef.org/resources/dataset/malaria/

表 9.2: 各数据集的训练超参数

数据集	残差密集块数	初始学习率	批次大小	最大轮数
CIFAR10	8	5×10^{-4}	16	50
Brain	6	1×10^{-3}	32	50
BUSI	6	5×10^{-4}	16	80
Kvasir	4	1×10^{-3}	32	30
Concrete	6	1×10^{-3}	64	30
Malaria	6	1×10^{-3}	64	30

(3) 评估指标

针对图像二分类任务, 选取准确率 (accuracy, ACC) 作为评估指标, 定义为

$$\mathrm{ACC}=\frac{\mathrm{TP}+\mathrm{TN}}{\mathrm{TP}+\mathrm{TN}+\mathrm{FP}+\mathrm{FN}}, \tag{9.16}$$

其中, TP (true positive) 表示真正例 (true positive), TN (true negative) 表示真负例, FP (false positive) 表示假正例, FN (false negative) 表示假负例.

9.4.2 与模型方法比较

表 9.3 列出了各方法在多数据集上的分类准确率, 最优结果以加粗标注. 由实验结果可见, 所提 AD-SMM 在全部测试数据集上均取得最优分类性能, 平均准确率高达 93.85%. 相较于次优的 SSMRe, 平均准确率提升 12.70 个百分点. 相较于基准 SMM, 提升幅度达 19.01 个百分点. 在 CIFAR10 数据集上, AD-SMM 分类准确率可达 93.63%, 而其余对比方法准确率均未突破 75%. 在 Concrete 数据集上, AD-SMM 实现近乎完美的分类效果, 性能显著领先其他方法. 对于小规模数据集 BUSI, 虽然 SSMRe 与 HL-SMM 取得相对较好的结果, 但与所提 AD-SMM 仍存在差距. 推测其原因在于数据集样本量有限, 网络易出现过拟合现象.

表 9.3: 各方法在不同数据集上的准确率对比 (%)

数据集	SMM	RSMM	SSMRe	ALMSNCG	HL-SMM	AD-SMM
CIFAR10	68.20	63.87	73.73	63.80	64.80	**93.63**
Brain	84.07	81.20	92.22	94.79	92.67	**98.06**
BUSI	76.53	72.45	85.71	81.40	85.71	**88.78**
Kvasir	74.20	73.67	79.33	76.24	78.00	**86.67**
Concrete	74.57	74.82	88.55	90.89	75.31	**99.88**
Malaria	64.21	63.58	67.37	66.83	76.83	**96.08**
Average	74.84	72.85	81.15	78.99	78.89	**93.85**

图 9.3 展示了各方法的平均运行时间对比. 可以看出, 相较于基于一阶优化算法的方法, ALMSNCG 引入牛顿迭代后计算效率明显改善. 而所提 AD-SMM 依托神经网络结构并结合

GPU 并行加速, 进一步大幅降低了运算耗时. 以 CIFAR10 数据集为例, ALMSNCG 耗时为 51.37 s, 而 AD-SMM 仅需 0.58 s. 综上, 相较于基于数学模型的支持矩阵机, 所提 AD-SMM 在分类精度与效率两方面均实现了显著提升.

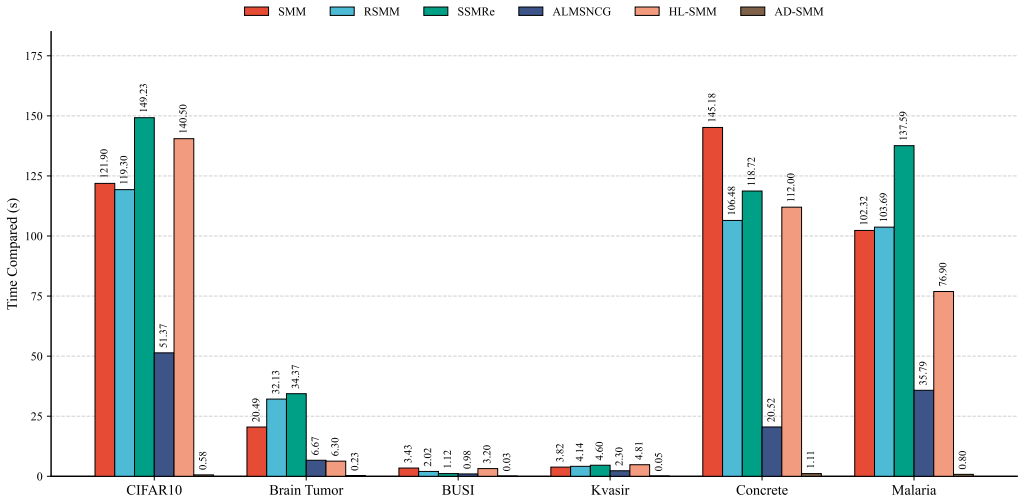


图 9.3: 各方法平均运行时间比较

9.4.3 与深度方法比较

本节选取表 9.1 中样本量超过 20，000 的数据集 CIFAR10、Concrete 以及 Malaria, 并与两类主流轻量化深度学习分类器 FasterNet[163]、MobileNetV4-S[164] 进行比较, 结果如表 9.4 所示. 可以看出, AD-SMM 表现出极具竞争力的分类精度, 在部分任务场景下性能甚至优于参数量更大的深度模型. 具体来说, 在 Concrete 与 Malaria 数据集上, AD-SMM 均取得最优分类准确率. 这表明 AD-SMM 能够有效提取特定领域的关键特征, 尤其在医疗影像这种对局部纹理敏感的场景中具有更强的鲁棒性.

表 9.4: 不同网络的准确率对比 (%)

数据集	FasterNet	MobileNetV4-S	AD-SMM
CIFAR10	**97.96**	97.04	93.63
Concrete	99.82	99.85	**99.88**
Malaria	95.89	95.89	**96.08**
Average	**97.89**	97.59	96.53

图 9.4 进一步刻画了模型分类性能与资源开销的权衡关系. 在模型规模方面, AD-SMM 具备压倒性轻量化优势, 参数量仅为 0.22M, 分别仅为 MobileNetV4-S 的 5.8%、FasterNet 的 2.9%. 极低的模型复杂度, 可大幅降低硬件存储开销. 在训练效率层面, AD-SMM 每轮迭代耗时仅约 11.3s, 显著低于 FasterNet 与 MobileNetV4-S. 这不仅加速了模型的迭代过程, 也预示着在实际部署时具备更高的吞吐量和更低的延迟.

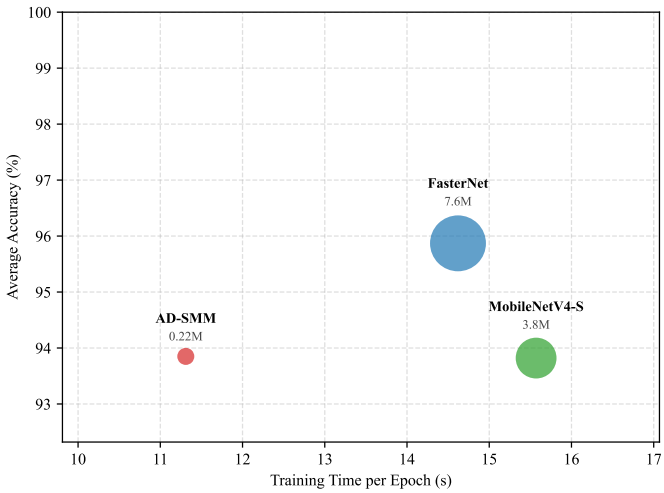


图 9.4: 模型参数量与平均精度对比

9.4.4 讨论

(1) 消融实验

为探究不同空间注意力模块对 AD-SMM 分类性能的影响, 本节选取 AHDRNet、CBAM、FFA 以及 SimAM 作为候选特征提取组件, 并将 AD-SMM 拓展到支持向量机变体 AD-SVM. 所有对比模型的结构仅前端接入的注意力模块存在差异, 其余结构与超参数均保持完全一致. 由表 9.5 可知, 尽管引入 CBAM 和 SimAM 模块在部分特定数据集上能带来一定的性能提升, 但 AHDRNet 模块在 4 个数据集上均取得最优分类成绩, 且在所有数据集上的表现保持良好稳健性. 实验结果同时表明, 注意力模块的性能并非随结构复杂度提升而增强, 印证了 AHDRNet 结构与当前分类框架的高度适配性. 此外, 表 9.6 列出了不同模块的参数量和训练时间, 进一步展示了 AHDRNet 的优势.

表 9.5: 消融实验结果准确率对比 (%)

数据集	AD-SVM	+AHDRNet	+CBAM	+FFA	+SimAM
CIFAR10	92.95	93.63	**93.80**	93.52	93.79
Brain	97.69	**98.06**	98.06	97.31	96.85
BUSI	85.71	**88.78**	85.71	82.65	80.61
Kvasir	85.33	86.67	**87.67**	87.33	87.00
Concrete	99.85	**99.88**	99.87	99.78	99.73
Malaria	95.86	**96.08**	95.04	95.24	95.24
Average	92.90	**93.85**	93.36	92.64	92.22

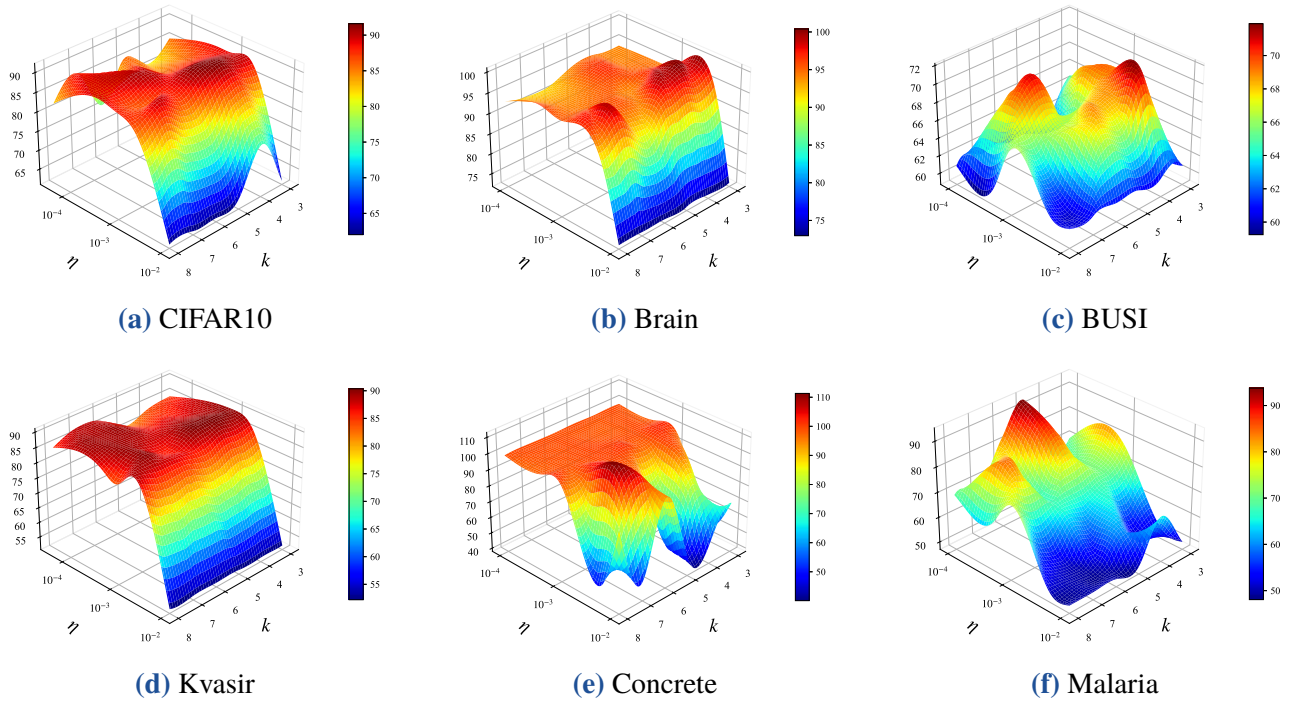
(2) 参数分析

本节开展参数敏感性分析, 重点探讨初始学习率 $\eta \in \{10^{-4}, 5\times10^{-4}, 10^{-3}, 5\times10^{-3}, 10^{-2}\}$ 与残差密集块数 $k \in \{3,4,5,6,7,8\}$ 对模型分类精度的影响. 图 9.5 记录了各参数组合下的最高

表 9.6: 不同模块参数量及训练时间对比

模型	参数量	训练时间
+AHDRNet	220,158	11.31
+CBAM	220,399	12.63
+FFA	220,454	12.27
+SimAM	220,301	11.36

分类准确率. 其中, 曲面平缓区域内起伏较小, 表明该范围内参数鲁棒性较强. 反之, 曲面起伏剧烈的区域则说明模型性能受参数变化的影响较大.



(a) CIFAR10 **(b)** Brain **(c)** BUSI

(d) Kvasir **(e)** Concrete **(f)** Malaria

图 9.5: 不同数据集敏感性曲线对比

通过分析可知, 学习率 η 是决定模型收敛状态的最敏感参数. 当设定过大的学习率时, 所有数据集上的模型分类性能均出现灾难性的断崖式下降. 例如, 在 Concrete 和 Malaria 数据集上, 高学习率导致准确率直接退化至 50% 左右. 这可能是因为过激的参数更新在跨越多层级联特征块时易引发梯度震荡, 进而破坏残差密集块对隐式先验的平滑拟合效果. 相比于学习率的剧烈响应, 深度展开层数 k 对网络性能的影响则呈现出典型的边际效益递减规律. 当展开层数较浅 (如 $k = 3$) 时, 由于优化迭代步数不足, 模型未能充分映射原始数据的复杂二阶结构, 导致分类指标普遍垫底. 随着层数逐步提升至 $k \in [4, 6]$ 的区间, 各项性能指标迅速饱和并达到全局最优, 例如 Brain 数据集在 $k = 4$、$\eta = 10^{-3}$ 时取得 98.06% 附近的高精度, 而 Concrete 数据集则在 $k = 5$ 左右逼近 99.66%. 然而, 继续盲目增加层级, 不仅无法带来显著的精度提升, 反而会导致模型计算负担大幅加重.

(3) 收敛性分析

图 9.6 展示了模型在 CIFAR10 数据集上的训练曲线, 其中蓝色曲线为训练指标, 橙色曲线为验证指标. 整体来看, 模型验证集曲线与训练集曲线走势高度贴合, 泛化间隙极小. 这表明 AD-SMM 虽结构极度轻量化, 仍具备良好的归纳偏置能力. 此外, 模型训练集与验证集准确率均在 30 至 50 轮次内完成收敛, 表现出了良好的快速收敛性.

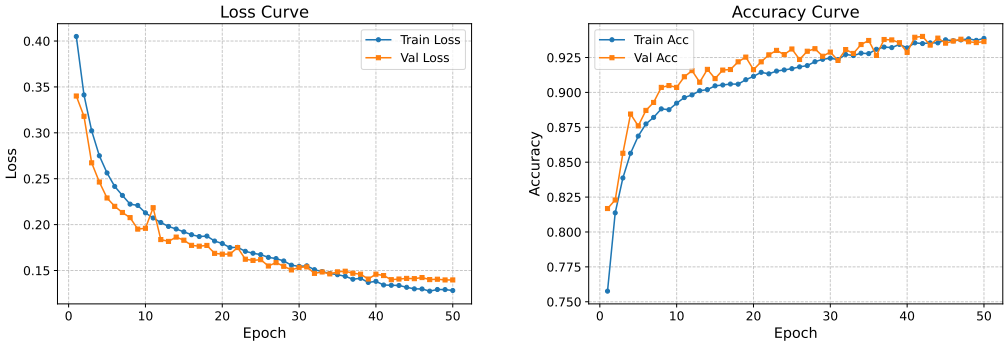


图 9.6: CIFAR10 数据集上的训练曲线

9.5 本章小结

本章针对传统支持矩阵机依赖先验正则项、需人工调参的问题, 提出了注意力引导的深度支持矩阵机 (AD-SMM). 该方法将传统优化算法进行网络化重构, 并通过融合空间注意力机制与残差密集块, 实现了对图像先验结构的精准捕获与背景噪声的有效抑制, 且所有参数均通过端到端方式自主学习. 实验结果表明, 所提 AD-SMM 在六个基准数据集上取得了 93.85% 的平均准确率, 显著优于传统支持矩阵机及其变体. 同时, AD-SMM 具有极高的计算效率, 其参数量仅为 0.22M, 表明了其在计算资源受限场景下进行高效图像识别的实用价值.

第 10 章 基于自适应稀疏的大语言模型剪枝

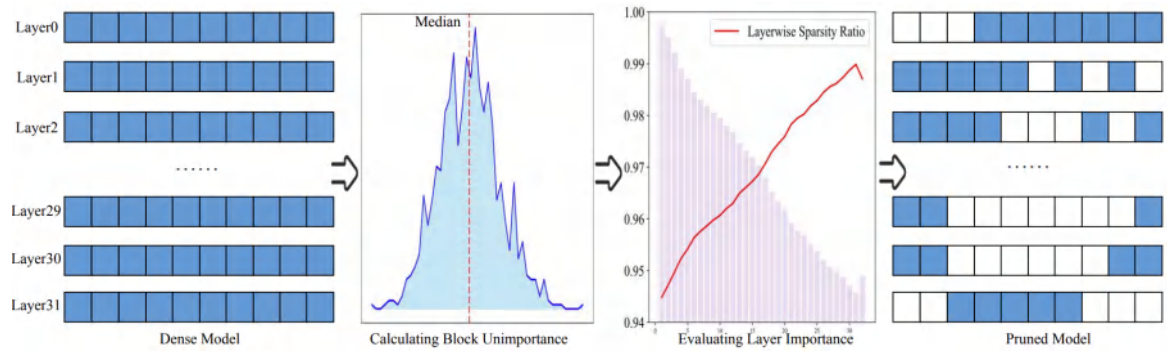
大语言模型在计算机、机器人及医学等领域应用广泛, 但庞大的参数量使其部署面临巨大挑战. 然而, 现有多数基于层间动态分配的结构化剪枝方法, 仅依靠权重幅值、激活特征等静态统计量评估神经元重要性, 难以精准刻画不同网络层对任务目标的动态贡献与敏感度差异. 为此, 本章提出梯度引导的自适应结构化剪枝 (gradient-guided adaptive pruning, GAP), 用于提升大语言模型的压缩效果与推理效率. 该方法在自适应稀疏分配框架中引入梯度敏感度分析与对数映射校准机制, 从而兼顾高精度与轻量化. 针对所构建的非线性组合优化模型, 基于边际收益分析设计了贪心求解算法, 并对其计算复杂度进行分析. 实验结果表明, 所提 GAP 在模型结构化压缩中具有明显优势, 同时展现出其在边缘计算平台的部署潜力.

10.1 引言

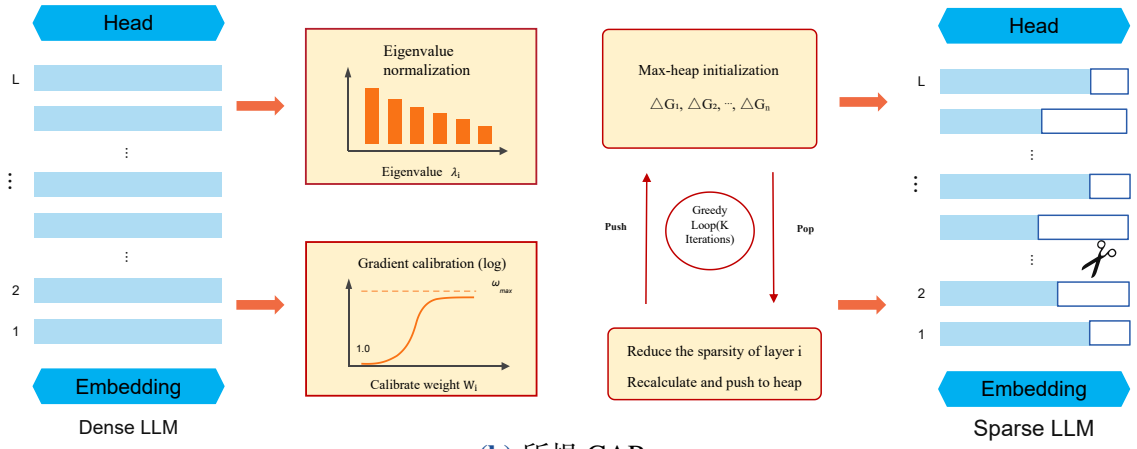
近年来, 以 Transformer[144] 为核心的大语言模型 (large language models, LLMs) 在自然语言处理领域实现了跨越式发展, 从 GPT-3[165] 的 1,750 亿参数到 Llama3[166] 的 4,050 亿参数, 模型规模呈现指数级增长. 同时, 这种规模扩张也带来了严峻的资源依赖与部署困境. 以 Llama3 为例, 其单次推理需处理数十亿次矩阵运算, 权重文件约 800GB, 远超普通服务器的存储与算力容量. 因此, 如何在最大限度保留模型性能的前提下实现高效压缩与推理加速, 已成为大语言模型领域的关键挑战. 根据综述文献[167-168], 现有模型压缩技术可分为四类: 剪枝、量化、知识蒸馏及低秩分解.

剪枝凭借直接移除冗余参数的能力, 成为大语言模型压缩领域的研究热点[169]. 与量化、知识蒸馏等间接方法不同, 剪枝通过物理性缩减模型参数量, 同时优化网络计算图结构. 依据权重移除粒度, 剪枝可分为非结构化剪枝与结构化剪枝. 早在神经网络剪枝的初期研究中, 提出了著名的最优脑损伤 (optimal brain damage, OBD)[170] 和最优脑外科 (optimal brain surgeon, OBS)[171] 策略. Frantar 等[172] 在 OBD 基础上引入近似处理技术, 设计了稀疏化策略 SparseGPT. 虽然会牺牲部分重构精度, 但该方法实现了在数小时内对超大规模模型的非结构化剪枝. 由于 SparseGPT 的权重更新过程计算复杂, Sun 等[173] 通过结合权重幅值与输入激活范数的乘积作为重要性度量, 提出了 Wanda (pruning by weights and activations) 策略. 该技术能够在完全不调整剩余权重的前提下, 对大语言模型实现高稀疏度剪枝. 最近, Bovza**bovza2024fast** 将剪枝转化为带约束的权重优化问题, 并通过交替方向乘子法 (alternating direction method of multipliers, ADMM) 实现了大语言模型的快速剪枝, 因此该方法也被称为 ADMM.

与非结构化剪枝相对, 结构化剪枝以通道、神经元、注意力头等完整的结构单元为对象, 剪枝后模型权重仍保持密集矩阵形式, 能够充分利用 NVIDIA Tensor Core 等专用计算单元, 因



(a) 现有 DLP



(b) 所提 GAP

图 10.1: 与现有 DLP 框架对比

此在大语言模型部署中更具工程价值. 从方法流程看, 结构化剪枝可分为层内裁剪和层间分配. 具体来说, 层内裁剪解决的是具体剪哪些的问题. 在给定每层预算后, 需要在该层内部识别冗余结构并执行物理删除. LLM-Pruner[174] 是首个面向大语言模型的结构化剪枝框架, 具有任务无关、数据需求少、速度快等优势. Ashkboos 等[175] 通过在每层输入输出之间插入正交变换矩阵, 将剪枝转化为主成分分析 (principal component analysis, PCA) 中的维度选择问题, 实现了均匀层内分配的结构化剪枝, 该方法称为 SliceGPT. 层间分配解决的是每层剪多少的问题. 由于不同层承担的表示功能并不一致, 统一的稀疏度难以兼顾压缩与性能. 最近, Chen 等[176] 通过计算层非重要性的中位数并转化为相对重要性, 实现动态层剪枝 (dynamic layerwise pruning, DLP). 此外, DLP 可以与多种层内剪枝方法结合, 从而对稀疏度进行自适应分配.

虽然大语言模型的结构化剪枝方法在层内裁剪方面已取得显著进展, 但在层间分配方面仍有较大的提升空间. 现有方法通常基于权重幅值、激活统计等静态特征进行层重要性评估, 缺乏与任务目标对齐的动态度量, 难以有效识别对最终性能贡献较大但统计特征不突出的关键层. 此外, 大语言模型层间梯度范数往往存在一定的差异, 若直接以原始梯度作为重要性依据, 极易导致部分高敏感层过度占用维度资源, 而其他层则被过度剪枝. 如何引入梯度信息并设计有效的校准机制以实现更精准的层间分配, 是提升大语言模型结构化剪枝性能的关键. 为此, 本章提出了梯度引导的自适应剪枝 (gradient-guided adaptive pruning, GAP), 其框架及与现

有 DLP 的对比如图 10.1 所示. 本章的主要贡献为

(1) 通过分析梯度敏感度精准评估各层对任务的贡献, 利用对数映射有效平滑跨数量级的梯度异常分布, 设计了梯度引导的结构化剪枝策略.

(2) 设计了一种基于最优边际收益且严格面向硬件对齐的贪心分配算法, 其中稀疏度采用从大到小、逐步回收的方式, 并且与硬件步长对齐.

(3) 在 Llama 和 Qwen 不同参数的大语言模型上验证了所提 GAP 的有效性, 并在 Jetson Orin NX 嵌入式平台完成了部署.

10.2 相关工作

10.2.1 Transformer

根据下游任务需求, Transformer 架构可分为 Encoder-only、Decoder-only 及 Encoder-Decoder. 其中, Encoder-only 架构适用于文本分类等判别式任务, Decoder-only 架构适用于文本生成等生成式任务, Encoder-Decoder 架构适用于机器翻译等序列到序列任务. 目前, Decoder-only 是主流大语言模型 (如 GPT 系列、LLaMA 系列) 的首选架构. 此外, Transformer 的每层均包含多头自注意力机制 (multi-head attention, MHA) 和逐位置前馈网络 (feed forward network, FFN). 为保障深层网络的稳定训练, 还引入了残差连接和层归一化 (layer normalization, LayerNorm).

设 $\boldsymbol{X}$ 为数据矩阵, $\boldsymbol{Z}$ 为降维后的特征表示, 为统一注意力模块与逐位置前馈神经网络的数学表达, 可将二者统一为

$$\boldsymbol{Z} = \phi(\boldsymbol{X}\boldsymbol{W}_{\text{in}} + \mathbf{1}\boldsymbol{b}_{\text{in}}^{T})\boldsymbol{W}_{\text{out}} + \mathbf{1}\boldsymbol{b}_{\text{out}}^{T}, \tag{10.1}$$

其中, $\boldsymbol{W}_{\text{in}}$ 和 $\boldsymbol{W}_{\text{out}}$ 分别表示输入端与输出端的权重矩阵, $\boldsymbol{b}_{\text{in}}$ 和 $\boldsymbol{b}_{\text{out}}$ 为对应的偏置项, $\phi(\cdot)$ 表示模块内部的非线性变换. 在该表达模板下, 注意力模块可视为特殊情形, 将输入端三个投影矩阵 $\boldsymbol{W}_q$、$\boldsymbol{W}_k$、$\boldsymbol{W}_v$ 合并为 $\boldsymbol{W}_{\text{in}}$, 输出投影记为 $\boldsymbol{W}_{\text{out}}$, 并令 ϕ 对应注意力映射.

10.2.2 SliceGPT

设 $\boldsymbol{Q} \in \mathbb{R}^{d\times d}$ 为正交矩阵, 满足 $\boldsymbol{Q}^T\boldsymbol{Q} = \boldsymbol{Q}\boldsymbol{Q}^T = \boldsymbol{I}$. 由于正交变换保持向量范数不变, 且均方根归一化 (root mean square layer normalization, RMSNorm) 仅依赖范数归一化, 可得均方根归一化与正交变换可交换, 即

$$\text{RMSNorm}(\boldsymbol{X}^{(l)}\boldsymbol{Q})\boldsymbol{Q}^T = \text{RMSNorm}(\boldsymbol{X}^{(l)}), \tag{10.2}$$

其中, $\boldsymbol{X}^{(l)}$ 为网络的第 l 层. 这意味着在相邻模块间插入 $\boldsymbol{Q}$ 与 $\boldsymbol{Q}^T$ 不会改变结构. 进一步, 可将这些变换吸收到线性层权重中, 得到如下与原网络等价的参数重写形式

$$\hat{\boldsymbol{W}}_{\text{embd}} = \boldsymbol{W}_{\text{embd}}\boldsymbol{Q},\ \hat{\boldsymbol{W}}_{\text{in}}^{(l)} = \boldsymbol{Q}^T\boldsymbol{W}_{\text{in}}^{(l)},\ \hat{\boldsymbol{W}}_{\text{out}}^{(l)} = \boldsymbol{W}_{\text{out}}^{(l)}\boldsymbol{Q},\ \hat{\boldsymbol{W}}_{\text{head}} = \boldsymbol{Q}^T\boldsymbol{W}_{\text{head}}, \tag{10.3}$$

其中, $\boldsymbol{W}_{\text{embd}}$ 为嵌入层权重, $\boldsymbol{W}_{\text{in}}^{(l)}$ 和 $\boldsymbol{W}_{\text{out}}^{(l)}$ 分别为第 l 层输入端与输出端权重, $\boldsymbol{W}_{\text{head}}$ 为输出头权重. 因此, 模型可在任意正交基下执行而保持输出不变, 这正是 SliceGPT 可行的理论基础. 上述不变性严格依赖均方根归一化连接, 对于采用层归一化的模型, SliceGPT 先进行等价转换, 即将层归一化中的缩放项吸收到后继输入矩阵, 将均值消除项并入前驱输出矩阵, 同时对嵌入层与输出头作一致重标定. 该过程本质是运算次序重排, 不改变函数映射, 是网络满足均方根归一化不变性分析前提. 设 m 为需保留的主成分数, 在将特征转换到新的正交基后, 模型可以通过删除矩阵 $\boldsymbol{D} \in \mathbb{R}^{d\times m}$ 对特征维度进行裁减, 具体为

$$\boldsymbol{Z} = \boldsymbol{X}\boldsymbol{Q}\boldsymbol{D},\ \hat{\boldsymbol{X}} = \boldsymbol{Z}\boldsymbol{D}^T\boldsymbol{Q}^T, \tag{10.4}$$

其中, $\hat{\boldsymbol{X}}$ 为通过反向变换重构的特征矩阵. 通过对每层输入输出特征的协方差矩阵进行特征分解, SliceGPT 能够识别出主成分并保留前 m 个主成分以实现层内裁剪.

