计 算 机 研 究 与 发 展
Journal of Computer Research and Development
https://crad.ict.ac.cn
62(9): 2139−2151, 2025

基于合成数据预训练基础模型的表格数据聚类方法

李培文[1] 李飞江[1,2] 王婕婷[1,2] 钱宇华[1,2]

[1](山西大学大数据科学与产业研究院 太原 030006)

[2](演化科学智能山西省重点实验室(山西大学) 太原 030006)

(202422407020@email.sxu.edu.cn)

Clustering Method for Tabular Data Based on Pretrained Foundation Models with Synthetic Data

Li Peiwen[1], Li Feijiang[1,2], Wang Jieting[1,2], and Qian Yuhua[1,2]

[1](*Institute of Big Data Science and Industry*, *Shanxi University*, *Taiyuan* 030006)

[2](*Shanxi Key Laboratory of Evolutionary Science and Intelligence* (*Shanxi University*), *Taiyuan* 030006)

Abstract Driven by rapid advancements in data acquisition and storage technologies, vast amounts of unlabeled tabular data have been collected and stored across various industries. Clustering analysis serves as a fundamental method for uncovering the latent grouping structures within such data. Currently, traditional clustering algorithms still predominantly constitute the methods employed for processing this type of data. Deep learning techniques and large model technologies, known for their powerful representation and inference capabilities, are primarily employed for handling unstructured data types like images, text, and audio; their advantages remain largely underexploited in the context of structured tabular data processing. In 2025, Nature published TabPFN, a foundation model for tabular data capable of efficiently handling classification and regression tasks, offering a novel perspective for tabular data learning. Building upon TabPFN, this paper proposes a clustering method for tabular data based on pretrained foundation models with synthetic data. The method comprises two main phases: a pretraining phase and an iterative inference phase. The pretraining phase leverages traditional clustering algorithms and TabPFN model to obtain initial pseudo-labels for the unlabeled tabular data. The iterative inference phase then employs the fine-tuned TabPFN model to cyclically update these pseudo-labels until the final clustering results are obtained. Extensive experimental analyses conducted on benchmark datasets demonstrate that the proposed method significantly improves the clustering performance of seven representative clustering algorithms.

Key words clustering analysis; tabular data learning; foundation model; iterative inference; unsupervised learning

摘 要 随着数据采集与数据存储技术的飞速发展,各行业收集并存储了大量无标记的表格数据.聚类分析是挖掘这类数据潜在分组结构的重要方法.目前,处理表格数据的聚类方法多数仍然是传统聚类算法.深度学习技术和大模型技术主要用于处理非结构化的图像、文本、语音等数据类型,其强大的表示能力和推理能力在结构化的表格数据处理中仍难以发挥优势.2025 年,《Nature》刊发的 TabPFN 是一种可用于高效处理分类和回归任务的表格数据基础模型,为表格数据学习提供了新的基础.受此启发,提出了一种基于合成数据预训练基础模型的表格数据聚类方法,主要包括预训练阶段和迭代推理阶段.其中,预训练阶段基于传统数据聚类算法和 TabPFN 模型获得无标记表格数据的初始伪标签,迭代推理阶段基于微调

收稿日期: 2025-06-06; **修回日期**: 2025-07-16

基金项目: 国家自然科学基金项目(62441239, 62476160, 62306170, 62136005, U24A20253)

This work was supported by the National Natural Science Foundation of China (62441239, 62476160, 62306170, 62136005, U24A20253).

通信作者: 钱宇华(jinchengqyh@126.com)

后的 TabPFN 模型循环更新伪标签以得到聚类结果. 在基准数据集上的大量实验分析表明, 改进方法显著提高了 7 种代表性聚类算法的性能.

关键词 聚类分析; 表格数据学习; 基础模型; 迭代推理; 无监督学习

中图法分类号 TP311.13; TP309

DOI: 10.7544/issn1000-1239.202550405 **CSTR:** 32373.14.issn1000-1239.202550405

表格数据以种类繁杂的不同形式广泛存在于各种学科和工程领域, 包括电子商务[1]、医疗保健[2]、金融[3]、医学分析[4] 等. 同时, 随着数据采集技术与数据存储技术的飞速发展, 各行业收集并存储了大量表格数据. 然而, 数据标注的滞后性导致这些数据标注匮乏. 聚类分析是挖掘无标注数据内涵分组结构的重要方法, 也是机器学习、计算机视觉、模式识别等领域广受关注的任务之一. 聚类分析旨在将相似度高的样本划分在同一类簇, 将不相似样本划分至不同类簇. 现阶段, 传统的聚类方法, 如 k-means[5]、层次聚类[6]、谱聚类[7]、高斯混合模型[8] 聚类等以及在其基础上发展出的大量聚类算法仍是处理表格数据聚类的主要方法.

近年来, 得益于深度学习强大的非线性表征与端到端学习能力, 以伪标签深度聚类[9]、自监督深度聚类[10]、对比深度聚类[11] 等模型为代表的深度聚类算法在图像、文本等非结构化数据中展现出优异性能. 这些方法通过卷积网络[12]、Transformer[13] 等架构有效挖掘了此类非结构化数据的潜在结构, 显著提升了聚类性能. 然而对于结构化表格数据学习领域, 深度聚类的表示能力难以被有效利用. 表格数据普遍存在特征异构性、高维稀疏性及弱局部相关性等特性, 与深度学习擅长处理的图像、文本数据具有较大差异, 导致深度聚类模型依赖的层次化特征抽象机制难以直接迁移. 全连接网络的粗暴应用也难以挖掘表格数据特征的内在关联机制. 因此, 表格数据中隐含的业务逻辑约束(如金融时序特征的相关性规则[14])和复杂特征交互模式(如交叉特征的组合效应[15])使得现有深度聚类方法往往陷入表征能力不足与先验知识利用不足的双重困境. 尽管部分研究尝试通过数据重构(如表格转图像[16])或浅层自编码器[17] 进行适配, 但这些方案尚未突破模型架构与数据特性的适配障碍. 这些局限严重制约了深度聚类技术在医疗诊断、金融风控等领域表格数据上的实际应用. 因此, 针对结构化数据特性设计新的深度学习聚类模式将进一步丰富表格数据聚类分析研究的发展.

近年来, 大模型领域的快速发展为改进聚类架构提供了新的思路. 文献 [18] 利用 CLIP 等大模型提供文本-图像对齐特征, 实现了跨模态增强, 提升了图像数据聚类性能. 文献 [19] 指出大模型生成的数据可通过聚类去重(如 stable diffusion[20] 生成内容去重), 实现了降噪与数据筛选. 文献 [21] 将大模型输出作为聚类初始监督信号, 提出了 Self-CC 框架, 采用高置信样本来优化目标分布, 代替原有的伪标签生成. 虽然, 目前大模型辅助聚类的研究仍然多数针对非结构化的图像或文本数据, 但这些研究为大模型赋能表格数据聚类提供了可借鉴思路.

2025 年, Hollmann 等人在 Nature 上的工作中提出了一个基于合成数据构建的表格数据基础模型 TabPFN[22], 把大语言模型(LLM)的核心架构 Transformer[13] 融入表格数据的分类和回归预测任务. 在新的任务场景中, 使用者仅需基于 TabPFN 模型进行快速训练拟合, 就可在该场景下对新数据进行预测. 得益于 TabPFN 在表格数据分类任务中的巨大成功, 近期的研究已揭示出其在广泛应用中的潜力, 包括表格数据生成[23-25]、数据扩充[26] 和时间序列预测[27] 等. 本文旨在基于 TabPFN 基础模型设计聚类方法.

综合上述讨论, 本文提出了一种基于合成数据预训练基础模型的聚类方法, 方法流程如图 1 所示, 该方法可分为预训练和迭代推理 2 个阶段. 预训练阶段旨在获得无标记表格数据的初始伪标签. 具体而言, 对于给定的无标签数据集, 随机抽取部分样本, 然后采用一个聚类算法得到采样样本的伪标签, 并基于此训练微调 TabPFN 模型, 最后预测整个数据集得到全部样本的伪标签. 迭代推理阶段旨在提升伪标签性能并得到最终聚类结果. 该过程是一个迭代推理循环过程, 具体而言, 每次循环从带伪标签数据按比例抽取训练集微调 TabPFN 模型, 然后对所有样本预测以更新伪标签, 最后循环至标签不变或达到最大迭代轮数并将预测标签输出为聚类结果. 我们在 16 组 UCI 基准表格数据集上的实验分析表明, 本文方法显著提高了 7 种代表性聚类算法的性能.

