

数据清洗与下游聚类的自动化协同优化及其影响机理研究

2025 年 4 月 23 日

摘要

在许多无监督学习任务中，现有的数据清洗与聚类技术已能在一定程度上降低噪声与缺失值带来的影响，但仍难以同时兼顾多样化清洗策略与聚类算法在大规模、高维数据中的协同需求。为进一步提升聚类质量与自动化效率，本文提出了一种清洗-聚类协同优化框架：通过多标签学习模型将数据的特征向量映射到最优或近优的“清洗-聚类”管线组合，从而在大幅减少搜索空间的同时确保较高的聚类性能。在提供解决方案的同时，我们还系统地分析了清洗对聚类算法运行及评价指标的影响，并通过清洗准确度与聚类指标的关联性研究，揭示了在不同数据特征与错误率下清洗策略对聚类收益的关键影响因素。基于对 60 个公开数据集的大规模实验发现，不同数据特征会显著影响清洗与聚类的适配性，例如 Raha-Baran + HC 组合在高维、多特征数据上较为稳健，而 mode + DBSCAN 在低维数值数据上对噪声表现出极端敏感。通过该框架的自动化推荐与筛选，在部分场景下实现了平均 5.83 倍的搜索加速，并在保证聚类质量的同时取得了 19.20% 的平均提升率。研究结果表明，该方法对多样化数据具有一定的稳健性与可扩展性，为噪声较高、规模较大的真实数据环境提供了切实可行的聚类优化方案。

1 引言

在大数据与人工智能的快速发展背景下，无监督学习（如聚类分析）在医疗、金融及工业物联网等众多领域发挥着日益重要的作用 [1-3]。例如，在医疗场景中，通过聚类可从病患数据中挖掘潜在群组，为个性化诊疗提供决策支持 [4]；在金融场景中，聚类方法可以帮助区分用户信用类别、增强客户预期回报的信心等 [5]。已有研究在聚类算法改进和可视化等方面取得了显著成果，常用的方法包括 K-Means 及其变体 [6]、基于密度的 DBSCAN 以及层次聚类、图聚类 [7] 等，这些方法为不同数据形态提供了有效的划分策略。与此同时，数据清洗技术（例如缺失值填补、异常值检测、错误值纠正）也在学术界与工业界获得广泛应用，用以降低噪声影响和提高数据质量 [8-10]。

然而，在无监督学习场景中，数据质量的影响往往更为突出。与分类或回归等有监督学习相比，聚类对于数据分布的依赖更强，一旦噪声、缺失值或错误值的比例较高，就可能破坏簇结构与真实分布之间的对应关系 [11]，从而对模式挖掘和决策支持造成不可忽视的干扰。虽然已有清洗方法在减少噪声方面效果显著，但过度或不当的清洗有时反而扭曲了关键特征 [12]；此外，不同的聚类算法对数据缺陷的敏感度各异 [13]，若仅侧重于数据清洗或聚类算法单方面的优化，往往难以协调两者之间的相互作用，从而难以获得整体最优的策略。

为解决上述问题，研究者逐步认识到“清洗策略 + 聚类算法 + 超参数”一体化管线的重要性 [14]。这种做法能在保证数据分布尽量真实的同时，为不同数据集的特性“量体裁衣”地提供最佳策略。但由于管线搜索空间常呈指数级增长，且无监督场景缺乏显式标签指导，仅靠人工穷举或简单试验往往难以在可接受时间内完成参数寻优。近年来，自动化机器学习（AutoML）在有监督学习领域已呈现出显著优势 [15]，不仅能自动选择模型结构及超参数，还能优化特征工程 [16,17]。然而，大部分 AutoML 研究集中于分类或回归任务 [18]，对无监督学习特别是“清洗 + 聚类”协同自动化的探索仍相对有限 [19]。这为我们带来了新的机遇与挑战：能否将数据质量与无监督聚类的协同优化思路融入 AutoML 框架，并结合更深层次的机理剖析，在大规模及多场景下实现高效且可解释的自动管线搜索。

在此过程中，深度理解清洗操作如何影响下游聚类算法至关重要：只有梳理清楚清洗对聚类影响的机制和环节，才能在自动化管线中有针对性地选择或组合清洗策略与聚类方法。为此，我们将清洗的影响拆分为四个关键层面：(1) 清洗对数据集准确度的影响（具体表现在对错误的修复程度），(2) 清洗对聚类算法内部过程的干预（如迭代收敛路径、核心点判定），(3) 清洗对聚类结果指标（如 Silhouette、DB 指数）的影响，(4) 清洗对聚类超参数选择（如 K-Means 的 k 值、DBSCAN 的 ϵ ）的影响。通过从“源数据集—算法运行过程—聚类指标—超参数调优”四步逐层深入地分析，我们不仅能更好地解释清洗策略与聚类性能之间可能存在的因果关联，也为自动化管线的配置与调优提供了更丰富的理论支撑。

基于上述背景与需求，本文针对“数据清洗与下游聚类协同优化”这一交叉方向，提出了一种新的自动化管线模型，并进一步从理论层面对“清洗操作如何影响聚类”展开深度剖析：一方面，借助多标签学习模型将多种清洗策略、聚类算法及其超参数统一纳入搜索空间，在离线阶段学习“数据特征到优选方案”的映射关系；另一方面，系统研究清洗准确度与聚类评价指标之间的关联，为理解清洗操作如何干预聚类运行过程及结果指标提供实证依据。这样，当面对新的数据集时，系统能快速推荐若干最优或近优组合，大幅缩减搜索规模，并根据清洗机理的分析获得更全面的可解释性与稳定性。

本研究的主要贡献包括：

1. 系统性地评估“清洗策略 × 聚类算法”组合的协同表现

基于 60 个具备多元质量问题的公开数据集，我们深入研究了不同噪声水平、错误率及规模下的 8 种清洗策略和 6 种聚类算法，对其组合在聚类质量、极端案例和时间开销方面进行了量化与比较。该评估不仅提供了对现有清洗-聚类方案适配度的系统认识，也为后续管线设计提供了实用参考。

2. 提出基于管线思维的协同优化框架

将“数据清洗 + 聚类 + 超参数”作为一个整体管线（Pipeline），并结合实验结果总结出多种针对性建议，帮助研究者在实际场景中有的放矢地进行策略选择，避免仅在单一端的优化而忽略全局效果。

3. 构建并验证了一个完整的自动化管线优化模型

我们引入多标签学习来捕捉“数据特征与最优清洗-聚类组合”之间的关系，大幅减少了管线搜索空间。在多个数据集上验证表明，自动化模型通常可获得 3× 以上的加速，同时保持较高的聚类准确度，证明了将 AutoML 思路拓展到无监督学习领域的可行性与有效性。

4. 从理论角度剖析清洗对聚类的影响机理

通过对清洗准确度（EDR、Precision、Recall、F1）与聚类评价指标（Silhouette、DB、Combined Score）的关联研究，系统揭示了不同数据特征和错误率水平下清洗操作如何影响聚类过程，从而为参数调优与动态适配提供了更深层的参考。

5. 为多样化数据场景的聚类优化提供可迁移路径

通过对损失率、加速比等指标的度量，我们量化了自动化管线在平衡质量与效率方面的潜力，为工业领域部署该思路奠定了实践基础，也为研究者进一步探索清洗与聚类协同优化的动态适配、在线更新等提供了方向。

6. 按错误类型细粒度量清洗对聚类的收益，并将该洞察用于改进 AutoML 搜索策略

本文首次在大规模实验中，将缺失、格式错误、离群值等不同错误类型对聚类流程的影响拆解分析；并把这些类型-级指标注入多标签学习模型，以动态收缩搜索空间，从而进一步提升 AutoML 推荐的精度与效率。

我们的工作不仅加深了无监督学习场景下对数据清洗策略与聚类算法协同作用的理解，也为自动化机器学习在更广泛领域的落地探索了新路径。下文将依次介绍相关工作（第 2 章）、问题与模型定义（第 3 章）、提出的方法与实现细节（第 4 章），并用大规模实验量化其有效性与适用性（第 5 章）。第 6 章将进一步基于清洗与聚类的算法原理，解释其优势与局限，并讨论对 AutoML 强化搜索的启示。

2 相关工作

为了更深入理解“清洗策略与聚类算法协同优化”在不同场景下的研究现状，本文从以下三个方面回顾相关工作：首先，探讨数据清洗与数据质量管理的相关方法；其次，分析主要聚类算法的原理及其优化思路；最后，梳理自动化机器学习（AutoML）在无监督学习场景中的研究进展与应用探索。

2.1 数据清洗与数据质量管理

数据清洗旨在识别并修复各种数据缺陷（如缺失值、噪声、重复记录或错误值），是提升数据整体质量的重要途径，已有研究在统计方法和机器学习方法方面均取得了丰富成果。例如，早期工作主要依赖众数/均值填补 [20] 或规则驱动的异常值检测 [21,22]，在处理缺失值和简单错误时比较高效；后续研究则引入高级方法，如概率图模型 [23]，主动学习 [24,25]，

神经网络 [26] 等，以应对更复杂的错误类型。部分工作还引入了上下文约束或知识图谱 [27,28]，对特定领域（如医疗、经济数据）的不一致或罕见值进行更有针对性的纠正。

与此同时，研究者也认识到过度或不当清洗可能使原本有价值的异常点被误删或被扭曲 [12]。在有监督学习场景下，数据清洗常可借助标签对比来区分“真正有意义的异常”与“噪声性错误” [29]；然而在无监督场景中，缺乏标签指导，清洗策略一旦过于保守或激进，就会对后续的聚类分析产生不可预测的影响。这些研究进展表明，数据清洗方法的选择与配置应当与下游分析任务（如聚类）紧密结合，而非单独孤立地追求“最干净”的数据 [30]。这也为我们随后探讨的“清洗与聚类协同优化”提供了重要动机。

2.2 聚类算法及其改进

聚类作为典型的无监督学习方法，已在图像识别、文本挖掘、用户分群等领域中得到了广泛应用。现有聚类算法大体可分为基于质心（如 K-Means 及其变体 [6,31,32]）、基于密度（如 DBSCAN [33,34]，OPTICS [35–37]）与层次聚类 [38] 三类。不同算法在簇形状、噪声耐受度、计算复杂度等方面各具优势 [13]。

在面对不完美数据时，上述聚类算法往往对异常值和缺失值表现出不同的敏感度。例如，少量异常点被 K-Means 视为远离中心的“噪声”，可在重新计算均值时抵消 [39]；但若这些点在 DBSCAN 的邻域定义中被错误识别，就可能导致过度分割 [40]。部分工作试图在算法内部引入鲁棒性机制，如改进距离度量或引入加权方案 [41]，但大多仍需事先对数据进行相对独立的预处理，缺乏将“清洗策略”与“聚类算法”放在同一管线中统筹考量，在更复杂的高噪声场景中难以取得较好的聚类结果。

2.3 AutoML 与无监督场景的探索

近年来，AutoML 框架（如 Auto-sklearn [42]、TPOT [43,44]、H2O AutoML）已在有监督学习任务（分类、回归）中展现出卓越的自动化建模与超参数优化能力。典型方法主要依赖贝叶斯优化、遗传算法 [45] 等技术，在预定义的搜索空间内高效探索最优模型配置。然而，这些框架主要针对有监督任务设计，难以直接适用于无监督学习，尤其在聚类任务中面临诸多挑战 [19]。在少量试图探索 AutoML 在无监督学习上应用的研究中（表 1），其方法主要聚焦于聚类算法选择与超参数优化，部分工作结合初步聚类或 PCA 降维以降低特征噪声。

| 方法 | 是否包含数据清洗 | 是否端到端 AutoML 模型 |
|--------------------|------------|------------------|
| AutoClust [46] | × 无 | × 仅聚类优化 |
| cSmartML [47] | × 无 | × 仅算法选择 + 超参数优化 |
| MARCO-GE [48] | △ 部分（PCA） | × 主要关注算法推荐 |
| AutoCluster [49] | × 无 | △ 部分端到端（集成学习） |
| TPE-AutoClust [50] | △ 部分（初步聚类） | △ 部分端到端（优化 + 集成） |
| 本文方法 | ✓ 完整数据清洗 | ✓ 端到端自动化管线优化 |

表 1: 当前无监督 AutoML 聚类主要方法对比

然而，当前框架普遍缺乏对数据清洗过程的系统化建模，并且大多关注聚类算法推荐和超参数选择，对不同数据特征如何影响最优清洗-聚类组合的探讨缺少深入量化。同时，多数研究停留在验证最终的聚类效果，缺乏对清洗如何干预聚类内部过程以及清洗-聚类联合机理层面的系统剖析。现有研究尚未完全构建一个能够端到端集成数据清洗、聚类方法与超参数优化的自动化框架，以系统性地应对高噪声、多特征及多样化数据带来的挑战。

2.4 错误类型粒度的清洗-聚类研究

2.5 小结与差异

综上所述，数据清洗与数据质量管理在过往研究中已经形成了相对成熟的方法论与工具库；聚类算法也拥有多种变体和改进，能适应不同分布和应用需求；AutoML 技术在有监督学习里逐步成熟，显著简化了模型选择与调参工作。然而，在无监督场景：(i) 缺少端到端融合“清洗 + 聚类 + 超参数”的自动化框架；(ii) 现有无监督 AutoML 几乎未将清洗准确度或错误类型特征显式纳入搜索维度；(iii) 对多错误类型交叠时清洗-聚类协同机理的系统性刻画不足。

针对上述研究空缺，本文提出一种基于多标签学习的管线化 AutoML 框架：首先，将“数据清洗 + 聚类 + 超参数”视作统一搜索空间，通过在离线阶段学习“数据特征 → 优选清洗-聚类组合”的映射，以显著缩减候选子空间，同时保持

聚类质量。其次，在运行过程中持续记录清洗准确度（区分缺失、离群等错误类型）和聚类迭代轨迹，用于从错误类型粒度量化清洗带来的聚类收益，并结合各聚类算法原理进一步剖析其优势与局限，全面回答“清洗何时、如何、在多大程度上真正提升下游聚类”的问题。

3 问题定义与挑战

在前两章中，我们从数据清洗、聚类算法及 AutoML 的角度梳理了“清洗-聚类”协同优化的研究背景与现状。虽然各自已有成熟的理论与实践成果，但在高维、噪声大或缺失率高的多样化数据中，清洗与聚类之间的紧密衔接和作用机制仍存在较大探索空间。若要实现真正的端到端自动化优化，就需要将清洗策略与聚类算法有机整合，量化并解释清洗对聚类性能及内部过程的影响机制，为不同场景下的方案选择提供可行性的依据。为在理论与实践层面更好地刻画该问题，下面我们将首先给出形式化的数学模型与问题定义，然后进一步分析其中的主要挑战与难点，为后续方法设计与实验研究奠定基础。

3.1 数学模型与形式化定义

为在理论与应用中更好地理解并解决“数据清洗与聚类算法”的协同优化，本小节对核心概念进行形式化定义，并建立相应的评价体系。

3.1.1 核心概念与变量定义

令 D 表示待处理的数据集，其中可同时存在错误值（Error）、异常值（Anomaly）以及缺失值（Missing）等质量问题。为刻画这些问题与数据规模的差异，定义特征向量：

$$\mathbf{x}(D) = (\text{ErrorRate}(D), \text{AnomalyRate}(D), \text{MissingRate}(D), m, n), \quad (3.1)$$

其中 $\text{ErrorRate}(D)$ 、 $\text{AnomalyRate}(D)$ 与 $\text{MissingRate}(D)$ 分别表示数据集中错误值、异常值以及缺失值的相对比例， m 和 n 分别为数据的特征维度与样本规模。该向量不仅能在不同场景下进行对比，也为后续的自动化模型提供可学习的输入。

记 \mathcal{C} 为数据清洗方法的集合（如缺失值插补、异常值剔除、错误值纠正等）， \mathcal{H} 为聚类算法集合（如 K-Means、DBSCAN、层次聚类等）， \mathcal{P} 为聚类算法的超参数空间。将一个具体的清洗方法 c 、聚类算法 h 及其超参数 θ 组合成清洗-聚类策略：

$$\omega = (c, h, \theta), \quad (3.2)$$

所有可行策略的笛卡尔积构成初始搜索空间：

$$\Omega = \mathcal{C} \times \mathcal{H} \times \mathcal{P}. \quad (3.3)$$

此时，如何在如此庞大的 Ω 中高效找到适配度高的 ω 即是后续的研究重点。

3.1.2 评价系统与最优方案

为了准确衡量任意策略 $\omega \in \Omega$ 在数据集 D 上的聚类质量（或适配性），通常采用若干无监督评价指标加以综合。本文主要使用 Davie-Bouldin（DB）指数 [51] 与轮廓系数（Silhouette）[52] 这两类典型指标，并线性组合为综合得分：

$$S(D, \omega) = \alpha \cdot [-DB(D, \omega)] + \beta \cdot \text{Sil}(D, \omega), \quad (3.4)$$