10.2.3 DLP

给定不重要性度量函数 $h(\cdot)$, 用于评估每层中权重的重要性. DLP 首先通过下式计算第 l 层的绝对不重要性

$$s^{(l)} = \sum_{i=1}^{c_{\text{out}}}\sum_{j=1}^{c_{\text{in}}} h(A_{ij}^{(l)}), \tag{10.5}$$

其中, c_{in} 和 c_{out} 为输入和输出的通道数, $A_{ij}^{(l)}$ 为第 l 层中第 i 个输出通道与第 j 个输入通道之间权重的重要性分数, 计算公式如下

$$A_{ij}^{(l)} = |W_{ij}^{(l)}| \cdot \|\boldsymbol{x}_j^{(l)}\|_2, \tag{10.6}$$

其中, $|W_{ij}^{(l)}|$ 为权重 $W_{ij}^{(l)}$ 的幅值, $\|\boldsymbol{x}_j^{(l)}\|_2$ 为第 l 层中第 j 个输入通道的激活范数, $h(\cdot)$ 可取不同统计形式 (如求和、平均值、中位数、最大值、方差、标准差). 为使不同层之间的重要性可比较, DLP 将不重要性分数归一化并转换为相对重要性, 即

$$y^{(l)} = 1 - \frac{s^{(l)}}{\sum_{j=1}^{L} s^{(j)}}, \tag{10.7}$$

其中, $y^{(l)}$ 为第 l 层的相对重要性, L 为总层数. 所有层的重要性向量 $\{y^{(1)}, y^{(2)}, \ldots, y^{(L)}\}$ 共同构成相对重要性分布 (relative importance distribution, RID). 该公式将不重要性取逆得到重要性, 确保重要性越高的层分配越低的稀疏度.

在确定各层的重要性后, DLP 引入动态稀疏度分配机制. 给定全局目标稀疏度 r 和超参数 α, 将每层的稀疏度控制在 $[r-\alpha, r+\alpha]$ 范围内, 并通过以下公式计算最终的层稀疏度

$$r^{(l)} = r + \bar{b} - b^{(l)}, \tag{10.8}$$

其中, $\bar{b}$ 为所有层 $b^{(l)}$ 的均值, $b^{(l)}$ 的定义为

$$b^{(l)} = \frac{y^{(l)} - y_{\min}}{y_{\max} - y_{\min}} \times 2\alpha, \tag{10.9}$$

且 $y_{\min}$ 和 $y_{\max}$ 表示 $y^{(l)}$ 的最小值与最大值. 该策略通过先缩放再中心化的方式, 使层稀疏度在给定区间内波动, 同时保证整体平均稀疏度为 r, 从而避免某一层被过度剪枝.

10.3 模型与算法

10.3.1 构建模型

事实上, DLP 通过中位数计算层非重要性并转化为相对重要性, 相较于均匀剪枝的等比例分配, 能够更准确地刻画层重要性差异. 但该方法仍存在一定局限性. 一方面, 度量指标缺乏任务导向, 仅基于权重幅值、激活统计等模型内在特征, 未与任务目标关联, 可能遗漏对任务重要但统计量不突出的层. 另一方面, 未引入梯度信息衡量层重要性, 忽视不同层对损失函数的敏感差异, 无法有效识别参数量精简但梯度敏感的重要层.

本章提出了梯度引导的自适应剪枝框架, 将层重要性度量直接对齐任务目标, 通过梯度敏感度分析实现更精准的稀疏度分配. 为便于表述, 设 i 为层索引 (同上一节的记号 l), n 为总层数, d 为隐藏层维度, m 为保留维度, $s_i \in [0,1)$ 为该层分配的稀疏度, 则该层保留维度可写为 $m_i = \lfloor (1-s_i)d \rfloor$, 其中 $m_i = \lfloor \cdot \rfloor$ 表示向下取整. 设 $\lambda_{i,j}$ 为第 i 层的第 j 个特征值, 令 $\bar{\lambda}_{i,k} = \lambda_{i,k} / \sum_{j=1}^{d} \lambda_{i,j}$ 为归一化的特征值. 给定全局稀疏度 s, 构建如下 GAP 模型

$$\begin{aligned} \max_{\{s_i\}} \quad & \sum_{i=1}^{n} \beta_i f_i(s_i) \\ \text{s.t.} \quad & \frac{1}{n} \sum_{i=1}^{n} s_i \geq s, \\ & s_i \in [0,1),\ i \in \{1,2,\ldots,n\}, \end{aligned} \tag{10.10}$$

其中, 目标函数中

$$f_i(s_i) = \sum_{k=1}^{\lfloor (1-s_i)d \rfloor} \bar{\lambda}_{i,k}, \tag{10.11}$$

$\beta_i > 0$ 为梯度先验权重. 第一个约束条件表示所有层的平均稀疏度需要大于全局稀疏度 s. 与 DLP 相比, GAP 在目标函数构造上由静态统计驱动转向梯度驱动, 通过梯度权重 β_i 刻画损失函数对各层的敏感程度, 使得模型能够更准确反映不同层对最终任务性能的真实贡献, 也能在相同全局稀疏约束下实现更优的层间资源分配.

10.3.2 对数映射

在大语言模型中, 底层的 Embedding 层往往具有较小的梯度范数, 而顶层的分类头则梯度范数极大[177]. 如果直接使用原始梯度作为权重, 敏感度极高的层会完全垄断维度资源, 而敏感度较低的层则被过度压缩, 导致模型性能崩溃. 对数函数具有压缩大数值、放大小数值的特性, 能够有效缓解长尾分布不平衡. 当梯度范数跨多个数量级时, 线性归一化易将大部分数值压缩至极小区间, 导致细节丢失. 而对数归一化在保持数据相对顺序的同时, 能让不同量级的梯度得到更充分的表达. 为此, 本章采用对数映射进行权重校准.

若记第 i 层所有参数梯度敏感度的均值为 g_i, 并记 $g_{\min}$ 和 $g_{\max}$ 分别为所有层在对数空间中 $\log(g_i)$ 的最小值与最大值. 则校准后的权重 β_i 计算如下

$$\beta_i = 1 + (\beta_{\max} - 1) \cdot \frac{\log(g_i) - g_{\min}}{g_{\max} - g_{\min}}. \tag{10.12}$$

该映射将梯度范数平滑映射到 $[1, \beta_{\max}]$ 区间. 这种处理既能保留层间敏感度的相对顺序, 又能弱化长尾效应. 参数 $\beta_{\max}$ 控制权重分配的动态范围. 当 $\beta_{\max} = 1$ 时, 所有层权重相同, 方法退化为均匀分配. 通过对数映射校准, 模型能够更合理地分配维度资源, 避免过度集中于少数高敏感层, 从而在全局稀疏约束下实现更优的性能表现.

10.3.3 贪心求解

式 (10.10) 是一个非线性组合优化问题, 精确求解属于 NP 难问题. 因此, 采用贪心策略寻求近似最优解, 从极大稀疏到逐步回收. 初始化时将除第一层外的所有层设为最大稀疏度 $s_{\max}$, 随后通过 K 轮迭代, 每轮选择降低稀疏度可带来最大边际性能增益的层进行更新. 为满足硬件对齐要求, 定义稀疏度步长 $\Delta s = r/d$, 其中 r 为硬件对齐步长, 即稀疏度仅能以 Δs 为单位进行调整. 当第 i 层的稀疏率从 s_i 降低到 $s_i - \Delta s$ 时, 边际性能增益定义为

$$\Delta G_i(s_i) = \beta_i(f_i(s_i - \Delta s) - f_i(s_i)), \tag{10.13}$$

贪心分配流程分为边界初始化、动态注入与收敛产出三个阶段. 完整的迭代过程见算法 2.

(1) 边界初始化

将除第一层外所有层的稀疏度初始化为最大允许上限 $s_{\max}$, 并计算当前平均稀疏度与全局目标稀疏度 s 之间的配额盈余, 即

$$\mathcal{R} = (n-1)s_{\max} - ns. \tag{10.14}$$

配额盈余表示需要回收的稀疏度总量, 以满足全局稀疏度目标.

(2) 动态注入

将盈余配额折算为可迭代步数 $K = \lfloor(\mathcal{R} \cdot n \cdot d)/r\rfloor$, 算法维护最大堆存储各层边际收益

算法 2 求解式 (10.10) 的贪心算法

输入: 模型层数 n, 全局稀疏度 s, 梯度权重 $\{\beta_i\}$, 步长 Δs

初始化: $s_i^0 = s_{\max}$, $\forall i \in \{2, 3, \ldots, n\}$, $s_1^0 = 0$, 计算配额盈余 $\mathcal{R}$, 迭代步数 K

当 $k \le K$ **时**

1: 通过式 (10.13) 计算各层边际收益 $\Delta G_i(s_i^{k-1})$
2: 通过式 (10.15) 选择收益最大的层 i^*
3: 通过式 (10.16) 更新稀疏度 $s_{i^*}^k$
4: 保持其他层不变 $s_i^k = s_i^{k-1}$, $\forall i \neq i^*$

结束循环

输出: 各层稀疏度 $\{s_i^*\}$

$\Delta G_i(s_i)$. 每轮迭代中, 弹出收益最大的层

$$i^* = \operatorname*{argmax}_i \Delta G_i(s_i). \tag{10.15}$$

若更新后稀疏度低于最小约束 $s_{\min}$ 则移除该层, 否则将该层稀疏度更新为

$$s_{i^*}^{(k+1)} = s_{i^*}^{(k)} - \Delta s, \tag{10.16}$$

重新计算下一阶边际收益后压回堆中.

(3) 收敛产出

重复迭代直至耗尽步数 K, 完成收敛产出, 最终得到符合硬件对齐要求且全局性能最优的差异化稀疏配置 $\{s_1^*, s_2^*, \ldots, s_n^*\}$.

10.3.4 复杂度分析

设网络层数为 n, 隐藏维度为 d, 校准批次数为 B, 每批有效词元 (Token) 数为 T, 硬件对齐步长为 r, 分配迭代步数为 K. 离线阶段包含特征值谱采集与梯度敏感度采集, 计算复杂度为 $T_1 = O(n(BTd^2 + d^3)) + O(BnTd^2)$, 其中, 前一项对应各层激活统计与协方差构建, nd^3 对应各层特征分解, 后一项对应逐层前向缓存与反向传播的梯度敏感度采集开销. 分配阶段的时间复杂度为 $T_2 = O(K \log n)$, 其中主导开销来自 K 轮堆弹出与回插操作, 初始化与建堆项在大规模设定下通常可忽略于主项. 因此, 总时间复杂度为 $T = T_1 + T_2$. 一般而言, 离线阶段通常主导总体耗时, 而分配阶段相对轻量.

10.4 数值实验

为验证所提 GAP 方法的有效性, 本节选取 SparseGPT[172]、Wanda[173]、ADMM[bovza2024fast]三种非结构化剪枝方法, 以及 SliceGPT[175]、DLP[176] 两种结构化剪枝方法, 开展稀疏剪枝方法的

评估. 所有实验均在工作站平台上完成, CPU 采用 Intel Core Ultra 9 285K 处理器, GPU 为 NVIDIA RTX 5090 并配备 32GB 显存, 内存大小为 64GB DDR5, 操作系统为 Ubuntu 22.04.4 LTS, CUDA 版本为 12.8, 深度学习框架采用 PyTorch 2.7.1, Python 版本为 3.11. 此外, 所提方法开源代码见链接 https://github.com/xianchaoxiu/GAP.

10.4.1 实验设置

(1) 数据集

实验选用 Llama 系列与 Qwen 系列中参数量介于 1B 至 8B 的 6 个预训练模型, 各模型的参数量、网络层数、隐藏层维度及注意力头数等详细配置如表 10.1 所示.

表 10.1: 大模型的参数信息

模型	参数量	层数	隐藏层维度	注意力头数
Llama3.2-1B	1.23B	16	2,048	32
Llama3.2-3B	3.21B	28	3,072	24
Llama3.1-8B	8.03B	32	4,096	32
Qwen2.5-1.5B	1.54B	28	1,536	12
Qwen2.5-3B	3.09B	36	2,048	16
Qwen2.5-7B	7.61B	28	3,584	28

WikiText2 数据集[178] 由高质量维基百科文本构成, 包含约 200 万训练词与 20 万验证词, 是自然语言处理领域中评估语言建模效果的经典基准数据集. 为保证实验对比的公平性, 本章参照 SliceGPT 的实验设置, 从 WikiText2 数据集中随机选取 128 条样本作为校准数据, 每条样本包含 2,048 个词元. 在剪枝过程中, 设置最大稀疏度 $s_{\max} = s + 0.05$, 最小稀疏度 $s_{\min} = 0$, 权重动态范围 $\beta_{\max} = 10$, 硬件对齐步长 $r = 8$, 确保剪枝配置满足实际部署需求.

(2) 评估指标

困惑度 (perplexity, PPL) 是评估语言模型性能的指标, 其定义为

$$\mathrm{PPL} = \exp\left(-\frac{1}{N}\sum_{i=1}^{N}\log\ p(x_i \mid x_1, x_2, \ldots, x_{i-1})\right), \tag{10.17}$$

其中, N 为测试集的序列长度, $p(x_i \mid x_1, x_2, \ldots, x_{i-1})$ 为模型对第 i 个词元的预测概率. 困惑度数值越低, 表明模型对序列的预测能力越强[179].

10.4.2 结果分析

(1) 性能对比

表 10.2 和表 10.3 分别给出了不同方法在 Llama 系列与 Qwen 系列模型上的困惑度对比结果, 其中 Dense 代表剪枝前的原始模型状态, SliceGPT、DLP 与 GAP 的稀疏度为 0.3. 可以看出, 尽管各方法在不同模型上的表现存在差异, 但在平均意义下, 所提 GAP 取得了与经典结构化剪枝方法 SliceGPT 与 DLP 相当的困惑度性能. 特别地, 针对 Llama3-8B 模型, GAP 的困惑度较 SliceGPT 降低 6.45, 较 DLP 降低 1.87, 优势更为明显. 上述结果验证了 GAP 所采用的稀疏自适应分配策略的有效性, 能够有效弥补结构化约束带来的模型表达能力下降. 此外, 与非结构化剪枝方法相比, 所提 GAP 性能优于 Wanda, 并与 SparseGPT、ADMM 基本持平. 这说明, 尽管 GAP 属于约束更强的结构化剪枝框架, 但其在多数模型上仍能实现优异的性能.

表 10.2: Llama 模型剪枝后的困惑度, 其中最优结构化剪枝结果以加粗标注

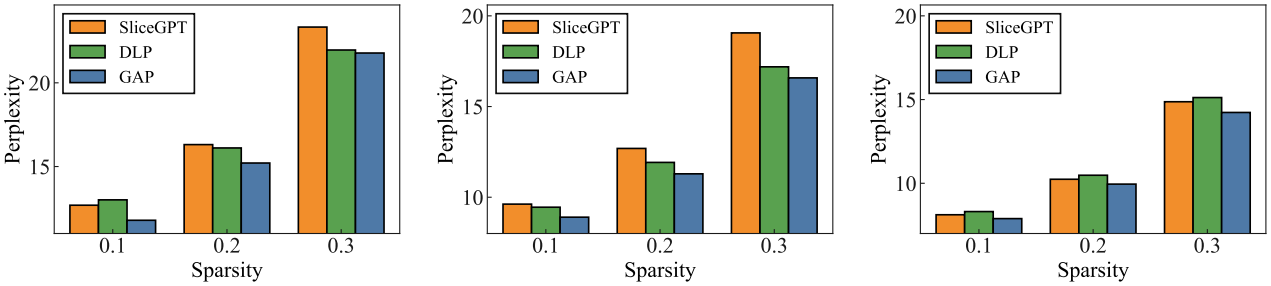
方法	稀疏度	Llama3.2-1B	Llama3.2-3B	Llama3-8B	平均
Dense	0	9.75	7.81	6.14	7.90
SparseGPT	2:4	24.92	16.06	12.24	17.74
Wanda	2:4	94.77	30.62	20.21	48.53
ADMM	2:4	20.86	14.91	11.26	15.68
SliceGPT	0.3	23.33	19.06	19.40	20.60
DLP	0.3	21.96	17.19	14.82	17.99
GAP	0.3	**21.78**	**16.58**	**12.95**	**17.10**

表 10.3: Qwen 模型剪枝后的困惑度, 其中最优结构化剪枝结果以加粗标注

方法	稀疏度	Qwen2.5-1.5B	Qwen2.5-3B	Qwen2.5-7B	平均
Dense	0	9.26	8.03	6.85	8.05
SparseGPT	2:4	17.26	12.76	9.33	13.12
Wanda	2:4	36.36	21.54	13.02	23.64
ADMM	2:4	15.61	12.09	9.07	12.26
SliceGPT	0.3	20.87	15.04	**10.55**	15.49
DLP	0.3	20.95	15.44	10.68	15.69
GAP	0.3	**19.56**	**14.89**	10.79	**15.08**

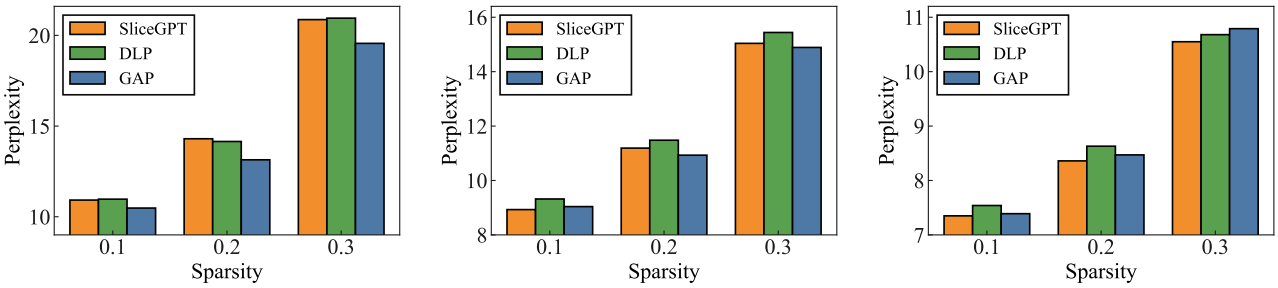
(2) 稀疏度分析

为量化不同结构化剪枝方法的性能差异, 图 10.2 和图 10.3 以柱状图形式呈现了 SliceGPT、DLP 与 GAP 三种方法在 Llama 系列与 Qwen 系列模型上不同稀疏度下的困惑度对比结果. 由结果可知, 当稀疏度从 0.1 逐步提升至 0.3 时, 各方法的困惑度均呈上升趋势, 但在绝大多数实验场景中, 所提 GAP 的性能表现普遍优于 SliceGPT 与 DLP. 从模型规模看, GAP 在小参数量模型上的性能表现稳健. 相比之下, SliceGPT 在高稀疏与大模型场景下性能下降剧烈, 鲁棒性不足. DLP 虽整体优于 SliceGPT, 但与 GAP 方法相比仍存在一定的差距. 上述实验结果表明, 所提 GAP 能够在模型压缩效率与性能保持之间实现更优的平衡.



(a) Llama3.2-1B **(b)** Llama3.2-3B **(c)** Llama3.1-8B

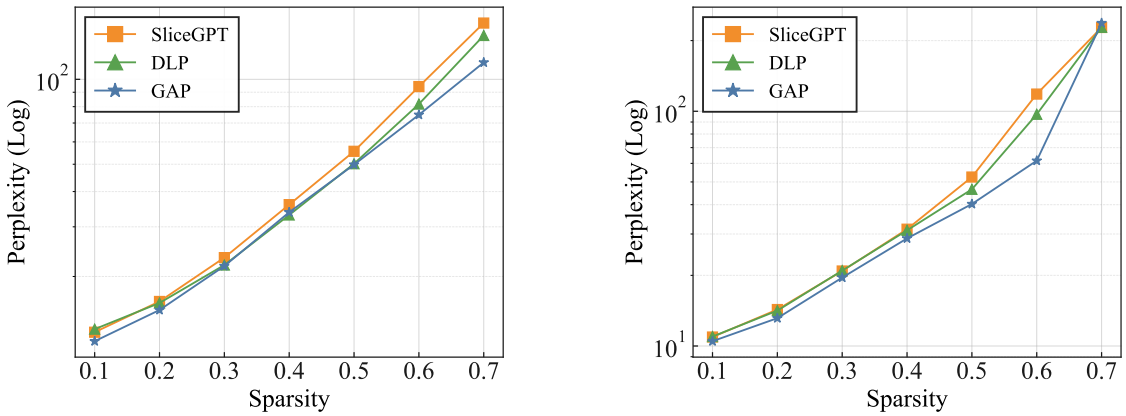
图 10.2: Llama 模型结构化剪枝后的困惑度



(a) Qwen2.5-1.5B **(b)** Qwen2.5-3B **(c)** Qwen2.5-7B

图 10.3: Qwen 模型结构化剪枝后的困惑度

为了进一步验证所提 GAP 在高稀疏度条件下的性能优势, 选取 Llama3.2-1B 与 Qwen2.5-1.5B 两个轻量化模型, 在稀疏度 0.1 至 0.7 的区间内开展困惑度评估实验, 实验结果如图 10.4 所示. 尽管所有方法的困惑度均不可避免地随着稀疏度的增加而上升, 但所提 GAP 在整个测试稀疏度区间内, 几乎始终保持着最低的困惑度水平. 尤其是在 0.5 至 0.7 的高稀疏度区间, 当对比方法出现性能崩塌时, GAP 依然具有较好的鲁棒性.



(a) Llama3.2-1B **(b)** Qwen2.5-1.5B

图 10.4: 高稀疏度时的困惑度比较

10.5 实际部署

本节将在 Seeed Studio reComputer J30/40 系列设备上完成边缘部署, 具体型号为 reComputer J4012, 其计算模组为 Jetson Orin NX 16GB, 设备实物图如图10.5所示.

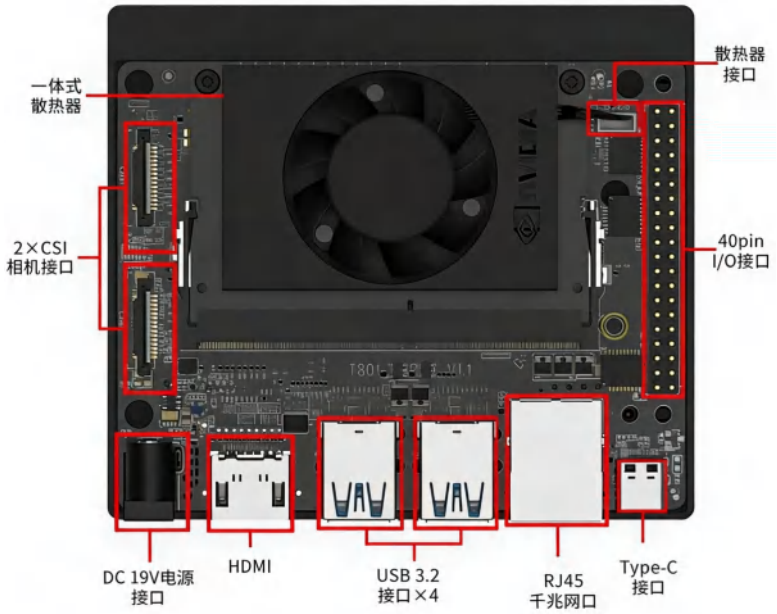


图 10.5: Jetson Orin NX 设备实物图

该平台在功耗、体积与推理性能之间实现了良好的工程平衡, 能够较真实地反映受限算力条件下大模型的实际部署环境. 相较于云端 GPU 服务器, 该平台更贴近边缘端的实际应用形态, 可有效验证剪枝模型在端侧的可部署性与实时响应性能. 相较于纯 CPU 边缘终端, 其具备更强的并行推理能力, 便于系统全面评估剪枝方法在推理吞吐、延迟及硬件资源占用等方面的综合收益. 表 10.4 给出了该平台软硬件参数.

表 10.4: 软硬件环境配置

组件	配置
设备型号	reComputer J4012
计算模组	NVIDIA Jetson Orin NX 16GB
AI 性能	100 TOPS
GPU	1024-core NVIDIA Ampere GPU + 32 Tensor Cores
CPU	8-core Arm Cortex-A78AE 64-bit
内存	16GB 128-bit LPDDR5 (102.4 GB/s)
存储	128GB NVMe SSD (预装 JetPack)
系统软件	JetPack 6.2 + Ubuntu 22.04
网络与外设	1×GbE, 4×USB 3.2, HDMI 2.1, 2×CSI
扩展接口	M.2 Key E, M.2 Key M, CAN, GPIO

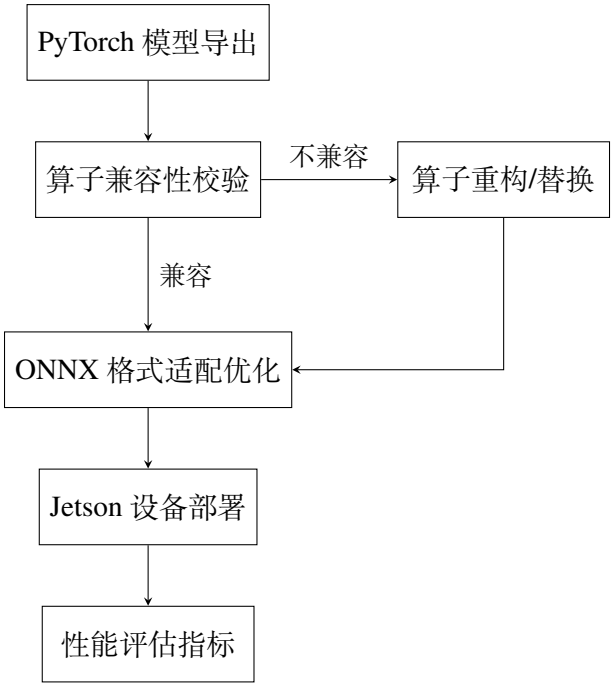


图 10.6: Jetson 部署流程

10.5.1 部署流程

部署流程共有五个步骤, 依次为模型导出、格式适配、模型优化、设备部署及性能评估, 如图 10.6 所示, 各步骤具体实施细节如下.

(1) 模型导出

完成剪枝后, 将 PyTorch 模型导出为 ONNX (Open Neural Network Exchange) 格式. ONNX 是一种开放的模型交换格式, 支持跨平台部署. 导出时显式指定输入输出节点名称, 涵盖词元序列、注意力掩码、位置编码及各层 KV 缓存, 并将批大小、序列长度和 KV 缓存长度声明为动态维度, 以支持变长推理. 模型权重以外部数据文件形式存储, 便于在存储受限的边缘设备上管理大体积参数.

(2) 格式适配

针对 Jetson Orin NX 平台的硬件特性, 对导出的 ONNX 模型进行格式适配优化. 重点检查模型算子兼容性, 替换平台不支持的高阶算子为兼容算子, 确保模型能够被 ONNX Runtime 正常解析, 同时调整模型输入输出的数据类型, 匹配边缘设备的计算精度需求, 避免因格式不兼容导致的推理失败或性能损耗.

(3) 模型优化

导出过程中开启常量折叠优化, 使导出器在追踪阶段预先计算图中的固定值子图, 消除推理期间的冗余计算. 部署到 Jetson 设备后, ONNX Runtime 在加载模型时会自动执行运行时图优化, 包括算子融合和内存访问合并, 进一步降低推理延迟.

(4) 设备部署

将适配优化后的 ONNX 模型文件传输至 Jetson 设备, 需要进行依赖检查、磁盘空间校验及模型文件完整性验证, 确保部署环境就绪后加载模型. 同时配置 ONNX Runtime 的运行参数, 指定模型加载路径和硬件推理后端, 为后续推理测试做好准备.

(5) 性能评估

使用 ONNX Runtime 的推理接口分别在 CPU 和 GPU 两种后端上运行测试, 批处理大小设为 1 (边缘部署场景). 推理前执行若干轮预热以消除冷启动误差, 随后进行多轮正式测试, 记录吞吐量、首字延迟和逐字延迟等性能指标, 完成剪枝模型在边缘平台的部署性能验证.

10.5.2 性能对比

与前述仿真实验中使用困惑度不同, 本节选取吞吐量、首词元延迟、单词元生成时间及加载时间等四个性能指标.