本文的主要贡献包括: 建立了一种基于合成数

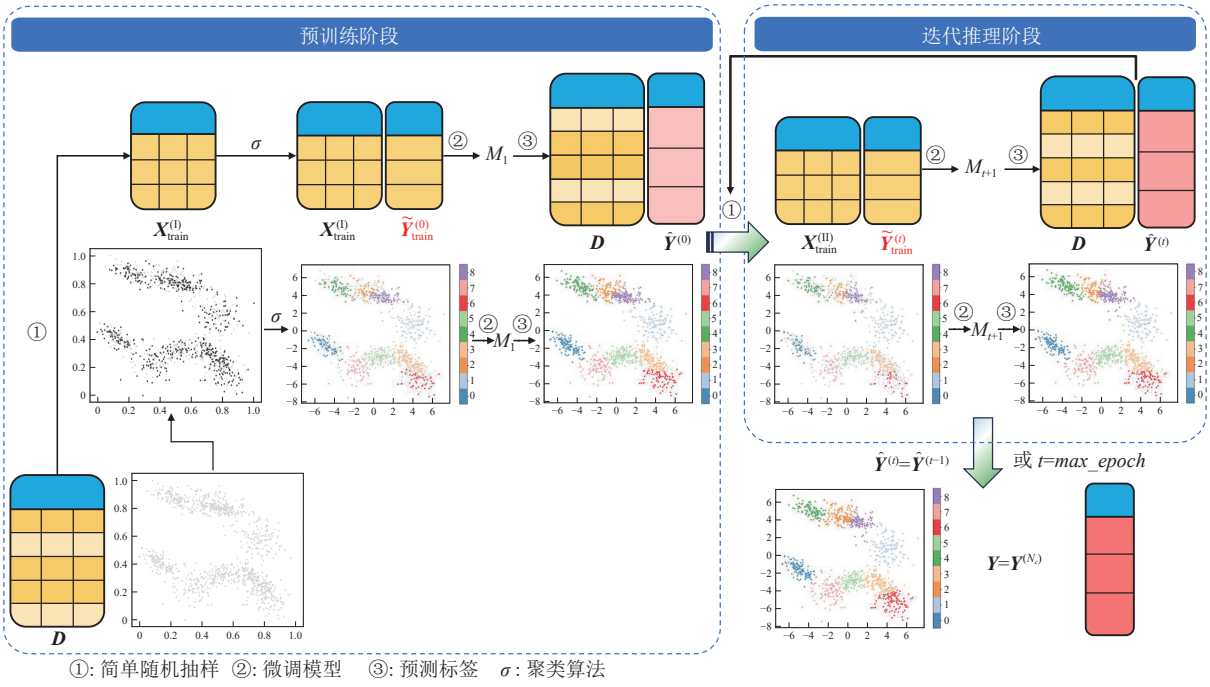


Fig. 1 Flow chart of clustering method for tabular data based on pretrained foundation models with synthetic data

图 1 基于合成数据预训练基础模型的表格数据聚类方法流程图

据预训练基础模型的数据聚类方法，利用了表格数据基础模型的推理能力，实验分析表明所提方法可有效提升 7 种代表性聚类方法的聚类性能，揭示了表格数据基础模型对表格数据聚类的促进作用.

1 相关工作

1.1 表格数据聚类

主流经典聚类方法中，k-means 聚类以最小化数据点与中心间的距离为目标，迭代优化聚类中心来完成划分[5]，计算效率高，但对初始中心敏感，且只能捕获球形分布，适合处理低维、球形簇分布的表格数据. 层次聚类（hierarchical clustering）通过构建树状结构（凝聚或分裂）逐步合并或分割簇[6]，可解释性强且适应任意形状，但计算复杂度高，适合中小型数据且对噪声敏感的场景. 谱聚类（spectral clustering）基于图论，通过构造相似度矩阵并映射到低维空间进行图分割，一大优势是能识别非凸分布簇，对数据分布假设较少，但计算相似度矩阵的时间空间开销较大，且对核函数参数敏感[7]，擅长处理复杂结构的中小型表格数据. 高斯混合模型（Gaussian mixture model, GMM）聚类假设数据由多个高斯分布混合生成[28]，通过 EM 算法[29]优化参数实现概率软聚类，其优点在于提供簇分配概率，适用于椭球型分布，但依赖高斯分布假设，易陷入局部最优，适合具有明确概率特征的表格数据. 基于密度的聚类如 DBSCAN[30]、DP[31] 和 OPTICS[32]，都是根据局部邻域范围的密度划分簇，该类方法抗噪性强且能发现任意形状的类簇，但都对参数敏感，适合处理密度差异显著的中小型表格数据. AP（affinity propagation）[33]聚类通过消息传递自动确定代表点，用相似度矩阵驱动聚类过程. 其优势在于无需预设簇数且对初始值不敏感，但计算复杂度（$O(n^2)O(n^2)$）和内存消耗高[33]，且性能受偏好参数（控制簇数）和相似度度量影响，适合分布多样但规模较小的表格数据.

这些方法在处理中小型表格数据集时受参数选择、数据分布、计算开销等因素影响，各有其局限性. 随着深度学习的飞速发展，研究者自然想到将可传播梯度的网络结构融入聚类模型，深度聚类应运而生. 现有主流深度聚类方法通过神经网络自动提取高维数据的低维嵌入特征，联合优化特征提取器损失和嵌入空间聚类损失来完成聚类[34-35]；所用数据一般为图像（面向 CV 领域的图像分割、物体识别）[10,36]、文本（主题建模、文档聚类）[37] 和图数据（面向社交网络社区发现）[38] 以及混合类型的多视图数据[39-41]. 相比传统聚类，深度聚类利用非线性映射降维弥补了在高维数据中“维度灾难”的缺陷，能捕获非凸、流形结构，且深度聚类通过端到端学习最优特征实现了自动化特征提取. 但也有其局限性，包括计算成本高，如对比学习需构建大量样本对，时间成本远超传

统方法;低维数据优势不明显,在低维结构化表格数据中,深度聚类可能不如 k-means 高效;依赖数据质量,如噪声数据[42]或小样本场景下,深度聚类易过拟合.综合看来,现有的深度聚类模型在表格数据聚类方面仍值得关注.

大模型在非结构化数据学习中展现出独特价值.其强大的上下文推理能力可捕捉特征间的隐性关联,基于预训练知识迁移的范式为稀疏小样本场景提供数据增强的可能,而注意力机制的多层交互特性有助于解析高维表格数据的复杂分布模式.然而,深度学习方法常因表格数据的特殊含义特征与离散属性陷入表征学习[43]困境,大模型固有的参数量与数据需求又难以适配表格场景隐含的特定逻辑关系,这使得传统非结构化数据的大模型架构在处理结构化表格数据时面临计算效率与泛化性能的双重瓶颈.

在此背景下,基于大量合成数据先验学习推理能力的表格数据基础模型 TabPFN 框架实现了表格数据的分类、回归下游任务的良好适配.该模型通过构建概率化的神经网络结构,将贝叶斯推断与 Transformer 注意力机制有机融合[44],在保持小样本快速收敛的同时,成功实现了表格特征空间的高效建模.该模型为大模型赋能表格数据聚类提供了模型基础.

1.2 TabPFN 表格合成分类器

文献 [22] 提出了 TabPFN 模型,将注意力机制引入表格数据预测中,该模型重点解决样本容量在 10 000 以内、特征个数不超过 500 个、类别数不超过 10 个的表格数据分类任务,并提供了密度估计、回归等多种模块功能[22].在与多种基于树的分类方法对比后,TabPFN 体现出明显的性能优势.该模型是一个先验数据拟合网络(prior-data fitted network, PFN),旨在用 PFN 来近似贝叶斯神经网络(Bayesian neural networks, BNN)中的 PPD(posterior predictive distribution)[44-45].TabPFN 主要分为预训练和推理 2 个阶段.在预训练阶段,通过最小化训练样本的预测标签与其真实标签间的差异,使用因果推断模型 SCMs(structural causal models)[22,45]合成表格数据集,在数以亿计的合成数据集上反复对模型进行预训练.在推理阶段,给定一个新的任务和真实世界一组带标签的表格实例作为提示,PFN 利用上下文学习直接预测测试样本的标签,而不需要进一步的参数更新[45].

预训练阶段,可以在 PFN 中自由设计对期望先验进行编码的数据集生成算法,用一个先验采样方案来指定先验:

$$p(\boldsymbol{D})=\mathbb{E}_{\phi\sim p(\phi)}\left[p(\boldsymbol{D}|\phi)\right], \tag{1}$$

其中,假设空间 $\boldsymbol{\Phi}$ 定义了一组输入样本 x 和输出标签 y 之间的关系,每个假设 $\phi\in\boldsymbol{\Phi}$ 都可以被看作一种产生数据分布的机制,PFN 从中抽取样本形成数据集.该公式首先用 $p(\phi)$ 对假设 ϕ(即生成机制)做采样,然后用 $p(\boldsymbol{D}|\phi)$ 采样给定假设条件下的合成数据集 $\boldsymbol{D}$.获得先验信息后就可根据贝叶斯定理结合训练样本的似然度计算后验概率分布,在反复采样生成新的观测数据过程中,对示例标签分布的预测值逐渐逼近其目标类别分布,PFN 的参数 θ 也被不断优化.在推理阶段 TabPFN 通过

$$p(y|x,\boldsymbol{D})\propto\int_{\phi}p(y|x,\phi)\,p(\boldsymbol{D}|\phi)\,p(\phi)\,\mathrm{d}\phi \tag{2}$$

导出待测数据的标签软分配概率.

具体训练过程中,训练集中的每一对点 (x_i,y_i) 都被转换成 Transformer 中的一个 token,使得每个特征向量和标签彼此之间相互注意[44].

实际应用中,导入并初始化 TabPFN 模型后,输入下游领域任务中的表格数据和对应的标签,调用模型进行领域数据拟合,即可在新的领域对数据进行推理预测.

2 本文模型

本文所提聚类方法主要包括 2 个阶段,分别为预训练阶段和迭代推理阶段,对应阶段(I)和阶段(II),具体流程如图 1 所示.其中,阶段(I)基于基础模型 TabPFN 和任意一个聚类算法,旨在获得无标记表格数据的初始伪标签.阶段(II)主要基于阶段(I)微调所得模型,旨在迭代提升伪标签性能并得到最终聚类结果.本文所用到的数学符号如表 1 所示.

2.1 预训练

预训练阶段,我们输入表格数据集 $\boldsymbol{D}$,经过一轮微调预测过程得到伪标签 $\hat{\boldsymbol{Y}}^{(0)}$,将初始无监督任务转化成一个伪标记指导的监督学习任务.下面是具体流程.