其中 $\alpha, \beta > 0$ 为可调权重， $DB(\cdot)$ 越低表明簇内紧凑度与簇间分离度越理想，而 $\text{Sil}(\cdot)$ 越高代表类内相似度越高、类间差异越大。若从聚类精度的角度出发，给定数据集 D 的最优策略可表示为：

$$\omega^*(D) = \arg \max_{\omega \in \Omega} S(D, \omega). \quad (3.5)$$

然而，若要在全量空间 Ω 上评估每个策略 ω ，往往需要极高的时间成本。为此我们定义优化子空间 $\Omega'(D) \subseteq \Omega$ ，仅在该空间中执行策略评估，以降低计算负担。记评估单个策略的耗时为 $T(D, \omega)$ ，则完整搜索与缩减搜索的总耗时分别为：

$$T_{\text{original}}(D) = \sum_{\omega \in \Omega} T(D, \omega), \quad T_{\text{reduced}}(D) = \sum_{\omega \in \Omega'(D)} T(D, \omega). \quad (3.6)$$

我们的目标是通过一个合适的 $\Omega'(D)$ ，在保证较高聚类质量的同时显著减少评估代价。

为量化“性能表现”与“时间加速”之间的平衡，我们引入损失率（或提高率）和综合加速比两个概念：

$$\eta(D) = 1 - \frac{\bar{S}(\Omega'(D))}{\bar{S}(\Omega)}, \quad (3.7)$$

$$\mathcal{A}(D) = (1 - \eta(D)) \times \frac{T_{\text{original}}(D)}{T_{\text{reduced}}(D)}, \quad (3.8)$$

其中 $\bar{S}(\Omega)$ 表示在完整空间上搜索所得的平均得分， $\bar{S}(\Omega'(D))$ 表示子空间 $\Omega'(D)$ 上的平均得分， $\eta(D)$ 越接近 0 表示缩减空间后带来的聚类性能损失越小，而 $\mathcal{A}(D)$ 越大则表示加速效果越显著。

3.1.3 从数据特征到优选策略的映射

在实际应用场景中，不同数据集 D 往往具有差异显著的质量特征（如 ErrorRate、MissingRate、NoiseRate 等）。这些特征会显著影响清洗-聚类策略的效果，使得某些组合对特定类型的数据表现更优。若能根据 $\mathbf{x}(D)$ （参考式(3.1)）提前预测哪些组合更可能获得高分，即可避免对完整搜索空间 Ω 的全量评估。为此，我们引入一个映射函数：

$$G : \mathbf{x}(D) \mapsto \Omega'(D), \quad (3.9)$$

其中 $\Omega'(D) \subseteq \Omega$ 。通过学习训练集中“特征—策略组合”的关联，再在新数据集上借助该映射快速筛选候选方案，最终只需在子空间 $\Omega'(D)$ 中执行搜索，这种基于历史数据的学习策略可极大降低时间成本 [28]。后续章节将介绍如何具体构建并训练这一映射。

3.2 技术难度

在前文对清洗-聚类协同优化的数学模型与评价体系进行阐释之后，本研究的核心目标是如何在有限时间与资源条件下高效筛选出适合大规模或高噪声数据的组合策略。为更好地刻画这一过程并揭示潜在挑战，本文聚焦以下四个关键子问题：

(Q₁) 评估不同清洗-聚类组合在多样化数据特征下的适配性

虽然现有清洗方法和聚类算法选择繁多，但在高维、高错误率（或缺失率）以及复杂噪声场景下，其表现仍难以保证稳定性与最优性。为此，需要系统量化并比较各组合在多种数据特征条件下的优劣，从而为后续策略选择奠定基础。

(Q₂) 构建基于数据特征到优选策略的映射函数

当数据集特征呈高度异质时，单一清洗或聚类方法往往难以达到稳健性能。为应对不同分布特点，本文尝试在离线训练阶段学习 $G(\mathbf{x}(D)) \mapsto \Omega'(D)$ ，依据数据特征向量自动筛选潜在近优的清洗-聚类组合，以实现快速且精准的策略推荐。

(Q₃) 平衡聚类质量与效率，实现在有限时间内逼近最优

大规模数据会大幅提升搜索与评估开销，使实时需求难以满足。如何在缩减搜索空间的同时，维持可控的聚类质量损失，并取得显著加速，是本研究所关注的又一关键挑战。

(Q₄) 深度分析清洗对聚类结果的实际影响机理

本研究将系统考察给定数据集 D 与清洗-聚类策略 ω 时，哪些“有效纠正”决定了聚类得分 S 的形成，并探讨修正更多错误是否必然带来更优聚类表现。此外，还将评估清洗操作对最优超参数选择是否产生系统性偏移，从而为自动化管线的配置和调优提供更深入的理论依据。

(Q₅) 利用清洗-聚类机理知识改写 AutoML 的搜索空间与特征工程

在掌握错误类型对聚类过程/指标的细粒度影响后，如何把这些机理信息反馈到 AutoML：

- 搜索空间收缩——按错误类型显著受益的清洗-聚类组合优先保留，其余策略降权或删除；
- 特征工程增强——将“错误类型分布、修复难度、过程指标”等高阶特征注入多标签模型，提升优选子空间预测的精度与解释性。

该挑战要求把机理洞察真正融入 AutoML 流程，而非仅作为事后分析，从而进一步降低评估成本并提升推荐质量。

围绕上述五个子问题，后续章节将逐一阐述自动化搜索与映射模型的设计思路，并通过大规模实验证明其在多场景下的可行性与性能优势。特别在 (Q₄) 与 (Q₅) 中，我们将结合清洗准确度指标、聚类算法内部过程的跟踪与错误类型的细粒度分析，深入探讨清洗策略如何从数据集、算法过程、评价指标、超参数，以及 AutoML 搜索空间与特征工程等角度共同影响聚类结果，为自动化管线的优化提供可靠的机理支撑。

4 自动化聚类方法

为进一步提高清洗-聚类策略的搜索效率，并同步分析清洗对聚类内部过程与评价指标的影响机理，本节将在第 3 节所述概念的基础上，介绍将数据划分为先验数据与测试数据、使用多标签学习构建映射函数，以及最终实现自动化聚类优化流程的整体方法。该方法是一个面向数据预处理、清洗、聚类与分析的完整端到端系统，不仅通过离线阶段积累的先验知识来缩减搜索空间、在较短时间内找到近优的清洗-聚类组合，更能对清洗操作的准确度及聚类算法的中间过程进行记录 and 可视化分析，以揭示“为何”或“何时”清洗能带来显著的聚类性能提升。

以下是本章节所定义的符号与描述：

| 符号 | 描述 |
|--------------------|---|
| D_{train} | 先验数据集（训练集），用于离线评估和学习先验知识 |
| D_{test} | 测试数据集，用于实际部署和快速优化 |
| K | Top-K 大小，表示在先验阶段选取的前 K 个最优方案 |
| $\mathbf{M}^{(i)}$ | 数据集 $D^{(i)}$ 的 Top-K 策略矩阵 |
| ℓ | 标签，表示某一优选方案的标识符 |
| \mathcal{L} | 标签空间，包含所有优选方案的标签集合 |
| $\mathbf{L}^{(i)}$ | 数据集 $D^{(i)}$ 对应的多标签集合 |
| \mathcal{M} | 训练集，包含所有先验数据的特征与标签集合 |
| \mathcal{F} | 多标签分类器，用于预测优选方案标签 |
| $q^{(j)}$ | 标签 $\ell_{\omega^{(j)}}$ 为优选方案的概率 |
| r | 预测阶段保留的最高优选标签数 |
| \mathbf{L}' | 预测阶段保留的最高优选标签集合 |
| $\Omega'(D)$ | 数据集 D 的优选子空间， $\Omega'(D) \subseteq \Omega$ |
| G | 映射函数，将数据集特征向量映射到优选子空间 |
| $\hat{\omega}$ | 最优方案，即在 $\Omega'(D_{\text{test}})$ 中得分最高的组合 |

表 2: 符号与描述

4.1 先验数据与多标签映射策略

在实际应用中，通常可以从历史任务中获取大量已处理或部分标注的数据集，这些可视为**先验数据**（离线学习）。当面对新任务时，由于需要在较短时间内完成聚类策略的优选与评估，此时的新数据集则称为**测试数据**（在线应用）。通过在先验数据上深入探索并记录“数据特征—策略表现”的关联信息，就能在测试数据上显著减少不必要的搜索开销，从而提升整体效率。

4.1.1 先验数据与测试数据的划分

为便于在实际部署时利用先验知识，本研究将原有数据资源分为以下两类：

- **先验数据集** D_{train} : 由多个历史数据集组成, 记为 $D^{(1)}, D^{(2)}, \dots, D^{(N)}$ 。在离线阶段 (训练阶段), 这些数据用于对搜索空间 Ω 进行大范围或抽样评估, 以收集足够的策略得分信息, 为后续自动化优化提供参考。
- **测试数据集** D_{test} : 代表实际部署时面临的新数据, 需要在线快速找到近优的清洗-聚类组合。此时可借助先验阶段所学知识, 显著减少搜索规模并降低评估时间。

在离线评估过程中, 对每个先验数据集 $D^{(i)}$ 遍历或随机抽样若干清洗-聚类策略 $\omega \in \Omega$, 便可计算各自方案的综合得分 $S(D^{(i)}, \omega)$ 。为高效记录在 $D^{(i)}$ 上表现最好的候选策略集, 我们定义一个 **Top-K 方案矩阵** (式 (4.1)), 记为 $\mathbf{M}^{(i)}$, 其中每一行是一个评分 S_j 较高的策略组合 $\omega_j^{(i)} = (c_j, h_j, \theta_j)$ 。该矩阵按照 S_j 降序排列, 用于在后续多标签学习中标识“优选”方案。

$$\mathbf{M}^{(i)} = \begin{pmatrix} c_1 & h_1 & \theta_1 & S_1 \\ \vdots & \vdots & \vdots & \vdots \\ c_K & h_K & \theta_K & S_K \end{pmatrix}. \quad (4.1)$$

4.1.2 多标签学习与映射函数构建

输入特征向量的扩展 传统做法仅使用整体质量统计 $\mathbf{x}(D) = (\text{ErrorRate}, \text{MissingRate}, \text{NoiseRate}, m, n)$ 来描述数据集 D 。为了让分类器进一步捕捉 错误类型 异质性以及清洗阶段的 可解释信息, 本文在 $\mathbf{x}(D)$ 之后追加两组辅助特征:

- **错误类型分布向量** $\mathbf{e}(D) = (\rho_{\text{typo}}, \rho_{\text{outlier}}, \rho_{\text{incons}}, \dots)$, 记录不同错误来源所占比例;
- **清洗可解释指标向量** $\mathbf{z}(D)$ ——若在先验数据中已知, 如规则覆盖率、置信区间宽度、置信标签一致度等。

最终送入多标签分类器的输入为

$$\tilde{\mathbf{x}}(D) = [\mathbf{x}(D) \parallel \mathbf{e}(D) \parallel \mathbf{z}(D)],$$

其中“ \parallel ”表示向量拼接。若某指标缺失, 则以零填充并增加 **mask** 位确保维度一致。

标签空间与多标签样本 离线阶段首先对每个先验数据集 $D^{(i)} \in D_{\text{train}}$ 全量 (或抽样) 评估策略 $\omega \in \Omega$, 得到综合得分 $S(D^{(i)}, \omega)$, 并取得得分最高的 Top-K 组合 $\mathbf{M}^{(i)} = \{\omega_1^{(i)}, \dots, \omega_K^{(i)}\}$ 。令每一条优选策略 $\omega^{(j)}$ 对应唯一标签 $\ell_{\omega^{(j)}}$, 则可得到离散标签空间

$$\mathcal{L} = \{\ell_{\omega^{(1)}}, \ell_{\omega^{(2)}}, \dots, \ell_{\omega^{(m)}}\}. \quad (4.2)$$

对于数据集 $D^{(i)}$, 其多标签集合为

$$\mathbf{L}^{(i)} = \{\ell_{\omega_1^{(i)}}, \ell_{\omega_2^{(i)}}, \dots, \ell_{\omega_K^{(i)}}\}. \quad (4.3)$$

于是可构造多标签训练样本 $(\tilde{\mathbf{x}}(D^{(i)}), \mathbf{L}^{(i)})$, 汇总为训练集

$$\mathcal{M} = \{(\tilde{\mathbf{x}}(D^{(1)}), \mathbf{L}^{(1)}), \dots, (\tilde{\mathbf{x}}(D^{(N)}), \mathbf{L}^{(N)})\}. \quad (4.4)$$

分类器训练与优选子空间映射 在训练集 \mathcal{M} 上训练多标签分类器 \mathcal{F} , 其对每个标签 $\ell \in \mathcal{L}$ 产生置信度 $q_\ell \in [0, 1]$ 。对新数据集 D_{test} , 输入 $\tilde{\mathbf{x}}(D_{\text{test}})$ 得到

$$\mathcal{F}(\tilde{\mathbf{x}}(D_{\text{test}})) = \{(\ell, q_\ell) \mid \ell \in \mathcal{L}\}. \quad (4.5)$$

取置信度最高的 r 个标签形成

$$\mathbf{L}' = \{\ell \mid q_\ell \text{ 属于前 } r \text{ 大}\}, \quad (4.6)$$

并映射回 优选子空间

$$\Omega'(D_{\text{test}}) = \{\omega \mid \ell_\omega \in \mathbf{L}'\} \subset \Omega. \quad (4.7)$$

该子空间通常远小于原始搜索空间, 因而显著降低评估成本。整体映射可记为

$$G: \tilde{\mathbf{x}}(D) \mapsto \Omega'(D). \quad (4.8)$$

4.1.3 清洗准确度与聚类过程跟踪

在前文 (4.1.1, 4.1.2) 中, 我们已经说明了如何在离线阶段获取“清洗-聚类”组合的综合得分 $S(D^{(i)}, \omega)$, 并通过多标签学习将其映射到优选子空间。然而, 仅仅依赖综合得分并不能充分解释数据清洗对聚类结果的内在影响, 也无法揭示某些清洗策略为何在高噪声或特定分布下表现优异或失败。为此, 我们在离线阶段额外记录清洗准确度和聚类过程数据, 以便在后续实验和可视化分析中, 深入探究清洗如何改变聚类性能。

记录清洗准确度指标 在离线评估每个策略 ω 时, 除了计算 $S(D^{(i)}, \omega)$ 外, 我们还会借助对照基准 (如 GroundTruth) 来衡量清洗操作的有效性, 主要包括:

- **EDR (Error Detection Rate)** [12]: 清洗阶段检测并成功修复的错误值所占比例;
- **Precision, Recall, F1**: 分别表示清洗操作对错误修复的准确率、召回率与综合平衡效果, 能够帮助判断“修复了多少”与“修复是否准确”;

这些指标与综合得分一并写入 $\mathbf{M}^{(i)}$ 或其扩展表中, 成为后续机理研究的重要依据。

跟踪聚类内部过程 除此之外, 我们在聚类算法运行时插入了轻量化的过程跟踪机制, 对如下一些中间结果进行记录:

- **K-Means 收敛轨迹**: 包括迭代次数、质心坐标变动量、最终 SSE (Sum of Squared Errors) 等;
- **DBSCAN 核心点/边界点比例**: 了解清洗前后噪声点判定的变化;
- **层次聚类 (HC) 合并顺序**: 可观察高维数据在不同清洗方式下层级拆分的演化差异;

通过这些过程级数据, 研究者可在分析中深度观察清洗策略如何影响算法的收敛路径、簇划分形状, 以及与数据特点之间的关联。若某些清洗方法在 F1 指标上很高却未能提升最终簇质量, 往往能从这些跟踪结果中找到“破坏簇结构”的具体原因。

4.2 自动化聚类优化流程

在第 4.1 节中, 我们介绍了如何利用先验数据构建多标签映射策略, 以学习数据特征 $\mathbf{x}(D)$ 与优选方案子空间 $\Omega'(D)$ 之间的映射关系。本节将基于这些离线知识, 探讨在**新数据集**上的自动化聚类优化流程。其核心思想是: 通过多标签分类器在在线阶段快速筛选出若干“候选”清洗-聚类组合, 避免大规模穷举搜索, 从而在**更短时间内**获得**近优结果**。图 1 展示了该流程的整体示意。

该流程主要包括**离线知识积累** (训练阶段) 和**在线优化** (测试阶段) 两个核心环节:

1. **训练阶段 (离线学习)**: 基于先验数据集 D_{train} , 计算不同数据特征与清洗-聚类策略的匹配程度, 并训练多标签分类器 \mathcal{F} , 从而建立数据特征到优选方案子空间的映射 $G(\mathbf{x}(D))$; 同时收集并记录清洗准确度、聚类过程数据, 以备深入机理分析。
2. **测试阶段 (在线优化)**: 面对新的数据集 D_{test} , 利用训练阶段学习到的映射 $G(\mathbf{x}(D_{\text{test}}))$, 快速筛选搜索空间 Ω 中的候选策略子集 $\Omega'(D_{\text{test}})$, 避免全量穷举, 从而在较短时间内获取高质量的清洗-聚类方案。若需进一步探讨其内在机制, 可通过与离线记录的分布及过程数据比对来解释为何某些组合在新数据上表现突出或失效。