- 吞吐量 (Token/s): 衡量模型推理效率的重要指标, 定义为每秒生成的词元数量. 固定生成序列长度 $T = 100$, 记录模型生成该序列的总耗时 t_{total}, 通过公式 T/t_{total} 计算得出.
- 首词元延迟 (time to first Token, TTFT/ms): 反映用户初始等待体验的关键指标, 定义为从输入请求发出到模型输出第一个词元的时间间隔, 直接测量该时间差即可获得.
- 单词元生成时间 (time per output Token, TPOT/ms): 影响文本生成流畅度的重要指标, 定义为输出第二个 Token 及之后每个词元的平均生成延迟, 通过计算第二个词元至第 T 个词元的生成时间均值得到.
- 加载时间 (load time, LT/ms): 衡量模型冷启动效率的指标, 定义为从模型文件从磁盘读取并加载至内存、完成初始化的总时间, 通过记录模型加载函数的完整执行时长进行测量.

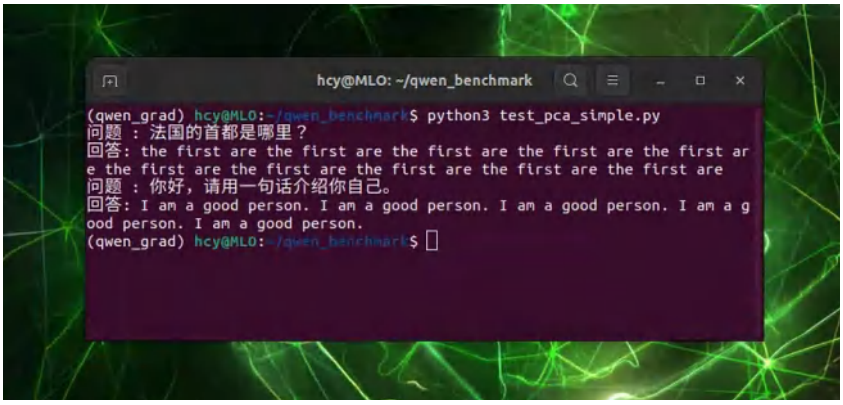
表 10.5: CPU 推理性能

方法	吞吐量 ↑	首词元延迟 ↓	单词元生成时间 ↓	加载时间 ↓
Dense	2.2	455.7	236.2	22,924.3
SliceGPT	2.7	376.0	434.3	**19,057.4**
DLP	2.4	424.1	**344.4**	19,516.3
GAP	**2.8**	**351.9**	372.6	20,248.6

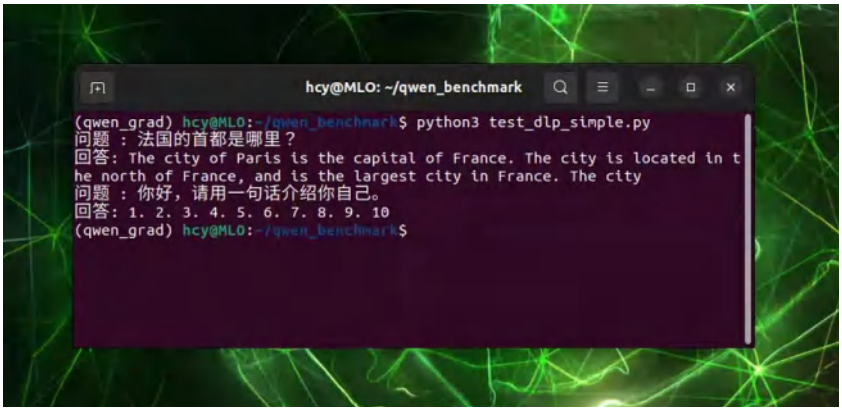
表 10.6: GPU 推理性能

方法	吞吐量 ↑	首词元延迟 ↓	单词元生成时间 ↓	加载时间 ↓
Dense	12.9	77.5	70.8	8,663.5
SliceGPT	15.7	63.7	56.4	**7,323.1**
DLP	**15.9**	**62.8**	**55.7**	9,550.2
GAP	15.8	63.2	57.3	7,446.0

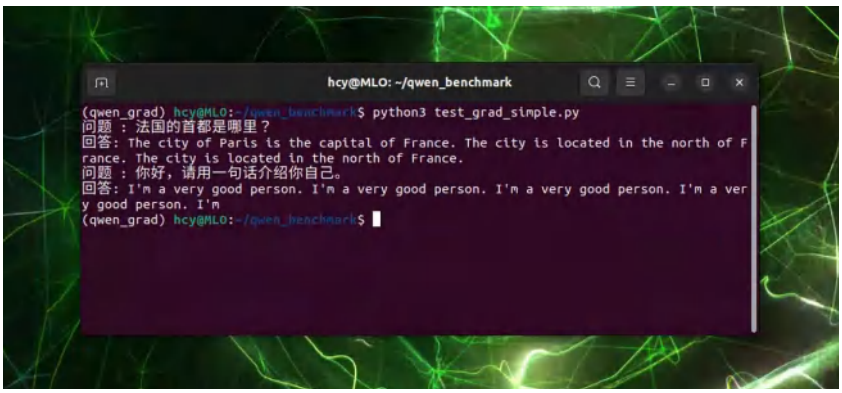
表 10.5 和表 10.6 分别展示了 CPU 与 GPU 两种运行模式下各方法的推理性能测试结果. 从 CPU 模式结果可以看出, 三种结构化剪枝方法优于原始模型. 与原始模型相比, GAP 吞吐量提升 29.7%, 后续词元延迟降低首词元延迟降低 22.8%, 加载时间降低 11.7%. 值得注意的是, 剪枝操作引入的稀疏结构会导致内存访问模式变得不规则, 降低缓存效率, 因此单词元生成时间均出现不同程度的上升. 从 GPU 模式结果可以看出, 三种结构化剪枝方法同样优于原始模型. 其中相比于原始模型, GAP 的吞吐量提升 22.6%, 首词元延迟降低 18.5%, 后续词元延迟降低 19.1%, 加载时间降低 14.1%.



(a) SliceGPT



(b) DLP



(c) GAP

图 10.7: Jetson Orin NX 上的对话结果

综上, 所提 GAP 的困惑度指标显著优于 SliceGPT 和 DLP, 而在 CPU 和 GPU 推理速度上, GAP 与后两者的表现几乎无差异, 这表明 GAP 能够同时兼顾模型性能与推理效率.

10.5.3 对话演示

为直观验证剪枝模型在实际场景中的部署效果, 本节开发了基于 Jetson Orin NX 的对话演示. 通过预设的 prompt, 模型实时生成回答. 以问题 “法国的首都是哪里？” “你好, 请用一句话介绍你自己。” 为例, 实验结果如图 10.7 所示. SliceGPT 回答第一个问题时出现错误, DLP 回答第二个问题时出现错误, 而所提 GAP 均回答正确, 这表明 GAP 在生成质量上表现更为稳定. 需要指出的是, 由于预训练阶段的基础模型参数量本身较少, 在此基础上进行剪枝操作后, 模型的生成质量与输出多样性会受到较为显著的影响.

10.6 本章小结

本章针对现有层间结构化剪枝方法依赖静态统计量、未充分利用梯度信息的问题, 提出了梯度引导的自适应剪枝方法. 相较于基于权重幅度和激活统计的层重要性度量, 所提方法通过分析损失函数对各层参数的梯度敏感度来评估层贡献, 并设计了对数映射校准机制, 有效缓解层间梯度敏感度波动. 在求解层面, 采用基于最优边际收益的贪心算法, 借助最大堆数据结构实现了高效稀疏配置. 通过在 Llama3 和 Qwen2.5 系列模型上的数值实验, 验证了方法的压缩性能. 剪枝后, 进一步在 NVIDIA Jetson Orin NX 边缘平台上完成了端到端的部署, 实现了剪枝模型在受限算力环境下的实际推理.

第三部分

拓展篇

第 11 章 基于深度展开的图像反问题求解方法

图像反问题是一类典型的病态问题, 旨在从退化的观测图像中恢复未知的原始清晰图像. 近年来, 深度展开方法通过将迭代优化算法映射为含可训练参数的多阶段展开网络, 不仅能够自适应调节正则参数、惩罚参数、步长等, 还可有效学习图像的先验结构, 为图像反问题的求解提供了全新的视角. 本章围绕图像反问题的深度展开求解方法展开综述, 首先介绍几种典型的迭代优化算法, 然后系统梳理参数学习型、结构学习型及生成式驱动型三类深度展开方法, 概括相关研究的发展脉络与主要特点. 在此基础上, 结合图像压缩感知重建实验, 对不同深度展开方法的性能进行对比.

11.1 引言

给定退化观测图像 $\boldsymbol{y} \in \mathbb{R}^m$, 待恢复的原始清晰图像 $\boldsymbol{x} \in \mathbb{R}^n$ 以及退化算子 $\boldsymbol{A} \in \mathbb{R}^{m\times n}$, 图像反问题可归纳为如下数学模型

$$\min_{\boldsymbol{x}} \ \frac{1}{2}\|\boldsymbol{A}\boldsymbol{x}-\boldsymbol{y}\|_2^2 + \lambda g(\boldsymbol{x}), \tag{11.1}$$

其中, 第一项用于约束重建结果与观测数据的一致性, $g(\boldsymbol{x})$ 用于刻画图像的先验知识, $\lambda > 0$ 为正则参数, 用于平衡数据一致性项与先验信息项的权重. 图像反问题广泛应用于医学影像、遥感测绘、工业检测等领域, 具有重要的研究意义与科研价值.

围绕上述模型, 研究者已发展出多种经典迭代优化算法, 其中具有代表性的包括迭代收缩阈值法 (iterative shrinkage-thresholding algorithm, ISTA) 及其加速形式 (fast iterative shrinkage-thresholding algorithm, FISTA)[180]、交替方向乘子法 (alternating direction method of multipliers, ADMM)[50], 以及原始-对偶混合梯度法 (primal-dual hybrid gradient, PDHG)[181] 等. 这些算法具有较强的可解释性和收敛性保证, 但在复杂图像分布、未知退化以及跨任务等实际应用中, 其固定的迭代规则与手工设计的先验函数, 往往难以满足大规模图像反问题求解的高精度、高效率与高鲁棒性需求[182].

神经网络技术的快速发展为图像反问题的高效求解提供了全新思路, 通过从大量训练数据中自适应学习复杂的图像先验特征与非线性映射关系, 有效弥补了手工先验表达能力不足的问题[183]. 同时, 迭代优化算法为神经网络结构设计提供了明确的来源, 使得模型与数据的融合成为可能[184]. 在此背景下, 深度展开 (deep unfolding, DU) 方法应运而生. 它将迭代优化算法的有限步更新过程映射为多阶段展开网络结构, 在每个阶段中通过数据驱动的方式学习步长、阈值、变换算子、近端映射等关键参数, 从而在保留迭代优化算法结构可解释性的基础上, 提升图像恢复的精度与效率. 感兴趣的读者可参考相关综述[185-188]. 需要说明的是, 除深度展开

方法外, 图像反问题求解还存在即插即用 (plug-and-play, PnP) 及端到端等其他思路, 但上述方法并非本章的研究重点.

经过文献梳理, 本章依据网络学习内容与建模方式的差异, 将图像反问题中的现有深度展开求解方法归纳为以下三类:

(1) 参数学习型: 通常保留迭代优化算法的基本更新形式与固定解析先验, 主要学习步长、阈值、线性变换矩阵等迭代参数.

(2) 结构学习型: 将近端映射、变换算子、正则化结构或去噪模块替换为可学习的神经网络, 从而能够以隐式学习的先验信息提升重建性能.

(3) 生成式驱动型: 通过在深度展开迭代中引入由生成式模型刻画的数据分布先验, 利用数据一致性约束和生成式先验共同改善反问题的求解效果.

11.2 迭代优化算法

本节简要介绍迭代收缩阈值法、交替方向乘子法、原始–对偶混合梯度法等几种典型的迭代优化算法, 为后续的深度展开奠定基础.

11.2.1 迭代收缩阈值法

当式 (11.1) 中的正则项 $g(\boldsymbol{x})$ 为简单非光滑函数, 且其近端映射易于计算时, 可采用近端梯度法对该优化问题进行求解[189]. 取 $g(\boldsymbol{x}) = \|\boldsymbol{x}\|_1$, 则可得到如下模型

$$\min_{\boldsymbol{x}} \ \frac{1}{2}\|\boldsymbol{A}\boldsymbol{x} - \boldsymbol{y}\|_2^2 + \lambda\|\boldsymbol{x}\|_1. \tag{11.2}$$

该式在统计上又称 Lasso 问题. 数据一致性项 $\frac{1}{2}\|\boldsymbol{A}\boldsymbol{x} - \boldsymbol{y}\|_2^2$ 的梯度为

$$\boldsymbol{A}^T(\boldsymbol{A}\boldsymbol{x} - \boldsymbol{y}). \tag{11.3}$$

若选取步长 $0 < \eta \le 1/L$, 且 $L = \|\boldsymbol{A}^T\boldsymbol{A}\|_2$ 为梯度的 Lipschitz 常数, 则 ISTA 的迭代更新形式可表示为

$$\boldsymbol{x}^{k+1} = \mathcal{S}_{\eta\lambda}(\boldsymbol{x}^k - \eta\boldsymbol{A}^T(\boldsymbol{A}\boldsymbol{x}^k - \boldsymbol{y})), \tag{11.4}$$

其中, $\mathcal{S}_\theta(\cdot)$ 为软阈值算子, 其逐元素定义为 $\mathcal{S}_\theta(z_i) = \operatorname{sgn}(z_i)\max(|z_i| - \theta, 0)$.

从迭代结构上看, ISTA 的每次迭代均由一次线性梯度步和一次非线性阈值步组成, 前者利用数据一致性项修正估计, 后者通过近端映射引入稀疏先验. 由于其更新过程仅涉及线性变换、步长调节与阈值操作等简单计算步骤, ISTA 成为深度展开方法中最具代表性的框架之一. 此外, FISTA 在 ISTA 基础上引入动量外推技巧, 将收敛速度从 $O(1/k)$ 提升至 $O(1/k^2)$[180].

11.2.2 交替方向乘子法

当优化问题的正则项包含复合结构, 或其近端映射难以直接计算时, 可通过变量分裂技巧将原复杂问题分解为若干个更易求解的子问题, 进而采用 ADMM 迭代求解[boyd2011admm]. 仍以式 (11.2) 为例, 引入辅助变量 $\boldsymbol{z}$, 可将原问题改写为如下约束优化问题

$$\begin{aligned} \min_{\boldsymbol{x},\boldsymbol{z}} \quad & \frac{1}{2}\|\boldsymbol{A}\boldsymbol{x}-\boldsymbol{y}\|_2^2+\lambda\|\boldsymbol{z}\|_1 \\ \text{s.t.} \quad & \boldsymbol{x}=\boldsymbol{z}. \end{aligned} \tag{11.5}$$

对应的增广拉格朗日函数为

$$L_\rho(\boldsymbol{x},\boldsymbol{z},\boldsymbol{v})=\frac{1}{2}\|\boldsymbol{A}\boldsymbol{x}-\boldsymbol{y}\|_2^2+\lambda\|\boldsymbol{z}\|_1+\boldsymbol{v}^T(\boldsymbol{x}-\boldsymbol{z})+\frac{\rho}{2}\|\boldsymbol{x}-\boldsymbol{z}\|_2^2, \tag{11.6}$$

其中, $\boldsymbol{v}\in\mathbb{R}^n$ 为拉格朗日乘子, $\rho>0$ 为罚参数. 为简化迭代表述, 引入缩放对偶变量 $\boldsymbol{u}=\boldsymbol{v}/\rho$, 此时 ADMM 的迭代更新公式可表示为

$$\begin{cases} \boldsymbol{x}^{k+1}=\operatorname{argmin}\limits_{\boldsymbol{x}} \ \frac{1}{2}\|\boldsymbol{A}\boldsymbol{x}-\boldsymbol{y}\|_2^2+\frac{\rho}{2}\|\boldsymbol{x}-\boldsymbol{z}^k+\boldsymbol{u}^k\|_2^2, \\ \boldsymbol{z}^{k+1}=\operatorname{argmin}\limits_{\boldsymbol{z}} \ \lambda\|\boldsymbol{z}\|_1+\frac{\rho}{2}\|\boldsymbol{x}^{k+1}-\boldsymbol{z}+\boldsymbol{u}^k\|_2^2, \\ \boldsymbol{u}^{k+1}=\boldsymbol{u}^k+\boldsymbol{x}^{k+1}-\boldsymbol{z}^{k+1}, \end{cases} \tag{11.7}$$

其中, $\boldsymbol{x}$ 子问题对应如下线性系统

$$(\boldsymbol{A}^T\boldsymbol{A}+\rho\boldsymbol{I})\boldsymbol{x}^{k+1}=\boldsymbol{A}^T\boldsymbol{y}+\rho(\boldsymbol{z}^k-\boldsymbol{u}^k). \tag{11.8}$$

该线性系统可利用 Sherman-Morrison-Woodbury 公式进行加速计算, 而 $\boldsymbol{z}$ 子问题可直接通过软阈值操作求解.

与 ISTA 依赖梯度步和近端映射不同, ADMM 通过变量分裂将数据一致性项、正则项以及约束关系分别处理, 能够将问题分解为原变量 $\boldsymbol{x}$ 的更新、辅助变量 $\boldsymbol{z}$ 的更新和对偶变量 $\boldsymbol{u}$ 的更新, 具有更强的模块化特征, 便于引入不同类型的先验信息. 其局限性在于, 罚参数 ρ 的取值会显著影响算法的收敛速度, 且各子问题的求解精度也会直接影响整体的迭代性能.

11.2.3 原始–对偶混合梯度法

Chambolle-Pock 算法是 PDHG 在图像反问题领域中应用最广泛的形式, 适用于全变差 (total variation, TV)、图像去噪以及其他含线性算子正则的优化模型[181]. 考虑如下问题

$$\min_{\boldsymbol{x}} \ F(\mathcal{A}\boldsymbol{x},\boldsymbol{y})+G(\boldsymbol{x}), \tag{11.9}$$

其中, $F(\mathcal{A}\boldsymbol{x},\boldsymbol{y})$ 表示与 $\boldsymbol{y}$ 相关的数据一致性项, $\mathcal{A}$ 为线性算子, $G(\cdot)$ 表示定义在原始变量上的正则项. Chambolle-Pock 算法的典型迭代形式为

$$\begin{cases} \boldsymbol{z}^{k+1} = \mathrm{prox}_{\sigma F^*}(\boldsymbol{z}^k + \sigma \mathcal{A}\bar{\boldsymbol{x}}^k), \\ \boldsymbol{x}^{k+1} = \mathrm{prox}_{\tau G}(\boldsymbol{x}^k - \tau \mathcal{A}^* \boldsymbol{z}^{k+1}), \\ \bar{\boldsymbol{x}}^{k+1} = \boldsymbol{x}^{k+1} + \omega(\boldsymbol{x}^{k+1} - \boldsymbol{x}^k), \end{cases} \tag{11.10}$$

其中, $\boldsymbol{z}^k$ 表示第 k 次迭代中的对偶变量, F^* 表示 F 关于其第一变量的凸共轭函数, σ 和 τ 分别表示对偶步长和原始步长, ω 表示外推系数.

Chambolle-Pock 型 PDHG 的特点在于将复合项 $F(\mathcal{A}\boldsymbol{x}, \boldsymbol{y})$ 转化到对偶空间中进行处理, 使每次迭代由对偶变量 $\boldsymbol{z}$ 的更新、原始变量 $\boldsymbol{x}$ 的更新和外推 $\bar{\boldsymbol{x}}$ 的更新组成. 相比部分需要求解复杂线性系统的 ADMM, 原始-对偶框架在处理梯度算子、投影算子和 TV 正则等特定问题时更加直接高效, 也便于利用算子的结构特性降低计算成本.

11.3 参数学习型方法

本章以 LISTA (learned ISTA) 及其相关变体为研究对象, 探讨参数学习型深度展开的基本结构、收敛性、泛化性及近期进展.

11.3.1 基本结构

LISTA 将经典 ISTA 算法的迭代过程映射为展开网络结构, 通过训练数据对线性变换矩阵与阈值参数进行学习, 不仅保留了优化算法的可解释性, 还能有效提升图像的恢复精度与计算效率[140]. 作为深度展开方法的早期代表, LISTA 展示了将迭代算法网络化的基本范式[190].

LISTA 以式 (11.2) 为展开对象, 取常数 $\xi \geq \|\boldsymbol{A}^T\boldsymbol{A}\|_2$, 则对应的迭代公式可表示为

$$\boldsymbol{x}^{k+1} = \mathcal{S}_{\lambda/\xi}(\boldsymbol{x}^k - \frac{1}{\xi}\boldsymbol{A}^T(\boldsymbol{A}\boldsymbol{x}^k - \boldsymbol{y})), \ k = 0, 1, \ldots, K-1, \tag{11.11}$$

其中, ξ 为不小于 $\|\boldsymbol{A}^T\boldsymbol{A}\|_2$ 的常数, $\mathcal{S}_\theta(\cdot)$ 表示软阈值算子, K 为最大迭代次数. 若记

$$\boldsymbol{W}_1 = \frac{1}{\xi}\boldsymbol{A}^T, \ \boldsymbol{W}_2 = \boldsymbol{I} - \frac{1}{\xi}\boldsymbol{A}^T\boldsymbol{A}, \ \theta = \frac{\lambda}{\xi}, \tag{11.12}$$

则式 (11.11) 可简化为

$$\boldsymbol{x}^{k+1} = \mathcal{S}_\theta(\boldsymbol{W}_1\boldsymbol{y} + \boldsymbol{W}_2\boldsymbol{x}^k), \ k = 0, 1, \ldots, K-1. \tag{11.13}$$

ISTA 算法的迭代过程可自然对应为一个多阶段的展开网络, 其中每个阶段对应一次 ISTA 迭代更新, 软阈值算子则作为网络中的非线性激活模块. 进一步, 若允许网络各阶段的参数随迭代次数自适应调整, 可将式 (11.13) 推广为

$$\boldsymbol{x}^{k+1} = \mathcal{S}_{\theta(k)}(\boldsymbol{W}_1^k\boldsymbol{y} + \boldsymbol{W}_2^k\boldsymbol{x}^k), \ k = 0, 1, \ldots, K-1, \tag{11.14}$$

由此得到多阶段展开网络, 其整体结构如图 11.1 所示, 记号略有不同.

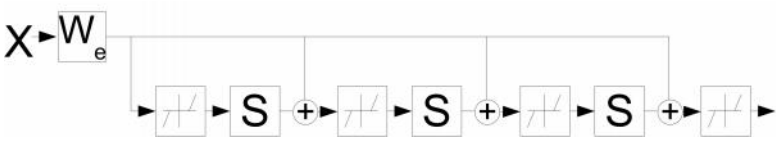


图 11.1: LISTA 展开网络结构示意图[140]

以正则参数 λ 为例, ISTA 算法中通常需通过网格搜索确定, 且参数一旦设定便固定不变. 针对不同的数据样本, 还需手动调整参数取值, 该过程往往耗费大量时间与人力成本. 而在 LISTA 中, 正则参数 λ 可通过网络训练自主学习确定, 且不同阶段对应不同的参数取值. 实验结果表明, 采用动态参数的 LISTA 模型, 其性能表现优于采用固定参数的 ISTA 算法.

11.3.2 收敛性分析

作为迭代算法的拓展形式, 深度展开可从优化理论的角度分析解的收敛性. LISTA 的收敛性研究主要围绕参数耦合关系、误差递减速率及支撑选择等开展. Chen 等[191] 指出, LISTA 中逐层学习得到的权重矩阵渐近满足如下耦合关系

$$\boldsymbol{W}_2^k = \boldsymbol{I} - \boldsymbol{W}_1^k \boldsymbol{A}. \tag{11.15}$$

基于该耦合关系, 式 (11.11) 对应的展开网络形式可进一步简化为

$$\boldsymbol{x}^{k+1} = \mathcal{S}_{\theta^k}(\boldsymbol{x}^k - (\boldsymbol{W}^k)^T(\boldsymbol{A}\boldsymbol{x}^k - \boldsymbol{y})). \tag{11.16}$$

定理 11.1 *设 $\boldsymbol{x}^* \in \{\boldsymbol{x} \in \mathbb{R}^n \mid \|\boldsymbol{x}\|_0 \leq s,\ \|\boldsymbol{x}\|_\infty \leq B\}$, 且 $\|\boldsymbol{\eta}\|_1 \leq \sigma$. 若 $\boldsymbol{A}$ 满足相应相干性条件, 且稀疏度 s 足够小, 则存在参数序列 $\{\boldsymbol{W}^k, \theta^k\}_{k=0}^{K-1}$, 使得式 (11.16) 在初始值 $\boldsymbol{x}^0 = \boldsymbol{0}$ 下生成的迭代序列满足*

$$\|\boldsymbol{x}^k - \boldsymbol{x}^*\|_2 \leq sB\exp(-ck) + C\sigma, \tag{11.17}$$

其中, $c > 0$ 和 $C > 0$ 是仅依赖于线性算子 $\boldsymbol{A}$ 和稀疏度 s 的常数.

该定理表明, 在适当参数选择下, LISTA 能够产生线性收敛的迭代序列, 并在噪声存在时收敛到由噪声水平控制的误差邻域. 相比之下, 传统 ISTA 算法通常仅能达到次线性收敛速度. 不过, 上述结论本质上属于存在性结果, 即存在一组最优参数可实现收敛加速, 但这并不意味着通过经验训练得到的参数必然满足该收敛性质.

进一步, Chen 等[191] 提出了带支撑选择的改进模型 LISTA-SS (LISTA with support selection), 其迭代形式为

$$\boldsymbol{x}^{k+1} = \mathcal{S}^{\mathrm{ss}}_{p^k,\theta^k}(\boldsymbol{x}^k - (\boldsymbol{W}^k)^T(\boldsymbol{A}\boldsymbol{x}^k - \boldsymbol{y})), \tag{11.18}$$

其中, $\mathcal{S}^{\mathrm{ss}}_{p^k,\theta^k}(\cdot)$ 表示带支撑选择的阈值算子, 其支撑保留比例通常按下式确定

$$p^k = \min\{p \cdot k,\ p_{\max}\}. \tag{11.19}$$

定理 11.2 在与式 (11.17) 对应定理相同的条件下, 存在参数序列 $\{\boldsymbol{W}^k, \theta^k\}_{k=0}^{K-1}$, 使得式 (11.18) 在初始值 $\boldsymbol{x}^0 = \boldsymbol{0}$ 且支撑保留比例 p^k 按式 (11.19) 选取时, 生成的迭代序列满足

$$\|\boldsymbol{x}^k - \boldsymbol{x}^*\|_2 \le sB \exp\left(-\sum_{i=0}^{k-1} \tilde{c}^i\right) + \tilde{C}\sigma, \tag{11.20}$$

其中, $\tilde{c}^i \ge c$ 对所有 i 成立, 且 $\tilde{C} \le C$. 在附加信噪比条件下, 当 i 足够大时, 还可满足 $\tilde{c}^i > c$ 和 $\tilde{C} < C$.

该定理验证了支撑选择能够改善 LISTA 的收敛常数, 从而提升有限层展开网络的求解效率. 此外, Liu 等[192] 提出了 ALISTA (analytic LISTA), 将带支撑选择的 LISTA 进一步简化为

$$\boldsymbol{x}^{k+1} = \mathcal{S}^{\mathrm{ss}}_{p^k,\theta^k}(\boldsymbol{x}^k - \gamma^k \boldsymbol{W}^T(\boldsymbol{A}\boldsymbol{x}^k - \boldsymbol{y})), \tag{11.21}$$

其中, 矩阵 $\boldsymbol{W}$ 可预先固定且与训练数据无关, 这显著减少了可学习的参数量, 同时保持了 LISTA 类深度展开的线性收敛特性. 在此基础上, Chen 等[193] 引入动量项对模型进行改进, 提出 HyperLISTA, 其迭代形式为

$$\boldsymbol{x}^{k+1} = \mathcal{S}^{\mathrm{ss}}_{p^k,\theta^k}(\boldsymbol{x}^k - \gamma^k \boldsymbol{W}^T(\boldsymbol{A}\boldsymbol{x}^k - \boldsymbol{y}) + \beta^k(\boldsymbol{x}^k - \boldsymbol{x}^{k-1})). \tag{11.22}$$

值得说明的是, HyperLISTA 在特定参数配置下可实现超线性收敛. 除此之外, 还有很多 LISTA 变体, 例如, Step-LISTA[194] 从步长自适应设计角度对 LISTA 结构进行改进, GLISTA[195] 则通过引入门控机制增强展开结构的特征表达能力. 限于篇幅, 这里不再逐一赘述.

11.3.3 泛化性分析

作为一类由迭代算法展开得到的神经网络, 深度展开需重点刻画其泛化误差. 与固定参数的迭代误差分析不同, 泛化性分析主要考察由有限训练样本学习得到的展开网络, 能否在总体数据分布上保持稳定的求解性能. 设训练集为 $S = \{(\tilde{\boldsymbol{y}}_i, \tilde{\boldsymbol{x}}_i)\}_{i=1}^N$, 其中, 样本满足独立同分布假设. 对于展开网络 h, 定义其经验误差 $L_E(h)$ 为

$$L_E(h) = \frac{1}{N}\sum_{i=1}^{N} \ell(h(\tilde{\boldsymbol{y}}_i), \tilde{\boldsymbol{x}}_i), \tag{11.23}$$

其中, $h(\tilde{\boldsymbol{y}}_i)$ 表示展开网络以观测数据 $\tilde{\boldsymbol{y}}_i$ 为输入时输出的重建结果, $\tilde{\boldsymbol{x}}_i$ 为对应的清晰图像, $\ell(\cdot)$ 表示逐样本损失函数, N 表示训练样本数量. 同理, 总体误差 $L_P(h)$ 为

$$L_P(h) = \mathbb{E}[\ell(h(\boldsymbol{y}), \boldsymbol{x})]. \tag{11.24}$$

于是, 泛化误差定义为

$$\mathrm{Gen}(h) = |L_P(h) - L_E(h)|. \tag{11.25}$$

Behboodi 等[196] 讨论了一类模型驱动展开网络的泛化误差行为. 与 LISTA 直接假设目标信号本身稀疏不同, 该工作假设待恢复信号在某个正交字典下具有稀疏表示. 对应的展开网络与 LISTA 相近, 但字典参数由数据学习并在各层之间共享, 因此其泛化分析还依赖于测量矩阵和字典结构的相应条件. 其代表性结论可概括为如下定理.