读入表格数据 $\boldsymbol{D}$ 后,首先做一次不放回的简单随机抽样,得到的训练样本定义为 $\boldsymbol{X}^{(\mathrm{I})}_{\mathrm{train}}=(\boldsymbol{s}^{(\mathrm{I})}_1,\boldsymbol{s}^{(\mathrm{I})}_2,\cdots,\boldsymbol{s}^{(\mathrm{I})}_{n^{(\mathrm{I})}})^{\mathrm{T}}$.具体执行中,指定抽样比例 $\pi^{(\mathrm{I})}$ 来控制从全集 $\boldsymbol{D}$ 中随机采样出的训练集大小为 $n^{(\mathrm{I})}=\pi^{(\mathrm{I})}\cdot N$,通过抽样出的序号 $S^{(\mathrm{I})}\sim Uniform(\mathcal{P}_{n^{(\mathrm{I})}}(\{1,2,\cdots,N\}))$ 可以生成阶段(I)的训练样本,设 $h^{(\mathrm{I})}_i$ 是从集合 $S^{(\mathrm{I})}$ 中抽取的第 i 个序号,如式(3)所示.

$$\boldsymbol{X}^{(\mathrm{I})}_{\mathrm{train}}=(\boldsymbol{x}_{h^{(\mathrm{I})}_1},\boldsymbol{x}_{h^{(\mathrm{I})}_2},\cdots,\boldsymbol{x}_{h^{(\mathrm{I})}_{n^{(\mathrm{I})}}})^{\mathrm{T}}. \tag{3}$$

Table 1 Summary of Mathematical Symbols

表 1 数学符号汇总

符号	含义
N	样本总数
$\boldsymbol{D}$	输入的无标记表格数据：$\boldsymbol{D}=(\boldsymbol{x}_1,\boldsymbol{x}_2,\cdots,\boldsymbol{x}_N)^{\mathrm{T}}$
$\boldsymbol{x}_i$	表格数据$\boldsymbol{D}$的第i行
N_c	模型收敛所需迭代轮数
t	当前迭代轮次
M_t	第t轮的模型
$\theta^{(t)}$	第t轮的模型参数
k	目标类簇个数
σ	所选基线聚类算法
f	本文模型的单次微调过程
g	对M的前向推理过程
$\mathcal{P}_n$	所有大小为n的整数子集构成的集合
$\boldsymbol{Y}^{(t)}$	第t轮的全集硬标签：$\boldsymbol{Y}\subset C=\{c\mid c\in N^*,c\leqslant k\}$
$\hat{\boldsymbol{Y}}^{(t)}$	第t轮的全集软标签：$\hat{\boldsymbol{Y}}^{(t)}=(\hat{\boldsymbol{y}}_i^{(t)})^{\mathrm{T}},i\leqslant N$
$\boldsymbol{X}_{\mathrm{train}}^{(\mathrm{I})}$	阶段（I）的训练样本
$\pi^{(\mathrm{I})}$	阶段（I）的随机抽样比例
$n^{(\mathrm{I})}$	阶段（I）的训练集大小
$\boldsymbol{X}_{\mathrm{train}}^{(\mathrm{II})}$	阶段（II）的训练样本
$\pi^{(\mathrm{II})}$	阶段（II）的随机抽样比例
$n^{(\mathrm{II})}$	阶段（II）的训练集大小
$\tilde{\boldsymbol{L}}_{\mathrm{train}}^{(t)}$	第t轮的训练集伪标签
$\boldsymbol{D}_{\mathrm{train}}^{(t)}$	第t轮的训练集
max_epoch	最大迭代轮数
$\boldsymbol{D}_t$	第t轮的全集
η	参数更新过程的学习率
$\mathcal{L}_\theta^{(t)}$	第t轮的损失函数

接下来，基于选定聚类算法σ，对$\boldsymbol{X}_{\mathrm{train}}^{(\mathrm{I})}$聚类得到初始的训练标签：

$$\widetilde{\boldsymbol{L}}_{\mathrm{train}}^{(0)}=\sigma\left(\boldsymbol{X}_{\mathrm{train}}^{(\mathrm{I})}\right). \tag{4}$$

这里定义为$\widetilde{\boldsymbol{L}}_{\mathrm{train}}^{(0)}=(\widetilde{\boldsymbol{l}}_1^{(0)},\widetilde{\boldsymbol{l}}_2^{(0)},\cdots,\widetilde{\boldsymbol{l}}_{n^{(\mathrm{I})}}^{(0)})^{\mathrm{T}}$,其中每个元素都是一个概率形式的标签向量：$\widetilde{\boldsymbol{l}}_i^{(0)}=(\widetilde{l}_{i,j}^{(0)}),\widetilde{l}_{i,j}^{(0)}\in[0,1]$, $j=1,2,\cdots,k$.

由于该标签是迭代过程中模型预测的结果，不同于监督学习的真实标签，因此将该类中间预测结果通称为伪标签. 合并训练样本和伪标签就得到了阶段(I)的训练集$\boldsymbol{D}_{\mathrm{train}}^{(0)}=(\boldsymbol{X}_{\mathrm{train}}^{(\mathrm{I})},\widetilde{\boldsymbol{L}}_{\mathrm{train}}^{(0)})$，用小规模数据集$\boldsymbol{D}_{\mathrm{train}}^{(0)}$和基础模型$M_0$(即 TabPFN)微调优化参数，得到第一轮模型$M_1$. 所采用的损失函数为：

$$\mathcal{L}_\theta^{(0)}=\frac{1}{n^{(\mathrm{I})}}\sum_{i=1}^{n^{(\mathrm{I})}}J^{(0)}(p_i^{(0)},\widetilde{\boldsymbol{l}}_i^{(0)}), \tag{5}$$

其中$J^{(0)}$是阶段(I)单个训练样本的损失函数，这里采用交叉熵损失：

$$J^{(0)}(p_i^{(0)},\widetilde{\boldsymbol{l}}_i^{(0)})=-\sum_{c=1}^{k}\widetilde{l}_{i,j}^{(0)}\log(p_{i,c}^{(0)}), \tag{6}$$

其中$p_{i,c}^{(0)}$是训练样本$s_i^{(\mathrm{I})}$经过模型M_0的 softmax 层后对类别c的簇分配概率：

$$p_{i,c}^{(0)}=\frac{\exp\left(z_{i,c}^{(0)}\right)}{\sum_{j=1}^{k}\exp\left(z_{i,j}^{(0)}\right)}, \tag{7}$$

其中$z_{i,c}^{(0)}$是$s_i^{(\mathrm{I})}$输入M_0后对类别c的输出. 基于式(5)~(7)，阶段(I)的参数更新过程可描述为：

$$\theta^{(1)}=\theta^{(0)}-\eta\cdot\nabla_{\theta^{(0)}}\mathcal{L}_\theta^{(0)}. \tag{8}$$

利用微调一轮后的模型M_1，进行一次前向推理即可预测出无标注数据集$\boldsymbol{D}$的伪标签$\hat{\boldsymbol{Y}}^{(0)}$，如式(9)所示.

$$\begin{aligned}&\hat{\boldsymbol{y}}_i^{(0)}=(\hat{y}_{i,j}^{(0)})=g_{\theta^{(1)}}(x_i),\\&\hat{y}_{i,j}^{(0)}\in[0,1],\ j=1,2,\cdots,k.\end{aligned} \tag{9}$$

实际应用中，可能由于缺乏足够的先验信息，聚类效果仍有较大提升空间，因此我们在M_1基础上继续进行一个自监督的迭代推理循环，即阶段(II)的迭代推理过程.

预训练阶段的流程如算法 1 所示.

算法 1. 阶段(I)预训练的步骤.

输入：无标注表格数据集$\boldsymbol{D}$，类簇数k，初始模型M_0；

输出：$\boldsymbol{D}$的预测标签$\boldsymbol{Y}^{(0)}$.

①根据式 (3) 从$\boldsymbol{D}$中抽取训练样本$\boldsymbol{X}_{\mathrm{train}}^{(\mathrm{I})}$；

②利用聚类算法σ对$\boldsymbol{X}_{\mathrm{train}}^{(\mathrm{I})}$进行聚类，得到训练标签$\widetilde{\boldsymbol{L}}_{\mathrm{train}}^{(0)}$；

③创建训练集$\boldsymbol{D}_{\mathrm{train}}^{(0)}=(\boldsymbol{X}_{\mathrm{train}}^{(\mathrm{I})},\widetilde{\boldsymbol{L}}_{\mathrm{train}}^{(0)})$；

④根据式(5)~(8)在训练集$\boldsymbol{D}_{\mathrm{train}}^{(0)}$上微调模型$M_0$参数，得到$M_1$；

⑤利用M_1根据式 (9) 预测出$\boldsymbol{D}$的伪标签$\hat{\boldsymbol{Y}}^{(0)}$.

2.2 迭代推理

迭代推理阶段，我们输入表格数据$\boldsymbol{D}$和其对应伪标签$\hat{\boldsymbol{Y}}^{(0)}$，经过训练集更新，用训练集做微调、对全集样本做预测，然后迭代抽取训练集更新伪标签预测的过程，得到最终聚类结果$\boldsymbol{Y}$. 具体流程如下.

根据阶段(I)的输出值，将$\boldsymbol{D}$和$\hat{\boldsymbol{Y}}^{(0)}$合并可创建一个包含全部样本的有监督数据集，我们称它为第 1 轮的全集：$\boldsymbol{D}_1=(\boldsymbol{D},\hat{\boldsymbol{Y}}^{(0)})$，相当于将上一轮的伪标签作为本轮微调的监督信号，以此类推有

$$\boldsymbol{D}_t=\left(\boldsymbol{D},\hat{\boldsymbol{Y}}^{(t-1)}\right)=\left(\left(\boldsymbol{x}_1,\hat{\boldsymbol{y}}_1^{(t-1)}\right),\left(\boldsymbol{x}_2,\hat{\boldsymbol{y}}_2^{(t-1)}\right),\cdots,\left(\boldsymbol{x}_N,\hat{\boldsymbol{y}}_N^{(t-1)}\right)\right)^{\mathrm{T}},\ t=1,2,\cdots,N_c, \tag{10}$$

往后每轮迭代的训练集都是对上一轮的全集做不放回的简单随机抽样产生的.