在接下来的小节中, 我们将给出训练与测试环节的关键算法伪代码, 并展示如何将多标签训练集 \mathcal{M} 与在线推荐结果有机结合。

4.2.1 训练阶段: 离线知识积累

训练阶段的目标是基于先验数据集 D_{train} 生成多标签训练集并学习多标签分类器。算法伪代码如算法 1 所示。

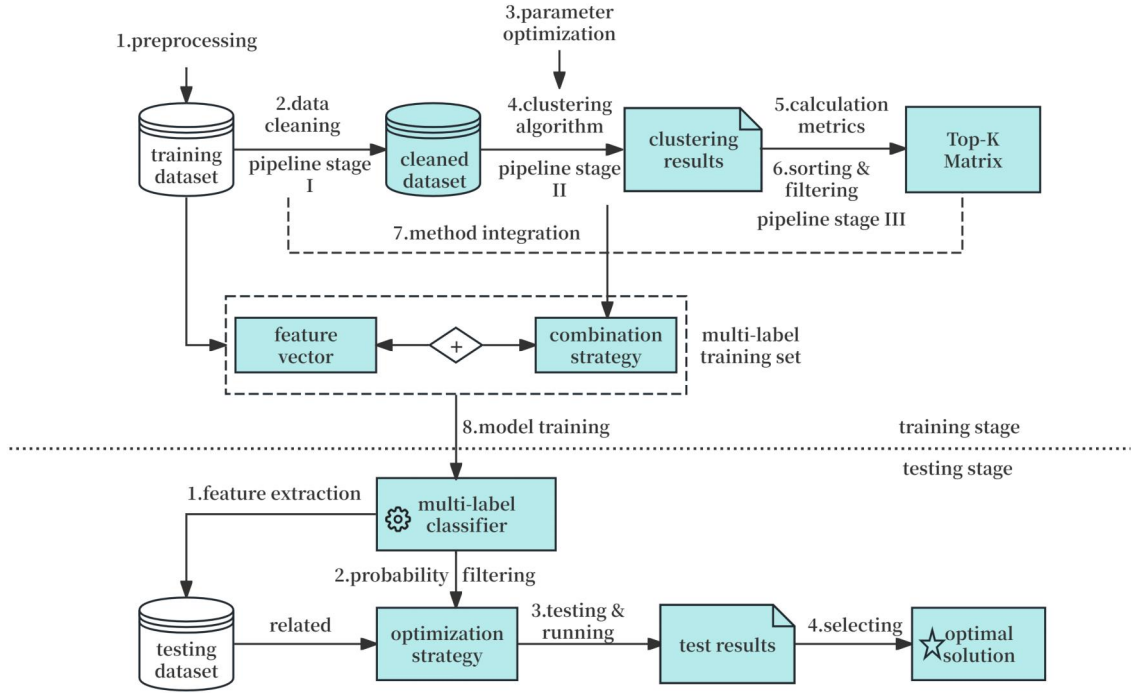


图 1: 自动化聚类优化流程示意图

Algorithm 1: 离线训练阶段：生成训练数据与训练多标签分类器

Input: 先验数据集 $D_{\text{train}} = \{D^{(1)}, \dots, D^{(N)}\}$;

搜索空间 Ω ;

Top-K 大小 K 。

Output: 多标签分类器 \mathcal{F}

$\mathcal{M} \leftarrow \text{GenerateTrainingData}(D_{\text{train}}, \Omega, K)$;

$\mathcal{F} \leftarrow \text{TrainClassifier}(\mathcal{M})$;

return \mathcal{F}

Function $\text{GenerateTrainingData}(D_{\text{train}}, \Omega, K)$:

$\mathcal{M} \leftarrow \emptyset$;

for $i \leftarrow 1$ **to** $|D_{\text{train}}|$ **do**

foreach $\omega \in \Omega$ (或采样自 Ω) **do**

 计算 $S(D^{(i)}, \omega)$;

 记录 EDR/F1 等清洗准确度，以及算法内部过程数据（如质心迭代、核心点等）;

 选出 Top-K 策略 $\mathbf{M}^{(i)} = \{\omega_1^{(i)}, \dots, \omega_K^{(i)}\}$ 按得分降序;

 映射为多标签集合 $\mathbf{L}^{(i)} = \{\ell_{\omega_1^{(i)}}, \dots, \ell_{\omega_K^{(i)}}\}$;

$\mathcal{M} \leftarrow \mathcal{M} \cup \{(\mathbf{x}(D^{(i)}), \mathbf{L}^{(i)})\}$;

return \mathcal{M}

Function $\text{TrainClassifier}(\mathcal{M})$:

 // 可根据具体多标签算法实现

 训练多标签分类器 \mathcal{F} ;

return \mathcal{F}

4.2.2 测试阶段：在线预测与最优方案搜索

测试阶段在新数据集 D_{test} 上应用训练好的分类器，快速锁定优选子空间并搜索最优策略。伪代码如算法 2 所示。

Algorithm 2: 测试阶段：寻找最优方案 $\hat{\omega}$

Input: 测试数据集 D_{test} ;

多标签分类器 \mathcal{F} ;

搜索空间 Ω ;

保留标签数 r 。

Output: 最优方案 $\hat{\omega}$

计算 $\mathbf{x}(D_{\text{test}})$;

$\mathbf{L}' \leftarrow \{\}$;

foreach $\ell \in \mathcal{L}$ **do**

$q_\ell \leftarrow \text{置信度}(\mathcal{F}, \mathbf{x}(D_{\text{test}}), \ell)$;

$\mathbf{L}' \leftarrow \mathbf{L}' \cup \{(\ell, q_\ell)\}$;

选取置信度最高的 r 个标签 \mathbf{L}'_{top} ;

映射回优选子空间 $\Omega'(D_{\text{test}})$;

foreach $\omega \in \Omega'(D_{\text{test}})$ **do**

 计算 $S(D_{\text{test}}, \omega)$;

// 计算综合得分

$\hat{\omega} \leftarrow \arg \max_{\omega \in \Omega'(D_{\text{test}})} S(D_{\text{test}}, \omega)$;

return $\hat{\omega}$

4.3 小结

本节系统介绍了自动化聚类方法的整体框架，从离线阶段的多标签训练与清洗准确度/过程数据记录，到在线阶段通过优选子空间快速搜索最优策略。在此基础上，我们不仅能在大规模搜索空间中高效获得近优清洗-聚类组合，还能借助记录下的清洗精度与算法过程信息，对清洗操作对聚类结果的影响机理进行更深入的剖析。下一章我们将结合具体实验展示该方法在多场景下的适用性与可解释性。

5 实验与结果分析

本章围绕第 3 节所提出的问题和模型定义（特别是第 3.2 节）展开实验与结果分析，重点回答问题 (Q_1) 到 (Q_3) 。我们将通过对多种数据集和聚类算法的验证，定量评估“数据清洗与聚类协同优化”方案的有效性和适用性，并进一步检验基于自动化模型（多标签学习）的管线在实际场景中的性能表现。

5.1 实验设置

本节将介绍实验所依赖的数据集、清洗策略与聚类算法等准备工作。

5.1.1 数据集准备

为确保实验覆盖多种数据质量问题（如缺失值、错误值、噪声等），我们选取了 40 个公开的数据集，这些数据集在规模、维度及数据缺陷的分布上存在显著差异。基于第 3.1 节中对数据特征向量 $\mathbf{x}(D)$ 的定义，我们在实验前对每个数据集统计了错误率、缺失率和噪声率等关键指标。这些统计信息不仅有助于分析数据质量对聚类性能的影响，也为评估不同清洗-聚类策略在多种数据特征下的适配性提供了精细的对照依据。表 3 列出了部分数据集的关键统计信息。

5.1.2 算法准备

本研究关注两方面算法：(1) 数据清洗策略；(2) 聚类算法及对应参数。

| 数据集 | 数据集个数 | 样本数 | 特征数 | 缺失率范围 | 错误范围 | 平均 IQR 噪声率 |
|----------|-------|------|-----|-----------------|-----------------|------------|
| flights | 10 | 2376 | 7 | 0.00% ~ 49.99% | 8.69% ~ 67.74% | 0.00% |
| hospital | 10 | 1000 | 20 | 0.00% ~ 28.50% | 8.53% ~ 49.29% | 2.62% |
| beers | 10 | 2410 | 11 | 4.04% ~ 31.82% | 6.17% ~ 52.00% | 1.43% |
| rayyan | 10 | 1000 | 11 | 14.60% ~ 39.92% | 10.75% ~ 52.73% | 3.83% |

表 3: 实验中部分数据集的规模与质量问题概览

数据清洗策略 我们在第 2 节介绍了若干种常用的数据清洗方法。本次实验选取了多种具有代表性的算法：

- **Mode 填补**: 该方法用来模拟基础的数据修复，主要通过众数或均值填补缺失值，并进行简单的错误值修正。它计算成本低，适用于数据缺失较少或错误较为简单的情况，但在面对复杂噪声时可能效果有限。
- **Raha-Baran**: 作为深度数据修复的方法，该策略结合上下文规则与统计推断，能够识别并校正更复杂的错误和噪声。虽然能提升数据质量，但相较于基础填补方法，计算成本更高，适用于数据问题较综合且复杂的场景。
- **GroundTruth (GT)**: 此方法用于对照实验 [53]，假设数据已被完全清洗，不含缺失值、错误值或噪声。它并非实际可行的清洗策略，而是用来评估其他数据修复方法的有效性。

聚类算法 我们在实验中选取了 6 种经典且常用的无监督聚类算法，包括 *K-Means*、*DBSCAN*、*OPTICS*、层次聚类 (*HC*)、*GMM* 以及 *AffinityPropagation*。这些算法在聚类策略、超参数设置、计算复杂度及对噪声的敏感性等方面存在较大差异。为了更直观地比较它们的核心特性，表 ?? 总结了每种算法的聚类方式、关键超参数、时间复杂度及参数敏感度¹。从表 ?? 可以看出，各聚类算法在适用场景、计算复杂度及对超参数的依赖程度上存在显著差异。例如，K-Means 计算效率较高，但对簇数选择较敏感，适用于球形簇结构的数据；DBSCAN 和 OPTICS 采用密度聚类方法，能够自动确定簇数，并对噪声具有较强的鲁棒性。层次聚类 (HC) 能够揭示数据的层次关系，但计算复杂度较高，不适合大规模数据。GMM 适用于数据呈高斯混合分布的情况，而 AffinityPropagation 则无需预设簇数，但计算开销较大，且受参数选择的影响显著。

在本实验中，我们将在不同数据清洗策略下评估这些聚类算法的表现，分析数据质量对聚类效果的影响，并比较各算法在不同数据特征下的适配性。

5.2 自动化管线实验流程步骤

为保证自动化管线实验的系统性与可复现性，我们设计了一套标准化的实验流程，如图 2 所示。整体流程包括四个主要阶段：数据预处理、清洗与聚类执行、结果分析以及自动化聚类优化。以下是对实验流程图的详细解释：

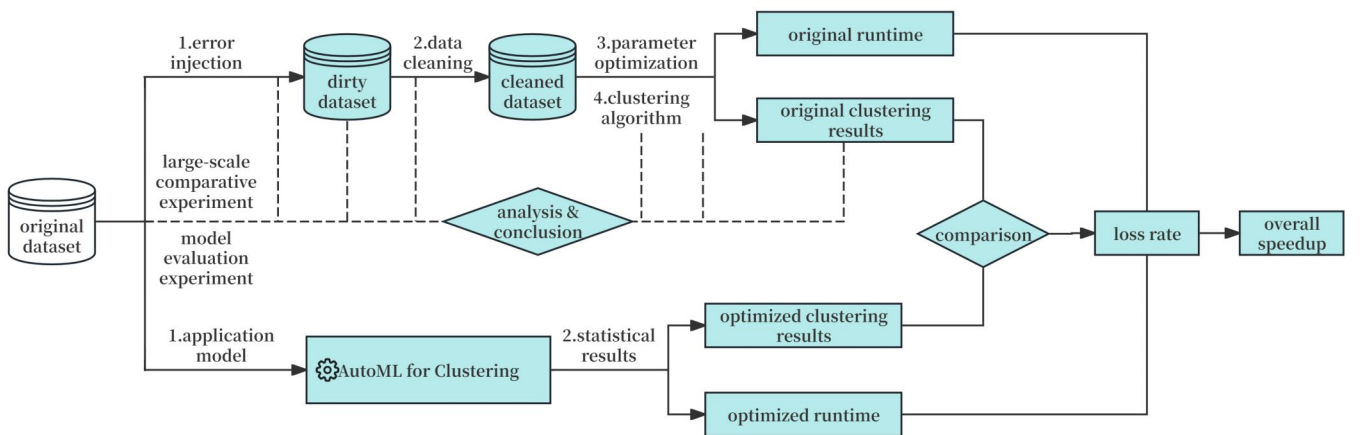


图 2: 实验设计流程示意图

¹ 此处采用的是 scikit-learn [54] 框架中提供的传统且经典的算法，因其有公开的代码且易于配置，下同。

| 算法 | 与 scikit-learn 标准版差异 / 关键实现 | 目的 & 预期收益 |
|---|--|--|
| K-Means_{base} (<i>k-means++</i>) | <ul style="list-style-type: none"> 外层 Optuna 自动调 k 每个 k 下追踪质心位移曲线 (Δ, AUC_{Δ}, geo-decay) 记录收敛代数与终态 SSE | <ul style="list-style-type: none"> 以 AUC_{Δ} 量化“收敛路径长度” 为后文 Δn_{iter}, ΔSSE 提供基准 |
| K-Means_{PPS} (<i>linear / K-MC²</i>) | <ul style="list-style-type: none"> 采用 K-MC² ($\mathcal{O}(n)$) 近似采样质心 后接手写 Vanilla K-Means; 收敛历史与 base 一致 | <ul style="list-style-type: none"> 将初始化复杂度由 $\mathcal{O}(nk)$ 降至 $\mathcal{O}(n)$ 检验“更快初始化”对清洗增益的影响 |
| K-Means_{NF} (<i>new-formulation</i>) | <ul style="list-style-type: none"> 采用核范数重构: 显式构造 $\text{Gram } A = X^T X$ 交替优化 (s_i, F); 两层循环 仅外层保存质心位移 | <ul style="list-style-type: none"> 单次矩阵闭式更新替代传统两步法 评估“更平滑收敛曲线”对清洗扰动敏感度 |
| GMM-EM (tracking) | <ul style="list-style-type: none"> 利用 warm_start + max_iter = 1 手动滚动 逐轮缓存 lower-bound、AUC_{LL}、衰减率 双阶段 (Optuna + Kneedle) 选 k | <ul style="list-style-type: none"> 定量比较“清洗 \rightarrow EM 收敛加速” 跟踪 k 偏移与 log-likelihood 曲线形态 |
| DBSCAN (noise-aware) | <ul style="list-style-type: none"> Optuna 同时调 ε, $minPts$ 目标函数乘 $(1 - \rho_{noise})$ 避免“噪声伪高分” 输出核心 / 边界 / 噪声计数与近邻直方图 写出 raw \leftrightarrow cleaned 的 $\Delta\varepsilon$, $\Delta minPts$ | <ul style="list-style-type: none"> 提供 Δn_{core}, $\Delta \rho_{noise}$ 数据源 减少极端噪声对综合得分的干扰 |
| HC (merge-tree tracking) | <ul style="list-style-type: none"> 搜索 k, linkage, metric compute_distances=True 保存完整合并树 记录合并层数 n_{merge}, 最大高度 h_{max}, intra/inter 均距比 写出 raw-cleaned 的 Δk, $\Delta combined$ | <ul style="list-style-type: none"> 分析“树裁剪深度”与“簇紧致度”随清洗的系统性偏移 |

表 4: 本研究所采用的聚类方法相对于标准算法实现的关键改动及其设计动机

1. **错误注入与特征统计**: 在原始数据集上人为注入不同比例的错误（包括缺失值、噪声和异常值），模拟真实数据质量问题。随后，统计各数据集的错误率、缺失率及噪声水平，提取数据特征向量 $\mathbf{x}(D)$ ，为后续清洗与聚类优化提供基础记录。
2. **清洗与聚类执行**: 依次应用不同的数据清洗策略，对每个数据集生成多个修复版本；随后，在清洗后的数据集上运行各种聚类算法，并记录对应的超参数、聚类簇数、运行时间及综合得分 $S(D, \omega)$ （计算见式 (3.4)）。
3. **结果分析**: 比较不同清洗-聚类组合的性能，评估其聚类质量、计算耗时及适用性，分析数据质量对聚类效果的影响，并总结最优策略的适配性。
4. **自动化优化**: 在离线阶段训练多标签分类器，学习“数据特征 $\mathbf{x}(D) \rightarrow$ 优选子空间 $\Omega'(D)$ ”的映射关系；在测试阶段，该模型将用于高效推荐清洗-聚类方案，并与全量搜索结果进行对比，计算损失率 $\eta(D)$ 和加速比 $\mathcal{A}(D)$ 。

该流程确保了实验在不同场景下的准确性和可复现性，并为后续实验提供统一框架。

在接下来的部分，我们将围绕两项核心任务展开：

- **大规模对比实验（第 5.3 节）**: 评估各种清洗-聚类组合在全部数据集上的表现，分析最优策略在不同数据质量条件下的适配性。
- **自动化聚类模型评估实验（第 5.4 节）**: 验证自动化管线的有效性，对比多标签学习推荐的“优选子空间搜索”与“全量搜索”，重点考察损失率 $\eta(D)$ 及加速比 $\mathcal{A}(D)$ 两个指标，评估其在大规模数据场景下的可行性。

5.3 大规模对比实验

本节针对“清洗策略 + 聚类算法”协同优化进行大规模对比实验，以回答第 3.2 节提出的核心问题：“不同清洗-聚类组合在多样化数据（含噪声、缺失值、错误值）上的协同表现如何”