定理 11.3 设 $\boldsymbol{A}$ 的谱范数有界且 $\|\boldsymbol{x}^*\|_2$ 有界, 则 K 阶段的 LISTA 展开网络的泛化误差在高概率下满足

$$\mathrm{Gen}(h) \le O\left(\sqrt{\frac{mn\log K + n^2 \log K}{N}}\right). \tag{11.26}$$

该定理揭示了 LISTA 类展开网络的泛化误差仅与展开网络阶段数 K 呈对数关系, 而传统神经网络的泛化误差随层数呈指数增长, 这说明在相应假设条件下, 深度展开网络在泛化行为上具有一定优势. 需要指出的是, 式 (11.26) 仍然显式依赖维度 m 和 n, 且尚未直接刻画稀疏度对泛化能力的影响, 因此该类结果仍有进一步细化的空间. 此外, Schnoor 等[197] 的研究进一步验证, 软阈值非线性操作在保障深度展开网络泛化性能中起着关键作用. 通过合理选取参数, 泛化误差会随网络层数增加而衰减, 而标准 ReLU 网络并不存在此特性.

11.3.4 近期进展

相关研究热点从结构构造与收敛加速, 逐渐延伸至训练优化、稳定性约束、初始化策略及理论分析等方向. Shah 等[198] 从训练优化角度, 分析了采用平滑软阈值算子的有限层 LISTA 展开网络, 在过参数化、适当初始化和网络宽度条件下, 经验损失在初始化邻域内可满足修改后的 Polyak-Łojasiewicz 性质, 从而为梯度下降达到低训练误差提供理论依据. Hadou 等[199] 通过逐层下降约束构造随机下降展开网络, 分析了其在未见样本上的收敛性与鲁棒性. Kouni 等[200] 将延续策略 (continuation) 用于关键结构量的热启动初始化, 以改善过参数化深度展开的泛化表现. Chen 等[201] 则通过引入历史梯度信息增强网络各阶段间的信息传递效率, 提出深度记忆展开网络 (DeMUN), 进一步提升了反问题求解性能. 此外, Sucker 等[202] 首次利用 PAC (probably approximately correct) 贝叶斯理论, 为深度展开提供了严格的泛化性能保证. Sambharya 等[203] 在此基础上给出了更紧的泛化误差界, 并提出了数据驱动的计算方法. 值得说明的是, 现有研究结论多聚焦于 LISTA 类展开网络, 针对更一般结构展开网络的研究仍较为匮乏, 相关理论支撑也有待完善.

11.4 结构学习型方法

结构学习型方法不再局限于学习迭代参数, 而是将近端映射、变换算子及正则结构等设计为可学习的模块, 从而增强了深度展开对复杂图像先验的表达能力.

11.4.1 基于 ISTA 的深度展开

Zhang 等[141] 通过卷积神经网络 (convolutional neural network, CNN) 构造可学习的非线性变换及其逆变换, 以替代传统 ISTA 中依赖固定稀疏变换和软阈值算子的近端映射. 该方法称为 ISTA-Net, 其第 k 个阶段迭代表达式为

$$\begin{cases} \boldsymbol{r}^{k+1} = \boldsymbol{x}^k - \rho^k \boldsymbol{A}^T(\boldsymbol{A}\boldsymbol{x}^k - \boldsymbol{y}), \\ \boldsymbol{x}^{k+1} = \widetilde{\mathcal{F}}^k(\mathcal{S}_{\theta^k}(\mathcal{F}^k(\boldsymbol{r}^{k+1}))), \end{cases} \tag{11.27}$$

其中, ρ^k 为可学习步长, $\mathcal{F}^k(\cdot)$ 与 $\widetilde{\mathcal{F}}^k(\cdot)$ 分别表示前向变换与逆变换模块, θ^k 为可学习阈值参数. 由式 (11.27) 可知, ISTA-Net 的变化并不在于放弃 ISTA 的基本迭代框架, 而在于将 ISTA 中的显式近端映射替换为由可学习变换、阈值收缩与逆变换共同构成的阶段模块. ISTA-Net 的整体展开形式如图 11.2 所示, 记号略有不同.

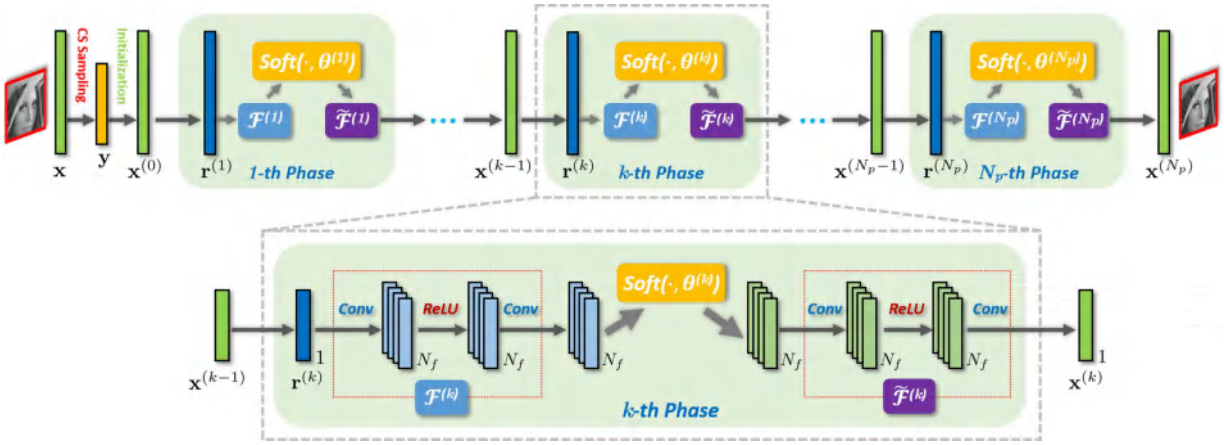


图 11.2: ISTA-Net 展开网络结构示意图[141]

对比 LISTA 与 ISTA-Net 不难发现, LISTA 主要针对基础的稀疏恢复任务, 核心是学习线性迭代参数以加快算法收敛. 而 ISTA-Net 则针对图像压缩感知重建任务, 核心是将近端映射本身网络化, 以增强对复杂图像先验的表达能力. 此外, 在网络结构设计方面, 两者差异同样显著. LISTA 采用全连接网络架构, ISTA-Net 则选用卷积神经网络, 因此 ISTA-Net 不仅运算效率更高, 在泛化性能上也更具优势.

ISTA-Net 为后续结构学习型方法提供了重要借鉴, 在此基础上, 各类改进版本不断涌现. 例如, FISTA-Net[204] 利用 FISTA 中的动量外推思想, 以提升展开网络的收敛速度和重建效率.

ISTA-Net++[142] 针对不同采样率的重建需求, 引入动态展开以增强模型的多采样率适应能力. OPINE-Net[205] 则将采样矩阵学习与图像重建过程联合建模, 进一步拓展了 ISTA 类展开方法在压缩感知成像中的应用形式. 尽管这些改进方法取得了优异的性能表现, 但相关的收敛性分析与泛化性探讨仍较为匮乏.

11.4.2 基于 ADMM 的深度展开

基于式 (11.7) 所示的 ADMM 迭代更新规则, Yang 等[206] 针对压缩感知磁共振成像 (magnetic resonance imaging, MRI) 重建任务, 将 ADMM 的迭代过程展开为多阶段神经网络, 提出了深度 ADMM 网络 (DeepADMM-Net). 每个迭代步骤对应网络中的一个阶段, 其单阶段更新过程可概括为

$$\begin{cases} \boldsymbol{x}^{k+1} = \mathcal{R}^k(\boldsymbol{z}^k, \boldsymbol{u}^k, \boldsymbol{y}), \\ \boldsymbol{z}^{k+1} = \mathcal{S}_{\theta^k}(\boldsymbol{W}^k\boldsymbol{x}^{k+1} + \boldsymbol{u}^k), \\ \boldsymbol{u}^{k+1} = \boldsymbol{u}^k + \eta^k(\boldsymbol{W}^k\boldsymbol{x}^{k+1} - \boldsymbol{z}^{k+1}), \end{cases} \tag{11.28}$$

其中, $\mathcal{R}^k(\cdot)$ 表示重建模块, $\boldsymbol{W}^k$ 表示可学习变换算子, θ^k 和 η^k 分别表示可学习阈值参数和对偶更新参数. 相较于式 (11.7) 的迭代, DeepADMM-Net 不再固定变换算子、收缩函数和惩罚参数, 而是通过端到端训练对这些模块进行学习, 既保留了变量分裂框架又增强了展开网络的表达能力.

Shultzman 等[207] 表明 ADMM 展开网络的泛化误差上界并不只由展开阶段数决定, 还受到各阶段权重矩阵范数、训练样本规模、数据分布以及软阈值的共同影响. 借助局部 Rademacher 复杂度, 给出了网络的泛化误差界理论.

定理 11.4 *考虑 k 阶段 ADMM 展开网络函数类 $\mathcal{H}^k$. 设训练集由 m 个独立同分布样本构成, 损失函数满足相应的 Lipschitz 连续性假设, 且各阶段权重矩阵范数有界, 则其泛化误差满足*

$$G(\mathcal{H}^k) \le 2\widetilde{B}_k \mathbb{E}_S G^{k-1}, \tag{11.29}$$

其中, $G(\mathcal{H}^k)$ 表示函数类 $\mathcal{H}^k$ 的泛化误差上界, $\widetilde{B}_k$ 表示由第 k 阶段权重范数及 ADMM 更新参数共同决定的上界系数, $\mathbb{E}_S$ 表示对训练样本集 S 取期望, G^{k-1} 表示由前 $k-1$ 个阶段递推得到的复杂度上界项.

自 DeepADMM-Net 提出以来, ADMM 展开思想被广泛拓展至多个图像反问题领域. Yang 等[123] 将此思想用于压缩感知重建, 构建了基于 ADMM 的压缩感知网络 (ADMM-CSNet), 在保持模型可解释性的同时提升了压缩感知图像重建质量和推理效率. Kouni 等[208] 将 ADMM 展开与解析稀疏模型相结合, 通过联合学习分析字典增强压缩感知重建中的先验表达, 形成了基于 ADMM 的深度分析字典网络 (ADMM-DADNet), 从而提升了分析稀疏模型在不同信号恢复任务中的重构表现. An 等[209] 提出微分方程启发的加速线性化 ADMM 展开网络, Hao 等[210]

提出长短期残差 ADMM 压缩感知网络 (LSRA-CSNet).

总而言之, 基于 ADMM 的深度展开方法的主要特点在于继承了变量分裂和交替优化的模块化结构. 相比主要围绕单变量梯度步和阈值映射展开的 ISTA-Net, 这种 ADMM 展开网络更适合处理复合线性算子、复杂约束和多变量先验结构. 但受限于 ADMM 算法引入乘子变量的迭代更新, 该类展开网络易出现参数规模偏大的问题.

11.4.3 基于 PDHG 的深度展开

在深度展开方法中, 原始-对偶框架代表了另一类重要的技术路线. 相应地, 第 k 个阶段的原始-对偶展开网络可抽象写为

$$\begin{cases} \boldsymbol{z}^{k+1} = \Gamma_d^k(\boldsymbol{z}^k, \mathcal{A}(\bar{\boldsymbol{x}}^k), \boldsymbol{y}), \\ \boldsymbol{x}^{k+1} = \Gamma_p^k(\boldsymbol{x}^k, \mathcal{A}^*(\boldsymbol{z}^{k+1}), \boldsymbol{y}), \\ \bar{\boldsymbol{x}}^{k+1} = \boldsymbol{x}^{k+1} + \omega_k(\boldsymbol{x}^{k+1} - \boldsymbol{x}^k), \end{cases} \tag{11.30}$$

其中, $\Gamma_d^k(\cdot)$ 与 $\Gamma_p^k(\cdot)$ 分别表示对偶更新模块和原始更新模块, ω_k 表示可学习或可调的外推系数. 与式 (11.10) 所示的 Chambolle-Pock 型 PDHG 迭代相比, 主要区别在于, PDHG 中的 $\mathrm{prox}_{\sigma F^*}$ 和 $\mathrm{prox}_{\tau G}$ 是由优化模型确定的解析近端映射, 而展开网络中的 Γ_d^k 和 Γ_p^k 则由卷积网络、残差模块或其他可学习结构参数化.

相关工作最早可追溯到 Vogel 等[211] 提出的面向低层视觉问题的原始-对偶网络. 在图像反问题求解领域, Adler 等[212] 提出的学习型原始-对偶重建方法 (learned primal-dual, LPD) 是该方向的代表性工作. 该方法针对 CT (computed tomography) 重建任务, 将原始-对偶算法展开为有限阶段的深度网络, 并将原始空间与对偶空间中的更新映射替换为卷积神经网络模块, 同时在网络内部嵌入前向算子及其伴随算子, 从而保留成像物理模型的信息. LPD 的更新过程可写为

$$\begin{cases} \boldsymbol{\alpha}^{k+1} = \Gamma_{\vartheta_k^d}(\boldsymbol{\alpha}^k, \mathcal{A}(\boldsymbol{\beta}^k), \boldsymbol{y}), \\ \boldsymbol{\beta}^{k+1} = \Lambda_{\vartheta_k^p}(\boldsymbol{\beta}^k, \mathcal{A}^*(\boldsymbol{\alpha}^{k+1})), \end{cases} \tag{11.31}$$

其中, $\boldsymbol{\alpha}^k$ 和 $\boldsymbol{\beta}^k$ 表示对偶特征和原始特征, $\Gamma_{\vartheta_k^d}(\cdot)$ 和 $\Lambda_{\vartheta_k^p}(\cdot)$ 分别表示对偶更新网络和原始更新网络. 值得说明的是, 由于该类方法同时涉及原始空间、对偶空间及线性算子作用下的多变量耦合更新, 引入可学习参数后的理论分析变得更为复杂, 相关理论仍有进一步完善空间.

11.5 生成式驱动型方法

典型的生成式方法包括变分自编码器 (variational autoencoder, VAE)、生成对抗网络 (generative adversarial network, GAN)、流模型以及扩散模型[213]. 与前述结构学习型展开方法不同,

生成式先验更注重对训练图像的整体数据分布与隐空间结构, 并将其用于修正中间重建结果. 从研究发展历程来看, 生成式模型最初多作为先验直接参与图像反问题求解, 随后逐步嵌入深度展开结构, 形成了以流模型与扩散模型为主的深度展开方法.

11.5.1 生成式先验的反问题求解

考虑如下数学模型

$$\hat{\boldsymbol{x}} = \underset{\boldsymbol{x}}{\operatorname{argmin}} \frac{1}{2}\|\boldsymbol{A}\boldsymbol{x} - \boldsymbol{y}\|_2^2 + \mathcal{R}_\Theta(\boldsymbol{x}), \tag{11.32}$$

其中, $\mathcal{R}_\Theta(\boldsymbol{x})$ 表示由生成式模型学习得到的图像先验项, Θ 为参数.

基于变分自编码器和生成对抗网络的方法, 通过隐变量表示或生成映射约束待恢复图像的可行解空间, 从而降低反问题的搜索维度. 例如, Bora 等[214] 将压缩感知重建问题转化为生成器隐变量的搜索问题, Peng 等[215] 则从自编码器的表示角度探讨了反问题的求解思路, González 等[216] 进一步从贝叶斯后验最大化视角出发, 将自编码器先验引入反问题求解过程, 并通过图像变量与隐变量刻画生成式先验约束. 与之不同, 归一化流模型 (normalizing flow, NF) 借助可逆映射特性与可计算的概率密度, 图像先验可通过 $-\log p_\Theta(\boldsymbol{x})$ 嵌入最大后验估计 (maximum a posteriori, MAP) 目标函数中, 进而获得清晰的统计意义解释[217-218]. 此外, 扩散模型通过逐步加噪与反向去噪过程刻画图像分布, 可在反问题求解中作为生成式先验引导重建结果逼近真实图像分布. 此类方法通常利用预训练的扩散模型, 在采样、去噪或迭代校正过程中融入数据一致性约束, 实现对退化图像的恢复. 典型代表包括扩散恢复模型 (DDRM)[219]、扩散后验采样 (DPS)[220]以及去噪扩散零空间模型 (DDNM)[221].

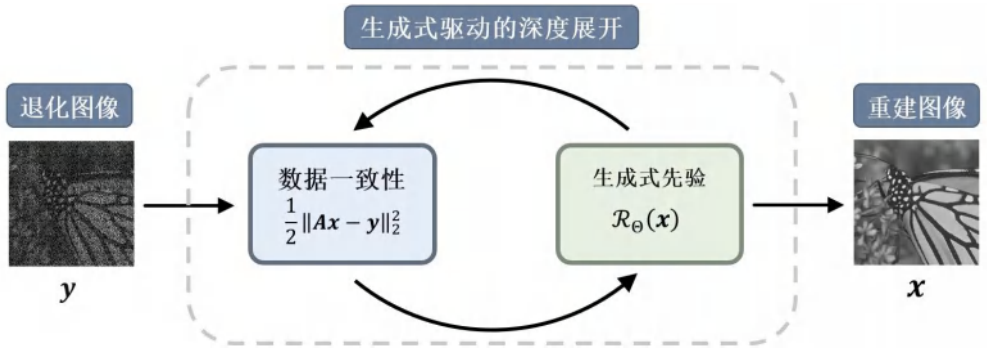


图 11.3: 生成式驱动深度展开方法求解反问题的流程

然而, 生成式模型的表示范围与生成流形约束会直接影响反问题的求解效果, 过度依赖隐空间可能带来表示误差[222]. 从贝叶斯反问题角度看, 生成式先验对真实分布的近似误差, 会进一步影响重建结果的稳定性[223]. 因此, 若仅将生成式先验作为外部约束使用, 虽能提升图像分布的表达能力, 但难以充分挖掘观测模型、迭代更新与生成式先验三者之间的内在关联. 深度展开方法恰好提供了一种阶段化建模框架, 其具体求解流程可归纳为图 11.3.

11.5.2 基于流模型先验的深度展开

Wei 等[224] 首次将深度展开与归一化流模型相结合并应用于图像去噪任务, 随后针对更通用的线性反问题, 提出了基于归一化流模型先验的深度展开方法 (NF-Unfolding)[225]. 将归一化流模型嵌入至展开近端梯度算法中, 使得近端步骤不再局限于普通卷积网络的近似拟合, 而是通过可逆生成式模型提供显式的概率先验. 设归一化流模型的正向映射与逆映射分别表示为

$$\boldsymbol{\zeta} = f_\psi(\boldsymbol{x}),\ \boldsymbol{x} = g_\psi(\boldsymbol{\zeta}), \tag{11.33}$$

其中, $\boldsymbol{x}$ 表示图像空间变量, $\boldsymbol{\zeta}$ 表示归一化流模型的隐空间变量, ψ 表示模型的可学习参数. 由于归一化流模型具有可逆性, 其概率密度函数可通过变量变换公式推导得出, 即

$$\log p_\psi(\boldsymbol{x}) = \log p_\zeta(\boldsymbol{\zeta}) + \log \left| \det \frac{\partial f_\psi(\boldsymbol{x})}{\partial \boldsymbol{x}} \right|. \tag{11.34}$$

在高斯噪声观测模型下, 引入归一化流模型先验后的最大后验估计问题可表示为

$$\hat{\boldsymbol{x}} = \underset{\boldsymbol{x}}{\operatorname{argmin}} \frac{1}{2\sigma_n^2} \|\boldsymbol{A}\boldsymbol{x} - \boldsymbol{y}\|_2^2 - \log p_\psi(\boldsymbol{x}). \tag{11.35}$$

NF-Unfolding 并未在图像空间中直接求解, 而是利用归一化流模型在图像空间与隐空间之间的可逆映射特性, 将近端更新操作转化至隐空间中执行, 有效降低了计算复杂度.

- 第一步: 第 k 个展开阶段, 执行数据一致性更新

$$\tilde{\boldsymbol{x}}^{k+1} = \boldsymbol{x}^k - \mu^k \boldsymbol{A}^T(\boldsymbol{A}\boldsymbol{x}^k - \boldsymbol{y}), \tag{11.36}$$

  其中, μ^k 为可学习步长.

- 第二步: 将得到的中间结果映射到隐空间, 即

$$\tilde{\boldsymbol{\zeta}}^{k+1} = f_\psi^{k+1}(\tilde{\boldsymbol{x}}^{k+1}), \tag{11.37}$$

  并在隐空间中执行收缩形式的近端更新

$$\boldsymbol{\zeta}^{k+1} = \frac{\tilde{\boldsymbol{\zeta}}^{k+1}}{1 + \kappa^{k+1}}, \tag{11.38}$$

  其中, κ^{k+1} 为可学习收缩参数.

- 第三步: 通过归一化流模型的逆映射将优化后的隐变量映射回图像空间, 得到该阶段的最终重建结果

$$\boldsymbol{x}^{k+1} = g_\psi^{k+1}(\boldsymbol{\zeta}^{k+1}). \tag{11.39}$$

通过上述多阶段迭代更新过程, 即可构建基于归一化流模型先验的深度展开, 其整体结构如图 11.4 所示, 记号略有不同. 其中, 数据一致性步骤负责约束重建结果满足观测模型, 流模型

近端步骤则先将中间重建结果映射到隐空间并进行收缩更新, 再通过逆映射回到图像空间.

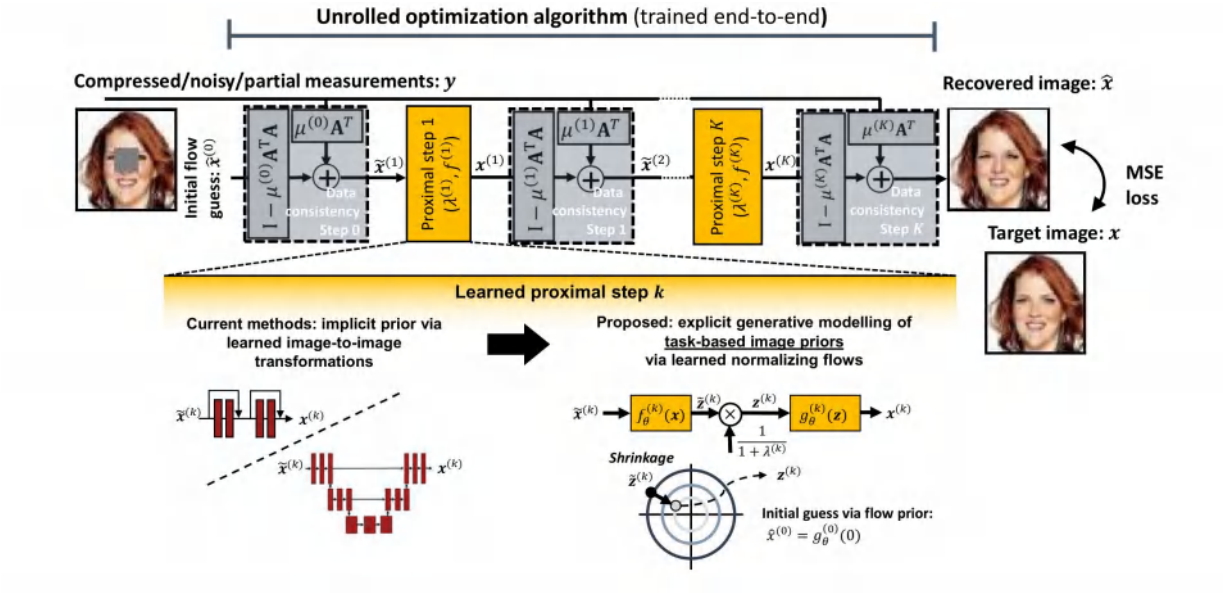


图 11.4: NF-Unfolding 框架示意图[225]

除归一化流模型外, Ai 等[226] 针对高光谱图像压缩重建任务, 通过条件流匹配生成式先验表示, 并借助先验引导的 Transformer 结构将其注入去噪模块, 从而增强空间-光谱细节恢复能力. 综上, 基于流模型先验的深度展开方法, 能够在保留深度展开结构固有可解释性的基础上, 充分利用流模型对复杂图像分布的表达能力, 有效提升深度展开网络的效果.

11.5.3 基于扩散模型先验的深度展开

Liao 等[227] 针对图像压缩感知任务, 将预训练扩散模型引入深度展开架构, 提出了基于扩散消息传递的深度展开网络 (diffusion message passing-based deep unfolding network, DMP-DUN). 设 k 为扩散反向过程中的时间步长 (类似于迭代数), 其迭代更新形式可表示为

$$\begin{cases} \boldsymbol{s}^k = \boldsymbol{x}^k - \sqrt{\alpha_k}\boldsymbol{A}^T(\boldsymbol{A}\boldsymbol{x}^k - \boldsymbol{y}), \\ \boldsymbol{r}^k = \mathcal{D}_k(\boldsymbol{s}^k + \sqrt{\alpha_k}o_k(\boldsymbol{u}^{k+1}, \boldsymbol{h}^{k+1})), \\ \boldsymbol{x}^{k-1} = \mathcal{P}_{\theta,k}(\boldsymbol{r}^k), \end{cases} \tag{11.40}$$

其中, $\boldsymbol{s}^k$ 表示经过测量一致性校正后的中间变量, $\boldsymbol{r}^k$ 表示送入预训练扩散模型的中间状态, $\mathcal{D}_k(\cdot)$ 表示高斯滤波校正算子, $o_k(\boldsymbol{u}^{k+1}, \boldsymbol{h}^{k+1})$ 表示与 Onsager 校正相关的项, $\mathcal{P}_{\theta,k}(\cdot)$ 表示由预训练扩散模型在时间步 k 实现的一步反向扩散更新. 因此, DMP-DUN 中的图像估计按照

$$\boldsymbol{x}^k \to \boldsymbol{s}^k \to \boldsymbol{r}^k \to \boldsymbol{x}^{k-1} \tag{11.41}$$

逐步更新, 即先由观测模型进行数据一致性校正, 再形成扩散模型输入状态, 最后由预训练扩散模型完成一步反向去噪更新.

为增强模型对复杂场景的适应, 采用可学习步长参数 η_k 代替原来的固定系数 $\sqrt{\alpha_k}$, 即

$$\boldsymbol{s}^k = \boldsymbol{x}^k - \eta_k \boldsymbol{A}^T(\boldsymbol{A}\boldsymbol{x}^k - \boldsymbol{y}). \tag{11.42}$$

对于难以直接计算的算子 $\mathcal{D}_t(\cdot)$ 和 Onsager 相关项 $\boldsymbol{o}_t(\cdot)$, 采用轻量残差模块进行近似拟合, 即

$$\{\boldsymbol{r}^k, \boldsymbol{v}^{k-1}\} = \mathcal{H}_{\text{Conv}}(\mathcal{H}_{\text{4-Res}}(\mathcal{H}_{\text{Conv}}(\{\boldsymbol{s}^k, \boldsymbol{v}^k\}))), \tag{11.43}$$

其中, $\mathcal{H}_{\text{Conv}}$ 表示卷积层, $\mathcal{H}_{\text{4-Res}}$ 表示由四个残差结构组成的轻量网络模块, $\boldsymbol{v}^k$ 表示阶段间传递的特征图. 随后, $\boldsymbol{r}^k$ 被送入预训练扩散模型, 完成一步反向扩散更新. 图 11.5 给出了 DMP-DUN 的阶段展开结构及整体图像重建流程, 记号略有不同.