我们根据划分比例$\pi^{(\mathrm{II})}$得到$n^{(\mathrm{II})}=\pi^{(\mathrm{II})}\cdot N$, 令阶段(II)的训练样本为: $\boldsymbol{X}_{\mathrm{train}}^{(\mathrm{II})}=(\boldsymbol{s}_1^{(\mathrm{II})},\boldsymbol{s}_2^{(\mathrm{II})},\cdots,\boldsymbol{s}_{n^{(\mathrm{II})}}^{(\mathrm{II})})^{\mathrm{T}}$, 训练标签为$\widetilde{\boldsymbol{L}}_{\mathrm{train}}^{(t)}=[\widetilde{\boldsymbol{l}}_1^{(t)},\widetilde{\boldsymbol{l}}_2^{(t)},\cdots,\widetilde{\boldsymbol{l}}_{n^{(\mathrm{II})}}^{(t)}]^{\mathrm{T}}$, 从全集$\boldsymbol{D}_1$中随机抽取$n^{(\mathrm{II})}$个样本, 对应序号集合$S^{(\mathrm{II})}\sim Uniform(\mathcal{P}_{n^{(\mathrm{II})}}(\{1,2,\cdots,N\}))$, 用这一可重复的采样过程可以生成每一轮的训练集, 包含训练样本和训练标签, 设$h_i^{(\mathrm{II})}$是从集合$S^{(\mathrm{II})}$中抽取的第i个序号:

$$D_{\mathrm{train}}^{(t)}=\left(\boldsymbol{X}_{\mathrm{train}}^{(\mathrm{II})},\widetilde{\boldsymbol{L}}_{\mathrm{train}}^{(t)}\right)=\left(\left(\boldsymbol{x}_{h_1^{(\mathrm{II})}},\widetilde{\boldsymbol{y}}_{h_1^{(\mathrm{II})}}^{(t-1)}\right),\left(\boldsymbol{x}_{h_2^{(\mathrm{II})}},\widetilde{\boldsymbol{y}}_{h_2^{(\mathrm{II})}}^{(t-1)}\right),\cdots,\left(\boldsymbol{x}_{h_{n^{(\mathrm{II})}}^{(\mathrm{II})}},\widetilde{\boldsymbol{y}}_{h_{n^{(\mathrm{II})}}^{(\mathrm{II})}}^{(t-1)}\right)\right)^{\mathrm{T}},\ t=1,2,\cdots,N_c, \tag{11}$$

在训练集$\boldsymbol{D}_{\mathrm{train}}^{(t)}$上继续微调模型$M_t$, 与阶段(I)中微调过程同理, 其中式(5)改为

$$\mathcal{L}_\theta^{(t)}=\frac{1}{n^{(\mathrm{II})}}\sum_{i=1}^{n^{(\mathrm{II})}}J^{(t)}(p_i^{(t)},\widetilde{\boldsymbol{l}}_i^{(t)}),\ t=1,2,\cdots,N_c, \tag{12}$$

$J^{(t)}$是阶段(II)单个训练样本的损失函数,

$$J^{(t)}(p_i^{(t)},\widetilde{\boldsymbol{l}}_i^{(t)})=-\sum_{c=1}^{k}\widetilde{l}_{i,j}^{(t)}\log(p_{i,c}^{(t)}), \tag{13}$$

其中$p_{i,c}^{(t)}$是训练样本$s_i^{(\mathrm{II})}$经过模型M_t的 softmax 层后对类别c的簇分配概率:

$$p_{i,c}^{(t)}=\frac{\exp\left(z_{i,c}^{(t)}\right)}{\sum_{j=1}^{k}\exp\left(z_{i,j}^{(t)}\right)}, \tag{14}$$

$z_{i,c}^{(t)}$是$s_i^{(\mathrm{II})}$输入M_t后对类别c的 logit(原始输出). 联立式(12)~(14)可得阶段(II)的参数更新公式:

$$\theta^{(t+1)}=\theta^{(t)}-\eta\cdot\nabla_{\theta^{(t)}}\mathcal{L}_\theta^{(t)}, \tag{15}$$

$$t=1,2,\cdots,N_c,$$

利用微调后的模型M_{t+1}, 进行一次前向推理即可预测出数据集$\boldsymbol{D}$的伪标签$\hat{\boldsymbol{Y}}^{(t)}$. 如式(16)所示.

$$\hat{\boldsymbol{y}}_i^{(t)}=(\hat{y}_{i,j}^{(t)})=g_{\theta^{(t+1)}}(x_i), \tag{16}$$

$$\hat{y}_{i,j}^{(t)}\in[0,1].$$

$$j=1,2,\cdots,k,\ t=1,2,\cdots,N_c,$$

进一步, 取最大值能得到硬划分标签:

$$\boldsymbol{y}_i^{(t)}=\arg\max_j(\hat{y}_{i,j}^{(t)}),\ j=1,2,\cdots,k, \tag{17}$$

$$\boldsymbol{Y}^{(t)}=\left(\boldsymbol{y}_i^{(t)}\right)^{\mathrm{T}},\ i=1,2,\cdots N. \tag{18}$$

循环执行微调模型(式(10)~(15))和前向推理操作(式(16)~(18))直至满足模型收敛条件: $\boldsymbol{Y}^{(t)}=\boldsymbol{Y}^{(t-1)}$, 此时$t=N_c$, 输出最后一轮预测的硬标签$\boldsymbol{Y}^{(N_c)}$作为聚类结果$\boldsymbol{Y}$.

为了尽可能兼顾效率和减少过拟合, 设定了一个最大迭代轮数max_epoch, 使得$N_c=\min(N_c,max_epoch)$, 这表明若本文模型对给定数据集聚类达到收敛所需的训练轮数较多, 我们就倾向于早停(early stop), 反之则正常收敛.

迭代推理阶段的算法流程如算法 2 所示.

算法 2. 阶段(II)迭代推理的步骤.

输入: 无标注表格数据集$\boldsymbol{D}$, 阶段(I)的预测标签$\hat{\boldsymbol{Y}}^{(0)}$;

输出: $\boldsymbol{D}$最终的收敛预测标签$\boldsymbol{Y}$.

① 令$t=0$, 利用式(17)(18)求出$\boldsymbol{Y}^{(0)}$;

② $t=1$;

③ while $\boldsymbol{Y}^{(t)}\neq\boldsymbol{Y}^{(t-1)}$ and $t<max_epoch$ do

④ $t=t+1$;

⑤ 利用式(10)得到本轮迭代的全集$\boldsymbol{D}_t$;

⑥ 根据式(11)从$\boldsymbol{D}_t$中抽取当前训练集$\boldsymbol{D}_{\mathrm{train}}^{(t)}$;

⑦ 根据式(12)~(15)在训练集$\boldsymbol{D}_{\mathrm{train}}^{(t)}$上微调模型$M_t$参数, 得到$M_{t+1}$;

⑧ end while

⑨ 利用M_{t+1}根据式(16)预测出$\boldsymbol{D}$的软标签$\hat{\boldsymbol{Y}}^{(t)}$, 根据式(17)(18)求出硬标签$\boldsymbol{Y}^{(t)}$;

⑩ $\boldsymbol{Y}=\boldsymbol{Y}^{(t)}$.

3 实验与分析

本节从实验角度分析所提方法的性能, 实验分析主要包括 4 方面, 分别为聚类性能对比分析、所提方法运行流程可视化分析、参数敏感性分析和初始方法性能影响分析. 本节首先介绍实验设置, 然后展示并分析实验结果.

3.1 实验设置

1)实验数据

我们在聚类性能对比实验部分使用了来自 UCI 基准数据集的 16 个真实表格数据集, 在可视化阶段使用了一个高斯分布合成数据. 本文使用的数据集信息如表 2 所示.

2)评价指标

本文采用主流的 3 个外部评价指标来量化数据集上的聚类效果, 即准确度(*Acc*)、标准化互信息(*NMI*)、调整兰德指数(*ARI*).

① 准确度(accuracy, *Acc*)

Acc 用于衡量聚类结果标签和真实标签的一致

Table 2 Statistics of the Datasets

表 2 数据集统计

序号	数据集	样本数量	特征数量	类个数
1	zoo	101	16	7
2	iris	150	4	3
3	wine	178	13	3
4	ecoli	336	7	8
5	synthetic-control	600	60	6
6	breast-cancer-wisc	699	9	2
7	energy-y1	768	8	3
8	mammographic	961	5	2
9	oocytes_merluccius_states	1 022	25	3
10	statlog-landsat_test	2 000	36	6
11	image-segmentation_test	2 100	18	7
12	statlog-image	2 310	18	7
13	pendigits_test	3 498	16	9
14	waveform	5 000	21	3
15	twonorm	7 400	20	2
16	pendigits_train	7 494	16	10
17	multi_gaussian	2 331	2	9

程度，公式表示为：

$$Acc=\frac{\sum_{i=1}^{n}\delta(\boldsymbol{y}_i,map(\hat{\boldsymbol{y}}_i))}{n},\tag{19}$$

其中，$\hat{\boldsymbol{y}}_i$，$\boldsymbol{y}_i$分别表示对数据x_i聚类获得的标签和真实标签，n为数据总个数，$\delta(\cdot)$表示指示函数，其定义为：

$$\delta(x,y)=\begin{cases}1, & \text{if } x=y,\\ 0, & \text{其他.}\end{cases}\tag{20}$$

$map(\cdot)$表示置换映射函数，其通过最优的一一映射将任意的簇标签重新分配到固定真实类别标签上，从而建立正确的对应关系，通常采用匈牙利算法(Hungarian algorithm)求解该最优映射.