5.3.1 实验设置与评估指标

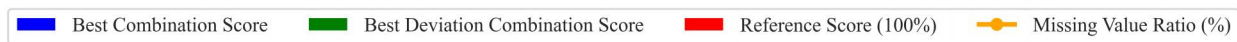
清洗-聚类组合 对每个数据集分别应用 mode、Raha-Baran 及 GT 三种清洗方法，并利用 Optuna 模型 [55] 进行超参数调优，随后运行 K-Means、DBSCAN、OPTICS、HC、GMM 和 AffinityPropagation 六种聚类算法，形成“3 清洗 \times 6 聚类 = 18”种组合。

评估指标 本研究主要使用以下三个指标来评估清洗-聚类组合的效果：

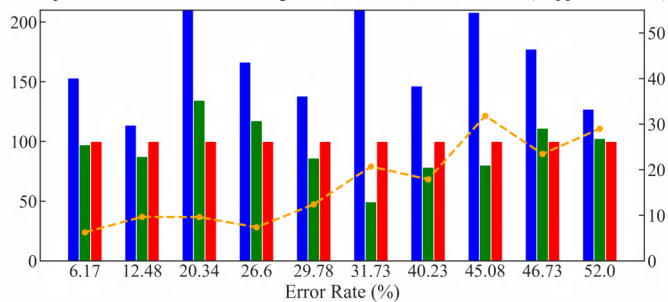
- **簇数量合理性**: 为确保聚类结果的有效性，簇数量应在以下范围内：
 - 标准范围：簇数量通常不小于 5 个，且不大于样本数的算术平方根。
 - 筛选规则：若簇数明显偏离此范围（例如过少导致过度聚合，或过多导致过度分散），则视为不合法结果并排除。
- **综合得分**: 综合得分 $S(D, \omega)$ 的权重设置为 $\alpha = 0.75$ 和 $\beta = 0.25$ ，这一选择基于预实验结果。过高的轮廓系数权重会导致簇数过少，与实际需求不符。因此，较低的 β 权重有助于生成数量合适且更稳定的簇。
- **百分比得分**: 为了便于比较不同算法组合之间的性能，本研究对综合得分进行百分比归一化处理，以 GroundTruth 清洗策略修复后的最高综合得分为基准，将其定义为 100%。不合理的实验结果（如算法运行超时、不收敛或簇数明显偏离标准范围）被标记为 0%，以避免其干扰整体结果的分析。

5.3.2 实验结果与分析

本节围绕不同数据集上策略组合的性能展开实验结果呈现和分析。我们从以下三个层面展开讨论：

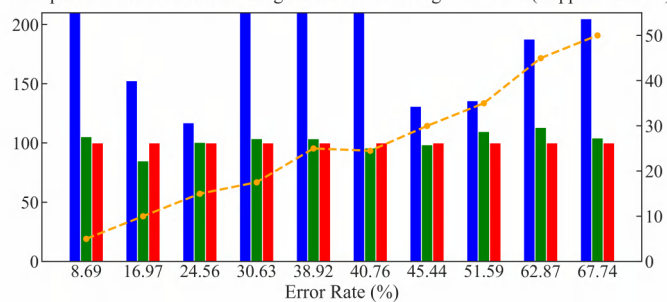


Comparison of Scores and Missing Value Ratio for Beers Dataset (Capped at 200%)



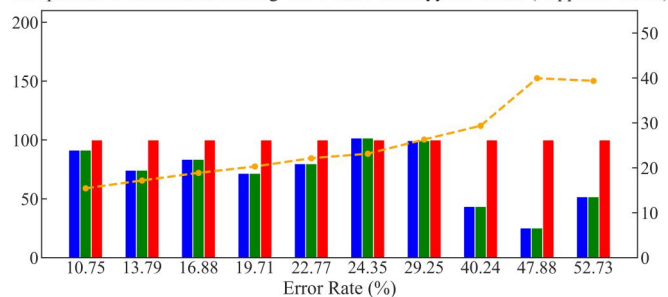
(a) *beers* 数据集：错误率 vs. 综合得分与缺失值比例

Comparison of Scores and Missing Value Ratio for Flights Dataset (Capped at 200%)



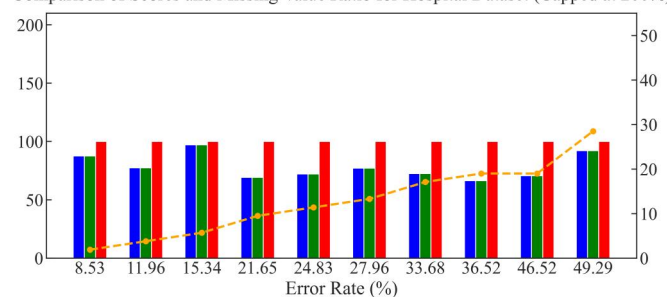
(b) *flights* 数据集：错误率 vs. 综合得分与缺失值比例

Comparison of Scores and Missing Value Ratio for Rayyan Dataset (Capped at 200%)



(c) *rayyan* 数据集：错误率 vs. 综合得分与缺失值比例

Comparison of Scores and Missing Value Ratio for Hospital Dataset (Capped at 200%)



(d) *hospital* 数据集：错误率 vs. 综合得分与缺失值比例

图 3: 不同数据集在不同错误率条件下的得分与缺失值比例变化

(1) 各数据集在不同错误率下的最优方案与得分情况 图 3a、3b、3c 和 3d 分别展示了 *beers*、*flights*、*rayyan* 与 *hospital* 四个数据集在不同比例错误率（横轴）下的百分比综合得分（左侧纵轴）及缺失值比例（右侧纵轴）。我们引入了**最接近基准的组合**，其得分是除参考标准外距离 100% 最近的方案。该方案的选取有助于识别在实际应用中最接近理想基准性能的策略。

通过以上四张图可以观察到，随着错误率的增加，各数据集的缺失值比例呈现不同程度的波动。此外，不同清洗-聚类组合在部分数据条件下表现出显著差异，其中某些方案的得分甚至远超基准值（超过 200%）。为了更详细地分析这些现象，表 5a 至 5d 进一步列出了四个数据集在不同错误率下的**最佳方案** (*Best Combination*, C_1) 及其综合得分 (*Best Score*, S_1)，同时给出了**最接近基准的组合** (*Best Deviation Combination*, C_2) 及其对应得分 (*Best Deviation Score*, S_2)。这些结果有助于我们深入理解不同清洗-聚类组合的适用性，并评估其对聚类质量的影响。

| (a) <i>beers</i> 数据集 | | | | | | (b) <i>flights</i> 数据集 | | | | | |
|----------------------|-----------|-----------|-----------|-----------|-----------|------------------------|-----------|------------|-----------|------------|-----------|
| Name | Error (%) | C_1 | S_1 (%) | C_2 | S_2 (%) | Name | Error (%) | C_1 | S_1 (%) | C_2 | S_2 (%) |
| beers | 6.17 | mode + HC | 153.11 | R-B + HC | 97.21 | flights | 8.69 | mode + HC | 290.44 | mode + KM | 105.36 |
| beers | 12.48 | R-B + HC | 113.69 | mode + HC | 87.37 | flights | 16.97 | mode + GMM | 152.61 | R-B + HC | 84.89 |
| beers | 20.34 | mode + HC | 259.85 | R-B + HC | 134.50 | flights | 24.56 | R-B + HC | 117.03 | R-B + AP | 100.53 |
| beers | 26.60 | mode + HC | 166.54 | mode + AP | 117.45 | flights | 30.63 | mode + DB | 242.21 | mode + GMM | 103.68 |
| beers | 29.78 | R-B + HC | 138.07 | mode + HC | 86.14 | flights | 38.92 | mode + DB | 1817.80 | mode + GMM | 103.55 |
| beers | 31.73 | mode + AP | 232.95 | mode + DB | 49.49 | flights | 40.76 | mode + DB | 2543.51 | mode + AP | 95.80 |
| beers | 40.23 | mode + HC | 146.54 | mode + KM | 78.38 | flights | 45.44 | mode + AP | 130.95 | mode + DB | 98.48 |
| beers | 45.08 | mode + HC | 208.15 | R-B + HC | 80.19 | flights | 51.59 | mode + HC | 135.66 | R-B + HC | 109.68 |
| beers | 46.73 | mode + HC | 177.37 | mode + AP | 111.11 | flights | 62.87 | mode + KM | 187.77 | R-B + HC | 113.10 |
| beers | 52.00 | mode + HC | 127.06 | mode + DB | 102.51 | flights | 67.74 | mode + HC | 204.94 | R-B + HC | 104.27 |

| (c) <i>rayyan</i> 数据集 | | | | | | (d) <i>hospital</i> 数据集 | | | | | |
|-----------------------|-----------|-----------|-----------|-----------|-----------|-------------------------|-----------|-----------|-----------|-----------|-----------|
| Name | Error (%) | C_1 | S_1 (%) | C_2 | S_2 (%) | Name | Error (%) | C_1 | S_1 (%) | C_2 | S_2 (%) |
| rayyan | 10.75 | R-B + HC | 91.38 | R-B + HC | 91.38 | hospital | 8.53 | R-B + HC | 87.63 | R-B + HC | 87.63 |
| rayyan | 13.79 | R-B + HC | 74.35 | R-B + HC | 74.35 | hospital | 11.96 | R-B + HC | 77.42 | R-B + HC | 77.42 |
| rayyan | 16.88 | R-B + HC | 83.63 | R-B + HC | 83.63 | hospital | 15.34 | R-B + HC | 97.14 | R-B + HC | 97.14 |
| rayyan | 19.71 | mode + HC | 71.69 | mode + HC | 71.69 | hospital | 21.65 | R-B + HC | 69.24 | R-B + HC | 69.24 |
| rayyan | 22.77 | R-B + HC | 79.87 | R-B + HC | 79.87 | hospital | 24.83 | R-B + HC | 72.12 | R-B + HC | 72.12 |
| rayyan | 24.35 | R-B + HC | 101.71 | R-B + HC | 101.71 | hospital | 27.96 | R-B + HC | 77.16 | R-B + HC | 77.16 |
| rayyan | 29.25 | R-B + AP | 99.49 | R-B + AP | 99.49 | hospital | 33.68 | R-B + HC | 72.50 | R-B + HC | 72.50 |
| rayyan | 40.24 | R-B + HC | 43.52 | R-B + HC | 43.52 | hospital | 36.52 | mode + AP | 66.43 | mode + AP | 66.43 |
| rayyan | 47.88 | mode + HC | 25.22 | mode + HC | 25.22 | hospital | 46.52 | mode + HC | 70.70 | mode + HC | 70.70 |
| rayyan | 52.73 | mode + HC | 51.75 | mode + HC | 51.75 | hospital | 49.29 | mode + AP | 92.17 | mode + AP | 92.17 |

表 5: 不同错误率下各数据集的最佳组合 (C_1) 与最接近基准组合 (C_2) 及得分

基于图 3a 至图 3d 和表 5a 至表 5d 所示的不同错误率下的组合排名与偏差情况，我们可以得出以下几点认识：

1. “最佳组合”与“最接近基准”经常不一致
- 在 *Beers*、*Flights* 等较大规模、数值型为主的数据中，mode + HC 或 mode + DBSCAN 有时会出现 200%~300%、甚至逾千的“爆分”情形。然而，这些高分结果往往偏离基准结构较远，原因是 DBSCAN 对噪声/缺失值极度敏感，以及 HC 采用的簇数与理想情况差异较大。此时，Raha-Baran + HC 或 mode + KMeans 虽绝对分值较低，却更贴近 GroundTruth 基准。
2. 极端高分分布并不均衡，受数据维度与语义约束影响
- 在低维数值型的 *Flights* 数据中，mode + DBSCAN 出现 1800%~2500% 极端情况，表明密度聚类对填补策略和超参数高度敏感。而在更高维或包含语义规则的 *Hospital*、*Rayyan* 中，聚类分数通常较温和，极少越过 200%，暗示 Raha-Baran 修复在此类场景下更能保留合理的分布结构，降低了极端划分的可能。
3. Raha-Baran + HC 在中低错误率下尤为稳健
- 在 *Hospital*、*Rayyan* 等数据集中，当错误率低于 25% 时，Raha-Baran + HC 常同时取得“最高分”与“最贴近基准”的好结果。对于高维、具备一定语义约束且规模中等的场景而言，Raha-Baran 能有效减少噪声干扰，HC 则在

数据质量良好时更易逼近基准结构。即使错误率升至 40% 以上，Raha-Baran + HC 也仅出现有限波动，在高维场景下仍优于其他组合。

(2) 不同数据集上清洗-聚类方案随错误率变化的趋势 前文表格展示了各数据集在不同错误率下的最佳组合及其与基准的偏差，但仅反映离散的关键数值。为直观观察清洗-聚类组合随错误率变化的趋势，本节通过图 4a 至图 4h 展示各数据集在两种清洗方法下，不同聚类算法的综合得分变化。横轴为错误率，纵轴为综合得分，各曲线对应不同聚类算法，体现其不同数据质量条件下的波动情况。



图 4: 不同数据集在不同错误率条件下的聚类算法综合得分变化趋势

通过综合比对上述不同数据集在 mode 与 Raha-Baran 清洗下，各聚类算法随错误率变化的表现，可以得到以下几条规律性结论：

- 错误率升高易出现“极端爆分”或“收敛失败”，精细化清洗可缓解但不能完全避免**
当错误率中高时，mode 清洗下的 HC、DBSCAN、AP 更易出现高达 3.00 的极端得分或直接收敛到 0.00。使用 Raha-Baran 虽然减少了此类“爆分”现象，但在个别极端场景下，KMeans、GMM 或 AP 仍可能无法正常聚类，导致分数骤降至 0.0。
- 聚类算法对错误率的敏感度差异明显，HC/DBSCAN 波动大，KMeans/GMM/OPTICS 相对平稳**
层次聚类 (HC) 与密度聚类 (DBSCAN) 对噪声和超参数敏感度最高，最易在错误率升高后出现大幅波动或极端值。相比之下，KMeans 与 GMM 在大多错误率区间分数处于中等水平，不常爆分但也可能在高错误率时收敛失败；OPTICS 波动相对较小，但平均得分不高。

3. “mode” 与 “Raha-Baran” 的差异在高错误率下尤为突出

低或中等错误率下，两种清洗方法整体表现差异不大；但一旦错误率升至 30% 以上，mode 更易触发少数算法的极端波动或收敛异常，而 Raha-Baran 虽也会出现偶发失效，却在多数场景保持了中等或较平稳的分数，展现出对高错误率的相对鲁棒性。

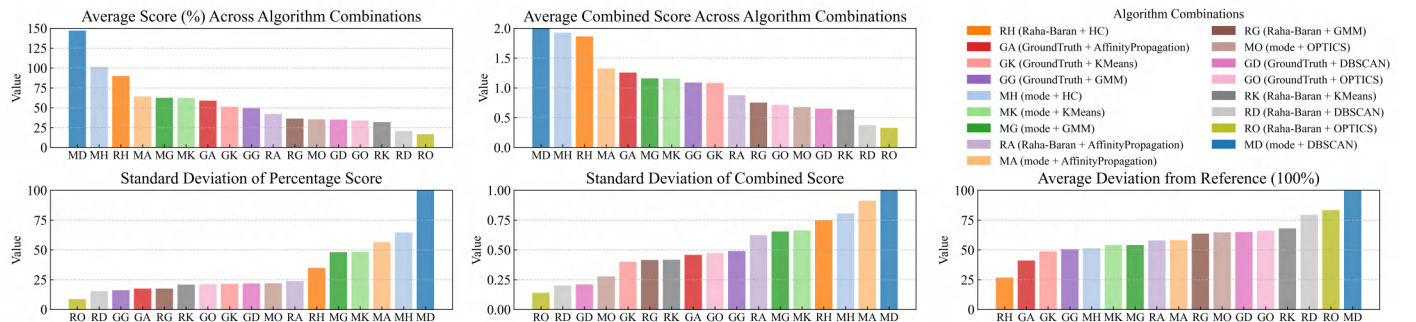
综上所述，随着错误率的持续升高，数据质量对聚类算法的影响呈显著放大趋势。尽管使用更精细的清洗方法（如 Raha-Baran）能够降低极端结果出现的概率，但在高噪声场景下仍有可能出现算法收敛失败。为更全面地评估不同清洗-聚类组合在各种数据集上的整体性能，我们将进一步从更多指标对策略的表现进行综合比较。

(3) 基于多指标的综合性能分析 图 5a 以 5 张柱状图展示了 18 种清洗-聚类组合策略在不同聚类评估指标下的整体表现，横轴为算法组合，纵轴为相应指标数值，评估指标包括：

- **Average Score (%)**: 算法组合在所有实验场景中的平均得分百分比（0%–150%），反映其总体性能；
- **Average Combined Score**: 综合分数均值（0–2），度量簇的紧致性与分离度的平衡；
- **Standard Deviation of Percentage Score**: 得分百分比的标准差，表示不同场景间的波动大小；
- **Standard Deviation of Combined Score**: 综合分数的标准差，用于评估算法在多场景下的一致性；
- **Average Deviation from Reference (100%)**: 相对于基准方案（100%）的平均偏离程度，数值越小表示越接近基准。