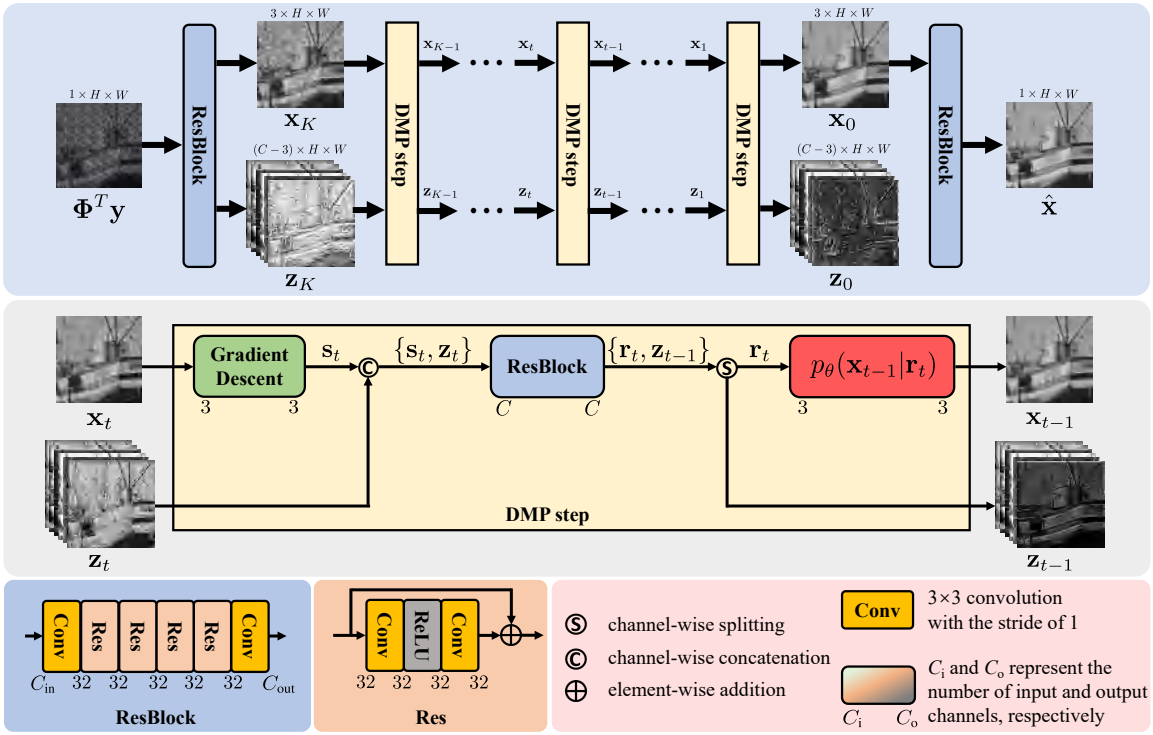


图 11.5: DMP-DUN 框架示意图[227]

近期, Wu 等[228] 针对快照光谱压缩成像任务, 利用隐空间扩散模型生成无退化先验, 并通过三分支 Transformer 将其与空间-光谱特征结合, 提出了隐空间扩散先验增强深度展开方法 (LADE-DUN). Zheng 等[229] 通过生成式先验模型获得紧凑先验表示, 再利用扩散模型推断该表示并嵌入迭代重建过程, 提出了图像压缩感知的生成式先验扩散深度展开算法 (GPD-CS). Wang 等[230] 进一步将物理测量算子显式引入模块化网络结构, 提出了基于扩散先验与后验评分学习的深度展开算法 (Diff-Unfolding).

需要指出的是, 扩散模型与深度展开的融合策略虽能有效提升图像重建性能, 却存在不可忽视的局限性. 预训练扩散模型的引入会带来高昂的训练开销, 大幅增加计算成本. 与之不同, 流模型依托轻量化的单一去噪流程, 可有效压缩训练时长.

11.6 数值实验

本节对图像压缩感知重建中的代表性方法进行对比, 包括迭代方法 FISTA[1]、ADMM-BPDN[2], 参数学习型方法 LISTA[3], 结构学习型方法 ISTA-Net+[4]、ADMM-CSNet[5]、LDAMP[6], 生成式驱动型方法 DMP-DUN+[7], 以及生成式模型方法 ICTM[8]、DDNM[9].

所有实验均在 Python 环境下完成, RTX PRO 6000 GPU (96 GB 显存) 和 25 vCPU Intel(R) Xeon(R) Platinum 8470Q 处理器.

11.6.1 实验设置

为保障对比实验的公平性, 所有对比方法均采用相同的测试图像、采样率及采样矩阵. 对于所有深度展开方法, 训练数据参考 ISTA-Net 的公开数据集, 共计 88,912 张尺寸为 33×33 的自然图像块, 图像块向量化后维度为 $n = 1,089$. 测试集选用 11 张经典的灰度自然图像, 如 Monarch、Barbara 等. 对于阶段数、去噪器或扩散时间步等, 均采用公开代码或原文推荐配置.

11.6.2 实验结果

表 11.1 汇总了各类对比方法在 50%、40%、25%、10% 四种采样率下的峰值信噪比 (peak signal-to-noise ratio, PSNR)、各采样率的平均指标及平均重建耗时, 表中最优结果以加粗形式标注. 需要说明的是, 迭代方法考虑 CPU 运行时间, 深度学习方法考虑 GPU 推理时间.

由实验结果可知, 深度学习方法相较于迭代优化方法在重建精度上整体具有明显优势. 其中, FISTA 与 ADMM-BPDN 的平均 PSNR 分别为 26.54 dB 和 26.46 dB, 在 10% 低采样率下均下降到约 21 dB 左右. 与之相比, LISTA 的平均 PSNR 提升至 28.84 dB, 在 10% 采样率下达到 24.96 dB, 验证了通过训练数据学习迭代参数能够改善迭代方法的重建效果. 进一步, ADMM-CSNet 的平均 PSNR 为 31.38 dB, 平均 GPU 重建时间为 0.0197 s, 这表明将近端映射网络化后不仅增强了图像结构表达能力, 还能提高网络的计算效率.

一个有意思的发现, 单纯将生成式模型直接引入图像反问题求解框架, 并不一定带来性能优势. 以流模型驱动的 ICTM 为例, 其平均 PSNR 仅为 29.73 dB, 重建精度低于 ISTA-Net+、

[1] https://github.com/jeankossaifi/fista

[2] https://github.com/bwohlberg/sporco

[3] https://github.com/jianzhangcs/ISTA-Net-PyTorch

[4] https://github.com/jianzhangcs/ISTA-Net-PyTorch

[5] https://github.com/yangyan92/Pytorch_ADMM-CSNet

[6] https://github.com/ricedsp/D-AMP_Toolbox

[7] https://github.com/FengodChen/DMP-DUN-CVPR2025

[8] https://github.com/YasminZhang/ICTM

[9] https://github.com/wyhuai/DDNM

表 11.1: 不同采样率下 PSNR 对比、平均及耗时

方法	采样率				平均	CPU/GPU
	50%	40%	25%	10%		
FISTA	30.81	28.74	25.48	21.13	26.54	0.0611/—
ADMM-BPDN	30.81	28.74	25.49	20.81	26.46	0.2520/—
LISTA	31.34	30.66	28.41	24.96	28.84	—/0.0258
ISTA-Net+	36.82	35.17	31.40	26.39	32.44	—/0.0219
ADMM-CSNet	35.41	33.35	30.86	25.90	31.38	—/**0.0197**
LDAMP	38.54	36.87	33.00	25.09	33.38	—/0.1003
ICTM	33.84	32.07	28.73	24.27	29.73	—/0.0473
DDNM	35.56	33.48	29.61	22.95	30.40	—/0.1423
DMP-DUN+	**41.86**	**39.83**	**36.41**	**30.93**	**37.26**	—/0.9141

ADMM-CSNet 等主流深度展开模型. 相比之下, DMP-DUN+ 在四个采样率下均取得最高 PSNR, 平均 PSNR 达到 37.26 dB. 该结果说明, 生成式模型的优势并不只是来自先验模型本身, 更关键的是将生成式先验嵌入具有明确数据一致性结构的深度展开框架中, 使先验校正和数据一致性更新能够协同作用. 尤其在 10% 低采样率下, DMP-DUN+ 达到 30.93 dB, 较 FISTA 高出 9.80 dB, 甚至比 50% 采样率下的 FISTA 还高一些. 需要注意的是, DMP-DUN+ 的平均重建时间达到 0.9141 s, 高于其他对比方法.

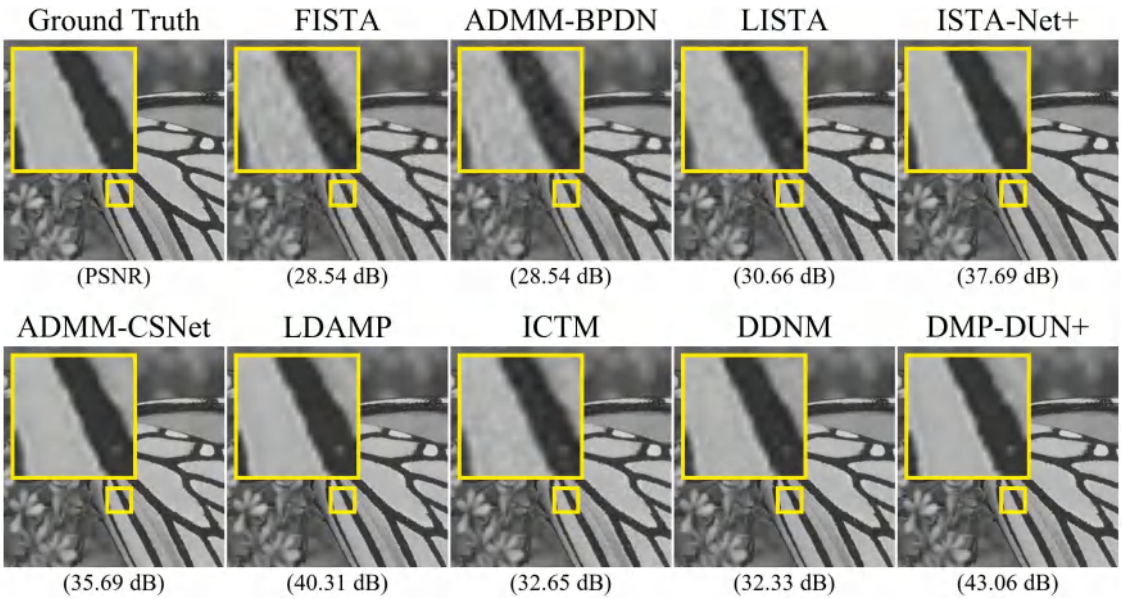


图 11.6: Monarch 图像在 50% 采样率下的重建可视化对比

11.6.3 可视化分析

为观察不同方法在图像压缩感知重建上的差异, 本节分别选取 50% 采样率的 Monarch 图像与 25% 采样率的 Barbara 图像, 开展可视化对比分析. 从图 11.6 可以看出, FISTA 与 ADMM-

BPDN 的局部区域较为模糊, 翅膀边缘和细小纹理难以完整恢复. ISTA-Net+ 与 ADMM-CSNet 虽有所改善, 但局部细节仍不够清晰. DMP-DUN+ 在放大区域中呈现出更清晰的边缘过渡和局部纹理, 说明将生成式先验嵌入展开结构后有助于改善在低采样率下的细节恢复质量. 图 11.7 展示了各方法在规则纹理场景下的差异. Barbara 图像的围巾区域包含密集条纹, 对高频纹理恢复要求较高. FISTA、ADMM-BPDN 及 LISTA 均出现一定纹理模糊, DDNM 在局部放大区域中也存在明显的纹理失真. 而 DMP-DUN+ 的周期纹理更接近真实图像, 进一步验证了该方法在图像恢复任务中的优势.

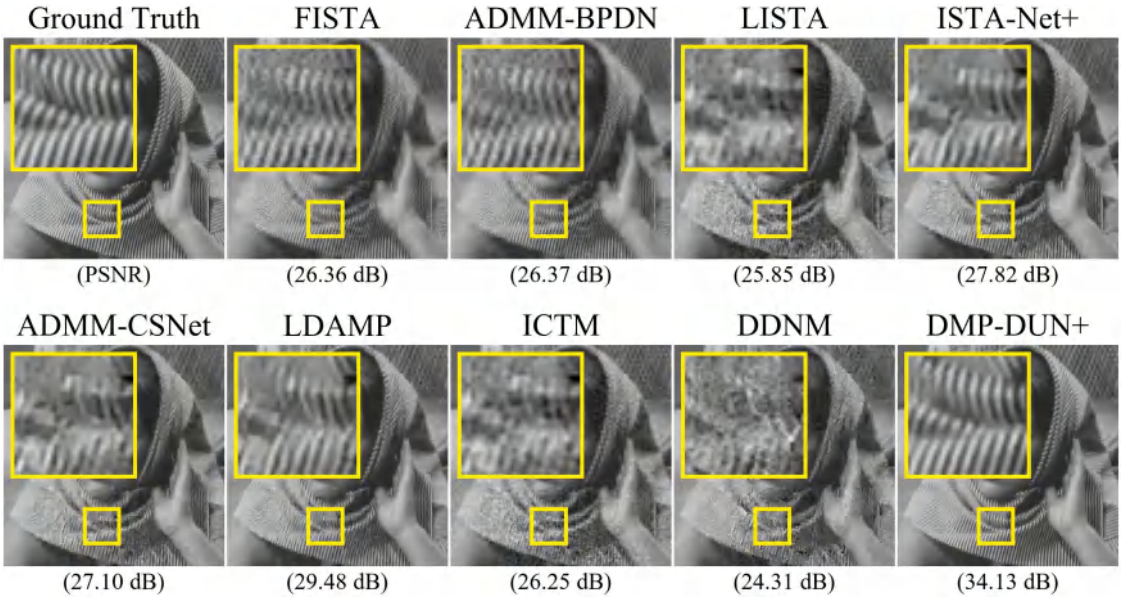


图 11.7: Barbara 图像在 25% 采样率下的重建可视化对比

进一步, 图 11.8 展示了 Barbara 图像在 25% 采样率下, 不同展开阶段数对应的 DMP-DUN+ 重建结果. 可以看出, 当展开阶段数从 1 增至 4 时, 图像重建质量显著提升, PSNR 指标由 28.82 dB 提高至 34.13 dB. 但若继续增加展开阶段数, 重建性能反而出现下降. 事实上, 深度展开通常仅需设置 3–6 个展开阶段即可取得最优重建效果, 相比之下, 传统迭代算法往往需要数百次甚至上千次迭代才能收敛.

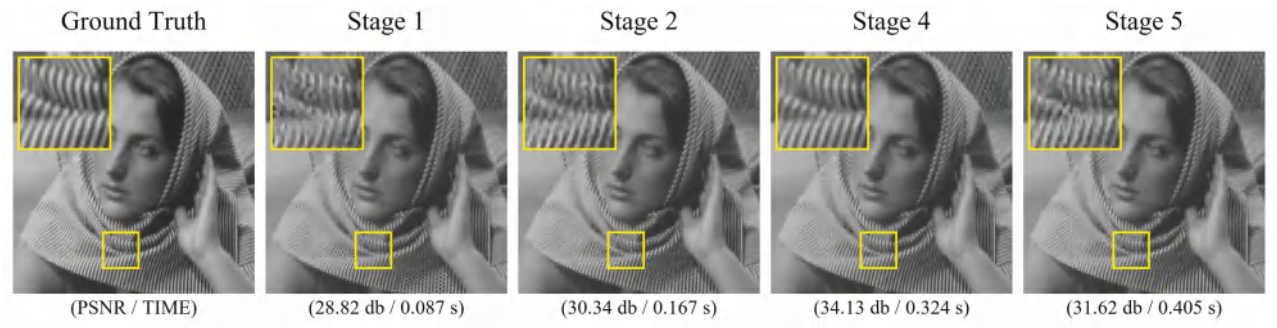


图 11.8: 25% 采样率下 DMP-DUN+ 不同展开阶段数的重建结果对比

11.6.4 参数量分析

除图像重建精度外, 模型参数量也是衡量深度展开方法实际部署的重要因素. 如图 11.9 所示, 横轴表示平均重建时间, 纵轴表示平均 PSNR, 气泡面积对应模型的参数量规模. 由对比结果可见, LISTA 的参数量偏高, 且平均 PSNR 低于后续结构学习型展开方法. 相比之下, ISTA-Net+ 与 ADMM-CSNet 则能够以轻量化模型规模, 获得较高的 PSNR 和较快的推理速度. 而 DMP-DUN+ 分布于图表右上角, 表明生成式先验的引入虽能获得最优重建精度, 但也伴随参数量与推理耗时激增的问题.

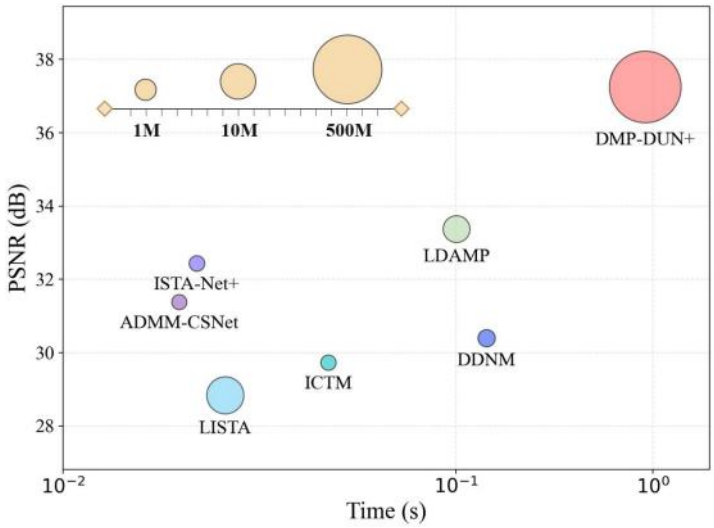


图 11.9: 重建性能、时间开销与参数量比较

11.7 本章小结

本章针对图像处理中最基本的反问题, 从参数学习型、结构学习型和生成式驱动型三个方面, 系统梳理了相关方法的基本思想、代表性模型和实验表现. 综合来看, 深度展开方法通过将优化算法的迭代过程转化为有限阶段的可训练网络, 有效平衡了模型可解释性、重建质量与计算效率的需求. 未来, 图像反问题的深度展开求解仍有多个方向值得深入研究.

(1) 优化理论分析　现有理论结果多集中于 LISTA 等结构相对清晰的模型, 对于含卷积近端模块、扩散先验或流模型先验展开网络, 相关理论十分匮乏. 因此, 亟需建立深度展开更一般的收敛性、复杂性、误差界等理论, 为图像反问题实际应用提供理论支撑.

(2) 二阶算法设计　半光滑牛顿驱动展开网络等近期工作表明, 二阶优化算法同样具备作为深度框架的潜力[231]. 相较于一阶算法, 二阶算法可以带来更快的收敛速度. 因此, 未来可尝试将半光滑牛顿、子空间牛顿等融入深度展开, 从而提升求解效率和收敛性能.

(3) 网络轻量化　将扩散模型引入深度展开, 虽能有效提升图像重建质量, 但也会大幅增加计算开销与模型复杂度. 因此, 应进一步探索基于流匹配等快速生成策略的深度展开融

合方案, 并结合参数高效微调技术, 研发兼具高性能与轻量化的架构.

(4) 视觉模型融合　Transformer、Mamba 及视觉语言模型等新型视觉模型, 为增强展开网络的特征表征能力提供了全新思路. 例如, Song 等[232] 将 cross-attention Transformer 嵌入压缩感知展开过程, Chen 等[233] 则利用 CLIP 先验实现医学图像恢复和分割.

第 12 章 基于大语言模型的优化问题求解方法

优化是支撑复杂系统决策问题的核心方法, 在数学、计算机和管理等学科中具有重要的应用价值. 传统优化方法高度依赖专家经验, 在面对大规模和多约束问题时往往难以高效求解. 近年来, 大语言模型 (large language models, LLMs) 的快速发展为突破上述瓶颈提供了新的研究范式. 本章旨在对大语言模型驱动的优化方法进行综述. 首先, 介绍了优化问题的定义和大语言模型的原理; 在此基础上, 重点分析了大语言模型在模型构建、算法设计和方案验证等优化问题求解流程中的作用.

12.1 引言

大语言模型凭借其超大规模的参数体系和高质量的训练数据, 在多任务场景下表现出强大的泛化能力, 标志着人工智能向通用智能迈出了重要步伐. 近年来, 大语言模型在数学推理、代码编程、算法设计和数据分析等多个复杂领域中展现出广阔的发展前景, 为人工智能辅助科学 (AI for Science) 研究提供了全新思路. 在此背景下, 大语言模型驱动的运筹优化得到了学术界和工业界的广泛关注, 涌现出一批代表性工作, 如图 12.1 所示. 大语言模型不仅减少了解决问题所需的人力, 还提高了解决方案的效率, 在一定程度上重塑了运筹优化问题的研究路径.

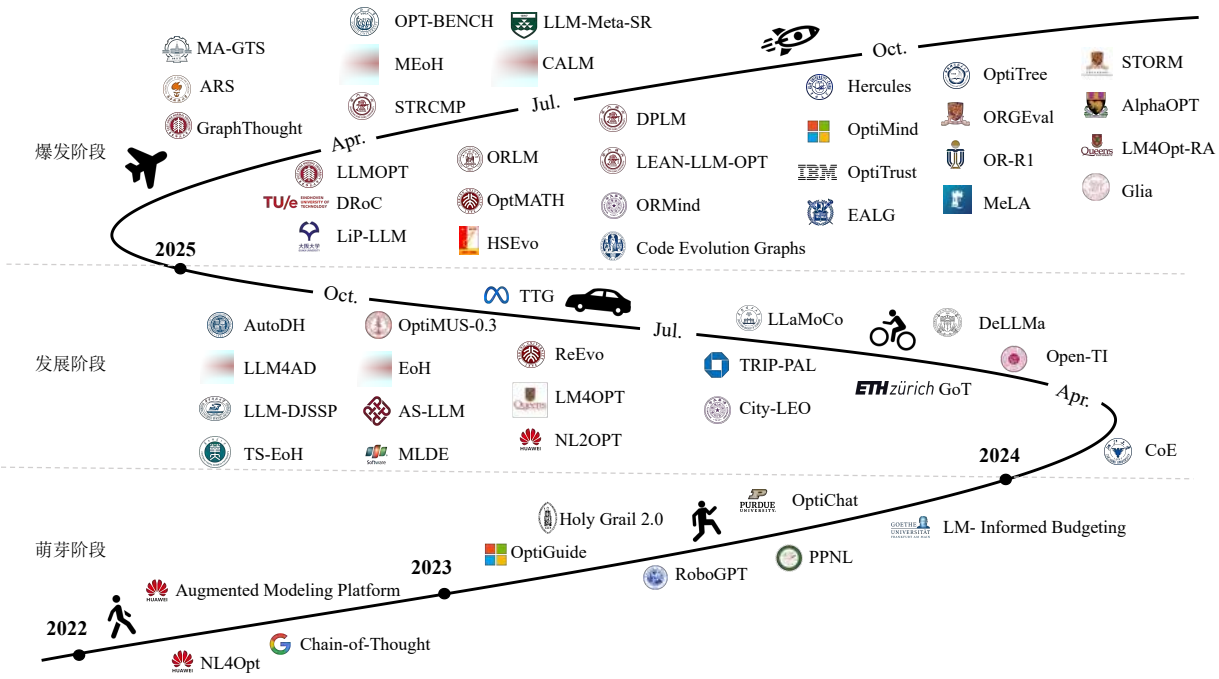


图 12.1: 时间发展图 (2022-2025)

优化问题的求解流程可归纳为三个步骤, 即模型构建、算法设计和方案验证, 如图 12.2 所示. 下文将具体分析大语言模型在各流程中的功能.

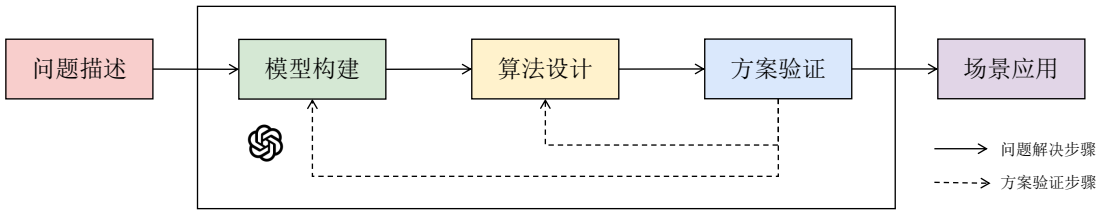


图 12.2: 优化问题求解流程

12.2 模型构建

模型构建是优化过程的基础环节, 目标在于将自然语言描述的问题形式化为数学模型. 所构建模型的质量在很大程度上影响后续算法选择和求解效率, 并最终决定结果的准确性. 大语言模型能够从自然语言中提取问题背景、目标函数和约束条件, 并转化为数学公式表达. 围绕该方向逐渐形成了两类主要研究路径: 提示方法和学习方法.

12.2.1 提示方法

提示方法通过设计专业的提示词, 并利用多代理框架、检索增强生成 (retrieval-augmented generation, RAG) 等技术, 在无需依赖大规模数据进行参数调整的情况下, 引导大语言模型更有效地完成任务, 是自然语言处理中的一种典型研究范式. 该方法充分利用了大语言模型在预训练过程获取的通用知识, 在面对任务时仅凭提示词就能展现出强大的零样本或少样本能力. 提示方法可以分为思维框架、代理框架和知识框架, 如图 12.3 所示.

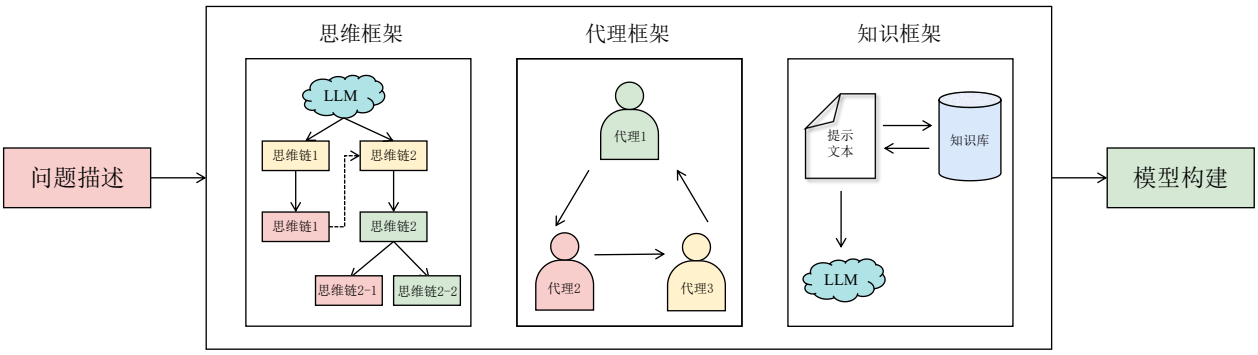


图 12.3: 提示方法三类框架示意图

(1) 思维框架

思维框架旨在引导大语言模型进行有效推理, 从而提出更合适的解决方案, 包括思维链、思维树和思维图. 虽然这些方法最初是为通用任务设计的, 但也已成功地应用于模型构建. 受 NL4Opt 竞赛启发, Tsouros 等[234] 开发了从数学建模到求解程序的完整框架. Wang 等[235] 提出了一种结合思维链方法的递归动态温度参数策略, 能够获得更优的可行解. Liu 等[236] 构建了依据问题层次和复杂程度进行分类的建模思维树, 将复杂问题分解为一系列的简单子问题.

(2) 代理框架

大语言模型的代理框架通过将不同任务分配给由大语言模型扮演的多个代理 (如公式器、代码器和评估器), 使得每个代理能够独立工作又互相配合. 为有效处理长描述和复杂数据问题而避免冗长的提示词, Ahmaditeshnizi 等[237] 设计了模块化代理框架 OptiMUS 0.2. 通过几个代理之间的交叉验证并使用连接图独立处理目标函数和每个约束条件, 显著提升了大语言模型数学建模的性能, 该研究的工作流程如图 12.4 所示. Hao 等[238] 设计了多代理框架 LLMFP, 在 2 个大语言模型在 9 个任务上验证了有效性. 在多代理框架的基础上, 研究人员引入更多机制来提高大语言模型建模的能力. Xiao 等[239] 采用前向思维构建和后向反思机制, 提出了多代理框架 CoE, 用来协调代理间的任务. Mostajabdaveh 等[240] 通过多个代理的投票机制改进决策过程, 提出了代理之间通过协作和竞争来对结果进行验证的策略, 从而增强了模型对复杂问题的建模能力. Astorga 等[241] 结合蒙特卡洛树搜索 (Monte Carlo tree search, MCTS) 开发了多代理 Autoformulator, 并引入剪枝技术删除琐碎的等价公式. 针对代理框架高计算延迟与协同能力不足, Berto 等[242] 设计了并行自回归框架 PARCO, 包含 Transformer 通信层、多指针机制以及基于优先级冲突处理, 在路径规划、取送货和车间调度等任务上展现出较好的性能.

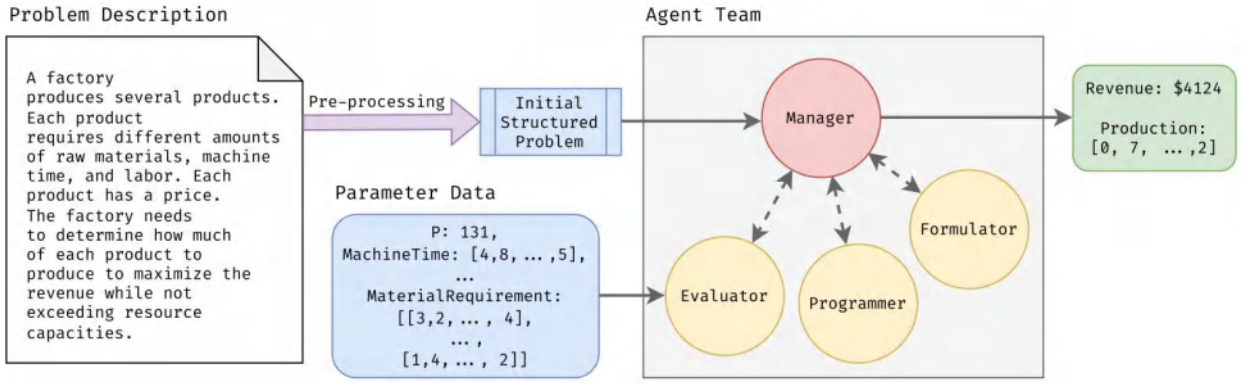


图 12.4: OptiMUS 工作流程[237]

(3) 知识框架

检索增强生成技术, 通过引入外部知识库为大语言模型提供额外的信息支撑, 显著提升优化求解的性能. 基于先前的多代理框架并引入检索增强生成技术, Jiang 等[243] 提出了约束分解检索框架 DROC, 能够分解复杂约束进而降低建模复杂度. Peng 等[244] 通过将本地大语言模型与特定领域的知识库结合, 设计的框架在飞机表皮制造案例中成功应用, 同时确保了数据隐私和计算效率. 在结合少量样本提示和多代理协作机制的基础上, Liang 等[245] 引入检索增强生成技术提出了 LEAN-LLM-OPT, 能够高效解决包含长文本描述和外部数据输入的复杂优化问题, 在新加坡航空公司的实际业务中得到了验证.