② 标准化互信息(normalized mutual information, *NMI*)

*NMI*用于度量聚类结果和真实标签的信息共享程度，给定数据集X，假设真实标签为$\boldsymbol{Y}$，预测标签为$\hat{\boldsymbol{Y}}$，*NMI*计算公式为：

$$NMI(\boldsymbol{Y},\hat{\boldsymbol{Y}})=\frac{2\times MI(\boldsymbol{Y},\hat{\boldsymbol{Y}})}{H(\boldsymbol{Y})+H(\hat{\boldsymbol{Y}})},\tag{21}$$

其中$H(\cdot)$表示熵函数，$H(\boldsymbol{Y})$和$H(\hat{\boldsymbol{Y}})$的定义为：

$$H(\boldsymbol{Y})=-\sum_{i=1}^{|Y|}P(i)\times\mathrm{lb}(P(i)),\tag{22}$$

$$H(\hat{\boldsymbol{Y}})=-\sum_{j=1}^{|\hat{Y}|}P(j)\times\mathrm{lb}(P(j)),\tag{23}$$

其中$P(\cdot)$表示边缘概率，$P(i,j)$表示联合概率. 此外，式(21)中的$MI(\cdot)$是预测值和真实值间的互信息，定义为：

$$MI(\boldsymbol{Y},\hat{\boldsymbol{Y}})=\sum_{i=1}^{|Y|}\sum_{j=1}^{|\hat{Y}|}P(i,j)\times\mathrm{lb}\left(\frac{P(i,j)}{P(i)\times P(j)}\right).\tag{24}$$

③ 调整兰德指数(adjusted rand index, *ARI*)

*ARI*用于评估聚类结果标签和真实标签的匹配程度，定义为：

$$ARI=\frac{RI-E(RI)}{\max(RI)-E(RI)},\tag{25}$$

$$RI=\frac{a+b}{\binom{n}{2}},\tag{26}$$

其中，$E(\cdot)$表示期望. a表示在真实标签和聚类结果中均属于同一类的样本对数，b表示在真实标签和聚类结果中均属于不同类的样本对数.

3)实验配置

本文采用的实验配置如下. ① 编程环境：Python 3.12；② 运行环境：64 位 Windows11 操作系统，32 GB 内存，20 核的 Intel® Core™ i7-14700F 处理器，NVIDIA GeForce RTX 4060Ti 显卡，CUDA 版本为 12.1；③主要依赖包版本：torch=2.4.1+cu121，tabpfn=2.0.8，scikit-learn=1.6.1.

参数配置包括：为了实验可重复，我们将涉及随机初始化的聚类算法随机数种子统一设为 42，在迭代推理循环中每一轮随机划分的随机数种子设为当前迭代轮数. 4 组实验的 *max_epoch* 统一设为 17，除参数敏感性分析实验之外，其余实验中$\pi^{(\mathrm{I})}$和$\pi^{(\mathrm{II})}$分别设为 0.5 和 0.4，对于每种基于本文模型的改进算法，我们用相同的随机数种子值 42 初始化基础模型 TabPFN.

3.2 对比分析

对于每种基线聚类算法，我们在表格数据集 1~16 上对比基于本文模型的改进方法聚类效果和直接聚类的区别，用 3 个外部评价指标度量聚类性能.

我们选取了 7 种代表性的基线聚类算法，包括：1)基于划分的聚类算法——k-means 聚类(KM)；2)基于层次的聚类算法——凝聚层次聚类[6](agglomerative clustering (Agg))；3)基于图的聚类算法——谱聚类[7](spectral clustering(Spec))；4)基于模型的聚类算法——GMM 聚类[28](Gaussian mixture model clustering (GMM))；5)基于信息传播的聚类算法——AP 聚类[33](affinity propagation clustering(AP))；6)基于密度的聚类算法——DP 聚类[31](density-peak clustering(DP))；

7)基于模糊分类的聚类算法——FCM 聚类[46](fuzzy c-means clustering(FCM)). 用加前缀 T 的名称代表相应基线算法基于本文模型的改进方法.

为了减少随机性的影响，我们将基线聚类算法中涉及随机初始化的算法重复 30 次，随机数种子设为当前重复次数，最终评价指标为多次运行结果的均值. 我们对于每种基线聚类方法分别设置实验组并单独绘制表格，统计在所用 16 个实验数据集上基于本文模型改进后的聚类方法和原算法的聚类指标并计算平均值，通过对比每个实验组上改进前后方法的性能变化来验证本文模型的优越性.

表 3 展示了 k-means 聚类算法(KM)和基于 k-means 改进方法(T-KM)在每个数据上的聚类性能. 可以看到，在几乎所有数据集上，T-KM 指标性能相比 KM 指标性能取得了提升，且平均 *Acc*、*NMI*、*ARI* 指标均有提升.

进一步，我们通过图 2~4 展示了本文改进方法基于 7 种聚类算法在 16 组数据上的整体性能. 具体而言，根据所有数据集上聚类指标的均值和方差绘制的水平条形对比图，纵坐标列出了全部 7 种基线聚类算法及其改进方法，横坐标分别为相应方法在数据集 1~16 上聚类结果的 *Acc*、*NMI* 和 *ARI* 均值，条形图右侧末端向左右等距离延伸的横线位置对应指标为均值±方差.

从图 2~4 可直观看出，改进方法在所有数据集上

Table 3 Comparison of Method Evaluation Indicators Based on k-means Clustering

表 3 基于 k-means 聚类的方法评价指标比较

数据集序号	*Acc*		*NMI*		*ARI*	
	KM	T-KM	KM	T-KM	KM	T-KM
1	0.808 6	**0.815 5**	**0.597 9**	0.597 4	**0.452 4**	0.449 4
2	0.824 2	**0.873 3**	0.652 7	**0.706 0**	0.611 0	**0.688 2**
3	**0.885 0**	0.881 2	0.828 4	**0.845 0**	0.744 3	**0.846 0**
4	0.976 8	**0.976 9**	0.841 1	**0.842 1**	0.909 3	**0.909 7**
5	0.950 6	**0.955 1**	0.839 3	**0.852 8**	0.854 4	**0.868 7**
6	0.954 2	**0.967 1**	0.716 5	**0.779 9**	0.823 4	**0.871 5**
7	0.718 4	**0.758 3**	0.740 1	**0.800 9**	0.584 7	**0.647 3**
8	0.697 8	**0.704 7**	0.685 5	**0.717 7**	0.526 3	**0.537 9**
9	0.651 0	**0.667 6**	0.615 5	**0.628 1**	0.481 7	**0.504 8**
10	0.789 2	**0.791 9**	0.264 6	**0.269 7**	0.333 8	**0.340 1**
11	0.729 3	**0.738 0**	0.614 3	**0.615 7**	0.519 9	**0.535 5**
12	0.654 0	**0.788 3**	0.610 0	**0.702 7**	0.478 3	**0.629 2**
13	0.783 0	**0.807 2**	**0.334 1**	0.329 9	0.286 1	**0.294 0**
14	0.530 6	**0.540 4**	0.364 1	**0.374 3**	0.253 5	**0.259 7**
15	0.740 2	**0.791 3**	0.694 5	**0.757 3**	0.574 0	**0.672 5**
16	**0.812 9**	0.812 5	0.575 1	**0.591 5**	0.571 9	**0.650 9**
平均值	0.781 6	**0.804 3**	0.623 4	**0.650 7**	0.562 8	**0.606 6**

注：黑体数值表示最优值.

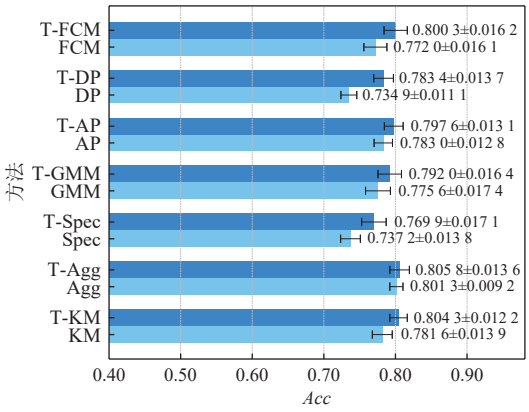


Fig. 2 Horizontal bar chart comparing mean accuracies of the algorithms before and after enhancement

图 2 改进前后算法 *Acc* 均值的水平条形对比图

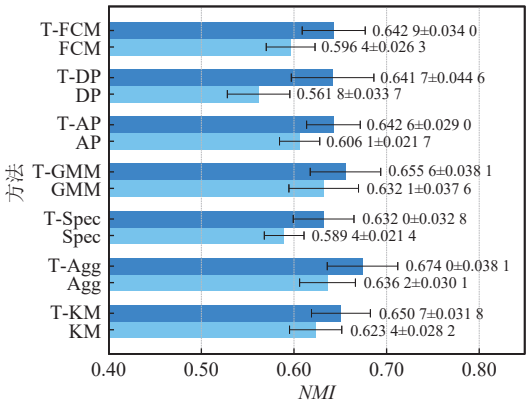


Fig. 3 Horizontal bar chart comparing *NMI* mean values of the algorithms before and after enhancement

图 3 改进前后算法 *NMI* 均值的水平条形对比图

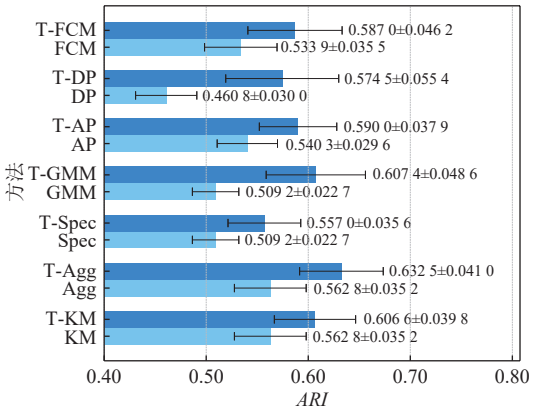


Fig. 4 Horizontal bar chart comparing *ARI* mean values of the algorithms before and after enhancement

图 4 改进前后算法 *ARI* 均值的水平条形对比图

的 3 种聚类指标平均值均超过基线. 此外，基于本文模型改进的聚类方法几乎在所有实验数据集上取得了最优性能. 从平均性能视角，相比 7 种基线算法，所提聚类方法在 *Acc*、*NMI*、*ARI* 指标上均取得了性能提升，在 *NMI*、*ARI* 指标性能提升显著，其中 *Acc* 平均提升超过 2.3 个百分点，*NMI* 平均提升 4.2 个百分点，*ARI* 平均提升 6.8 个百分点. 本节实验结果验证了本文改进方法的有效性，也验证了表格数据基础模型对提升聚类方法性能具有促进作用.