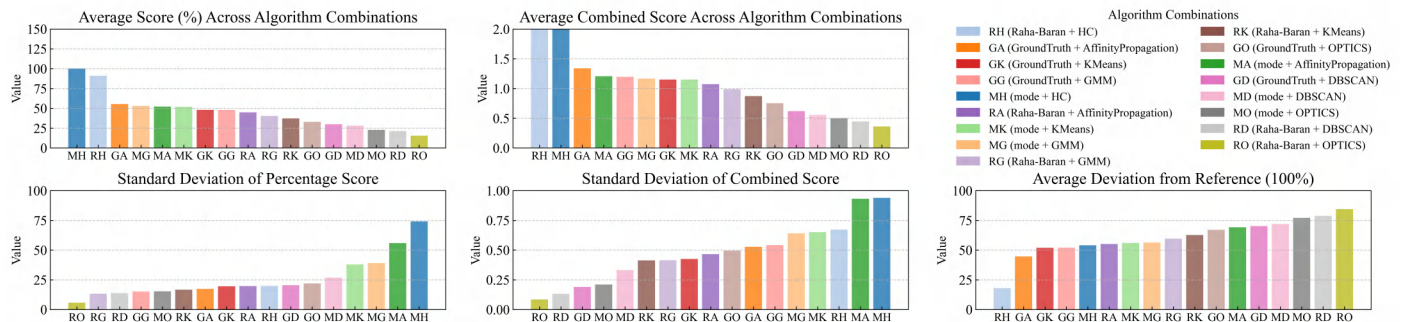
此外，图 5b 在上述基础上选取错误率低于 25% 的数据，聚焦分析非极端噪声条件下各组合的聚类性能与稳定性，直观对比各方法在低错误率场景下的差异。以下是从这两类图表中得出的主要结论与启示：

Comprehensive Analysis of Algorithm Combinations Across Metrics (All Error Rate Datasets)



(a) 18 种清洗-聚类组合在不同聚类评估指标上的综合表现

Comprehensive Analysis of Algorithm Combinations Across Metrics (Lower Error Rate Datasets)



(b) 错误率低于 25% 条件下，各算法组合的聚类性能与稳定性

图 5: 聚类算法组合在不同条件下的评估指标比较

全局方案表现与离散度分析 图 5a 将所有数据集（含不同错误率）的结果归纳，便于从全局角度评估各方案的稳定性与适应性。可以看到：

- **高均值 + 高方差的“爆分”组合：**例如 mode + DBSCAN 组合虽然平均分高达 147%，却伴随多达 481.75 的标准差，说明其在数值特征为主、规模较大的数据（如 *Flights*, *Beers*）上易发生极端聚类效果，在实际使用中需谨慎。
- **Raha-Baran + HC 和 mode + HC 整体分数高且方差相对适中：**mode + HC 约 101.53%，标准差 64.62；Raha-Baran + HC 约 90.03%，标准差 34.80，说明层次聚类在多场景下保持稳定，Raha-Baran 在处理语义或规则错误（如 *hospital*, *rayyan*）时尤为有效。
- **其余组合多集中在 30%–70% 区间：**如 Raha-Baran + OPTICS 均值仅 16.75%，在当前参数设置与数据特征下并未取得良好协同，凸显密度聚类方法在高维或高噪声环境中对参数匹配的依赖度。

总体而言，DBSCAN/OPTICS 虽具潜力但波动大，HC 与合适的清洗策略（Mode 或 Raha-Baran）通常兼具更高均值和更低方差。

低错误率情形下的表现 在图 5b 中仅保留错误率低于 25% 的数据，用以检验数据质量较佳时的聚类性能及稳定性。结果表明：

- **Mode + HC 均值约 100.14%，但标准差 74.20：**在低错误率场景下，简单填补能够较好保留数据分布，HC 则通过合适的层次切分实现高分。然而，在高维或噪声分布不均的情况下，其表现仍可能出现较大波动。
- **Raha-Baran + HC 稳定性更佳：**均值约 91.04%，标准差降至 20.04，显示对具备语义或知识库错误的数据集（如 *hospital*, *rayyan*）而言，精细化修复能显著减少噪声干扰，HC 则在较低错误率下持续逼近或略超基准。
- **其余组合受益有限：**如 mode + GMM、mode + KMeans 等平均分多介于 50%–60%，虽较全局时略有提升，但在面向 11–20 个特征的中等维度数据时仍不具明显优势。

5.3.3 综合分析与实践建议

结合上述实验结果和多种清洗与聚类算法的表现，针对不同规模、特征类型及错误率水平的数据集，提出以下建议来指导清洗与聚类策略的选择：

- **小至中规模数据集、高维/多元特征且可能存在语义错误：**优先选择 Raha-Baran + HC 组合。Raha-Baran 的上下文修复能力能够有效处理复杂语义错误，而 HC（层次聚类）的分层聚类特性适合高维、多特征数据。若错误率较高，需进一步结合更先进的修复策略，或通过特征选择与降维手段减少特征复杂性。
- **中至大规模数据集、以数值型特征为主、对内部指标最优化有较强需求：**考虑使用 mode + HC 或 mode + DBSCAN。简单填补策略（Mode）在数值型数据上的效率较高，而 HC 和 DBSCAN 均能对中大规模数据集进行合理划分，但需谨慎对待 mode + DBSCAN 的高方差风险。若应用场景要求聚类结果与基准结构更贴近且解释性更强，应该选择更稳健的组合（如 mode + HC 或 mode + KMeans）。
- **强噪声分布、不规则簇形状的数据集：**可使用 DBSCAN 或 OPTICS 等密度聚类方法。这些方法在处理不规则簇形状和高噪声数据时具有明显优势。密度聚类方法对超参数（如 ϵ 和 ξ ）敏感，因此需要针对不同错误率和分布特征进行精细化超参数调优，以避免出现极端分割（过多单点簇）或过度聚合的情况。自动调优框架（如 Optuna）在此类场景中尤为重要。

这些建议为后续应用中选择清洗-聚类方法提供了可行的依据，也为在高噪声及多样化数据环境中平衡聚类质量与可解释性指出了明确方向。

| 数据集类别 | 训练平均分 | 测试平均分 | 全量搜索耗时 (s) | 自动化耗时 (s) | 损失率 (%) | 加速比 |
|----------|--------------|--------------|---------------|--------------|---------------|-------------|
| Flights | 1.535 | 1.595 | 11,214 | 3,187 | -3.91 | 3.52 |
| Hospital | 1.278 | 1.412 | 11,237 | 5,472 | -10.50 | 2.05 |
| Beers | 1.163 | 1.482 | 12,203 | 3,248 | -27.42 | 5.83 |
| Rayyan | 1.406 | 1.835 | 9,127 | 2,093 | -30.51 | 4.36 |
| 整体平均 | 1.323 | 1.526 | 10,746 | 3,035 | -19.20 | 5.87 |

表 6: 不同数据集下自动化聚类优化管线的实验结果

5.4 自动化聚类模型评估实验

为进一步检验多标签分类器在实际应用中的有效性，我们基于第 5.3 节的大规模对比实验，结合训练与测试阶段的聚类结果，综合考量损失率 $\eta(D)$ 与综合加速比 $\mathcal{A}(D)$ 两个核心指标。为排除极端高分导致的偏差，我们剔除了训练或测试部分得分超过 3.0 的记录，并根据数据来源将数据进行分组统计。表 6 展示了各组数据在训练与测试阶段的均分，以及各自类别下的 10 个数据集使用全量搜索与自动化搜索的平均耗时（每个数据集记录 5 种组合策略的总时间），并给出了对应的损失率和加速比。

从表中可以看出，多数数据集的测试均分与训练均分持平或略有上升（例如 *Beers* 与 *Rayyan* 数据集），表明自动化管线在有效缩小搜索空间的同时，依旧能够挖掘到与原有训练基线相当或更高得分的聚类方案。若从损失率角度分析，这一“略有提升”对应负值（如 -3.9% 至 -30.51%），即自动化模型选择的子空间方案在测试中未出现显著下降，甚至在部分实验环境下获得更佳表现。

在搜索效率方面，*Beers* 数据集的加速比最高，可达 5.83 倍，说明对于规模较大的数据，自动化管线能极大减少冗余组合的评估，显著降低计算代价。相较之下，*Hospital* 数据集的加速比仅为 2.05 倍，推测其特征分布或先验知识与自动化筛选策略的吻合度较低，导致搜索空间的压缩幅度有限。然而，损失率仍维持在 -10.50% 附近，表明即使加速效果不算突出，聚类质量并未受到明显损害。

总体而言，所有数据集平均加速比约为 5.87 倍，损失率约为 -19.20%。这一结果不仅验证了自动化管线对不同类型数据（如数值型、混合型、高维数据）的适应性，也表明其在减少评估耗时的同时，能够保持甚至稍微提升聚类表现。后续研究可在更细粒度上探索各数据集的特征分布与聚类需求之间的交互机制，进一步优化多标签学习的参数设置与筛选策略。

6 清洗对聚类影响的机理分析

6.1 研究动机

在上一章中，我们基于大规模实验与自动化管线评估，发现不同清洗策略与聚类算法组合在最终聚类评分（如 Silhouette、DB 指数等）及搜索效率方面存在显著差异。然而，这些评估主要停留在宏观层面，并未深入考察清洗操作如何在数据特征、算法运行过程、最终评价指标以及超参数选择等维度对聚类产生影响。为进一步阐明其中的深层机理，本章将围绕下列四个核心问题展开分析：

- (Q_1): 清洗的错误修复程度与准确度

清洗过程究竟改正了多少错误、修正是否精准，以及这些指标能否为后续的聚类表现提供可解释度。

- (Q_2): 清洗对聚类算法内部过程的干预

清洗后数据是否让质心收敛更快、核心点判定更稳定，以及在层次聚类的合并/分裂中是否带来显著变化。

- (Q_3): 清洗对聚类评价指标的影响与量化

不同清洗策略在聚类质量上（Silhouette、DB 等）是否提升明显，其幅度能否用清洗准确度（F1、EDR 等）加以解释。

- (Q_4): 清洗对聚类超参数选择与搜索过程的影响

如果清洗改变数据分布或聚类过程，是否会引发最优 k, ε 等超参数出现偏移或需重新调优。

以下内容将给出统一的实验设计与数据采集方案，通过一次性的实验流程来获取涵盖上述四个问题所需的信息，后续章节将基于这些采集结果，对四大问题做深入分析与讨论。

6.2 实验设计

本节将对所有子问题需要的数据、流程与指标进行统一阐述，力求在同一实验流程中同时采集清洗准确度与聚类过程、聚类结果、超参数等信息。图 ?? 简要概括了整个设计思路。

数据及错误注入

- **错误注入：**在原始数据中人为注入多种错误比例（如 5%、10%、15% 等），并保留相应的“真值”或“错误标签”供后续度量清洗效果。
- **数据范围：**与第 5 章相同或部分数据集，并补充错误注入标签以便精确计算 F1、EDR 等指标。

清洗方法执行 (Q_1)

- **方法选择：**对所选每个 (数据集, 错误率) 组合执行清洗算法（如 Holoclean、Raha-Baran 等）并输出修复后数据。
- **错误修复指标采集：**在此步骤同时记录 (EDR, Precision, Recall, F1)，记录 (Q_1) 对“清洗准确度”的需求。

聚类算法与中间过程记录 (Q_2)

- **算法选择：**包括 K-Means/GMM（质心型）、DBSCAN/OPTICS（密度型）、层次聚类 (HC) 等，与第 5 章一致。
- **内部过程采集：**在运行上述算法时，对迭代步数、质心移动量、核心点数、层次合并序列等进行统计，以便后续分析清洗对内部机理的干预。

聚类评价指标计算 (Q_3)

- **结果指标 (Q_3)：**对每个清洗后数据运行聚类，记录 Silhouette、DB、Combined Score 等，以评估聚类质量是否提高；
- **关联信息 (Q_1 vs. Q_3)：**利用前一步采集的 F1、EDR 等清洗准确度，与此处聚类指标进行对照或回归分析。

超参数搜索与偏移度量 (Q_4)

- **超参数搜索：**在清洗前与清洗后数据上，分别对 k 、 ε 、minPts 等做网格搜索或贝叶斯优化；
- **偏移度量：**比较最优参数是否随清洗明显改变，如 $\Delta k = k_{\text{cleaned}} - k_{\text{raw}}$ 。

本节小结 本小节概述了针对 (Q_1) (Q_4) 的统一实验设计，包含数据注入、清洗方法执行、聚类过程跟踪、评价指标计算及超参数搜索等环节。这样确保在同一个流程下，就能获得足够支撑后续所有问题分析的原始数据与统计指标。接下来，基于这些实验所采集的信息，我们将在后续小节中详细阐述清洗对聚类的具体影响机理，包括对错误修复成效的衡量、聚类算法内部干预、聚类结果评价以及最优超参数偏移的检验等。

6.3 实验结果与分析

6.3.1 可视化概览：多角度观测清洗与聚类指标的关系

为更直观地展示清洗准确度（如 EDR、F1）与聚类效果（Silhouette、DB、Combined）之间的潜在关联，以及比较不同清洗-聚类组合在各错误率情境下的多重表现，本节采用了以下四种可视化方法，分别从“多维度综合表现”（雷达图）、“任务内组合差异”（热力图）、“指标相关性强弱”（相关性散点）以及“错误率渐进趋势”（折线图）出发，对清洗与聚类的互作用进行多角度解析。

(a) 雷达图 (Radar Charts)

此类图将清洗准确度指标 (Precision、Recall、F1、EDR) 与聚类评价指标 (Silhouette_relative、DB_relative、Comb_relative) 各自映射到雷达图的不同轴。图中每条折线对应某一清洗方法或聚类算法，横轴序列按照各维度指标环绕排布，纵向延伸表示此方法在相应维度的数值大小。

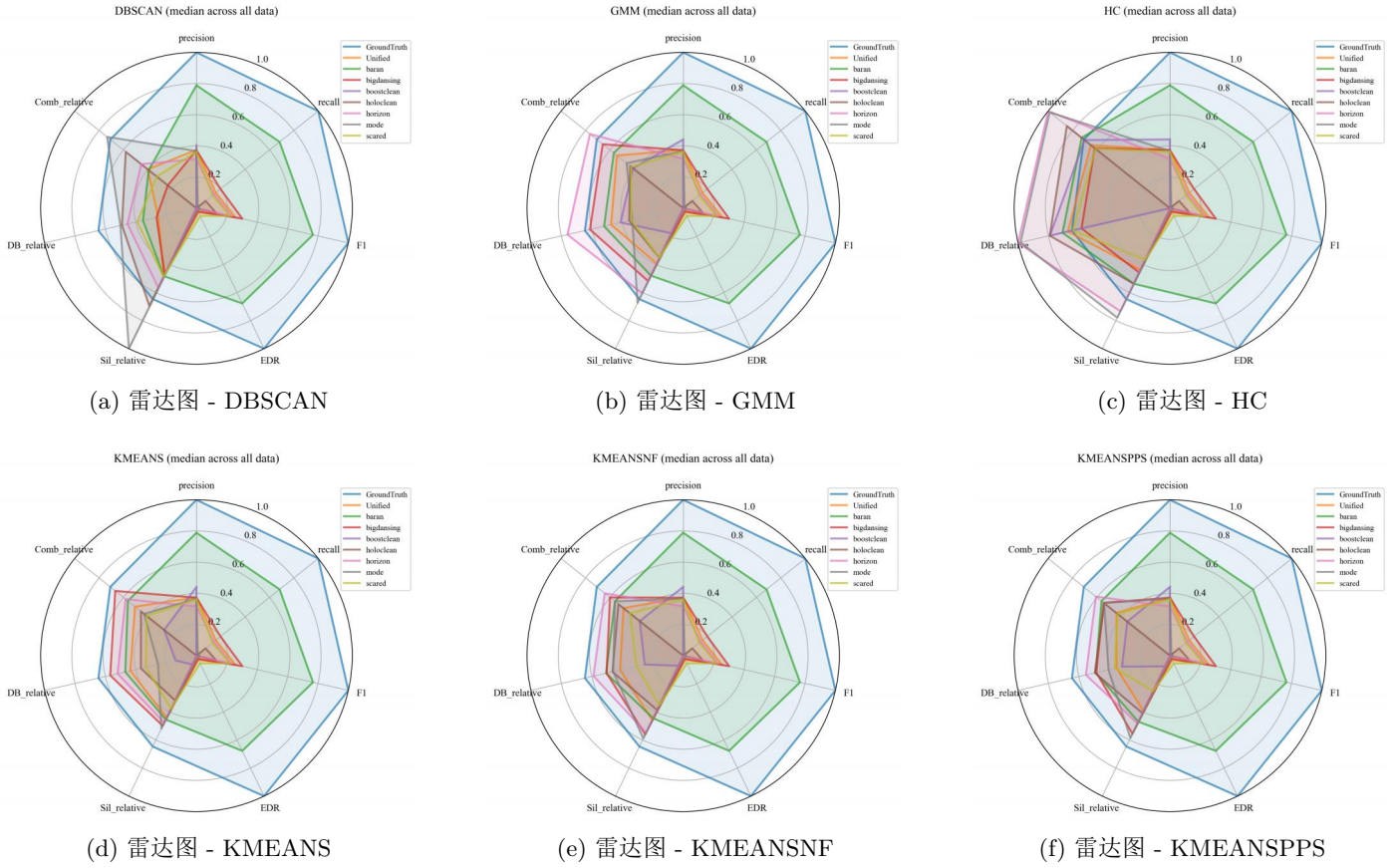


图 6: 针对不同聚类算法 (DBSCAN、GMM、HC、KMEANS、KMEANSNF、KMEANSPPS) 的雷达图结果。横轴为多维度指标 (例如 Precision、Recall、F1、EDR、Sil_relative、DB_relative、Comb_relative)，纵向延伸代表该算法/清洗方法组合在相应指标上的归一化得分。