12.2.2 学习方法

与提示方法不同, 学习方法通过在预训练大语言模型的基础上进一步引入监督微调、强化学习和参数高效微调等策略, 使模型能够有效地适应特定任务和领域需求, 如图 12.5 所示.

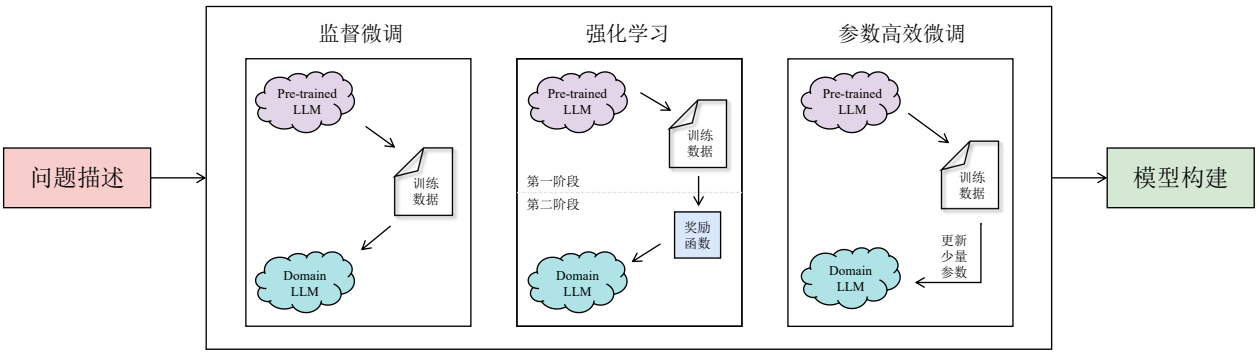


图 12.5: 学习方法三类框架示意图

(1) 监督微调

在监督微调范式下, 研究者围绕数据构建和训练策略展开了大量探索. Amarasinghe 等[246] 提出的 AI Copilot 使用监督微调后的大语言模型进行建模, 通过设计 9 个子模块并应用提示工程策略缓解大语言模型的 Token 限制. 基于 NL4Opt 数据集, Li 等[247] 扩展了更多的问题描述和约束类型, 并利用扩展后的数据集对 ChatGPT 和 Google Bard 进行监督微调, 提升了建模准确率. Masoud 等[248] 的研究展示了 GPT-3.5 Turbo 微调模型在旅行商问题的有效性, 并采用自集成方法对解决方案改进. 除直接扩展数据集外, Yang 等[249] 设计了从数学模型到问题描述的反向数据合成方法 ReSocratic, 主要利用生成的数据集对多个开源模型进行训练. 类似地, Huang 等[250] 在 ORLM 中也介绍了其设计的数据合成方法 OR-INSTRUCT, 从扩展与增强两种策略构建用于微调的数据集: 前者通过扩展问题场景和类型提升数据覆盖范围, 后者通过改写目标函数与约束条件、重述问题和引入多种建模技术增强数据的多样性, 并在多个模型上性能取得提升. Ma 等[251] 在微调前引入对比学习 (contrastive learning, CL) 作为预热阶段提出了 LLaMoCo, 从而改善了微调过程的训练成本和稳定性.

(2) 强化学习

在监督微调的基础上, 引入强化学习策略以增强模型的鲁棒性. 针对大语言模型的幻觉现象, Jiang 等[252] 引入模型对齐 (Kahneman-Tversky optimization, KTO) 与自校正机制, 进而提出了 LLMOPT, 在涵盖健康、环境、能源和制造等 20 个领域的 6 个真实数据集上得到了验证. 无独有偶, Zhou 等[253] 提出的 DPLM 同样采用两阶段方法进行模型训练: 第一阶段使用监督微调, 第二阶段使用直接偏好优化 (direct preference optimization, DPO) 进行离线训练或使用组相对策略优化 (group relative policy optimization, GRPO) 进行在线训练, 在动态规划问题的建模任务中取得了较好的性能表现. 与之类似, Ding 等[254] 采用监督微调和测试时分组相对策略优化

(test-time group relative policy optimization, TGRPO) 的方法构建了 OR-R1, 能够引导大语言模型同时利用稀缺的标注数据和丰富的未标注数据进行有效训练.

(3) 参数高效微调

考虑到监督微调需要更新全量参数, 往往带来较高的时间开销和计算资源消耗, 因此参数高效微调得到了许多研究者的青睐, 其中 LoRA 微调是当下应用最广泛的参数高效微调方法. Lu 等[255] 通过采用抑制采样机制对合成数据进行验证与筛选的策略提出了 OptMATH, 从而使不同规模 (0.5B–32B) 的 LoRA 微调模型在多个任务中均表现出稳定的性能. Zhang 等[256] 借助外部求解器进行辅助决策提出了 OptLLM, 支持多轮交互式建模, 显著提升了 LoRA 微调后的 Qwen 模型的建模准确率. Wu 等[257] 通过复杂性进化和范围进化的双重策略指导大语言模型生成高质量多样化数据, 设计的 EVO-STEP-INDUCT 在 Llama-3-8B 与 Mistral-7B 的 LoRA 微调实验中展示了问题结构化验证和逐步改进机制相结合的有效性.

提示方法充分发挥了大语言模型的推理能力, 通过思维框架、代理框架和知识框架, 使大语言模型在无需或仅需极少训练的情况下即可完成优化问题的模型构建, 具有实现成本低和部署灵活等优势, 能够快速迁移至不同类型的优化任务. 学习方法则凭借高质量的数据合成策略和多种训练机制, 通过监督微调、强化学习和参数高效微调等策略, 驱动大语言模型从数据中学习建模规律, 从而提升准确率和稳定性.

12.3 算法设计

算法设计是优化研究的关键环节, 直接影响到问题求解的效率和精度. 传统方法通常依赖专家进行设计, 其性能在很大程度上受经验影响. 为解决上述局限, 研究者开始尝试将大语言模型引入自动算法设计过程, 以提升算法生成的效率和质量. 这一趋势使得优化算法从由经验驱动逐步迈向由大语言模型驱动生成, 不仅降低了算法开发的门槛, 也为构建通用的算法体系提供了新的思路, 在各种经典优化问题上性能表现优异, 甚至优于专家设计的算法, 同时具备较好的泛化性. 根据大语言模型在算法设计中所承担的角色, 可将其划分为三类: 评估者、优化者和设计者, 如图 12.6 所示. 本章主要回顾大语言模型在组合优化和连续优化两类问题算法设计中的应用进展.

12.3.1 组合优化

在组合优化问题中, 大语言模型能够改进搜索策略, 还可与传统优化方法融合, 从而在算法设计中发挥重要作用.

(1) 评估者

评估者, 指的是大语言模型作为优化过程中的辅助模块, 用于进行算法评估和选择等环节.

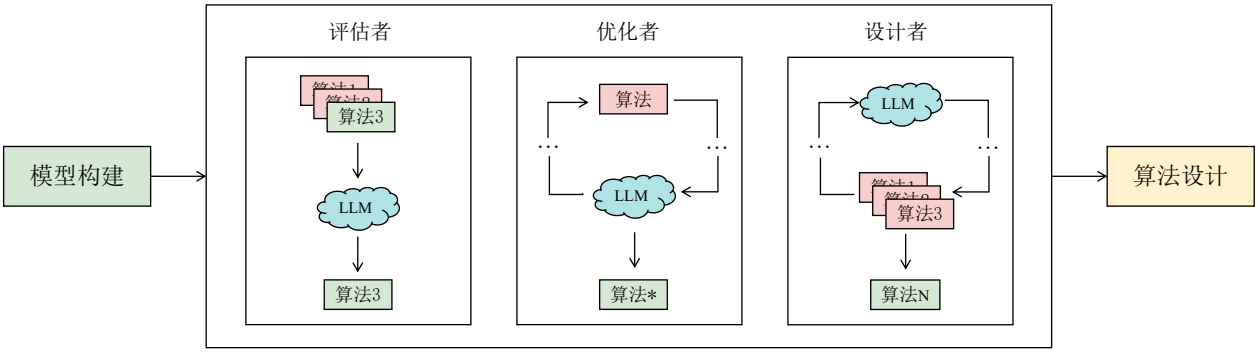


图 12.6: 算法设计三类角色示意图

Nie 等[258] 设计了基于大语言模型的求解器, 从历史优化轨迹中合成方向性反馈 (类似于传统优化方法中的一阶导数信息), 以在迭代过程中实现可靠的改进. Wu 等[259] 提出了算法选择框架 AS-LLM, 引导大语言模型捕捉算法的结构和语义, 通过给定问题和不同算法间的匹配程度确定算法. 类似地, Li 等[260] 结合图神经网络与大语言模型设计了 STRCMP, 使得大语言模型能够从组合优化问题实例中提取结构嵌入, 从而识别高性能的算法. 此外, Wu 等[261] 提出了少样本性能预测提示框架 Hercules, 对大语言模型生成算法与已有算法间的语义相似度进行分析, 实现了对启发式算法性能的快速评估.

(2) 优化者

优化者, 指的是大语言模型能够主动搜索、变异或交叉已有算法. Mao 等[262] 引导大语言模型对人工设计的节点评分函数进化, 生成的新函数在识别网络关键节点的决策任务展现出良好的适应性. Yu 等[263] 利用大语言模型和进化算法协同, 开发的 AutoRNet 用以设计鲁棒网络. 也有研究引入微调等技术提升大语言模型改进算法的能力. 例如, Zhang 等[264] 采用了监督微调和从编译器反馈中进行强化学习 (reinforcement learning from compiler feedback, RLCF) 的两阶段训练方式. Surina 等[265] 则在结合进化搜索的基础上结合强化学习微调, 驱动大语言模型改进算法. Sartori 等[266] 的两项研究展示了大语言模型在提升已有算法性能上的潜力, 在 10 个经典算法 (包括元启发式、确定性和精确算法等) 的实验中进行了验证.

(3) 设计者

设计者, 指的是大语言模型直接设计算法. Romera 等[267] 通过将预训练语言模型与系统评估器结合, 设计的 FunSearch 能够实现函数空间的搜索, 在帽集问题 (cap set problem, CSP) 和在线装箱问题 (online bin packing problem, OBPP) 上取得了不错的效果, 工作流程如图 12.7 所示. 此外, Liu 等[268] 提出了一种结合大语言模型与进化计算的框架 EOH, 旨在驱动模型自主设计算法及其代码. 随后, 该框架在多个领域得到扩展, 如 Yao 等[269] 提出的 MEOH 也在旅行商问题和在线装箱问题中展示了有效性. Ye 等[270] 结合进化搜索和大语言模型反思机制设计了 ReEvo, 能够在启发式空间中进行高效探索. Dat 等[271] 设计的 HSEvo 则在算法稳定性与种群多样性之间实现了平衡, 相较于 EOH 展现出更高的稳定性, 而在多样性上优于 FunSearch 和

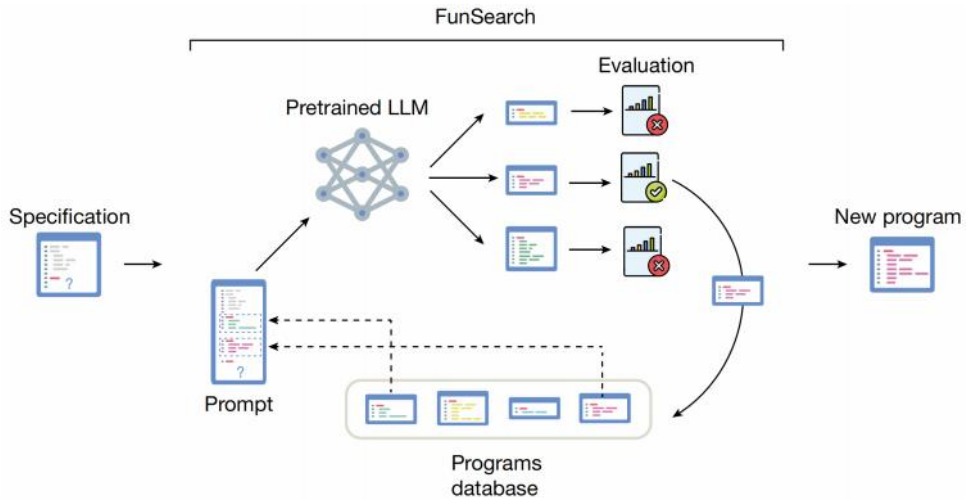


图 12.7: FunSearch 工作流程[267]

ReEvo. 为大语言模型提供统一的算法设计接口, Liu 等[272] 提出了 LLM4AD, 该框架能够生成多类任务的求解算法. Yu 等[273] 对算法设计中个体表示、变异算子和适应度评估三个关键组件进行了深入分析, 并结合大语言模型和进化算法以设计启发式算法. Shi 等[274] 提出了启发式元优化策略 MOH, 通过元学习自动构建多样化的算法, 在多种组合优化问题上展示了良好的性能. 此外, Huang[275] 等将语言引导和数值评估结合提出了 CALM, 并通过强化学习对大语言模型进行微调, 增强其生成启发式算法的质量.

12.3.2 连续优化

在连续优化领域, 大语言模型同样可以按照评估者、优化者和设计者的角色划分进行归纳. 在评估者中, Hao 等[276] 将模型辅助选择任务公式化为分类问题或回归问题, 利用大语言模型基于历史数据评估新解的质量, 进而提出了 LAEA. 部分工作围绕优化者角色进行研究. Wang 等[277] 在约束型多目标优化框架 CMOEA-LLM 中, 提出了利用提示工程将解的目标值与约束违反信息微调大语言模型的策略, 使模型能够判断候选解质量并生成更优解. Van 等[278] 设计的 LLaMEA 能够自动生成用于黑盒优化 (black-box optimization, BBO) 问题算法. Brahmachary 等[279] 提出了基于大语言模型的数值优化方法 LEO, 在超音速喷嘴形状优化、热传递和风场布局优化等多个工业工程问题上得到了可靠验证. 此外, Liu 等[280] 将大语言模型作为多目标进化算法的黑盒搜索算子提出了 MOEA 与 D-LMO, 驱动大语言模型生成候选解, 进一步设计出白盒算子以解释大语言模型的行为.

在算法结构确定之后, 参数配置在优化流程中同样至关重要, 直接影响算法的求解效率. 凭借对问题参数、约束条件和目标函数的理解能力, 大语言模型能够自动生成合理的参数方案, 并在反馈回路中持续改善配置策略[281]. 相关研究表明, 即使在搜索预算 (如算力、时间) 受限或数据稀缺的条件下, 该方法仍可取得与传统调参方法相当甚至更优的性能[282]. 总体而言, 大语言模型在算法设计阶段通过承担评估者、优化者与设计者等多重角色, 并扩展至参数配置

过程, 在一定程度上缓解了优化算法对人工经验的依赖, 在复杂组合优化与连续优化任务中展现出良好的泛化性.

12.4 方案验证

方案验证是确保模型构建、算法设计符合实际需求的关键环节. 与传统优化相比, 大语言模型输出的模型和算法虽然具备高度自动化, 但可能存在约束遗漏或逻辑不一致等问题. 因此, 建立高效的方案验证机制, 是保障大语言模型驱动的优化可靠性的关键. 当前研究主要包括外部验证、内部反思和代理协同三类方法, 如图 12.8 所示.

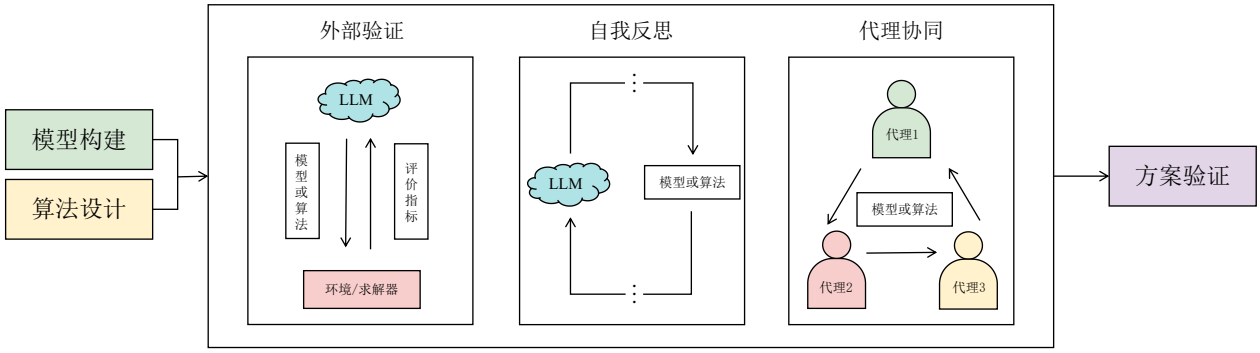


图 12.8: 三类验证方法示意图

12.4.1 外部验证

在外部验证阶段, 研究者普遍采用大语言模型与优化求解器的交互反馈或强化学习的奖励信号作为校验机制, 以评估生成方案. Chen 等[283] 聚焦于混合整数线性规划模型的可行性诊断, 借助大语言模型与 Gurobi 求解器交互识别不可约不可行的问题子集, 开发的 OptiChat 能够在建模层面完成自动纠错. 在研究的最新阶段, 外部验证机制进一步与强化学习深度结合. Ma 等[284] 通过引入一种新颖的类强化学习奖励机制设计了 AutoDH, 该框架能够综合权衡方案质量、时间成本以及调用大语言模型的经济开销, 从而自适应选择专家设计或大语言模型生成的启发式算法. Chen 等[285] 将外部优化求解器作为奖励信号的来源, 提出的 Solver-Informed RL 能够引导大语言模型在强化学习过程中能够依据验证反馈持续改进问题求解策略. 此外, Huang 等[286] 通过启发式引导的前向搜索或与求解器对齐的后向推理策略生成有效的推理序列, 用于求解图组合优化问题.

12.4.2 内部反思

内部反思方法致力于驱动大语言模型识别潜在错误, 并自动执行修复流程. Huang 等[287] 引入自我反思与调试的机制, 驱动大语言模型自动识别不一致的推理步骤并进行修正. Zhang

等[288] 提出的人工智能代理 OR-LLM-Agent, 能够在沙箱环境中实现代码执行到错误纠正的自主闭环. Fornies 等[289] 通过聚类分析与搜索空间反思相结合, 设计了多目标启发式反思进化框架 REMoH, 能够引导大语言模型生成多样化、高质量的启发式算法. Chen 等[290] 提出了 HeuriGym, 构建了由代码生成、环境执行、错误反馈的多轮迭代循环机制, 图 12.9 展示了该框架的工作流程.

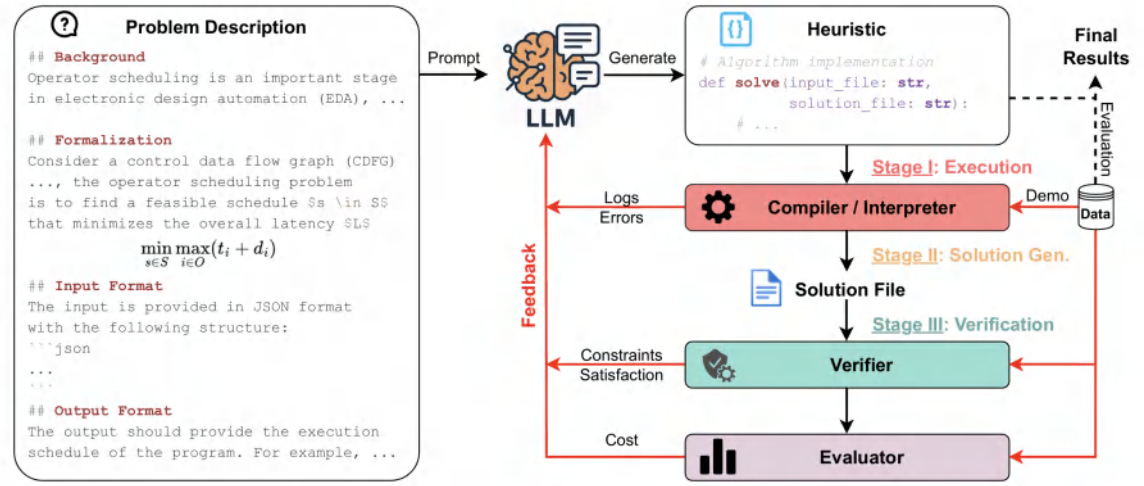


图 12.9: HeuriGym 工作流程[290]

12.4.3 代理协同

在多智能体系统中, 验证任务由专门代理负责, 作为自动方案评估和代码可行性检查的独立环节. 例如, Elhenawy 等[291] 在视觉推理场景中验证了多模态多代理系统在组合优化任务中的有效性. 通过知识集成和算法求解进行分层建模, Yuan[292] 等推出的 MA-GTS, 从文本逐步重构图结构, 能够自适应地调用优化算法. Yang 等[293] 设计的 HeurAgenix 框架, 验证了智能体能够评估该框架下生成、进化后的启发式算法. 从人类思考方式中得到启发, Wang[294] 等设计了 ORMind, 能够对大语言模型提出的求解方案进行多轮反思以达到最优. Lima 等[295] 利用多代理投票进行交叉验证的机制提出了 OptiTrust, 构建了完整的方案评估流程.

总体而言, 现有方案验证方法主要包括基于求解器的外部验证、基于模型一致性检查的内部反思, 以及基于多智能体分工的代理协同, 共同构成了大语言模型驱动优化的验证体系.

12.5 本章小结

本章系统梳理了大语言模型在优化领域中的最新研究进展, 重点探讨了其在模型构建、算法设计和方案验证中的关键作用. 总体而言, 大模型驱动的优化研究前景广阔, 期望本章内容能为相关领域研究者提供有价值的参考与启示. 然而, 该领域仍存在一些挑战性问题. 为此, 本章对其进一步的分析, 并展望了未来的研究趋势.

(1) 新框架探索　尽管大语言模型在求解已有优化问题方面展现出一定能力, 但在求解准确率和稳定性方面仍存在不足, 某些复杂数据集的准确率低于 50%. 此外, 现有研究多局限于中小规模数据集或局部性能提升, 难以处理百万变量工业级问题. 探索面向复杂决策问题的全流程新框架, 正逐渐成为大语言模型驱动优化的重要研究方向.

(2) 高质量数据集　大语言模型在面对数据规模庞大和任务复杂度较高的问题时, 往往会出现性能急剧下降甚至严重幻觉的问题. 现有的基准数据集主要来源于教科书, 难以充分反映现实世界中更加复杂和类型多样的优化场景. 因此, 构建高质量并贴近真实应用的基准数据集, 有助于提升大语言模型在运筹优化任务中的性能.

(3) 数据隐私安全　工业数据通常具有高度敏感性, 如交通物流中的道路信息和金融投资中的交易记录, 这对基于云端的大语言模型在方案建模和应用开发构成了显著制约. 由于数据外传可能引发隐私泄露, 探索不依赖云端大语言模型架构和隐私安全方法, 是推动大语言模型在优化领域实际落地的关键.

(4) 多模态大模型　优化问题不仅涉及到文本刻画, 还可能包含图像、表格等多种数据形式. 而依赖单一语言模型的方案在处理多类数据方面仍存在一定局限, 从而限制了其在规划任务中的表现. 设计基于多模态大模型的应用框架, 有望为复杂环境中的优化决策提供更高效的解决思路.

参考文献

[1] WRIGHT J, MA Y. High-Dimensional Data Analysis with Low-Dimensional Models: Principles, Computation, and Applications[M]. Cambridge University Press, 2022.

[2] TILLMANN A M, BIENSTOCK D, LODI A, et al. Cardinality minimization, constraints, and regularization: A survey[J]. SIAM Review, 2024, 66(3): 403-477.

[3] BOUMAL N. An Introduction to Optimization on Smooth Manifolds[M]. Cambridge University Press, 2023.

[4] ABSIL P A, MAHONY R, SEPULCHRE R. Optimization Algorithms on Matrix Manifolds[M]. Princeton University Press, 2008.

[5] CHEN W, JI H, YOU Y. An augmented *L*agrangian method for ℓ_1-regularized optimization problems with orthogonality constraints[J]. SIAM Journal on Scientific Computing, 2016, 38(4): B570-B592.

[6] CHEN H, SUN Y, GAO J, et al. Solving partial least squares regression via manifold optimization approaches [J]. IEEE Transactions on Neural Networks and Learning Systems, 2019, 30(2): 588-600.

[7] XIAO N, LIU X, YUAN Y X. Exact penalty function for $\ell_{2,1}$ norm minimization over the Stiefel manifold[J]. SIAM Journal on Optimization, 2021, 31(4): 3097-3126.

[8] BRELOY A, KUMAR S, SUN Y, et al. Majorization-minimization on the Stiefel manifold with application to robust sparse PCA[J]. IEEE Transactions on Signal Processing, 2021, 69: 1507-1520.

[9] LI Z, NIE F, BIAN J, et al. Sparse PCA via $\ell_{2,p}$-Norm Regularization for Unsupervised Feature Selection[J]. IEEE Transactions on Pattern Analysis and Machine Intelligence, 2023, 45(4): 5322-5328.

[10] ZHOU Y, BAO C, DING C, et al. A semismooth Newton based augmented Lagrangian method for nonsmooth optimization on matrix manifolds[J]. Mathematical Programming, 2023, 201(1-2): 1-61.

[11] HUANG W, WEI M, GALLIVAN K A, et al. A Riemannian optimization approach to clustering problems[J]. Journal of Scientific Computing, 2025, 103(1): 8.

[12] QU W, CHEN H, XIU X, et al. Distributed sparsity constrained optimization over the Stiefel manifold[J]. Neurocomputing, 2024, 602: 128267.

[13] ZEBARI R, ABDULAZEEZ A, ZEEBAREE D, et al. A comprehensive review of dimensionality reduction techniques for feature selection and feature extraction[J]. Journal of Applied Science and Technology Trends, 2020, 1(1): 56-70.

[14] HE X, CAI D, NIYOGI P. Laplacian score for feature selection[C]//Advances in Neural Information Processing Systems: vol. 18. 2005: 507-514.

[15] YANG Y, SHEN H T, MA Z, et al. $\ell_{2,1}$-norm regularized discriminative feature selection for unsupervised learning[C]//IJCAI International Joint Conference on Artificial Intelligence. 2011: 1589-1594.

[16] LIU Y, YE D, LI W, et al. Robust neighborhood embedding for unsupervised feature selection[J]. Knowledge-Based Systems, 2020, 193: 105462.

[17] NIE F, ZHU W, LI X. Unsupervised feature selection with structured graph optimization[C]//Proceedings of the AAAI Conference on Artificial Intelligence: vol. 30: 1. 2016.

[18] GREENACRE M, GROENEN P J, HASTIE T, et al. Principal component analysis[J]. Nature Reviews Methods Primers, 2022, 2(1): 100.

[19] ZOU H, XUE L. A selective overview of sparse principal component analysis[J]. Proceedings of the IEEE, 2018, 106(8): 1311-1320.

[20] NIE F, TIAN L, WANG R, et al. Learning feature-sparse principal subspace[J]. IEEE Transactions on Pattern Analysis and Machine Intelligence, 2023, 45(4): 4858-4869.

[21] ZHENG J, ZHANG X, LIU Y, et al. Fast sparse PCA via positive semidefinite projection for unsupervised feature selection[J]. arXiv:2309.06202, 2023.

[22] GAO Y, WU Q, XU Z, et al. Principal component analysis with fuzzy elastic net for feature selection[J]. IEEE Transactions on Fuzzy Systems, 2024, 32(12): 6878-6890.

[23] ZHU Y, ZHANG X, WEN G, et al. Double sparse-representation feature selection algorithm for classification [J]. Multimedia Tools and Applications, 2017, 76: 17525-17539.

[24] JAIN P, KAR P, et al. Non-convex optimization for machine learning[J]. Foundations and Trends in Machine Learning, 2017, 10(3-4): 142-363.

[25] ZHOU S, XIU X, WANG Y, et al. Revisiting $L_q(0 \leq q < 1)$ norm regularized optimization[J]. arXiv:2306.14394, 2023.

[26] BECK A. First-Order Methods in Optimization[M]. Philadelphia, PA: SIAM, 2017.

[27] CAO W, SUN J, XU Z. Fast image deconvolution using closed-form thresholding formulas of L_q ($q = \frac{1}{2}, \frac{2}{3}$) regularization[J]. Journal of Visual Communication and Image Representation, 2013, 24(1): 31-41.

[28] BOLTE J, SABACH S, TEBOULLE M. Proximal alternating linearized minimization for nonconvex and nonsmooth problems[J]. Mathematical Programming, 2014, 146(1): 459-494.