为了验证本文的改进方法是否对原方法性能产生了显著提升，我们对改进前后方法的聚类指标差值做配对样本的单尾 t 检验，并绘制表格如表 4 所示. 这里以 0.05 作为显著性水平(α)阈值，若 $p<0.05$，认为差异具有统计学显著性，可以拒绝原假设(H_0：改进方法性能不超过原方法)，支持备择假设(H_1：改进方法性能提升)；若 $p\geqslant 0.05$，则认为观察到的差异可能仅由随机波动引起，不具有统计学显著性，无法拒绝原假设. 从表 4 可观察出，在 *NMI* 指标评价下，所有改进方法与原方法均有显著性差异，在 *Acc* 和 *ARI* 指标评价下，不少于 5 种改进方法与原方法表现出显著性差异. 该结果表明本文改进方法可有效提升相应方法的聚类性能.

Table 4 Significance Analysis Results of the Improved Clustering Methods

表 4 改进聚类方法的显著性分析结果

方法	*Acc*		*NMI*		*ARI*	
	P 值	是否显著	P 值	是否显著	P 值	是否显著
T-KM	0.009 2	√	0.001 1	√	0.000 9	√
T-Agg	0.323 6	×	0.009 0	√	0.002 6	√
T-Spec	0.080 5	×	0.049 2	√	0.081 2	×
T-GMM	0.006 5	√	0.027 9	√	0.010 7	√
T-AP	0.002 1	√	0.000 9	√	0.036 2	√
T-DP	0.015 3	√	0.002 9	√	0.000 9	√
T-FCM	0.003 5	√	0.002 4	√	0.007 8	√

注：“√”表示显著；“×”表示不显著.

在运行效率方面，即便 TabPFN 模型是目前效率最高的有监督表格数据学习模型，由于需要多次微调基础 TabPFN 模型并在所有样本上进行推理，所以会增加额外的时间消耗. 但是，对于时间复杂度较高的聚类算法，如 AP 聚类算法，本文模型在规模较大的数据上仍能提升算法效率. 以基于 AP 聚类的对比实验为例，表 5 展示了 AP 聚类及其改进方法 T-AP 的运行时间，加粗数值标出每个数据集上聚类所用最短运行时间. 从表 5 可看出，在前 6 个数据集上，由于数据集规模较小，AP 聚类所用时间略少于改进方法 T-AP. 在后 10 个数据集上，随着数据规模的增大，基于本文模型改进的 AP 聚类运行时间明显少于原算法. 综上分析，当基线聚类算法计算复杂度偏高时，基于本文模型改进的聚类方法在规模较大的表格数据集上具有效率优势.

Table 5 Comparison of Running Time of Methods Based on AP Clustering

表 5 基于 AP 聚类的方法运行时间比较 s

数据集序号	AP	T-AP
1	**0.35**	0.52
2	**0.26**	1.06
3	**0.48**	2.63
4	**0.93**	3.09
5	**3.86**	8.75
6	**5.23**	5.46
7	11.79	**1.81**
8	11.22	**3.09**
9	11.54	**11.35**
10	69.57	**32.68**
11	130.89	**34.62**
12	118.87	**34.73**
13	418.62	**80.84**
14	1 174.49	**281.02**
15	3 152.6	**566.93**
16	3 179.36	**543.14**

注：黑体值表示每个数据集上聚类所用最短运行时间.

3.3 运行机制可视化分析

为了直观呈现改进方法的运行机制，本节在合成的二维高斯分布数据上进行了可视化实验. 表 2 中数据集 17 为该部分实验所使用数据集，其类别分布如图 5 所示，不同类簇的点用不同颜色呈现.

图 6、图 7 展示了在该数据集上基于本文模型的改进 FCM 聚类方法迭代过程中的聚类结果，分别对应第 13 轮训练集上的预测结果和第 13 轮全集上的预测结果. 作为对比，我们展示了 FCM 算法在该数据上的聚类结果，如图 8 所示. 从可视化分析实验结果可以看到，在处理此类簇边界模糊、目标类簇较多的表格数据时，本文模型仍能在较少迭代轮次内捕获到理想分布，相比基线算法对局部簇边界的处理更加精确.

3.4 参数敏感性分析

为了探究 2 个划分比例对实验结果的影响，我们进行参数敏感性分析，改变两阶段随机划分训练集

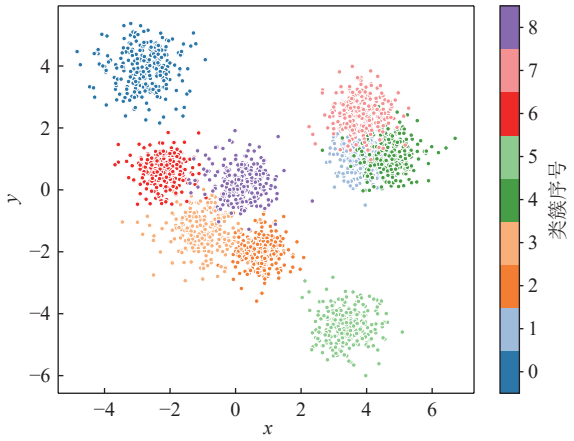


Fig. 5 Real labels of dataset 17

图 5 数据集 17 的真实标签

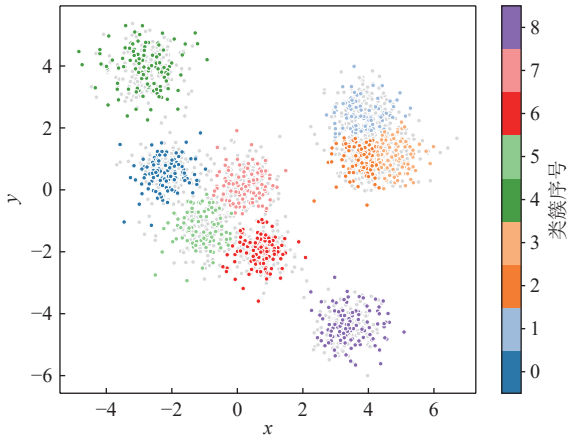


Fig. 6 Pseudo labels of T-FCM on training set

图 6 T-FCM 在训练集上的伪标签

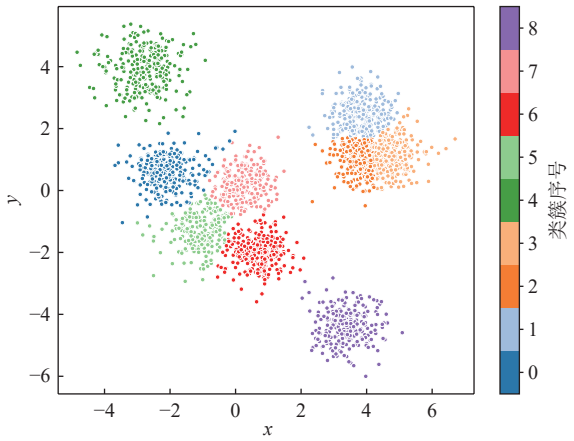


Fig. 7 Clustering results of T-FCM on dataset 17

图 7 T-FCM 对数据集 17 的聚类结果

的比例$\pi^{(\mathrm{I})}$和$\pi^{(\mathrm{II})}$, 记录每种参数组合对应性能及性能最优的超参数.