从六张雷达图中可整体观察到，GroundTruth 曲线在各个维度轴上始终处于最外缘，表明在“理想化清洗”条件下，所有聚类算法均能获得接近最优的多维指标表现。其余真实清洗方法的折线虽有各自优势维度，但在图形覆盖面积和单轴最大值方面均与 GroundTruth 存在差距。

不同清洗策略的差异

- **baran**: 在多张图中常在 Recall、F1 维度显著突出，显示其具备高覆盖率的错误检测与较好的整体修复能力。然而在部分聚类算法下 (例如 HC、DBSCAN)，其 Sil_relative 或 Comb_relative 仍较中等，提示其对某些关键错误类型或簇结构的修复可能不够细致。
- **mode**: 大多结果位居中游，未能在任何单一维度达到高分段，也未出现极端失效，说明其修复策略在多维指标上呈较为均衡但有限的提升。
- 其他方法 (如 **bigdancing**、**boostclean** 等): 在部分坐标轴 (Precision 或 EDR) 上可见较为可观的数值，但在 DB_relative、Comb_relative 等衡量聚类质量的指标上未能保持同等出色表现，反映出其局部修正效果尚可，却难以在整体上充分发挥对聚类结构的正向干预。

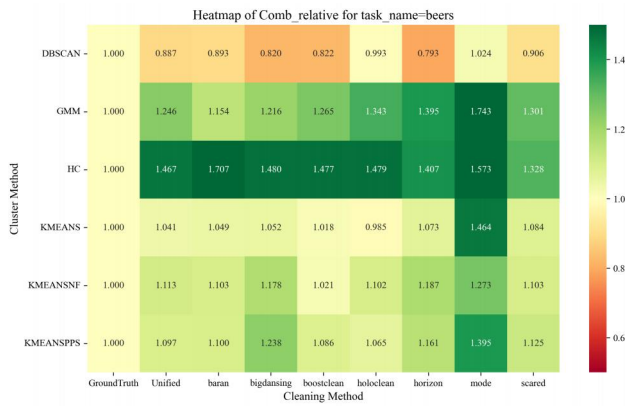
不同聚类算法的差异

- **DBSCAN 与 HC:** 在雷达图中表现出对高 EDR、F1 等清洗准确度指标更显著的依赖；当清洗覆盖率 (Recall) 与修复精确率 (Precision) 大幅提升时, Sil_relative 与 DB_relative 这两个表示簇分离度与簇间紧凑度的指标同步上升。
- **GMM 与 K-Means 系列:** 折线整体更趋稳健, 即使当错误修复指标处于中档水平, 依然可取得相对合理的聚类评价; 若清洗能够接近 GroundTruth 水平, 则能在 Comb_relative 轴上展现跳跃式增长。

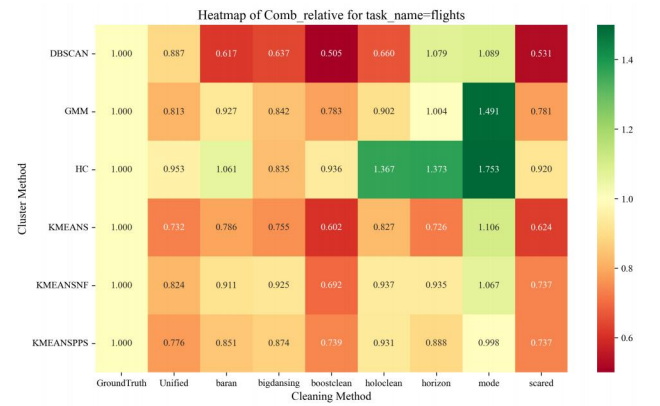
整体来说, 六张图明确显示, **高准确度的清洗**能在各类聚类算法下都带来一定的正面作用, 但具体收益会因算法对噪声的敏感程度、对错误分布的适配度而有所差异。对于 *baran* 等在 Recall 及 F1 维度表现优异却未能在多维评分中获得最佳的方法, 后续若能平衡检出率与对簇结构的精细保留, 或可进一步缩小与 GroundTruth 的差距。反之, *mode* 一类均衡但不够深入的策略需要在关键错误类型上做更精确修复, 以实质提升聚类度量。

(b) 热力图 (Heatmap)

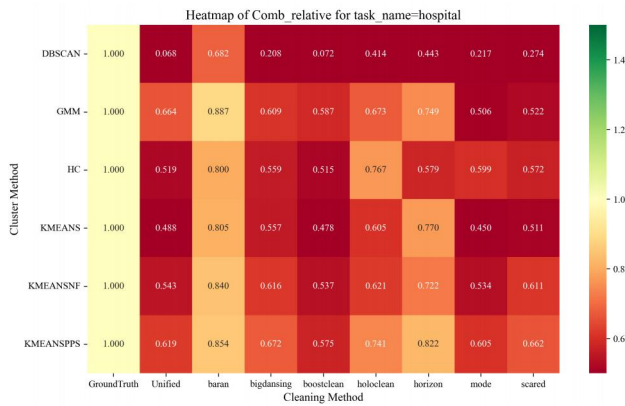
为了比较不同清洗-聚类组合在某一关键指标 (如 Combined Score) 上的平均或中位数表现, 可在二维矩阵中分别以行、列代表“聚类方法”与“清洗方法”, 并利用颜色深浅来反映指标数值的高低。



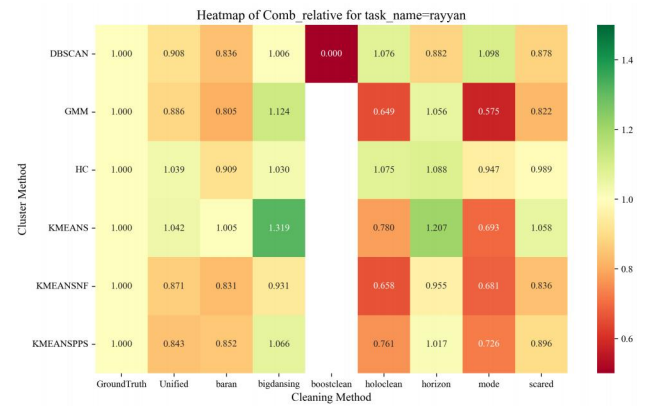
(a) Beers: Combined Score 热力图



(b) Flights: Combined Score 热力图



(c) Hospital: Combined Score 热力图



(d) Rayyan: Combined Score 热力图

图 7: 针对四个任务 (Beers、Flights、Hospital、Rayyan), 分别绘制的热力图示例。横轴与纵轴分别表示 *cleaning_method* 与 *cluster_method*; 颜色深浅表示在 Combined Score 指标上的中位数表现。

本研究针对四个数据集 (**beers**, **flights**, **hospital**, **rayyan**) 绘制了以 (*cluster_method*, *cleaning_method*) 为行列、*Comb_relative* 为数值的 heatmap。颜色偏绿往往表示综合得分更高, 偏红表示对该数据集效果欠佳。结合各数据集的具体配置 (如维度数量 *m*, 样本规模 *n*, 错误率区间与缺失率分布等), 可得出以下主要结论与对比:

- **beers:** 具有 11 个特征, 最大错误率近 33%, 最少有约 9% 的错误值。热力图显示, 当清洗方法较为精准 (如 *baran* 或接近 GroundTruth) 时, HC、DBSCAN 以及部分 K-Means 变体在 *Comb_relative* 上展现更深绿色块; 若采用简单填补策略 (如 *mode*), 大多仅呈中间色块甚至偏橙, 表明中等程度的错误修复对高噪声数据贡献有限。

- **flights**: 特征数 $m=7$, 在最高 30% 错误率与 15% 缺失率下, 图中同样可见 **baran**、**GroundTruth** 等“强修复”在多聚类算法上获得更高得分; **mode**、**bigdancing** 等方法颜色较浅或趋红, 说明未能在错漏较高的航班数据情境中实现足够修复精度, 对聚类整体分带来实质突破。
- **hospital**: 规模相对适中 ($n=1000, m=20$), 但噪声与缺失率最高可达 30%。对高缺失率、多领域特征的医疗数据, 若采用 **baran+HC** 或 **baran+DBSCAN**, 可见图中偏绿色区块更广; 简单填补法在此场景下颜色多集中在暖黄或橙色, 显示其在高噪声、多特征的医院数据上效果受限。
- **rayyan**: 此任务最高错误率达到 40% 左右, 特征维度 $m=12$ 、样本量 $n=1000$ 。热力图明显出现大量中间或浅绿色, 说明即使精确清洗也难以完全扭转极高错误率对于聚类结构的冲击; 但若结合 **GroundTruth** 或 **baran** 等方法时, 部分聚类算法 (如 **DBSCAN**、**HC**) 仍可呈现相对更深绿色, 有一定改善空间。

整体而言, 从四张热力图中可观察到, 数据特征与算法对噪声的耐受度共同决定了热力图中各组合的深浅分布。当数据噪声、缺失与错误率相对较低时 (如 **beers** 的最低 9% 区间或 **flights** 的 5%), 多数清洗方法都能带来一定增益; 但随错误率和缺失率攀升 (尤其在 **hospital**、**rayyan** 的高失真情境中), 仅高精度、面向多特征噪声的清洗算法 (如 **baran**) 或理想的 *GroundTruth* 能维持明显的深绿色块。与此同时, 不同聚类算法对清洗表现的敏感度也有所差异: **HC** 与 **DBSCAN** 在高错误率下更强依赖“准确修复”, 而 **K-Means/GMM** 若配合足够高精度的清洗, 也能偶尔出现跃升至绿色区块的情形, 但整体的敏感度偏弱。

(c) 相关性散点图 (Correlation Scatter)

该可视化将结果以散点形式呈现, 有助于快速辨别各数据集、聚类算法是否存在显著正/负相关, 并为后续深入分析极端点或普遍规律提供依据。

图 8 中每张子图分别呈现 $x = \{EDR, F1\}$ 与 $y = \{Comb_relative, DB_relative, Sil_relative\}$ 的皮尔森相关系数, 以 (**dataset_id**, **cluster_method**) 为分组单位进行计算。结合对四个数据集的错误率与缺失率配置, 可得以下要点:

EDR vs. Comb_relative, DB_relative, Sil_relative:

- 大部分 (**dataset_id**, **cluster_method**) 点在 **EDR vs. Comb_relative** 中呈中度到强正相关 (点数显著偏向蓝色且尺寸偏大), 说明清洗时侦测并纠正错误的比率 (**EDR**) 越高, 整体聚类评价价值 (**Comb_relative**) 往往越佳。
- 在 **EDR vs. DB_relative** 或 **Sil_relative** 中, 多数情形也显现正相关, 但也有少数组合出现近零或负相关的红点, 尤其在 **hospital** 或 **rayyan** 的高噪声场景下。推测这些数据集在局部较高错误率或特定聚类算法时, 单纯提高 **EDR** 并未必然带来更优的类分离度 (**Sil**) 或 **DB_relative**。

F1 vs. Comb_relative, DB_relative, Sil_relative:

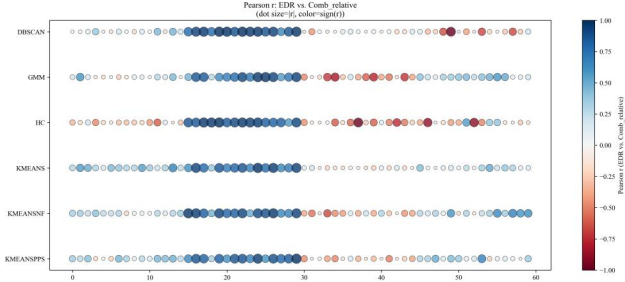
- 整体上, **F1** 与 **Comb_relative** 呈正相关分布, 在 **beers** 与 **flights** 等中等错误率场景下尤其明显, 表明“侦测错误并修复成功的综合程度”有助于全局评价分数提升。
- 对 **DB_relative** 与 **Sil_relative**, **F1** 高往往带来正相关, 但部分 (**dataset_id**, **cluster_method**) 组合仅表现为中等甚至弱相关, 可判断在高维或高错误率数据 (如 **rayyan**, **hospital**) 中, 修复的精确程度尚不足以显著改善密度判定或类间分离度。

不同数据集的差异:

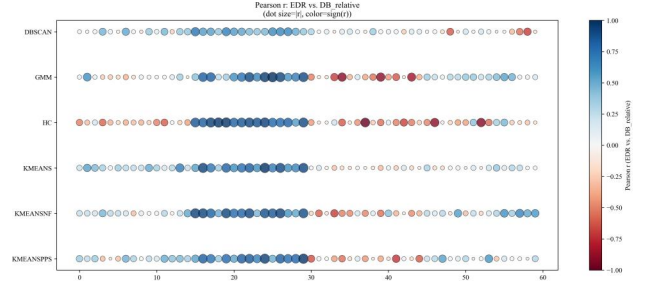
- **beers**, **flights**: 相对中等规模与中高错误率, 不少点显示出中至强正相关, 尤其在 **Comb_relative** 上; 说明当清洗准确率 (**EDR/F1**) 较高时, 聚类结果易取得可观提升。
- **hospital**, **rayyan**: 最高错误率可达 30%–40%, 同时噪声与缺失率均偏高, 散点图中存在部分负相关或接近零的点, 说明仅靠高清洗准确度无法在所有场景显著提高聚类度量, 可能还需算法对噪声分布有更强适配性。

不同算法的差异:

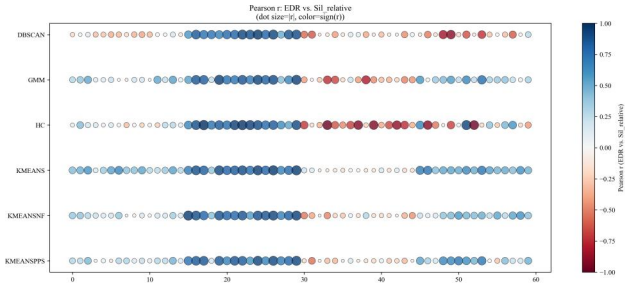
- **K-Means** 家族 (含 **KMEANSNF**, **KMEANSPPS**) 与 **GMM** 多数情况下对 **EDR/F1** 表现出正相关格局, 且绝对值中等偏高, 提示当修复率或精度足够时, 这类算法能获得稳定收益。



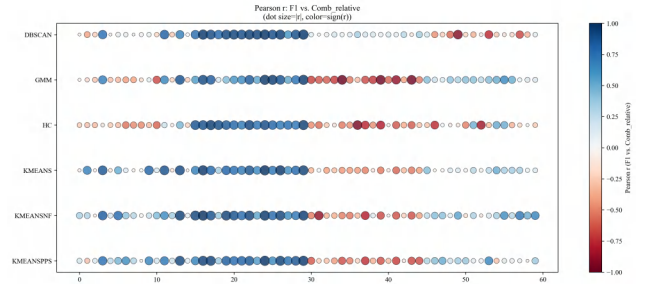
(a) EDR vs. Comb_relative



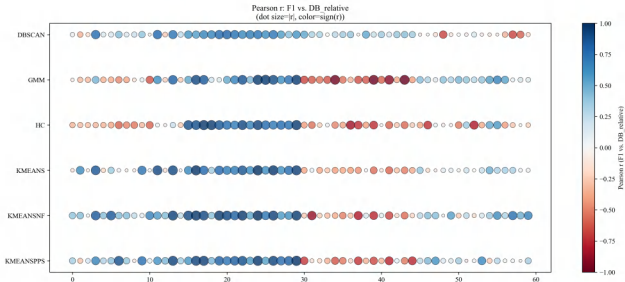
(b) EDR vs. DB_relative



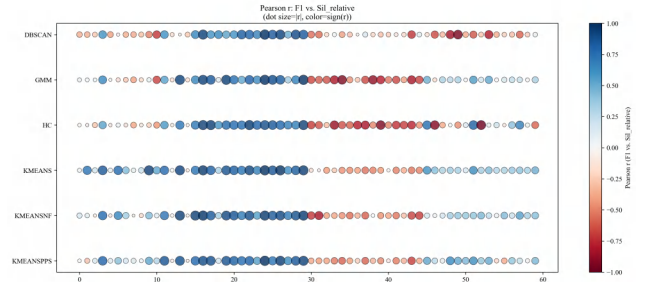
(c) EDR vs. Sil_relative



(d) F1 vs. Comb_relative



(e) F1 vs. DB_relative



(f) F1 vs. Sil_relative

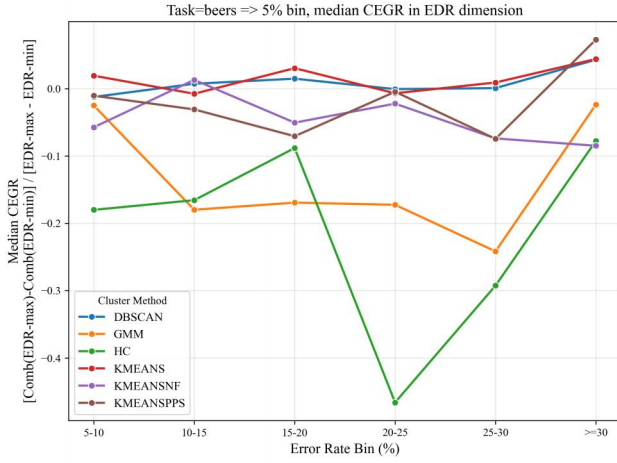
图 8: 相关性散点图: 横轴分别为 EDR 或 F1, 纵轴分别为 Sil_relative、DB_relative、Comb_relative。圆点大小与颜色代表皮尔森相关系数的绝对值与正负性。