[29] TIAN L, NIE F, WANG R, et al. Learning feature sparse principal subspace[J]. Advances in Neural Information Processing Systems, 2020, 33: 14997-15008.

[30] AOUEDI O, VU T H, SACCO A, et al. A survey on intelligent Internet of Things: Applications, security, privacy, and future directions[J]. IEEE Communications Surveys & Tutorials, 2025, 27(2): 1238-1292.

[31] TU J, YANG L, CAO J. Distributed machine learning in edge computing: Challenges, solutions and future directions[J]. ACM Computing Surveys, 2025, 57(5): 1-37.

[32] YALLI J S, HASAN M H, JUNG L T, et al. A systematic review for evaluating IoT security: A focus on authentication, protocols and enabling technologies[J]. IEEE Internet of Things Journal, 2025, 12(11): 18908-18928.

[33] IERACITANO C, ADEEL A, MORABITO F C, et al. A novel statistical analysis and autoencoder driven intelligent intrusion detection approach[J]. Neurocomputing, 2020, 387: 51-62.

[34] ZULFIQAR Z, MALIK S U, MOQURRAB S A, et al. DeepDetect: An innovative hybrid deep learning framework for anomaly detection in IoT networks[J]. Journal of Computational Science, 2024, 83: 102426.

[35] NGUYEN D C, DING M, PATHIRANA P N, et al. Federated learning for Internet of things: A comprehensive survey[J]. IEEE Communications Surveys & Tutorials, 2021, 23(3): 1622-1658.

[36] ZHANG Y, SULEIMAN B, ALIBASA M J, et al. Privacy-aware anomaly detection in IoT environments using FedGroup: A group-based federated learning approach[J]. Journal of Network and Systems Management, 2024, 32(1): 20.

[37] ZHOU X, WU J, LIANG W, et al. Reconstructed graph neural network with knowledge distillation for lightweight anomaly detection[J]. IEEE Transactions on Neural Networks and Learning Systems, 2024, 35(9): 11817-11828.

[38] NGUYEN T A, LE L T, NGUYEN T D, et al. Federated PCA on Grassmann manifold for IoT anomaly detection [J]. IEEE/ACM Transactions on Networking, 2024, 32(5): 4456-4471.

[39] WANG K, SONG Z. High-dimensional cross-plant process monitoring with data privacy: A federated hierarchical sparse PCA approach[J]. IEEE Transactions on Industrial Informatics, 2024, 20(3): 4385-4396.

[40] LUO G, CHEN N, HE J, et al. Privacy-preserving clustering federated learning for non-IID data[J]. Future Generation Computer Systems, 2024, 154: 384-395.

[41] CHEN Y, FAN J, MA C, et al. Bridging convex and nonconvex optimization in robust PCA: Noise, outliers, and missing data[J]. Annals of Statistics, 2021, 49(5): 2948.

[42] LIU Z, LV H, LIU X, et al. Similarity and diversity: PCA-based contribution evaluation in federated learning[J]. IEEE Internet of Things Journal, 2025, 12(12): 20393-20405.

[43] XIU X, HUANG C, SHANG P, et al. Bi-sparse unsupervised feature selection[J]. IEEE Transactions on Image Processing, 2025, 34: 7407-7421.

[44] ZOU H, HASTIE T, TIBSHIRANI R. Sparse principal component analysis[J]. Journal of Computational and Graphical Statistics, 2006, 15(2): 265-286.

[45] ROUSSEEUW P J, HUBERT M. Anomaly detection by robust statistics[J]. Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery, 2018, 8(2): e1236.

[46] TRILLES S, HAMMAD S S, ISKANDARYAN D. Anomaly detection based on artificial intelligence of things: A systematic literature mapping[J]. Internet of Things, 2024, 25: 101063.

[47] CHEN S, MA S, XUE L, et al. An alternating manifold proximal gradient method for sparse principal component analysis and sparse canonical correlation analysis[J]. INFORMS Journal on Optimization, 2020, 2(3): 192-208.

[48] XIAO X, LI Y, WEN Z, et al. A regularized semi-smooth Newton method with projection steps for composite convex programs[J]. Journal of Scientific Computing, 2018, 76: 364-389.

[49] HAGER W W. Updating the Inverse of a Matrix[J]. SIAM Review, 1989, 31(2): 221-239.

[50] BOYD S, PARIKH N, CHU E, et al. Distributed optimization and statistical learning via the alternating direction method of multipliers[J]. Foundations and Trends in Machine learning, 2011, 3(1): 1-122.

[51] LI J, MA S, SRIVASTAVA T. A Riemannian alternating direction method of multipliers[J]. Mathematics of Operations Research, 2025, 50(4): 3222-3242.

[52] MCMAHAN B, MOORE E, RAMAGE D, et al. Communication-efficient learning of deep networks from decentralized data[C]//Artificial Intelligence and Statistics. 2017: 1273-1282.

[53] LI T, SAHU A K, ZAHEER M, et al. Federated optimization in heterogeneous networks[J]. Proceedings of Machine Learning and Systems, 2020, 2: 429-450.

[54] LI T, HU S, BEIRAMI A, et al. Ditto: Fair and robust federated learning through personalization[C]//International Conference on Machine Learning. 2021: 6357-6368.

[55] LEE D D, SEUNG H S. Learning the parts of objects by non-negative matrix factorization[J]. Nature, 1999, 401(6755): 788-791.

[56] WANG J Y, ALMASRI I, GAO X. Adaptive graph regularized nonnegative matrix factorization via feature selection[C]//Proceedings of the 21st International Conference on Pattern Recognition. 2012: 963-966.

[57] VIRTANEN T. Monaural sound source separation by nonnegative matrix factorization with temporal continuity and sparseness criteria[J]. IEEE Transactions on Audio, Speech, and Language Processing, 2007, 15(3): 1066-1074.

[58] MOTOKI S, KAZUYOSHI T, SHUNSUKE M, et al. Sparse modeling of EELS and EDX spectral imaging data by nonnegative matrix factorization[J]. Ultramicroscopy, 2016, 170: 43-59.

[59] WANG Y, GUAN T, ZHOU G, et al. SOJNMF: Identifying multidimensional molecular regulatory modules by sparse orthogonality-regularized joint non-negative matrix factorization algorithm[J]. IEEE/ACM Transactions on Computational Biology and Bioinformatics, 2021, 19(6): 3695-3703.

[60] LI X, YANG Y, ZHANG W. Fault detection method for non-Gaussian processes based on non-negative matrix factorization[J]. Asia-Pacific Journal of Chemical Engineering, 2013, 8(3): 362-370.

[61] LI X, YANG Y, ZHANG W. Statistical process monitoring via generalized non-negative matrix projection[J]. Chemometrics and Intelligent Laboratory Systems, 2013, 121: 15-25.

[62] WANG Y, YUAN S, LING D, et al. Fault monitoring based on adaptive partition non-negative matrix factorization for non-Gaussian processes[J]. IEEE Access, 2019, 7: 32783-32795.

[63] REN Z, ZHANG W, ZHANG Z. A deep nonnegative matrix factorization approach via autoencoder for nonlinear fault detection[J]. IEEE Transactions on Industrial Informatics, 2019, 16(8): 5042-5052.

[64] XIU X, FAN J, YANG Y, et al. Fault detection using structured joint sparse nonnegative matrix factorization[J]. IEEE Transactions on Instrumentation and Measurement, 2021, 70: 1-11.

[65] ASTERIS M, PAPAILIOPOULOS D, DIMAKIS A G. Orthogonal NMF through subspace exploration[C]// International Conference on Neural Information Processing Systems. 2015: 1-9.

[66] JIANG B, MENG X, WEN Z, et al. An exact penalty approach for optimization with nonnegative orthogonality constraints[J]. Mathematical Programming, 2023, 198(1): 855-897.

[67] ZHANG C, JING L, XIU N. A new active set method for nonnegative matrix factorization[J]. SIAM Journal on Scientific Computing, 2014, 36(6): A2633-A2653.

[68] PAN L, ZHOU S, XIU N, et al. A convergent iterative hard thresholding for sparsity and nonnegativity constrained optimization[J]. Pacific Journal of Optimization, 2017, 13(2): 325-353.

[69] ZHAI L, ZHANG Y, GUAN S, et al. Nonlinear process monitoring using kernel nonnegative matrix factorization [J]. The Canadian Journal of Chemical Engineering, 2018, 96(2): 554-563.

[70] LIU Y, ZENG J, XIE L, et al. Structured joint sparse principal component analysis for fault detection and isolation [J]. IEEE Transactions on Industrial Informatics, 2018, 15(5): 2721-2731.

[71] DING S X. Data-Driven Design of Fault Diagnosis and Fault-Tolerant Control Systems[M]. Springer London, 2014.

[72] DOWNS J J, VOGEL E F. A plant-wide industrial process control problem[J]. Computers & Chemical Engineering, 1993, 17(3): 245-255.

[73] WANG B, LEI Y, LI N, et al. A hybrid prognostics approach for estimating remaining useful life of rolling element bearings[J]. IEEE Transactions on Reliability, 2018, 69(1): 401-412.

[74] LI Y, YANG M, ZHANG Z. A survey of multi-view representation learning[J]. IEEE Transactions on Knowledge and Data Engineering, 2019, 31(10): 1863-1883.

[75] YANG X, LIU W, LIU W, et al. A survey on canonical correlation analysis[J]. IEEE Transactions on Knowledge and Data Engineering, 2021, 33(6): 2349-2368.

[76] WITTEN D M, TIBSHIRANI R, HASTIE T. A penalized matrix decomposition, with applications to sparse principal components and canonical correlation analysis[J]. Biostatistics, 2009, 10(3): 515-534.

[77] ZHANG L, ZHAO Y, ZHU Z, et al. Mining semantically consistent patterns for cross-view data[J]. IEEE Transactions on Knowledge and Data Engineering, 2014, 26(11): 2745-2758.

[78] LV K, CAI J, HUO J, et al. Sparse generalized canonical correlation analysis: Distributed alternating iteration-based approach[J]. Neural Computation, 2024, 36(7): 1380-1409.

[79] KUMAR D, MAJI P. Discriminative deep canonical correlation analysis for multi-view data[J]. IEEE Transactions on Neural Networks and Learning Systems, 2024, 35(10): 14288-14300.

[80] YANG T, YI X, WU J, et al. A survey of distributed optimization[J]. Annual Reviews in Control, 2019, 47: 278-305.

[81] LUO Y, TAO D, RAMAMOHANARAO K, et al. Tensor canonical correlation analysis for multi-view dimension reduction[J]. IEEE Transactions on Knowledge and Data Engineering, 2015, 27(11): 3111-3124.

[82] REDDY T S, CHEPURI S P. Two-view and multi-view tensor canonical correlation analysis over graphs[J]. IEEE Transactions on Signal and Information Processing over Networks, 2025, 11: 535-550.

[83] SUN J, XIU X, LUO Z, et al. Learning high-order multi-view representation by new tensor canonical correlation analysis[J]. IEEE Transactions on Circuits and Systems for Video Technology, 2023, 33(10): 5645-5654.

[84] LUO T, HOU C, NIE F, et al. Semi-supervised feature selection via insensitive sparse regression with application to video semantic recognition[J]. IEEE Transactions on Knowledge and Data Engineering, 2018, 30(10): 1943-1956.

[85] DU L, ZHANG J, LIU F, et al. Mining high-order multimodal brain image associations via sparse tensor canonical correlation analysis[C]//2020 IEEE International Conference on Bioinformatics and Biomedicine (BIBM). 2020: 570-575.

[86] LEE G, BU F, ELIASSI-RAD T, et al. A survey on hypergraph mining: Patterns, tools, and generators[J]. ACM Computing Surveys, 2025, 57(8): 1-36.

[87] WANG R, WANG P, WU D, et al. Multi-view and multi-order structured graph learning[J]. IEEE Transactions on Neural Networks and Learning Systems, 2024, 35(10): 14437-14448.

[88] CHEN S, MA S, MAN-CHO SO A, et al. Proximal gradient method for nonsmooth optimization over the Stiefel manifold[J]. SIAM Journal on Optimization, 2020, 30(1): 210-239.

[89] ABDI H, WILLIAMS L J. Principal component analysis[J]. Wiley Interdisciplinary Reviews: Computational Statistics, 2010, 2(4): 433-459.

[90] KE Q, KANADE T. Robust ℓ_1-norm factorization in the presence of outliers and missing data by alternative convex programming[C]//2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition: vol. 1. 2005: 739-746.

[91] NG A Y. Feature selection, ℓ_1 vs. ℓ_2 regularization, and rotational invariance[C]//Proceedings of the Twenty-first International Conference on Machine Learning. 2004: 78.

[92] WANG Q, GAO Q, GAO X, et al. $\ell_{2,p}$-norm based PCA for image recognition[J]. IEEE Transactions on Image Processing, 2018, 27(3): 1336-1346.

[93] CHEN T, KORNBLITH S, NOROUZI M, et al. A simple framework for contrastive learning of visual representations[C]//International Conference on Machine Learning. 2020: 1597-1607.

[94] ZHANG H, QIANG W, ZHANG J, et al. Unified feature extraction framework based on contrastive learning[J]. Knowledge-Based Systems, 2022, 258: 110028.

[95] ZHOU Q, GAO Q, WANG Q, et al. Sparse discriminant PCA based on contrastive learning and class-specificity distribution[J]. Neural Networks, 2023, 167: 775-786.

[96] BOILEAU P, HEJAZI N S, DUDOIT S. Exploring high-dimensional biological data with sparse contrastive principal component analysis[J]. Bioinformatics, 2020, 36(11): 3422-3430.

[97] ZHOU Q, WANG Q, GAO Q, et al. Unsupervised discriminative feature selection via contrastive graph learning [J]. IEEE Transactions on Image Processing, 2024, 33: 972-986.

[98] CHEN T, SUN Y, SHI Y, et al. On sampling strategies for neural network-based collaborative filtering[C]// Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. 2017: 767-776.

[99] WU Z, XIONG Y, YU S X, et al. Unsupervised feature learning via non-parametric instance discrimination[C] //Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. 2018: 3733-3742.

[100] XIU X, YANG A, HUANG C, et al. Enhancing unsupervised feature selection via double sparsity constrained optimization[J]. arXiv:2501.00726, 2025.

[101] ECKART C, YOUNG G. The approximation of one matrix by another of lower rank[J]. Psychometrika, 1936, 1(3): 211-218.

[102] CAI X, NIE F, HUANG H. Exact top-k feature selection via $\ell_{2,0}$-norm constraint[C]//Proceedings of the Twenty-Third International Joint Conference on Artificial Intelligence. 2013: 1240-1246.

[103] BLUMENSATH T, DAVIES M E. Iterative hard thresholding for compressed sensing[J]. Applied and Computational Harmonic Analysis, 2009, 27(3): 265-274.

[104] VAN DER MAATEN L, HINTON G. Visualizing data using t-SNE[J]. Journal of Machine Learning Research, 2008, 9: 2579-2605.

[105] DABOV K, FOI A, KATKOVNIK V, et al. Image denoising by sparse 3-D transform-domain collaborative filtering[J]. IEEE Transactions on Image Processing, 2007, 16(8): 2080-2095.

[106] MAGGIONI M, KATKOVNIK V, EGIAZARIAN K, et al. Nonlocal transform-domain filter for volumetric data denoising and reconstruction[J]. IEEE Transactions on Image Processing, 2013, 22(1): 119-133.

[107] ZHANG H, HE W, ZHANG L, et al. Hyperspectral image restoration using low-rank matrix recovery[J]. IEEE Transactions on Geoscience and Remote Sensing, 2014, 52(8): 4729-4743.

[108] XU F, CHEN Y, PENG C, et al. Denoising of hyperspectral image using low-rank matrix factorization[J]. IEEE Geoscience and Remote Sensing Letters, 2017, 14(7): 1141-1145.

[109] CHANG Y, YAN L, ZHONG S. Hyper-Laplacian regularized unidirectional low-rank tensor recovery for multispectral image denoising[C]//2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR). 2017: 5901-5909.

[110] WANG Y, PENG J, ZHAO Q, et al. Hyperspectral image restoration via total variation regularized low-rank tensor decomposition[J]. IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing, 2018, 11(4): 1227-1243.

[111] HE W, YAO Q, LI C, et al. Non-local meets global: An integrated paradigm for hyperspectral denoising[C]// 2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). 2019: 6861-6870.

[112] ZHA Z, WEN B, YUAN X, et al. Nonlocal structured sparsity regularization modeling for hyperspectral image denoising[J]. IEEE Transactions on Geoscience and Remote Sensing, 2023, 61: 1-16.

[113] YUAN Q, ZHANG Q, LI J, et al. Hyperspectral image denoising employing a spatial–spectral deep residual convolutional neural network[J]. IEEE Transactions on Geoscience and Remote Sensing, 2019, 57(2): 1205-1218.

[114] WEI K, FU Y, HUANG H. 3D quasi-recurrent neural network for hyperspectral image denoising[J]. IEEE Transactions on Neural Networks and Learning Systems, 2021, 32(1): 363-375.

[115] MAFFEI A, HAUT J M, PAOLETTI M E, et al. A single model CNN for hyperspectral image denoising[J]. IEEE Transactions on Geoscience and Remote Sensing, 2020, 58(4): 2516-2529.

[116] ZHUANG L, NG M K, GAO L, et al. Eigen-CNN: Eigenimages plus eigennoise level maps guided network for hyperspectral image denoising[J]. IEEE Transactions on Geoscience and Remote Sensing, 2024, 62: 1-18.

[117] ZHUANG L, NG M K. FastHyMix: Fast and parameter-free hyperspectral image mixed noise removal[J]. IEEE Transactions on Neural Networks and Learning Systems, 2023, 34(8): 4702-4716.

[118] ZHANG K, ZUO W, ZHANG L. FFDNet: Toward a fast and flexible solution for CNN-based image denoising [J]. IEEE Transactions on Image Processing, 2018, 27(9): 4608-4622.

[119] XIONG F, ZHOU J, TAO S, et al. SMDS-Net: Model guided spectral-spatial network for hyperspectral image denoising[J]. IEEE Transactions on Image Processing, 2022, 31: 5469-5483.

[120] PENG J, WANG H, CAO X, et al. Learnable representative coefficient image denoiser for hyperspectral image [J]. IEEE Transactions on Geoscience and Remote Sensing, 2024, 62: 1-16.

[121] MONGA V, LI Y, ELDAR Y C. Algorithm unrolling: Interpretable, efficient deep learning for signal and image processing[J]. IEEE Signal Processing Magazine, 2021, 38(2): 18-44.

[122] WANG M, HONG D, HAN Z, et al. Tensor decompositions for hyperspectral data processing in remote sensing: A comprehensive review[J]. IEEE Geoscience and Remote Sensing Magazine, 2023, 11(1): 26-72.

[123] YANG Y, SUN J, LI H, et al. ADMM-CSNet: A deep learning approach for image compressive sensing[J]. IEEE Transactions on Pattern Analysis and Machine Intelligence, 2020, 42(3): 521-538.

[124] CAI J F, CANDÈS E J, SHEN Z. A singular value thresholding algorithm for matrix completion[J]. SIAM Journal on Optimization, 2010, 20(4): 1956-1982.

[125] CHEN Y, ZENG J, HE W, et al. Fast large-scale hyperspectral image denoising via noniterative low-rank subspace representation[J]. IEEE Transactions on Geoscience and Remote Sensing, 2024, 62: 1-14.

[126] ZHAO M, LI W, LI L, et al. Single-frame infrared small-target detection: A survey[J]. IEEE Geoscience and Remote Sensing Magazine, 2022, 10(2): 87-119.

[127] ZHU H, NI H, LIU S, et al. TNLRS: Target-aware non-local low-rank modeling with saliency filtering regularization for infrared small target detection[J]. IEEE Transactions on Image Processing, 2020, 29: 9546-9558.

[128] GAO C, MENG D, YANG Y, et al. Infrared patch-image model for small target detection in a single image[J]. IEEE Transactions on Image Processing, 2013, 22(12): 4996-5009.

[129] ZHANG L, PENG Z. Infrared small target detection based on partial sum of the tensor nuclear norm[J]. Remote Sensing, 2019, 11(4): 382.

[130] LIN F, GE S, BAO K, et al. Learning shape-biased representations for infrared small target detection[J]. IEEE Transactions on Multimedia, 2024, 26: 4681-4692.

[131] WEI Y, YOU X, LI H. Multiscale patch-based contrast measure for small infrared target detection[J]. Pattern Recognition, 2016, 58: 216-226.

[132] ZHANG T, LI L, CAO S, et al. Attention-guided pyramid context networks for detecting infrared small target under complex background[J]. IEEE Transactions on Aerospace and Electronic Systems, 2023, 59(4): 4250-4261.

[133] LIU Q, LIU R, ZHENG B, et al. Infrared Small Target Detection with Scale and Location Sensitivity[C]// Proceedings of the IEEE/CVF Computer Vision and Pattern Recognition. 2024.

[134] WU X, HONG D, CHANUSSOT J. UIU-Net: U-Net in U-Net for infrared small object detection[J]. IEEE Transactions on Image Processing, 2023, 32: 364-376.

[135] YANG Z, YU H, ZHANG J, et al. Deep learning based infrared small object segmentation: Challenges and future directions[J]. Information Fusion, 2025, 118: 103007.

[136] SHLEZINGER N, WHANG J, ELDAR Y C, et al. Model-based deep learning[J]. Proceedings of the IEEE, 2023, 111(5): 465-499.

[137] WU F, ZHANG T, LI L, et al. RPCANet: Deep unfolding RPCA based infrared small target detection[C]// Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision. 2024: 4809-4818.

[138] XIONG Z, ZHOU F, WU F, et al. DRPCA-Net: Make robust PCA great again for infrared small target detection [J]. IEEE Transactions on Geoscience and Remote Sensing, 2025, 63: 1-16.

[139] WU F, DAI Y, ZHANG T, et al. RPCANet++: Deep interpretable robust PCA for sparse object segmentation[J]. arXiv:2508.04190, 2025.

[140] GREGOR K, LECUN Y. Learning fast approximations of sparse coding[C]//Proceedings of the 27th International Conference on Machine Learning. 2010: 399-406.

[141] ZHANG J, GHANEM B. ISTA-Net: Interpretable optimization-inspired deep network for image compressive sensing[C]//Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. 2018: 1828-1837.

[142] YOU D, XIE J, ZHANG J. ISTA-Net++: Flexible deep unfolding network for compressive sensing[C]//2021 IEEE International Conference on Multimedia and Expo (ICME). 2021: 1-6.

[143] HAN S, YANG S, ZHANG X, et al. DISTA-Net: Dynamic closely-spaced infrared small target unmixing[J]. arXiv:2505.19148, 2025.

[144] VASWANI A, SHAZEER N, PARMAR N, et al. Attention is all you need[J]. Advances in Neural Information Processing Systems, 2017, 30.

[145] VIRMAUX A, SCAMAN K. Lipschitz regularity of deep neural networks: analysis and efficient estimation[J]. Advances in Neural Information Processing Systems, 2018, 31.

[146] RAHMAN M A, WANG Y. Optimizing intersection-over-union in deep neural networks for image segmentation [C]//International Symposium on Visual Computing. 2016: 234-244.

[147] LUO L, XIE Y, ZHANG Z, et al. Support matrix machines[C]//International Conference on Machine Learning. 2015: 938-947.

[148] LIANG S, HANG W, LEI B, et al. Adaptive multimodel knowledge transfer matrix machine for EEG classification[J]. IEEE Transactions on Neural Networks and Learning Systems, 2022, 35(6): 7726-7739.

[149] LI X, SHAO H, LU S, et al. Highly efficient fault diagnosis of rotating machinery under time-varying speeds using LSISMM and small infrared thermal images[J]. IEEE Transactions on Systems, Man, and Cybernetics: Systems, 2022, 52(12): 7328-7340.

[150] LI Y, WANG D, LIU F. The auto-correlation function aided sparse support matrix machine for EEG-based fatigue detection[J]. IEEE Transactions on Circuits and Systems II: Express Briefs, 2023, 70(2): 836-840.

[151] FENG R, XU Y. Support matrix machine with pinball loss for classification[J]. Neural Computing and Applications, 2022, 34(21): 18643-18661.

[152] XIU X, SUN S, LI X, et al. Heaviside low-rank support matrix machine[J]. arXiv:2603.00491, 2026.

[153] ZHENG Q, ZHU F, HENG P A. Robust support matrix machine for single trial EEG classification[J]. IEEE Transactions on Neural Systems and Rehabilitation Engineering, 2018, 26(3): 551-562.

[154] RAZZAK I, BOUADJENEK M R, SARIS R A, et al. Support matrix machine via joint $\ell_{2,1}$ and nuclear norm minimization under matrix completion framework for classification of corrupted data[J]. IEEE Transactions on Neural Networks and Learning Systems, 2024, 35(11): 16341-16352.

[155] KUMARI A, AKHTAR M, SHAH R, et al. Support matrix machine: A review[J]. Neural Networks, 2025, 181: 106767.

[156] WU C, LI D H, SUN D. Support matrix machine: exploring sample sparsity, low rank, and adaptive sieving in high-performance computing[J]. Mathematical Programming Computation, 2026: 1-46.

[157] YAN Q, GONG D, SHI Q, et al. Attention-guided network for ghost-free high dynamic range imaging[C]// Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. 2019: 1751-1760.

[158] WOO S, PARK J, LEE J Y, et al. CBAM: Convolutional Block Attention Module[C]//Proceedings of the European Conference on Computer Vision. 2018: 3-19.

[159] QIN X, WANG Z, BAI Y, et al. FFA-Net: Feature fusion attention network for single image dehazing[C]// Proceedings of the AAAI Conference on Artificial Intelligence: vol. 34: 07. 2020: 11908-11915.

[160] YANG L, ZHANG R Y, LI L, et al. Simam: A simple, parameter-free attention module for convolutional neural networks[C]//International Conference on Machine Learning. 2021: 11863-11874.

[161] ZHANG Y, TIAN Y, KONG Y, et al. Residual dense network for image super-resolution[C]//Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. 2018: 2472-2481.

[162] BENGIO Y, GOODFELLOW I, COURVILLE A, et al. Deep learning: vol. 1[M]. MIT press Cambridge, MA, USA, 2017.

[163] CHEN J, KAO S H, HE H, et al. Run, don't walk: chasing higher FLOPS for faster neural networks[C]// Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. 2023: 12021-12031.

[164] QIN D, LEICHNER C, DELAKIS M, et al. MobileNetV4: Universal models for the mobile ecosystem[C]// European Conference on Computer Vision. 2024: 78-96.

[165] FLORIDI L, CHIRIATTI M. GPT-3: Its nature, scope, limits, and consequences[J]. Minds and Machines, 2020, 30(4): 681-694.

[166] GRATTAFIORI A, DUBEY A, JAUHRI A, et al. The Llama 3 herd of models[J]. Neural Information Processing Systems, 2024.

[167] ZHU X, LI J, LIU Y, et al. A survey on model compression for large language models[J]. Transactions of the Association for Computational Linguistics, 2024, 12: 1556-1577.

[168] KIM G I, HWANG S, JANG B. Efficient compressing and tuning methods for large language models: A systematic literature review[J]. ACM Computing Surveys, 2025, 57(10): 1-39.

[169] CHENG H, ZHANG M, SHI J Q. A survey on deep neural network pruning: Taxonomy, comparison, analysis, and recommendations[J]. IEEE Transactions on Pattern Analysis and Machine Intelligence, 2024, 46(12): 10558-10578.

[170] LECUN Y, DENKER J, SOLLA S. Optimal brain damage[J]. Advances in Neural Information Processing Systems, 1989.

[171] HASSIBI B, STORK D. Second order derivatives for network pruning: Optimal brain surgeon[J]. Advances in Neural Information Processing Systems, 1992.

[172] FRANTAR E, ALISTARH D. SparseGPT: Massive language models can be accurately pruned in one-shot[C]// International Conference on Machine Learning. 2023: 10323-10337.

[173] SUN M, LIU Z, BAIR A, et al. A simple and effective pruning approach for large language models[C]// International Conference on Learning Representations. 2024.

[174] MA X, FANG G, WANG X. LLM-pruner: On the structural pruning of large language models[J]. Advances in Neural Information Processing Systems, 2023.

[175] ASHKBOOS S, CROCI M L, FRANTAR E, et al. SliceGPT: Compress large language models by deleting rows and columns[C]//International Conference on Learning Representations. 2024.

[176] CHEN Y, CHENG B, HAN J, et al. DLP: Dynamic layerwise pruning in large language models[C]//International Conference on Machine Learning. 2025: 7934-7956.

[177] TAO C, SHEN T, GAO S, et al. LLMs are also effective embedding models: An in-depth overview[J]. arXiv:2412.12591, 2024.

[178] MERITY S, XIONG C, BRADBURY J, et al. Pointer sentinel mixture models[C]//International Conference on Learning Representations. 2017.

[179] MEISTER C, COTTERELL R. Language model evaluation beyond perplexity[C]//Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing. 2021: 5328-5339.

[180] BECK A, TEBOULLE M. A fast iterative shrinkage-thresholding algorithm for linear inverse problems[J]. SIAM Journal on Imaging Sciences, 2009, 2(1): 183-202.

[181] CHAMBOLLE A, POCK T. A first-order primal-dual algorithm for convex problems with applications to imaging[J]. Journal of Mathematical Imaging and Vision, 2011, 40(1): 120-145.

[182] ELAD M, KAWAR B, VAKSMAN G. Image denoising: The deep learning revolution and beyond—a survey paper[J]. SIAM Journal on Imaging Sciences, 2023, 16(3): 1594-1654.