本节实验设定$\pi^{(\mathrm{I})}$和$\pi^{(\mathrm{II})}$的范围在[0.4,0.9]之间, 每隔 0.1 取一个值测试, 用两重循环遍历$\pi^{(\mathrm{I})}$和$\pi^{(\mathrm{II})}$的

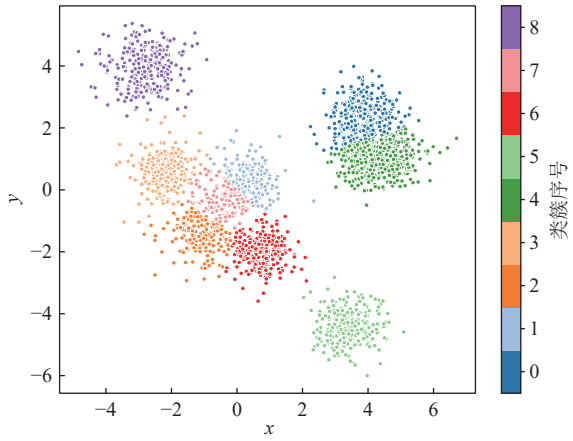


Fig. 8 clustering results of FCM on dataset 17

图 8 FCM 对数据集 17 的聚类结果

所有取值共 36 种组合. 我们将 a 值($\pi^{(\mathrm{I})}$)作为 x 轴, b 值($\pi^{(\mathrm{II})}$)作为 y 轴, 将不同 a、b 值组合聚类的评价指标作为 z 轴, 改变 2 个超参数来绘制柱状图, 观察它们对聚类指标 *Acc*、*NMI*、*ARI* 的影响. 我们统一选定基于本文模型改进的 k-means 聚类方法, 绘制了 15 张参数敏感性分析图, 分别刻画在数据集 1~5 上该算法的 3 种聚类指标随参数的变化情况, 每 3 张图对应 1 个数据集. 为了行文简洁, 这里只展示数据集 1~4 上改进 k-means 聚类方法的 *NMI* 值随参数变化图, 如图 9 所示. 由于参数$\pi^{(\mathrm{I})}$,$\pi^{(\mathrm{II})}$在取值分别为0.5和0.4时, 各数据集上的聚类都能取得较好结果, 且本着尽可能减少采样规模和提高模型运行效率的原则, 我们最终固定参数 $\pi^{(\mathrm{I})}=0.5$, $\pi^{(\mathrm{II})}=0.4$. 从图 9 中可以看出在所选数据集上, $\pi^{(\mathrm{I})}\in\{0.5, 0.6, 0.7\}$, $\pi^{(\mathrm{II})}\in\{0.4, 0.5, 0.6, 0.7, 0.8\}$时, 本文方法的性能相对稳定且良好. 同样地, 在更大的数据集上进行实验, 发现聚类结果较好的参数$\pi^{(\mathrm{I})}$和$\pi^{(\mathrm{II})}$也稳定在上述范围. 此外, 我们发现聚类性能受$\pi^{(\mathrm{I})}$值的变化影响较大, 而对$\pi^{(\mathrm{II})}$的取值不太敏感. 且在最小几个数据集上 2 个参数的变化对实验结果扰动最大, 样本量稍大后, 参数敏感性逐渐趋于平稳.

3.5 初始方法性能影响分析

为了量化改进后聚类性能的具体提升情况, 并分析获取伪标签时所用传统聚类方法的效果对大模型最终预测硬标签的影响, 我们进行性能提升分析. 本节绘制了 3 张散点图, 如图 10 所示: 分别以数据集 1~16 上各传统方法的聚类指标 x 为横坐标, 方法改进前后的聚类指标之差 y 为纵坐标, 并根据散点的位置分布趋势绘制拟合曲线. 由于 y 值体现了基于本文模型的改进方法对于原方法聚类效果的提升, 因此$y=0$即为性能提升的基准线, 观察散点高出 $y=0$

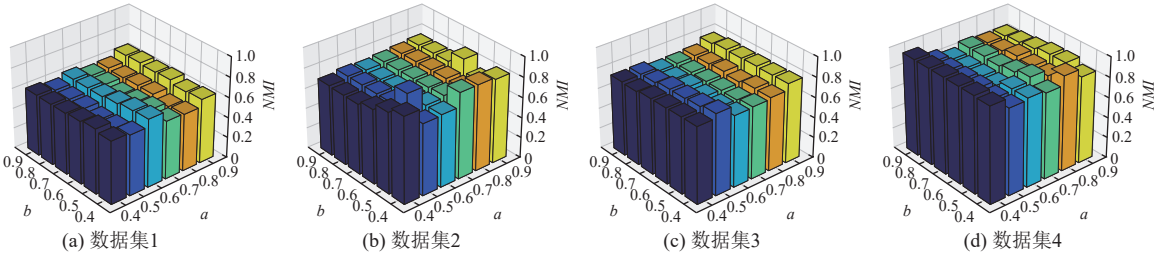


Fig. 9 *NMI* curves of the improved k-means clustering method under different parameter settings on datasets 1−4

图 9 数据集 1~4 上改进 k-means 聚类方法的 *NMI* 值随参数变化图

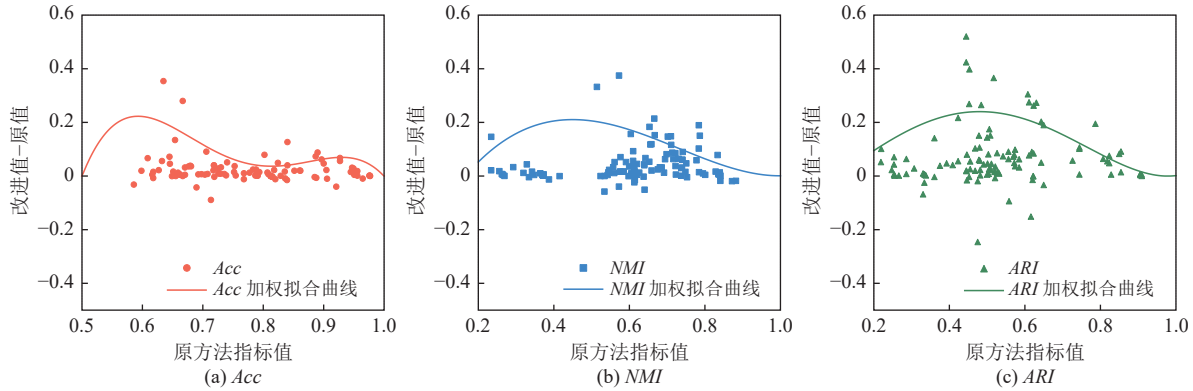


Fig. 10 Scatter plot of improvements in three clustering metrics

图 10 3 种聚类指标提升散点图

的幅度以及在基准线以上的散点密集分布区域即可总结出伪标签聚类效果对最终预测效果的影响规律.

总的来说, 3 张图的散点分布各不相同, 但大致可以按横坐标分成 3 段去理解. 第 1 段: 当 x 值较小时, 高于基准线的散点较少, 少数点略高于基准线, 这表明在本文所用数据集上, 当前伪标签所参考的传统聚类方法性能较差时, 基于本文模型的改进方法对聚类性能的提升效果也较弱, 推测是由于所选基线方法就不擅长捕获该类数据集的簇分布. 第 2 段: 随着 x 值增大, 高于基准线的散点逐渐增多, 散点的 y 值也普遍增大, 这一段的散点大概集中在原方法 *Acc* 和 *NMI* 值为 0.6~0.8、*ARI* 值为 0.4~0.6 的区间, 构成了散点分布的波峰, 这反映了原方法有一定聚类能力时, 本文的改进方法对于聚类性能有较好的提升. 我们认为这得益于 TabPFN 强大的上下文学习能力和表示拟合能力, 当初始伪标签达到一定可靠程度时, 说明所选传统聚类方法本身就比较适合处理该类数据, 在此基础上使用本文模型改进原方法, 就可以在迭代学习中实现对聚类结果的自我提升和完善, 最大限度发掘基线算法的潜力. 第 3 段: 在最后一小段散点图的末端, 散点的分布逐渐趋近基准线, 这说明当原方法聚类效果已经很好时, 改进方法的提升幅度也较小. 原因是原方法聚类结果已经接近真实标签, 提升空间受限. 综上, 当初始聚类方法性能较强或较弱时, 改进方法性能提升幅度较弱; 当初始聚类方法性能适中时, 改进方法性能提升幅度较强.

4 总结与展望

聚类分析是挖掘无标记数据潜在分组结果的重要方法. 目前, 针对表格数据的聚类分析方法仍然以传统基于距离的聚类算法为主. 深度学习技术和大模型技术强大的表示能力和推理能力在结构化的表格数据处理中仍难以发挥优势. 本文提出了一种基于合成数据预训练基础模型的表格数据聚类方法. 该方法通过在预训练阶段利用传统聚类和基于合成数据预训练基础模型 TabPFN 获取初始伪标签, 在迭代推理阶段基于微调后的 TabPFN 模型循环更新伪标签, 当达到预设迭代次数或伪标签不变时停止更新并将伪标签输出为聚类结果. 在 16 个真实基准表格数据上的实验分析显示, 改进方法有效提升了 7 个代表性聚类算法的聚类性能, 表明改进方法有效利用了基础模型的上下文学习优势和推理能力, 初步验证了表格数据基础模型在表格数据聚类任务上具有一定的促进作用.

未来将进一步关注两方面研究. 一是优化大规

模数据下的并行化推理架构，通过分布式计算解决当前对于大规模数据的效率瓶颈；二是探索将基于表格数据基础模型的方法思路拓展至图结构表格数据与动态表格分析.

作者贡献声明：李培文、李飞江提出了主要方法；李培文、李飞江、王婕婷、钱宇华参与了论文撰写；钱宇华改进了方法并提出修改意见.