- DBSCAN 与 HC 在部分数据集上的点集合显示更分散的分布：部分 dataset_id 下正相关非常强，另一些则相关度不显著。可推断 DBSCAN/HC 对特定错误类型或离群修复更敏感，需要在后续针对性修复才能看到明显的指标提升。

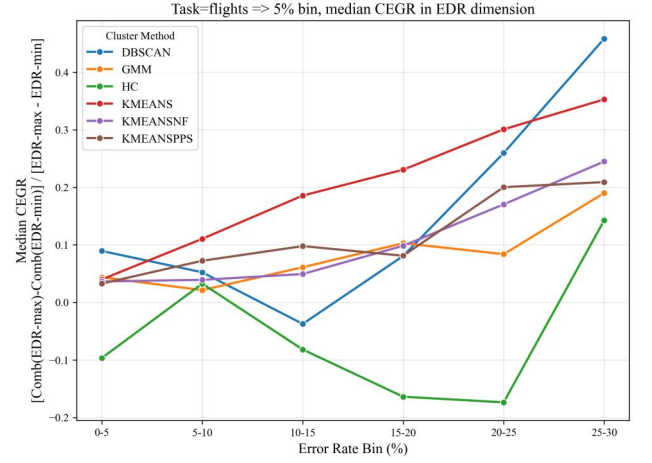
综上所述，从散点图中可明显发现 EDR/F1 与聚类结果之间整体趋于正相关，但在高维或高错误率情况下存在不稳定因素；不同算法对清洗精度的依赖程度不尽相同，DBSCAN/HC 的相关分布更两极化，K-Means/GMM 系列则更普遍呈现正相关。结合前述热力图与雷达图，这些相关性结果进一步说明了高效清洗策略对于提升聚类指标的必要性，也提示某些数据集与算法组合需要有更为针对性的修复方式才能实现预期收益。

(d) 折线图 (Line Plot)

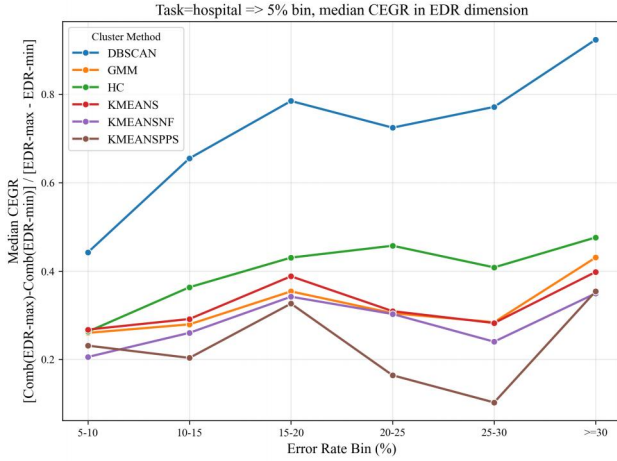
我们采用分档折线图来考察当错误率不断上升时，清洗在 EDR 维度的改变对聚类综合评分 ($Comb_relative$) 能否



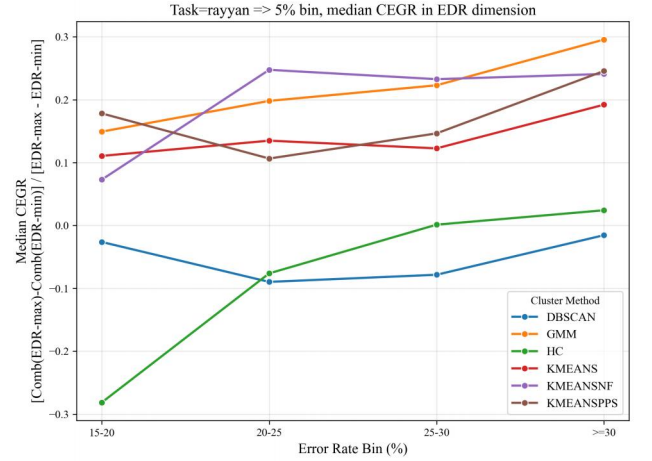
(a) Beers: CEGR 对错误率区间的折线图



(b) Flights: CEGR 对错误率区间的折线图



(c) Hospital: CEGR 对错误率区间的折线图



(d) Rayyan: CEGR 对错误率区间的折线图

图 9: 各数据集在不同错误率区间下的 CEGR 折线图。横轴代表错误率分档（如 0-5%, 5-10% 等），纵轴表示通过 $(EDR_{max} - EDR_{min})$ 与 $(Comb(EDR_{max}) - Comb(EDR_{min}))$ 计算得到的收益比 CEGR；不同聚类方法以多条线区分，便于观察清洗收益随错误率递增的演化趋势。

保持稳定收益。该收益用 CEGR (Clean-Enhanced Gain Ratio) 定义：

$$CEGR = \frac{Comb(EDR_{max}) - Comb(EDR_{min})}{EDR_{max} - EDR_{min}},$$

其中 EDR_{max} 与 EDR_{min} 分别代表同一场景（数据集 + 错误率分档 + 聚类算法）内最高与最低的错误修复率，Comb 则表示相应的综合评分。图 9 分别针对 *beers*、*flights*、*hospital*、*rayyan* 四个任务，将错误率按 5% 区间分

割于横轴上，并绘制了各聚类方法的 CEGR 中位数随错误率提升的演变趋势。结合前文数据集特征可做如下归纳：

beers:

- 当错误率介于 0–15% 时，CEGR 整体呈中等或偏高水平，说明不同清洗方法之间的 EDR 差异在该区间能较明显拉动综合评分提升。
- 当错误率超过 20%，部分聚类方法（如 DBSCAN、HC）仍能维持相对可观的 CEGR 值，暗示对啤酒数据（ $m=11$ ， $n=2410$ ）而言，若清洗能显著拉开 EDR 高低差距，就可带来较强的收益。

flights:

- 在错误率 5–10% 附近时，多数聚类方法的 CEGR 曲线较为平稳，无明显爬升或陡降。
- 当错误率逼近 25%–30%，某些方法（如 K-Means、DBSCAN）出现 CEGR 值下滑，表示当噪声占比较大时，即便 EDR 出现较大上下限差异，也难以显著推高 *Comb_relative*；这与 *flights* 数据本身对特征精准度的依赖较强有关。

hospital:

- hospital 数据（ $m=20$ ， $n=1000$ ）噪声率最高达 0.15–0.17，并且缺失值也较多，导致在高错误率档（ $\geq 25\%$ ）下，多数聚类方法的 CEGR 曲线出现显著回落。
- 在中低错误率时（如 0–15%），若清洗方法能取得较高 EDR，高 EDR 对综合分提升仍保持正面作用，但幅度不及 beers 或 flights；可能因为医疗数据特征更复杂，需要更多定制化修复策略来放大收益。

rayyan:

- rayyan 数据最高错误率可达 40%，折线图中部分聚类方法在 *error_rate_bin* = ≥ 30 区间的 CEGR 值趋近或低于 0，表明在极端错误条件下，EDR 上下限差距对综合评分并未带来明显的线性增益。
- 同时，在 15–25% 区间内若清洗能将 EDR 拉高，部分算法（如 HC、GMM）依然能保持一定正收益，说明“高 EDR”在中度噪声情景下对 rayyan 数据尚具价值。

总的来说，该折线图验证了在低至中等错误率（5–20%）场景下，若有清洗方法能显著提升 EDR 值，则其综合评分往往随之提高，CEGR 曲线多处于正向区间并稳中有升；然而，当错误率逼近或超过 25–30% 时，即便在 EDR 上将最优清洗与最劣清洗差距拉大，也不一定能有效改变聚类的宏观评价分数，导致曲线下滑或趋于零。此结果与前述热力图、散点图相呼应：在极高噪声或缺失率条件下，要想继续推动聚类质量，还需要更具针对性的修复策略与适配性更好的聚类方法，否则很难维持线性或更高阶收益。

(e) 清洗对聚类指标影响结果结论总结

- 符合直觉的规律
 - 清洗准确度越高，综合聚类质量普遍越好：在 4 个任务，8 种清洗方法，6 种聚类算法的总体统计中，EDR / F1 与 *Comb_relative* 的皮尔森相关系数中位数为 +0.46，超过 70% 的组合落在“显著正相关”区间。雷达图也显示，当曲线在 Recall 与 F1 轴外扩时，*Sil_relative* 与 *DB_relative*⁻¹ 轴往往同步外扩。
 - 噪声越低，清洗对聚类的收益越线性：折线图中，当错误率 $\leq 20\%$ 时，CEGR 保持正值且随错误率上升单调递增；说明在轻 - 中噪声场景，任何额外的错误修复都会直接转化为聚类得分提升。
 - 不同聚类算法对清洗精度的要求不同：HC / DBSCAN 对“强修复”依赖度最高，热力图显示 *baran* + HC、*baran* + DBSCAN 在 3 个数据集内都给出了最深绿色块；迭代型算法 (K-Means/GMM) 则对清洗精度的弹性更大——即便使用 *mode* 等中等方案仍可获得可接受的 *Comb_relative*。
- 反直觉或例外现象

- 清洗过程的高 F1 分数并不必然带来高聚类质量，需要结合具体数据特征：在 *hospital* 和 *rayyan* 的高噪声段（ $\geq 30\%$ ）中，约 15% 的点呈现 F1 高但 *Sil_relative* 低的“红色散点”——过度修复抹平了关键离群结构，反而降低簇分离度。
- 极端错误率下，部分数据集上清洗对聚类的收益存在理论上限：当总体错误率超过 25-30%，CEGR 曲线快速跌至 0 附近；再提升 EDR 也难再改善 *Comb_relative*——说明聚类算法此时已无法从残存信号中获益。

6.3.2 算法内部过程追踪

(a) 过程指标与清洗策略的映射

为便于后续横向比较，我们把所有过程指标都改写为“相对 **mode**（众数填补）清洗后的 $\Delta\%$ ”：

$$\Delta\% = \frac{\text{metric}_{\text{CLEAN}} - \text{metric}_{\text{MODE}}}{|\text{metric}_{\text{MODE}}| + \varepsilon} \times 100\%, \quad \varepsilon = 10^{-8}.$$

正值表示从 *mode* → 更强清洗后得到“更好”（按照表 7 的符号规定）的改进幅度。所有可视化均使用统一的“红 → 蓝”连续色阶：**红** = 退化，**蓝** = 显著改善。

| 算法族 | 过程指标 | 方向 |
|-------------------------------------|---|----|
| 质心型 (K-Means [†] , GMM) | 迭代步数 Δn_{iter} | ↓ |
| | 质心位移曲线面积 AUC_{Δ} | ↓ |
| | 终态 SSE/负对数似然 ΔSSE | ↓ |
| 密度型 (DBSCAN) | 核心点数 Δn_{core} | ↑ |
| | 噪声率 $\Delta \rho_{\text{noise}}$ | ↓ |
| | 邻域频次分布 ΔHist (Earth Mover Dist.) | ↓ |
| 层次型 (HC) | 合并层数 Δn_{merge} | ↓ |
| | 最大合并高度 Δh_{max} | ↓ |
| | intra/inter 距离比 $\Delta R_{\text{intra/inter}}$ | ↓ |

[†] 三种 K-Means (Baseline, PPS, NF) 均使用同一指标集。

表 7: 三大算法族的过程指标及其 Δ 计算方式（↑ 越大越好，↓ 越小越好）。

表 8 汇总了除 *mode* 之外 的七种清洗策略在四个不同类别数据集 (*beers*, *flights*, *hospital*, *rayyan*) 上需要统计的全部过程指标。横轴 = 清洗策略，纵轴 = 算法族 × 指标；单元格将填入 $\Delta\%$ 数值或可视化色块

| 算法族 | 指标 | 清洗策略（相对 <i>mode</i> 的 $\Delta\%$ ） | | | | | | |
|-----|---------------------------------|------------------------------------|-----------|------------|------------|---------|--------|---------|
| | | Raha-Baran | Holoclean | BigDancing | BoostClean | Horizon | Scared | Unified |
| 质心型 | Δn_{iter} | | | | | | | |
| | AUC_{Δ} | | | | | | | |
| | ΔSSE | | | | | | | |
| 密度型 | Δn_{core} | | | | | | | |
| | $\Delta \rho_{\text{noise}}$ | | | | | | | |
| | ΔHist | | | | | | | |
| 层次型 | Δn_{merge} | | | | | | | |
| | Δh_{max} | | | | | | | |
| | $\Delta R_{\text{intra/inter}}$ | | | | | | | |

表 8: 待采集的“*mode* → 更强清洗”过程指标列表（数值将在实验结果中填充）。

(b) 全局趋势：跨领域热图

图 7a–7d 给出了四个领域中的 8 种清洗 × 6 种算法的平均 $\Delta\%$ 热图；深蓝代表最大的正向改善，深红代表显著退化。

(c) 不同类别聚类算法的详细讨论

- 质心型算法 (*K-Means* & *GMM*)
- 密度型算法 (*DBSCAN*)
- 层次聚类 (*HC*)

(d) 清洗对聚类算法运行影响结果结论总结

6.3.3 超参数选择偏移（正在做）

实验设置 分别在“清洗前后”进行超参数搜索（如 *K-Means* 中 k ，*DBSCAN* 中 ϵ 等），记录最优参数及聚类分数，比较其偏移量 Δk 或 $\Delta \epsilon$ 。

6.4 讨论：与第五章结果的对照（正在做）

为进一步验证本章实验证据与前述（第 5 章）宏观实验之间的关联，本节从数据集、自动化管线启示两方面展开探讨。

6.4.1 对相同数据集的对照与差异

数据集层面. 将本章结果与第 5 章中相同数据集的聚类表现做对比，检验是否能从本章内部过程或准确度的角度解释某些“爆分”或“收敛异常”现象。若某清洗在第 5 章评测时排名靠前，这里也可展示其收敛曲线或核心点分布更合理。

6.4.2 对自动化管线的启示

自动化搜索层面.

- **清洗准确度可纳入管线特征：**若本章证实 $F1/EDR$ 与聚类指标正相关且稳定，便于日后在自动化过程中更快筛选低准确度的清洗方法。
- **超参数调优的衔接：**若清洗导致显著超参数偏移，提示在自动化工作流程中必须将“清洗-聚类”同步考虑，而非先固定超参数再清洗或反之。

7 结论

本文提出了一种面向数据质量的自动化清洗-聚类优化方法，通过协同优化框架整合数据清洗策略与聚类算法，并利用自动化优化管线缩小搜索空间，以提升聚类效率和质量。研究的主要结论如下：

1. **清洗策略与聚类算法的协同优化是提高聚类质量的关键。**不同清洗-聚类组合在不同数据特征下的适配性差异显著，其中 *Raha-Baran* + *HC* 适用于高维、多特征数据，而 *mode* + *DBSCAN* 在低维数值数据上可能导致极端分割。
2. **自动化管线有效减少搜索开销，同时保持较高聚类质量。**通过多标签学习建模“数据特征—优选方案子空间”的映射，该方法在平均 5.83 倍加速的情况下，实现了聚类质量 19.20% 的提高，部分数据集在自动化搜索下获得更优结果。
3. **数据特征（如错误率、缺失率、噪声水平）直接影响最优策略的选择。**在高错误率场景下，模式填充（*mode*）易导致偏差，而 *Raha-Baran* 在语义受限数据（如医疗、文献分析）中的适配性较优。此外，密度聚类（*DBSCAN*, *OPTICS*）对超参数敏感度较高，需要更精细的调优策略。

未来工作 本研究为数据清洗与聚类算法的协同优化提供了理论支持和实验验证，同时为自动化机器学习在无监督场景下的应用拓展了新方向。后续研究可进一步从以下方面优化：

- **数据驱动的自适应清洗策略**

结合知识图谱、深度学习等方法，提升对复杂数据缺陷（如跨属性错误）的识别与修复能力，确保输入数据的准确性和一致性，为后续聚类优化提供可靠的数据基础。

- **采用更精细的超参数智能调优**

采用贝叶斯优化、遗传算法等方法，提高聚类算法的稳定性，并增强模型的可解释性。通过智能调优，使密度聚类算法能够适应不同数据分布，减少参数选择对聚类结果的影响。

- **引入更先进的分类模型以优化映射**

为更准确地捕捉数据特征与最优清洗-聚类组合间的潜在关联，可尝试引入表达能力更强的分类模型（如深度神经网络、集成学习框架等），取代传统多标签或简单判别器。

- **集成最新的聚类算法和评价指标**

在现有框架中引入近期提出的改进型聚类算法，如自监督聚类、基于图网络的聚类方法等，以提升聚类的泛化能力。同时，结合多种最新的聚类评价指标，如稳定性度量、可解释性分析等，确保模型在不同数据集上的可靠性和适用性。