[183] ARRIDGE S, MAASS P, ÖKTEM O, et al. Solving inverse problems using data-driven models[J]. Acta Numerica, 2019, 28: 1-174.

[184] RAVISHANKAR S, YE J C, FESSLER J A. Image reconstruction: From sparsity to data-adaptive methods and machine learning[J]. Proceedings of the IEEE, 2020, 108(1): 86-109.

[185] SCARLETT J, HECKEL R, RODRIGUES M R D, et al. Theoretical perspectives on deep learning methods in inverse problems[J]. IEEE Journal on Selected Areas in Information Theory, 2022, 3(3): 433-453.

[186] MUKHERJEE S, HAUPTMANN A, ÖKTEM O, et al. Learned reconstruction methods with convergence guarantees: A survey of concepts and applications[J]. IEEE Signal Processing Magazine, 2023, 40(1): 164-182.

[187] CHEN X, LIU J, YIN W. Learning to optimize: A tutorial for continuous and mixed-integer optimization[J]. Science China Mathematics, 2024, 67(6): 1191-1262.

[188] SHLEZINGER N, SEGARRA S, ZHANG Y, et al. Deep unfolding: Recent developments, theory, and design guidelines[J]. arXiv:2512.03768, 2025.

[189] PARIKH N, BOYD S. Proximal algorithms[J]. Foundations and Trends in Optimization, 2014, 1(3): 123-231.

[190] MOREAU T, BRUNA J. Understanding trainable sparse coding with matrix factorization[C]//International Conference on Learning Representations. 2017.

[191] CHEN X, LIU J, WANG Z, et al. Theoretical linear convergence of unfolded ISTA and its practical weights and thresholds[J]. Advances in Neural Information Processing Systems, 2018, 31.

[192] LIU J, CHEN X, WANG Z, et al. ALISTA: Analytic weights are as good as learned weights in LISTA[C]// International Conference on Learning Representations. 2019.

[193] CHEN X, LIU J, WANG Z, et al. Hyperparameter tuning is all you need for LISTA[C]//Advances in Neural Information Processing Systems: vol. 34. 2021.

[194] ABLIN P, MOREAU T, MASSIAS M, et al. Learning step sizes for unfolded sparse coding[C]//Advances in Neural Information Processing Systems: vol. 32. 2019.

[195] WU K, GUO Y, LI Z, et al. Sparse coding with gated learned ISTA[C]//International Conference on Learning Representations. 2020.

[196] BEHBOODI A, RAUHUT H, SCHNOOR E. Compressive sensing and neural networks from a statistical learning perspective[G]//Compressed Sensing in Information Processing. Springer, 2022: 247-277.

[197] SCHNOOR E, BEHBOODI A, RAUHUT H. Generalization error bounds for iterative recovery algorithms unfolded as neural networks[J]. Information and Inference: A Journal of the IMA, 2023, 12(3): 2267-2299.

[198] SHAH S B, PRADHAN P, PU W, et al. Optimization guarantees of unfolded ISTA and ADMM networks with smooth soft-thresholding[J]. IEEE Transactions on Signal Processing, 2024, 72: 3272-3286.

[199] HADOU S, NADERIALIZADEH N, RIBEIRO A. Robust stochastically-descending unrolled networks[J]. IEEE Transactions on Signal Processing, 2024, 72: 5484-5499.

[200] KOUNI V. How to warm-start your unfolding network[C]//2025 International Conference on Sampling Theory and Applications (SampTA). 2025: 1-6.

[201] CHEN E, CHEN X, JALALI S, et al. Deep memory unrolled networks for solving imaging linear inverse problems [C]//2025 International Conference on Sampling Theory and Applications (SampTA). 2025: 1-6.

[202] SUCKER M, FADILI J, OCHS P. Learning-to-optimize with PAC-Bayesian guarantees: Theoretical considerations and practical implementation[J]. Journal of Machine Learning Research, 2025, 26(211): 1-53.

[203] SAMBHARYA R, STELLATO B. Data-driven performance guarantees for classical and learned optimizers[J]. Journal of Machine Learning Research, 2025, 26(171): 1-49.

[204] XIANG J, DONG Y, YANG Y. FISTA-Net: Learning a fast iterative shrinkage thresholding network for inverse problems in imaging[J]. IEEE Transactions on Medical Imaging, 2021, 40(5): 1329-1339.

[205] ZHANG J, ZHAO C, GAO W. Optimization-inspired compact deep compressive sensing[J]. IEEE Journal of Selected Topics in Signal Processing, 2020, 14(4): 765-774.

[206] YANG Y, SUN J, LI H, et al. Deep ADMM-Net for compressive sensing MRI[C]//Advances in Neural Information Processing Systems: vol. 29. 2016: 10-18.

[207] SHULTZMAN A, AZAR E, RODRIGUES M R D, et al. Generalization and estimation error bounds for model-based neural networks[C]//International Conference on Learning Representations. 2023.

[208] KOUNI V, PARASKEVOPOULOS G, RAUHUT H, et al. ADMM-DADNet: A deep unfolding network for analysis compressed sensing[C]//IEEE International Conference on Acoustics, Speech and Signal Processing. 2022: 1506-1510.

[209] AN W, LIU Y, SHANG F, et al. DEs-inspired accelerated unfolded linearized ADMM networks for inverse problems[J]. IEEE Transactions on Neural Networks and Learning Systems, 2025, 36(3): 5319-5333.

[210] HAO J, et al. Deep unfolding ADMM network for CS image reconstruction with long-short term residuals[J]. Signal Processing, 2026, 243: 110450.

[211] VOGEL C, POCK T. A primal dual network for low-level vision problems[C]//Lecture Notes in Computer Science: Pattern Recognition: vol. 10496. Springer, Cham, 2017: 189-202.

[212] ADLER J, ÖKTEM O. Learned primal-dual reconstruction[J]. IEEE Transactions on Medical Imaging, 2018, 37(6): 1322-1332.

[213] 郭田德, 幸天驰, 韩丛英, 等. 人工智能中的生成式方法: 数学模型、优化算法及其应用[J]. 运筹学学报 (中英文), 2025, 29(03): 1-33.

[214] BORA A, JALAL A, PRICE E, et al. Compressed sensing using generative models[C]//Proceedings of Machine Learning Research: Proceedings of the 34th International Conference on Machine Learning: vol. 70. 2017: 537-546.

[215] PENG P, JALALI S, YUAN X. Solving inverse problems via auto-encoders[J]. IEEE Journal on Selected Areas in Information Theory, 2020, 1(1): 312-323.

[216] GONZÁLEZ M, ALMANSA A, TAN P. Solving inverse problems by joint posterior maximization with autoencoding prior[J]. SIAM Journal on Imaging Sciences, 2022, 15(2): 822-859.

[217] WHANG J, LINDGREN E M, DIMAKIS A G. Composing normalizing flows for inverse problems[C]// International Conference on Machine Learning. 2021: 11158-11169.

[218] ZHAO Z, YE J C, BRESLER Y. Generative models for inverse imaging problems: From mathematical foundations to physics-driven applications[J]. IEEE Signal Processing Magazine, 2023, 40(1): 148-163.

[219] KAWAR B, ELAD M, ERMON S, et al. Denoising diffusion restoration models[C]//Advances in Neural Information Processing Systems: vol. 35. 2022: 23593-23606.

[220] CHUNG H, KIM J, MCCANN M T, et al. Diffusion posterior sampling for general noisy inverse problems[C]// International Conference on Learning Representations. 2023.

[221] WANG Y, YU J, ZHANG J. Zero-shot image restoration using denoising diffusion null-space model[C]// International Conference on Learning Representations. 2023.

[222] DUFF M A G, KEMETH F J P, DRISCOLL T A, et al. Regularising inverse problems with generative machine learning models[J]. Journal of Mathematical Imaging and Vision, 2024, 66(2): 173-208.

[223] HOSSEINI B, HUANG Z. Error analysis of Bayesian inverse problems with generative priors[J]. arXiv:2601.17374, 2026.

[224] WEI X, van GORP H, GONZALEZ CARABARIN L, et al. Image denoising with deep unfolding and normalizing flows[C]//Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing. 2022: 4123-4127.

[225] WEI X, van GORP H, GONZALEZ-CARABARIN L, et al. Deep unfolding with normalizing flow priors for inverse problems[J]. IEEE Transactions on Signal Processing, 2022, 70: 2963-2977.

[226] AI Y, CAI Y, ZHANG Y, et al. Flow-matching guided deep unfolding for hyperspectral image reconstruction[J]. arXiv:2510.01912, 2025.

[227] LIAO C, SHEN Y, LI D, et al. Using powerful prior knowledge of diffusion model in deep unfolding networks for image compressive sensing[C]//Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. 2025.

[228] WU Z, LU R, FU Y, et al. Latent diffusion prior enhanced deep unfolding for snapshot spectral compressive imaging[C]//European Conference on Computer Vision. 2024: 164-181.

[229] ZHENG B, SUN G, ZHANG H, et al. Deep unfolding architecture based on generative prior diffusion for image compressive sensing[J]. IEEE Signal Processing Letters, 2025, 32: 2878-2882.

[230] WANG Y, SHOUSHTARI S, KAMILOV U S. Diff-Unfolding: A model-based score learning framework for inverse problems[J]. arXiv:2505.11393, 2025.

[231] DENG X, ZHANG C, JIANG L, et al. DeepSN-Net: Deep semi-smooth Newton driven network for blind image restoration[J]. IEEE Transactions on Pattern Analysis and Machine Intelligence, 2025, 47(4): 2632-2646.

[232] SONG J, MOU C, WANG S, et al. Optimization-inspired cross-attention Transformer for compressive sensing [C]//Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. 2023: 6174-6184.

[233] CHEN P, HUANG Z, WANG X, et al. Vision-language controlled deep unfolding for joint medical image restoration and segmentation[J]. arXiv:2601.23103, 2026.

[234] TSOUROS D, VERHAEGHE H, KADIOGLU S, et al. Holy Grail 2.0: From natural language to constraint models[C]//Progress Toward the Holy Grail workshop at CP2023. 2023.

[235] WANG T, YU W Y, SHE R, et al. Leveraging large language models for solving rare MIP challenges[J]. arXiv:2409.04464, 2024.

[236] LIU H, WANG J, CAI Y, et al. OptiTree: Hierarchical thoughts generation with tree search for LLM optimization modeling[C]//Advances in Neural Information Processing Systems. 2025.

[237] AHMADITESHNIZI A, GAO W, UDELL M. OptiMUS: Scalable optimization modeling with (MI)LP solvers and large language models[C]//International Conference on Machine Learning. 2024.

[238] HAO Y, ZHANG Y, FAN C. Planning anything with rigor: General-purpose zero-shot planning with LLM-based formalized programming[C]//International Conference on Learning Representations. 2025.

[239] XIAO Z, ZHANG D, WU Y, et al. Chain-of-Experts: When LLMs meet complex operations research problems [C]//International Conference on Learning Representations. 2024.

[240] MOSTAJABDAVEH M, YU T T, RAMAMONJISON R, et al. Optimization modeling and verification from problem specifications using a multi-agent multi-stage LLM framework[J]. INFOR: Information Systems and Operational Research, 2024, 62(4): 599-617.

[241] ASTORGA N, LIU T, XIAO Y, et al. Autoformulation of mathematical optimization models using LLMs[C]// International Conference on Machine Learning. 2025.

[242] BERTO F, HUA C, LUTTMANN L, et al. PARCO: parallel autoregressive models for multi-agent combinatorial optimization[C]//Advances in Neural Information Processing Systems. 2025.

[243] JIANG X, WU Y, ZHANG C, et al. DRoC: Elevating large language models for complex vehicle routing via decomposed retrieval of constraints[C]//International Conference on Learning Representations. 2025.

[244] PENG M, CHEN Z, YANG J, et al. Automatic MILP model construction for multi-robot task allocation and scheduling based on large language models[C]//2025 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS). 2025: 20291-20296.

[245] LIANG K, LU Y, MAO J, et al. LLM for large-scale optimization model auto-formulation: A lightweight few-shot learning approach[J]. arXiv:2601.09635, 2026.

[246] AMARASINGHE P T, NGUYEN S, SUN Y, et al. AI-Copilot for business optimisation: A framework and a case study in production scheduling[J]. arXiv:2309.13218, 2023.

[247] LI Q, ZHANG L, MAK-HAU V. Synthesizing mixed-integer linear programming models from natural language descriptions[J]. arXiv:2311.15271, 2023.

[248] MASOUD M, ABDELHAY A, ELHENAWY M. Exploring combinatorial problem solving with large language models: A case study on the travelling salesman problem using GPT-3.5 turbo[J]. arXiv:2405.01997, 2024.

[249] YANG Z, WANG Y, HUANG Y, et al. OptiBench meets ReSocratic: Measure and improve LLMs for optimization modeling[C]//International Conference on Learning Representations. 2025.

[250] HUANG C, TANG Z, HU S, et al. ORLM: A customizable framework in training large models for automated optimization modeling[J]. Operations Research, 2025.

[251] MA Z, GONG Y J, GUO H, et al. LLaMoCo: Instruction tuning of large language models for optimization code generation[J]. IEEE Transactions on Evolutionary Computation, 2026.

[252] JIANG C, SHU X, QIAN H, et al. LLMOPT: Learning to define and solve general optimization problems from scratch[C]//International Conference on Learning Representations. 2025.

[253] ZHOU C, YANG J, XIN L, et al. Auto-formulating dynamic programming problems with large language models [J]. arXiv:2507.11737, 2025.

[254] DING Z, TAN Z, ZHANG J, et al. OR-R1: Automating modeling and solving of operations research optimization problem via test-time reinforcement learning[J]. arXiv:2511.09092, 2025.

[255] LU H, XIE Z, WU Y, et al. OptMATH: A scalable bidirectional data synthesis framework for optimization modeling[C]//International Conference on Machine Learning. 2025.

[256] ZHANG J, WANG W, GUO S, et al. Solving general natural-language-description optimization problems with large language models[C]//Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies. 2024: 483-490.

[257] WU Y, ZHANG Y, WU Y, et al. Evo-Step: Evolutionary generation and stepwise validation for optimizing LLMs in OR[Z]. 2025.

[258] NIE A, CHENG C A, KOLOBOV A, et al. Importance of directional feedback for LLM-based optimizers[C]// Advances in Neural Information Processing Systems. 2023.

[259] WU X, ZHONG Y, WU J, et al. Large language model-enhanced algorithm selection: Towards comprehensive algorithm representation[C]//Proceedings of the Thirty-Third International Joint Conference on Artificial Intelligence. 2024: 5235-5244.

[260] LI X, YANG J, WANG J, et al. STRCMP: Integrating graph structural priors with language models for combinatorial optimization[J]. arXiv:2506.11057, 2025.

[261] WU X, WANG D, WU C, et al. Efficient heuristics generation for solving combinatorial optimization problems using large language models[C]//Proceedings of the 31st ACM SIGKDD Conference on Knowledge Discovery and Data Mining V. 2. 2025: 3228-3239.

[262] MAO J, ZOU D, SHENG L, et al. Identify critical nodes in complex network with large language models[J]. arXiv:2403.03962, 2024.

[263] YU H, LIU J. AutoRNet: Automatically optimizing heuristics for robust network design via large language models [J]. arXiv:2410.17656, 2024.

[264] ZHANG Q, HONG X, TANG J, et al. GCoder: Improving large language model for generalized graph problem solving[J]. arXiv:2410.19084, 2024.

[265] SURINA A, MANSOURI A, QUAEDVLIEG L, et al. Algorithm discovery with LLMs: Evolutionary search meets reinforcement learning[J]. arXiv:2504.05108, 2025.

[266] SARTORI C C, BLUM C. Combinatorial optimization for All: Using LLMs to aid non-experts in improving optimization algorithms[J]. arXiv:2503.10968, 2025.

[267] ROMERA-PAREDES B, BAREKATAIN M, NOVIKOV A, et al. Mathematical discoveries from program search with large language models[J]. Nature, 2024, 625(7995): 468-475.

[268] LIU F, TONG X, YUAN M, et al. Evolution of heuristics: towards efficient automatic algorithm design using large language model[C]//International Conference on Machine Learning. 2024.

[269] YAO S, LIU F, LIN X, et al. Multi-objective evolution of heuristic using large language model[C]//AAAI Conference on Artificial Intelligence: vol. 39: 25. 2025: 27144-27152.

[270] YE H, WANG J, CAO Z, et al. ReEvo: Large language models as hyper-heuristics with reflective evolution[C]// Advances in Neural Information Processing Systems. 2024.

[271] DAT P V T, DOAN L, BINH H T T. Hsevo: Elevating automatic heuristic design with diversity-driven harmony search and genetic algorithm using LLMs[C]//AAAI Conference on Artificial Intelligence: vol. 39: 25. 2025: 26931-26938.

[272] LIU F, ZHANG R, XIE Z, et al. LLM4AD: A platform for algorithm design with large language model[J]. arXiv:2412.17287, 2024.

[273] YU H, LIU J. Deep insights into automated optimization with large language models and evolutionary algorithms [J]. arXiv:2410.20848, 2024.

[274] SHI Y, ZHOU J, SONG W, et al. Generalizable heuristic generation through large language models with meta-optimization[J]. arXiv:2505.20881, 2025.

[275] HUANG Z, WU W, WU K, et al. CALM: Co-evolution of algorithms and language model for automatic heuristic design[J]. arXiv:2505.12285, 2025.

[276] HAO H, ZHANG X, ZHOU A. Large language models as surrogate models in evolutionary algorithms: A preliminary study[J]. Swarm and Evolutionary Computation, 2024, 91: 101741.

[277] WANG Z, LIU S, CHEN J, et al. Large language model-aided evolutionary search for constrained multiobjective optimization[C]//International Conference on Intelligent Computing. 2024: 218-230.

[278] Van STEIN N, BÄCK T. LLaMEA: A large language model evolutionary algorithm for automatically generating metaheuristics[J]. IEEE Transactions on Evolutionary Computation, 2025, 29(2): 331-345.

[279] BRAHMACHARY S, JOSHI S M, PANDA A, et al. Large language model-based evolutionary optimizer: Reasoning with elitism[J]. Neurocomputing, 2025, 622: 129272.

[280] LIU F, LIN X, YAO S, et al. Large language model for multiobjective evolutionary optimization[C]//International Conference on Evolutionary Multi-Criterion Optimization. 2025: 178-191.

[281] MARTINEK A, LUKASIK S, GANDOMI A H. Large language models as tuning agents of metaheuristics.[C]// European Symposium on Artificial Neural Networks. 2024.

[282] ZHANG M, DESAI N, BAE J, et al. Using large language models for hyperparameter optimization[C]//Advances in Neural Information Processing Systems. 2023.

[283] CHEN H, CONSTANTE-FLORES G E, LI C. Diagnosing infeasible optimization problems using large language models[J]. INFOR: Information Systems and Operational Research, 2024, 62(4): 573-587.

[284] MA L, HAO X, YANG R, et al. Automatic algorithm design assisted by LLMs for solving vehicle routing problems[C]//International Conference on Signal Processing. 2024: 247-252.

[285] CHEN Y, XIA J, SHAO S, et al. Solver-Informed RL: Grounding large language models for authentic optimization modeling[J]. arXiv:2505.11792, 2025.

[286] HUANG Z, GUO L, LI W, et al. GraphThought: Graph combinatorial optimization with thought generation[J]. arXiv:2502.11607, 2025.

[287] HUANG Z, SHI G, SUKHATME G S. From words to routes: Applying large language models to vehicle routing [J]. arXiv:2403.10795, 2024.

[288] ZHANG B, LUO P. OR-LLM-Agent: Automating modeling and solving of operations research optimization problem with reasoning large language model[J]. arXiv:2503.10009, 2025.

[289] FORNIÉS-TABUENCA D, URIBE A, OTAMENDI U, et al. REMoH: A reflective evolution of multi-objective heuristics approach via large language models[J]. arXiv:2506.07759, 2025.

[290] CHEN H, WANG Y, CAI Y, et al. HeuriGym: An agentic benchmark for LLM-crafted heuristics in combinatorial optimization[C]//International Conference on Learning Representations. 2026.

[291] ELHENAWY M, ABUTAHOUN A, ALHADIDI T I, et al. Visual reasoning and multi-agent approach in multimodal large language models (MLLMs): Solving TSP and mTSP combinatorial challenges[J]. Machine Learning & Knowledge Extraction, 2024, 6(3): 1894-1920.

[292] YUAN Z, LIU M, WANG H, et al. MA-GTS: A multi-agent framework for solving complex graph problems in real-world applications[J]. arXiv:2502.18540, 2025.

[293] YANG X, ZHANG L, QIAN H, et al. HeurAgenix: Leveraging LLMs for solving complex combinatorial optimization challenges[J]. arXiv:2506.15196, 2025.

[294] WANG Z, CHEN B, HUANG Y, et al. ORMind: A cognitive-inspired end-to-end reasoning framework for operations research[J]. arXiv:2506.01326, 2025.

[295] LIMA V, PHAN D T, KALAGNANAM J, et al. Toward a trustworthy optimization modeling agent via verifiable synthetic data generation[J]. arXiv:2508.03117, 2025.

附录 A　英文缩写对照表

压缩感知	compressed sensing, CS
主成分分析	principal component analysis, PCA
近端交替最小化	proximal alternating minimization, PAM
归一化互信息	normalized mutual information, NMI
物联网	Internet of Things, IoT
联邦学习	federated learning, FL
交替方向乘子法	alternating direction method of multipliers, ADMM
交替流形近端梯度法	alternating manifold proximal gradient method, AManPG
半光滑牛顿	semi-smooth Newton, SSN
入侵检测系统	intrusion detection system, IDS
接收者操作特征	receiver operating characteristic, ROC
曲线下面积	area under the curve, AUC
故障检测	fault detection, FD
独立成分分析	independent component analysis, ICA
偏最小二乘	partial least squares, PLS
费舍尔判别分析	Fisher discriminant analysis, FDA
典型相关分析	canonical correlation analysis, CCA
非负矩阵分解	nonnegative matrix factorization, NMF
平方预测误差	squared prediction error, SPE
核密度估计	kernel density estimation, KDE
故障检测率	fault detection rate, FDR
近邻分类器	k-nearest neighbors, KNN
块匹配三维滤波	block matching 3D, BM3D
卷积神经网络	convolutional neural network, CNN
迭代收缩阈值算法	iterative shrinkage thresholding algorithm, ISTA
修正线性单元	rectified linear unit, ReLU
峰值信噪比	peak signal-to-noise ratio, PSNR
结构相似性	structural similarity, SSIM
光谱角度映射器	spectral angle mapper, SAM
离散余弦变换	discrete cosine transform, DCT
红外小目标检测	infrared small target detection, ISTD

鲁棒主成分分析	robust principal component analysis, RPCA
批归一化	batch normalization, BN
平均交并比	mean intersection over union, mIoU
支持向量机	support vector machine, SVM
支持矩阵机	support matrix machine, SMM
增广拉格朗日法	augmented lagrangian method, ALM
奇异值分解	singular value decomposition, SVD
近端梯度下降法	proximal gradient descent, PGD
大语言模型	large language models, LLMs
层归一化	layer normalization, LayerNorm
均方根归一化	root mean square layer normalization, RMSNorm
快速迭代收缩阈值法	fast iterative shrinkage thresholding algorithm, FISTA
原始-对偶混合梯度法	primal-dual hybrid gradient, PDHG
深度展开	deep unfolding, DU
即插即用	plug-and-play, PnP
全变差	total variation, TV
学习型迭代收缩阈值法	learned ISTA, LISTA
变分自编码器	variational autoencoder, VAE
生成对抗网络	generative adversarial network, GAN
归一化流模型	normalizing flow, NF
最大后验估计	maximum a posteriori, MAP
检索增强生成	retrieval-augmented generation, RAG
低秩适配	low-rank adaptation, LoRA
黑盒优化	black-box optimization, BBO
混合整数线性规划	mixed integer linear programming, MILP

* 按文中第一次出现的先后顺序排列, 仅列出部分

附录 B　优化求解器介绍

优化求解器是开展算法设计与智能决策的重要工具, 在工业生产、物流运输、能源规划等多个领域发挥着不可替代的作用. 当前全球优化求解器已形成 “国际主导、国产崛起、商业领先、开源追赶” 的发展格局, 不同类型软件在求解性能、使用成本、开发难度、行业适配性等方面存在差异, 下面对国内外主流优化求解器进行简单介绍.

B.1 商用求解器

商用求解器凭借成熟的算法内核、完善的技术支持及强大的大规模问题处理能力, 成为企业级决策优化的核心工具, 主要面向对求解效率、稳定性和集成性有较高要求的工业场景, 部分也提供学术版用于科研教学.

(1) Gurobi　由美国 Gurobi Optimization 公司开发, 在混合整数规划领域表现突出, 是目前工业界和科研界公认的高效求解工具之一. 其核心优势在于算法的高效性与稳定性, 能快速处理大规模线性规划、二次规划、混合整数线性/二次规划等各类优化问题, 尤其适用于高维约束、复杂调度等工业场景. 该软件支持多线程并行计算和云部署模式, 提供 C++、Java、Python、Matlab 等多语言 API 接口, 可与 Pyomo、AMPL 等主流建模语言无缝兼容, 方便用户嵌入现有系统进行二次开发. 此外, Gurobi 为高校教师和学生提供免费学术版, 降低了科研和教学门槛.

(2) CPLEX　1988 年推出首个版本, 2009 年被 IBM 公司收购, 现为 IBM 决策优化系统的关键组件. 该求解器具备全面的求解能力, 可高效处理线性规划、混合整数规划、二次规划、二阶锥规划等多种类型的优化问题, 尤其擅长大规模企业级复杂问题的求解. CPLEX 的优势在于稳定性强、兼容性好, 与 IBM Cloud、SPSS Modeler 深度关联, 提供完善的企业级 API 与数据集成方案, 适合需要集成至企业 IT 系统的复杂优化任务, 但该软件存在价格昂贵、许可证结构复杂、安装配置难度较高等问题.

(3) Xpress　1983 年正式发布, 是全球首个可在电脑端运行的商业线性规划与混合整数规划求解器, 2008 年被 FICO 公司收购, 在供应链优化、金融决策等场景中应用广泛. 该求解器擅长处理大型混合规划问题, 具备强大的商业建模平台, 支持多求解方法的自动切换, 配备可视化分析界面, 可深度嵌入 FICO 业务平台. 软件支持.NET、Java、C、Python、Matlab 等多种接口语言, 其内置的建模语言 Xpress Mosel 包含分布式计算功能, 可并行解决优化问题的多个场景, 能有效处理输入数据的不确定性.

(4) COPT　国内杉数科技自主研发的高性能商用求解器, 也是近年来国内运筹优化领域的重点突破产品, 其求解能力已达到国际主流商用软件水平. 该求解器支持线性规划、整数

规划、混合整数规划、二阶锥规划、半定规划等多种优化问题, 在高维约束问题中具备显著的加速效果. COPT 的突出优势是本地化支持完善, 提供中文技术文档与高效的技术响应服务, 同时支持 C、C++、Python、Julia 等多语言接口, 兼容 AMPL、GAMS、Pyomo 等主流建模工具, 适配国内企业的应用场景与需求.

B.2 开源求解器

开源求解器凭借免费、开源、可定制的优势, 成为科研人员、算法开发者及中小企业的首选工具. 尽管在大规模复杂问题的求解性能上略逊于商用软件, 但开源求解器在灵活性、可扩展性及二次开发方面具备明显优势, 部分优秀开源软件的求解性能已接近商业级水平.

(1) OR-Tools　由 Google 公司开源的运筹优化工具, 免费且灵活, 集成了多种求解器接口, 支持线性规划、约束规划、混合整数规划等多种优化问题, 适合中小企业和开发者用于原型验证. 其核心优势是轻量化、易集成, 可与 Python 数据生态无缝衔接, 便于快速构建优化模型并验证算法可行性, 但在大规模复杂模型的求解效率上不及专业商用求解器.

(2) SCIP　由德国 Zuse 研究所开发, 是目前学术领域最强大的非商业求解器之一. 主要针对混合整数非线性规划问题, 适合研究人员和算法开发者进行自定义算法开发或扩展模块研究. 其核心功能涵盖约束整数规划、分支定界、割平面等经典优化算法, 在学术研究中应用广泛, 但学习成本较高, 不适合初学者快速上手.

(3) HiGHS　由英国爱丁堡大学牵头开发的开源线性规划与混合整数规划求解器, 是国际上近年来崛起的高性能开源工具, 凭借高效的求解算法和简洁的架构, 逐步成为开源领域的重要选择. 相较于同品类开源求解器, HiGHS 在中大规模线性规划问题上表现突出, 求解性能可与部分商用求解器相媲美.

B.3 发展趋势

随着人工智能技术与运筹优化的深度融合, 优化求解器呈现以下发展趋势. 第一, 开源求解器在算法优化、功能完善上持续发力, 与商用求解器的性能差距逐步缩小. 第二, 人工智能与优化算法深度融合, 大幅提升了复杂动态场景的求解效率与鲁棒性. 第三, 国产求解器崛起, 如杉数 COPT、华为 OptVerse、阿里 MindOpt 等产品, 在算法性能、生态适配和本地化支持上逐步追平国际标准. 值得关注的是, 大语言模型的快速迭代与广泛应用, 正为优化求解器的创新发展注入全新动能, 推动其向更智能、更高效的方向迈进.