参 考 文 献

[1] Nederstigt L J, Aanen S S, Vandic D, et al. FLOPPIES: A framework for large-scale ontology population of product information from tabular data in e-commerce stores[J]. Decision Support Systems, 2014, 59: 296−311

[2] Hassan M R, Al-Insaif S, Hossain M I, et al. A machine learning approach for prediction of pregnancy outcome following IVF treatment[J]. Neural Computing and Applications, 2020, 32(7): 2283−2297

[3] Cao Lijuan, Tay Francis E H. Financial forecasting using support vector machines[J]. Neural Computing and Applications, 2001, 10: 184−192

[4] Schwartz L M, Woloshin S, Welch H G. The drug facts box: Providing consumers with simple tabular data on drug benefit and harm[J]. Medical Decision Making, 2007, 27(5): 655−662

[5] Zhao Wanlei, Deng Chenghao, Ngo C W. K-means: A revisit[J]. Neurocomputing, 2018, 291: 195−206

[6] Ran Xingcheng, Xi Yue, Lu Yonggang, et al. Comprehensive survey on hierarchical clustering algorithms and the recent developments[J]. Artificial Intelligence Review, 2023, 56(8): 8219−8264

[7] Ng A Y, Jordan M I, Weiss Y. On spectral clustering: Analysis and an algorithm[J]. Advances in Neural Information Processing Systems, 2001, 14: 849−856

[8] Reynolds D. Gaussian Mixture Models[M]. Boston, MA: Springer, 2015

[9] Niu Chuang, Shan Hongming, Wang Ge. SPICE: Semantic pseudo-labeling for image clustering[J]. IEEE Transactions on Image Processing, 2022, 31: 7264−7278

[10] Guo Xifeng, Liu Xinwang, Zhu En, et al. Deep clustering with convolutional autoencoders[C]//Proc of 24th Int Conf on Neural Information Processing (ICONIP 2017). Berlin: Springer Verlag, 2017: 373−382

[11] Jing Longlong, Tian Yingli. Self-supervised visual feature learning with deep neural networks: A survey[J]. IEEE Transactions on Pattern Analysis and Machine Intelligence, 2020, 43(11): 4037−4058

[12] Li Zewen, Liu Fan, Yang Wenjie, et al. A survey of convolutional neural networks: Analysis, applications, and prospects[J]. IEEE Transactions on Neural Networks and Learning Systems, 2021, 33(12): 6999−7019

[13] Vaswani A, Shazeer N, Parmar N, et al. Attention is all you need[J]. Advances in Neural Information Processing Systems, 2017, 30: 6000−6010

[14] Koval R, Andrews N, Yan Xifeng. Financial forecasting from textual and tabular time series[C]// Findings of the Association for Computational Linguistics: EMNLP 2024. Stroudsburg, PA: ACL, 2024: 8289−8300

[15] Luo Yuanfei, Wang Mengshuo, Zhou Hao, et al. AutoCross: Automatic feature crossing for tabular data in real-world applications[C]//Proc of the 25th ACM SIGKDD Int Conf on Knowledge Discovery & Data Mining. New York: ACM, 2019: 1936−1945

[16] Zhu Yitan, Brettin T, Xia Fangfang, et al. Converting tabular data into images for deep learning with convolutional neural networks[J]. Scientific Reports, 2021, 11(1): 11325

[17] Refinetti M, Goldt S. The dynamics of representation learning in shallow, non-linear autoencoders[C]//Proc of Int Conf on Machine Learning. Baltimore, Maryland: ML Research Press, 2022: 18499−18519

[18] Ma Jiawei, Huang P, Xie Saining, et al. MoDE: CLIP data experts via clustering[C]//Proc of the IEEE Computer Society Conf on Computer Vision and Pattern Recognition. Los Alamitos, CA: IEEE Computer Society, 2024: 26344−26353

[19] Draisbach U, Christen P, Naumann F. Transforming pairwise duplicates to entity clusters for high-quality duplicate detection[J]. Journal of Data and Information Quality, 2019, 12(1): 1−30

[20] Rombach R, Blattmann A, Lorenz D, et al. High-resolution image synthesis with latent diffusion models[C]//Proc of the IEEE Computer Society Conf on Computer Vision and Pattern Recognition. Los Alamitos, CA: IEEE Computer Society, 2022: 10674−10685

[21] Wang Hanxuan, Lu Na, Wang Zixuan, et al. Self-correcting clustering[J]. IEEE Transactions on Knowledge and Data Engineering, 2025, 37(3): 1439−1454

[22] Hollmann N, Müller S, Purucker L, et al. Accurate predictions on small data with a tabular foundation model[J]. Nature, 2025, 637(8045): 319−326

[23] Ma Junwei, Dankar A, Stein G, et al. TabPFGen-tabular data generation with TabPFN[J]. arXiv preprint, arXiv: 2406.05216, 2024

[24] Ma Junwei, Thomas V, Hosseinzadeh R, et al. TabDPT: Scaling tabular foundation models[J]. arXiv preprint, arXiv: 2410.18164, 2024

[25] Ma Junwei, Thomas V, Yu Guangwei, et al. In-context data distillation with TabPFN[J]. arXiv preprint, arXiv: 2402.06971, 2024

[26] Wydmański W, Movsum-zada U, Tabor J, et al. Vis TabNet: Adapting vision transformers for tabular data[C]//Proc of the 2025 SIAM Int Conf on Data Mining (SDM). Alexandria, VA: Society for Industrial and Applied Mathematics, 2025: 497−506

[27] Hoo S B, Müller S, Salinas D, et al. The tabular foundation model TabPFN outperforms specialized time series forecasting models based on simple features[J]. arXiv preprint, arXiv: 2501.02945, 2025

[28] He Xiaofei, Cai Deng, Shao Yuanlong, et al. Laplacian regularized Gaussian mixture model for data clustering[J]. IEEE Transactions on Knowledge and Data Engineering, 2011, 23(9): 1406−1418

[29] Moon T K. The expectation-maximization algorithm[J]. IEEE Signal

Processing Magazine, 1996, 13(6): 47−60

[30] Schubert E, Sander J, Ester M, et al. DBSCAN revisited, revisited: Why and how you should (still) use DBSCAN[J]. ACM Transactions on Database Systems, 2017, 42(3): 1−21

[31] Wang Yizhang, Qian Jiaxin, Hassan M, et al. Density peak clustering algorithms: A review on the decade 2014–2023[J]. Expert Systems with Applications, 2024, 238: 121860

[32] Agrawal K P, Garg S, Sharma S, et al. Development and validation of OPTICS based spatio-temporal clustering technique[J]. Information Sciences, 2016, 369: 388−401

[33] Wang Kaijun, Zhang Junying, Li Dan, et al. Adaptive affinity propagation clustering[J]. arXiv preprint, arXiv: 0805.1096, 2008

[34] Ren Yazhou, Pu Jingyu, Yang Zhimeng, et al. Deep clustering: A comprehensive survey[J]. IEEE Transactions on Neural Networks and Learning Systems, 2025, 36(4): 5858−5878

[35] Zhou Sheng, Xu Hongjia, Zheng Zhuonan, et al. A comprehensive survey on deep clustering: Taxonomy, challenges, and future directions[J]. ACM Computing Surveys, 2024, 57(3): 1−38

[36] Wang Qihong, Jia Hongjie, Huang Longxia, et al. Semantic contrastive clustering with federated data augmentation[J]. Journal of Computer Research and Development, 2024, 61(6): 1511−1524 (in Chinese)
(王气洪, 贾洪杰, 黄龙霞, 等. 联合数据增强的语义对比聚类[J]. 计算机研究与发展, 2024, 61(6): 1511−1524)

[37] Guan Renchu, Zhang Hao, Liang Yanchun, et al. Deep feature-based text clustering and its explanation[J]. IEEE Transactions on Knowledge and Data Engineering, 2022, 34(8): 3669−3680

[38] Tsitsulin A, Palowitch J, Perozzi B, et al. Graph clustering with graph neural networks[J]. Journal of Machine Learning Research, 2023, 24(127): 1−21

[39] Lin Yuxiu, Liu Hui, Yu Xiao, et al. A multi-view unified representation learning network for subspace clustering[J]. Journal of Computer Research and Development, 2025, 62(5): 1248−1261 (in Chinese)
(林毓秀, 刘慧, 于晓, 等. 面向子空间聚类的多视图统一表示学习网络[J]. 计算机研究与发展, 2025, 62(5): 1248−1261)

[40] Zhao Yuhan, Chen Songcan. Multi-view clustering algorithm via eliminating the complementarity controversy[J]. Journal of Computer Research and Development, 2025, 62(5): 1216−1225 (in Chinese)
(赵玉涵, 陈松灿. 消除互补性争议的多视图聚类算法[J]. 计算机研究与发展, 2025, 62(5): 1216−1225)

[41] Qian Luoxiong, Chen Mei, Ma Xueyan, et al. Multi-view clustering based on adaptive tensor singular value shrinkage[J]. Journal of Computer Research and Development, 2025, 62(3): 733−750 (in Chinese)
(钱罗雄, 陈梅, 马学艳, 等. 自适应张量奇异值收缩的多视角聚类[J]. 计算机研究与发展, 2025, 62(3): 733−750)

[42] Kim J, Huang Chao, Liu Xin. An empirical study on impact of label noise on synthetic tabular data generation[J]. Machine Learning, 2025, 114(3): 1234−1256

[43] Bengio Y, Courville A, Vincent P. Representation learning: A review and new perspectives[J]. IEEE Transactions on Pattern Analysis and Machine Intelligence, 2013, 35(8): 1798−1828

[44] Müller S, Hollmann N, Arango S P, et al. Transformers can do Bayesian inference[J]. arXiv preprint, arXiv: 2112.10510, 2021

[45] Hollmann N, Müller S, Eggensperger K, et al. TabPFN: A Transformer that solves small tabular classification problems in a second[J]. arXiv preprint, arXiv: 2207.01848, 2022

[46] Bezdek J C, Ehrlich R, Full W. FCM: The fuzzy c-means clustering algorithm[J]. Computers & Geosciences, 1984, 10(2-3): 191−203



Li Peiwen, born in 2002. Master candidate. Student member of CCF. His main research interest includes machine learning.
李培文, 2002 年生. 硕士研究生. CCF 学生会员. 主要研究方向为机器学习.



Li Feijiang, born in 1990. PhD, associate professor. Member of CCF. His main research interests include machine learning and unsupervised swarm intelligence.
李飞江, 1990 年生. 博士, 副教授. CCF 会员. 主要研究方向为机器学习、无监督群体智能.



Wang Jieting, born in 1991. PhD, associate professor. Member of CCF. Her main research interests include statistical machine learning and PAC learning theory.
王婕婷, 1991 年生. 博士, 副教授. CCF 会员. 主要研究方向为统计机器学习、可学习理论.



Qian Yuhua, born in 1976. PhD, professor. Member of CCF. His main research interests include artificial intelligence, big data, data mining, and machine learning.
钱宇华, 1976 年生. 博士, 教授. CCF 会员. 主要研究方向为人工智能、大数据、数据挖掘、机器学习.