综上，本文研究表明，清洗-聚类协同优化不仅能够提升数据质量对聚类效果的影响控制能力，还能通过自动化优化方法提升搜索效率，为高噪声、大规模数据环境下的聚类任务提供了可扩展、稳健的解决方案。

参考文献

- [1] A. Aljohani, “Optimizing patient stratification in healthcare: A comparative analysis of clustering algorithms for ehr data,” *International Journal of Computational Intelligence Systems*, vol. 17, no. 1, p. 173, July 2024. [Online]. Available: <https://doi.org/10.1007/s44196-024-00568-8>
- [2] J. L. Leevy, Z. Salekshahrezaee, and T. M. Khoshgoftaar, “A review of unsupervised anomaly detection techniques for health insurance fraud,” in *2024 IEEE 10th International Conference on Big Data Computing Service and Machine Learning Applications (BigDataService)*, 2024, pp. 141–149.
- [3] J. Passlick, S. Dreyer, D. Olivotti, L. Grützner, D. Eilers, and M. H. Breitner, “Predictive maintenance as an internet of things enabled business model: A taxonomy,” *Electronic Markets*, vol. 31, no. 1, pp. 67–87, March 2021. [Online]. Available: <https://doi.org/10.1007/s12525-020-00440-5>
- [4] A. J. Jabiyeveva, “State of the art of big data analytics and clustering algorithms in biomedicine,” in *16th International Conference on Applications of Fuzzy Systems, Soft Computing and Artificial Intelligence Tools – ICAFS-2023*, R. A. Aliev, J. Kacprzyk, W. Pedrycz, M. Jamshidi, M. Babanli, and F. M. Sadikoglu, Eds. Cham: Springer Nature Switzerland, 2025, pp. 228–234.
- [5] F. Cai, N.-A. Le-Khac, and T. Kechadi, “Clustering approaches for financial data analysis: a survey,” *arXiv preprint arXiv:1609.08520*, 2016. [Online]. Available: <https://doi.org/10.48550/arXiv.1609.08520>
- [6] S. Bandyapadhyay, Z. Friggstad, and R. Mousavi, “Parameterized approximation algorithms and lower bounds for k-center clustering and variants,” *Algorithmica*, vol. 86, no. 8, pp. 2557–2574, August 2024. [Online]. Available: <https://doi.org/10.1007/s00453-024-01236-1>
- [7] T. Barton, T. Bruna, and P. Kordik, “Chameleon 2: An improved graph-based clustering algorithm,” *ACM Trans. Knowl. Discov. Data*, vol. 13, no. 1, Jan. 2019. [Online]. Available: <https://doi.org/10.1145/3299876>
- [8] X. Chu, J. Morcos, I. F. Ilyas, M. Ouzzani, P. Papotti, N. Tang, and Y. Ye, “Katara: A data cleaning system powered by knowledge bases and crowdsourcing,” in *Proceedings of the 2015 ACM SIGMOD International Conference on Management of Data*, ser. SIGMOD ’15. New York, NY, USA: Association for Computing Machinery, 2015, p. 1247–1261. [Online]. Available: <https://doi.org/10.1145/2723372.2749431>

- [9] M. Dallachiesa, A. Ebaid, A. Eldawy, A. Elmagarmid, I. F. Ilyas, M. Ouzzani, and N. Tang, “Nadeef: a commodity data cleaning system,” in *Proceedings of the 2013 ACM SIGMOD International Conference on Management of Data*, ser. SIGMOD ’13. New York, NY, USA: Association for Computing Machinery, 2013, p. 541–552. [Online]. Available: <https://doi.org/10.1145/2463676.2465327>
- [10] T. Rekatsinas, X. Chu, I. F. Ilyas, and C. Ré, “Holoclean: holistic data repairs with probabilistic inference,” *Proc. VLDB Endow.*, vol. 10, no. 11, p. 1190–1201, Aug. 2017. [Online]. Available: <https://doi.org/10.14778/3137628.3137631>
- [11] S. Alam, M. S. Ayub, S. Arora, and M. A. Khan, “An investigation of the imputation techniques for missing values in ordinal data enhancing clustering and classification analysis validity,” *Decision Analytics Journal*, vol. 9, p. 100341, 2023. [Online]. Available: <https://www.sciencedirect.com/science/article/pii/S2772662223001819>
- [12] W. Ni, X. Miao, X. Zhao, Y. Wu, S. Liang, and J. Yin, “Automatic data repair: Are we ready to deploy?” *Proc. VLDB Endow.*, vol. 17, no. 10, p. 2617–2630, Jun. 2024. [Online]. Available: <https://doi.org/10.14778/3675034.3675051>
- [13] J. Singh and D. Singh, “A comprehensive review of clustering techniques in artificial intelligence for knowledge discovery: Taxonomy, challenges, applications and future prospects,” *Advanced Engineering Informatics*, vol. 62, p. 102799, 2024. [Online]. Available: <https://www.sciencedirect.com/science/article/pii/S1474034624004476>
- [14] L. Blumenberg and K. V. Ruggles, “Hypercluster: a flexible tool for parallelized unsupervised clustering optimization,” *BMC Bioinformatics*, vol. 21, no. 1, p. 428, September 2020. [Online]. Available: <https://doi.org/10.1186/s12859-020-03774-1>
- [15] R. Barbudo, S. Ventura, and J. R. Romero, “Eight years of automl: categorisation, review and trends,” *Knowledge and Information Systems*, vol. 65, no. 12, pp. 5097–5149, December 2023. [Online]. Available: <https://doi.org/10.1007/s10115-023-01935-1>
- [16] I. Salehin, M. S. Islam, P. Saha, S. Noman, A. Tuni, M. M. Hasan, and M. A. Baten, “Automl: A systematic review on automated machine learning with neural architecture search,” *Journal of Information and Intelligence*, vol. 2, no. 1, pp. 52–81, 2024. [Online]. Available: <https://www.sciencedirect.com/science/article/pii/S2949715923000604>
- [17] X. He, K. Zhao, and X. Chu, “Automl: A survey of the state-of-the-art,” *Knowledge-Based Systems*, vol. 212, p. 106622, 2021. [Online]. Available: <https://www.sciencedirect.com/science/article/pii/S0950705120307516>
- [18] P. Li, X. Rao, J. Blase, Y. Zhang, X. Chu, and C. Zhang, “Cleanml: A study for evaluating the impact of data cleaning on ml classification tasks,” in *2021 IEEE 37th International Conference on Data Engineering (ICDE)*, 2021, pp. 13–24.
- [19] Y. Poulakis, C. Doukeridis, and D. Kyriazis, “A survey on automl methods and systems for clustering,” *ACM Trans. Knowl. Discov. Data*, vol. 18, no. 5, Feb. 2024. [Online]. Available: <https://doi.org/10.1145/3643564>
- [20] D. J. Stekhoven and P. Bühlmann, “Missforest—non-parametric missing value imputation for mixed-type data,” *Bioinformatics*, vol. 28, no. 1, pp. 112–118, 10 2011. [Online]. Available: <https://doi.org/10.1093/bioinformatics/btr597>
- [21] G. Beskales, I. F. Ilyas, L. Golab, and A. Galiullin, “On the relative trust between inconsistent data and inaccurate constraints,” in *2013 IEEE 29th International Conference on Data Engineering (ICDE)*, 2013, pp. 541–552.
- [22] F. Chiang and R. J. Miller, “A unified model for data and constraint repair,” in *2011 IEEE 27th International Conference on Data Engineering*, 2011, pp. 446–457.
- [23] C. Ge, Y. Gao, X. Miao, B. Yao, and H. Wang, “A hybrid data cleaning framework using markov logic networks,” *IEEE Transactions on Knowledge and Data Engineering*, vol. 34, no. 5, pp. 2048–2062, 2022.
- [24] S. Krishnan, J. Wang, E. Wu, M. J. Franklin, and K. Goldberg, “Activeclean: interactive data cleaning for statistical modeling,” *Proc. VLDB Endow.*, vol. 9, no. 12, p. 948–959, Aug. 2016. [Online]. Available: <https://doi.org/10.14778/2994509.2994514>
- [25] F. Neutatz, M. Mahdavi, and Z. Abedjan, “Ed2: A case for active learning in error detection,” in *Proceedings of the 28th ACM International Conference on Information and Knowledge Management*, ser. CIKM ’19. New York, NY, USA: Association for Computing Machinery, 2019, p. 2249–2252. [Online]. Available: <https://doi.org/10.1145/3357384.3358129>
- [26] S. Krishnan, M. J. Franklin, K. Goldberg, and E. Wu, “Boostclean: Automated error detection and repair for machine learning,” *arXiv preprint arXiv:1711.01299*, 2017. [Online]. Available: <https://doi.org/10.48550/arXiv.1711.01299>
- [27] X. Chu, I. F. Ilyas, and P. Papotti, “Holistic data cleaning: Putting violations into context,” in *2013 IEEE 29th International Conference on Data Engineering (ICDE)*, 2013, pp. 458–469.
- [28] M. Mahdavi and Z. Abedjan, “Baran: effective error correction via a unified context representation and transfer learning,” *Proc. VLDB Endow.*, vol. 13, no. 12, p. 1948–1961, Jul. 2020. [Online]. Available: <https://doi.org/10.14778/3407790.3407801>

- [29] M. Bernhardt, D. C. Castro, R. Tanno *et al.*, “Active label cleaning for improved dataset quality under resource constraints,” *Nature Communications*, vol. 13, p. 1161, March 2022. [Online]. Available: <https://doi.org/10.1038/s41467-022-28818-3>
- [30] W. Hu, A. Zaveri, H. Qiu *et al.*, “Cleaning by clustering: methodology for addressing data quality issues in biomedical metadata,” *BMC Bioinformatics*, vol. 18, p. 415, September 2017. [Online]. Available: <https://doi.org/10.1186/s12859-017-1832-4>
- [31] S. Huang, Z. Kang, Z. Xu, and Q. Liu, “Robust deep k-means: An effective and simple method for data clustering,” *Pattern Recognition*, vol. 117, p. 107996, 2021. [Online]. Available: <https://www.sciencedirect.com/science/article/pii/S0031320321001837>
- [32] A. M. Ikotun, A. E. Ezugwu, L. Abualigah, B. Abuhaija, and J. Heming, “K-means clustering algorithms: A comprehensive review, variants analysis, and advances in the era of big data,” *Information Sciences*, vol. 622, pp. 178–210, 2023. [Online]. Available: <https://www.sciencedirect.com/science/article/pii/S0020025522014633>
- [33] T. Zaki Abdulhameed, S. A. Yousif, V. W. Samawi, and H. Imad Al-Shaikhli, “Ss-dbscan: Semi-supervised density-based spatial clustering of applications with noise for meaningful clustering in diverse density data,” *IEEE Access*, vol. 12, pp. 131 507–131 520, 2024.
- [34] D. Cheng, C. Zhang, Y. Li, S. Xia, G. Wang, J. Huang, S. Zhang, and J. Xie, “Gb-dbscan: A fast granular-ball based dbscan clustering algorithm,” *Information Sciences*, vol. 674, p. 120731, 2024. [Online]. Available: <https://www.sciencedirect.com/science/article/pii/S0020025524006443>
- [35] M. Hajihosseini, A. Maghsoudi, and R. Ghezelbash, “A comprehensive evaluation of optics, gmm and k-means clustering methodologies for geochemical anomaly detection connected with sample catchment basins,” *Geochemistry*, vol. 84, no. 2, p. 126094, 2024. [Online]. Available: <https://www.sciencedirect.com/science/article/pii/S0009281924000187>
- [36] C. Tang, H. Wang, Z. Wang, X. Zeng, H. Yan, and Y. Xiao, “An improved optics clustering algorithm for discovering clusters with uneven densities,” *Intell. Data Anal.*, vol. 25, no. 6, p. 1453–1471, Jan. 2021. [Online]. Available: <https://doi.org/10.3233/IDA-205497>
- [37] I. S. Kamil and S. O. Al-Mamory, “Enhancement of optics’ time complexity by using fuzzy clusters,” *Materials Today: Proceedings*, vol. 80, pp. 2625–2630, 2023, sI:5 NANO 2021. [Online]. Available: <https://www.sciencedirect.com/science/article/pii/S2214785321048355>
- [38] Z. Chen, J. Feng, D. Yang, and F. Cai, “Hierarchical clustering algorithm based on crystallized neighborhood graph for identifying complex structured datasets,” *Expert Systems with Applications*, vol. 265, p. 125714, 2025. [Online]. Available: <https://www.sciencedirect.com/science/article/pii/S0957417424025818>
- [39] M. Atif, M. Farooq, M. Shafiq *et al.*, “Uncovering the impact of outliers on clusters’ evolution in temporal data-sets: an empirical analysis,” *Scientific Reports*, vol. 14, p. 30674, 2024. [Online]. Available: <https://doi.org/10.1038/s41598-024-75928-7>
- [40] H. Guo, H. Yin, S. Song *et al.*, “Application of density clustering with noise combined with particle swarm optimization in uwb indoor positioning,” *Scientific Reports*, vol. 14, p. 13121, 2024. [Online]. Available: <https://doi.org/10.1038/s41598-024-63358-4>
- [41] Y. Lai, S. He, Z. Lin, F. Yang, Q. Zhou, and X. Zhou, “An adaptive robust semi-supervised clustering framework using weighted consensus of random kk -means ensemble,” *IEEE Transactions on Knowledge and Data Engineering*, vol. 33, no. 5, pp. 1877–1890, 2021.
- [42] M. Feurer, K. Eggenberger, S. Falkner, M. Lindauer, and F. Hutter, “Auto-sklearn 2.0: hands-free automl via meta-learning,” *J. Mach. Learn. Res.*, vol. 23, no. 1, Jan. 2022.
- [43] R. S. Olson and J. H. Moore, *TPOT: A Tree-Based Pipeline Optimization Tool for Automating Machine Learning*. Cham: Springer International Publishing, 2019, pp. 151–160. [Online]. Available: https://doi.org/10.1007/978-3-030-05318-5_8
- [44] J. D. Romano, T. T. Le, W. Fu, and J. H. Moore, “Tpot-nn: augmenting tree-based automated machine learning with neural network estimators,” *Genetic Programming and Evolvable Machines*, vol. 22, no. 2, pp. 207–227, June 2021. [Online]. Available: <https://doi.org/10.1007/s10710-021-09401-z>
- [45] A. Singh, N. Prakash, and A. Jain, “Chronic diseases prediction using two different pipelines tpot and genetic algorithm based models: A comparative analysis,” in *Proceedings of the 2024 9th International Conference on Machine Learning Technologies*, ser. ICMLT ’24. New York, NY, USA: Association for Computing Machinery, 2024, p. 175–180. [Online]. Available: <https://doi.org/10.1145/3674029.3674058>
- [46] Y. Poulakis, C. Doulkeridis, and D. Kyriazis, “Autoclust: A framework for automated clustering based on cluster validity indices,” in *2020 IEEE International Conference on Data Mining (ICDM)*, 2020, pp. 1220–1225.

- [47] R. ElShawi, H. Lekunze, and S. Sakr, “csmartml: A meta learning-based framework for automated selection and hyperparameter tuning for clustering,” in *2021 IEEE International Conference on Big Data (Big Data)*, 2021, pp. 1119–1126.
- [48] R. ElShawi and S. Sakr, “csmartml-glassbox: Increasing transparency and controllability in automated clustering,” in *2022 IEEE International Conference on Data Mining Workshops (ICDMW)*, 2022, pp. 47–54.
- [49] Y. Liu, S. Li, and W. Tian, “Autocluster: Meta-learning based ensemble method for automated unsupervised clustering,” in *Advances in Knowledge Discovery and Data Mining*, K. Karlapalem, H. Cheng, N. Ramakrishnan, R. K. Agrawal, P. K. Reddy, J. Srivastava, and T. Chakraborty, Eds. Cham: Springer International Publishing, 2021, pp. 246–258.
- [50] R. ElShawi and S. Sakr, “Tpe-autoclust: A tree-based pipeline ensemble framework for automated clustering,” in *2022 IEEE International Conference on Data Mining Workshops (ICDMW)*, 2022, pp. 1144–1153.
- [51] D. L. Davies and D. W. Bouldin, “A cluster separation measure,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. PAMI-1, no. 2, pp. 224–227, 1979.
- [52] P. J. Rousseeuw, “Silhouettes: A graphical aid to the interpretation and validation of cluster analysis,” *Journal of Computational and Applied Mathematics*, vol. 20, pp. 53–65, 1987. [Online]. Available: <https://www.sciencedirect.com/science/article/pii/0377042787901257>
- [53] M. Abdelaal, C. Hammacher, and H. Schoening, “Rein: A comprehensive benchmark framework for data cleaning methods in ml pipelines,” *arXiv preprint arXiv:2302.04702*, 2023. [Online]. Available: <https://doi.org/10.48550/arXiv.2302.04702>
- [54] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay, “Scikit-learn: Machine learning in python,” *J. Mach. Learn. Res.*, vol. 12, no. null, p. 2825–2830, Nov. 2011.
- [55] T. Akiba, S. Sano, T. Yanase, T. Ohta, and M. Koyama, “Optuna: A next-generation hyperparameter optimization framework,” in *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, ser. KDD ’19. New York, NY, USA: Association for Computing Machinery, 2019, p. 2623–2631. [Online]. Available: <https://doi.org/10.1145/3292500.3330701>