

# 面向数据质量的清洗方法与下游聚类算法协同优化的自动化模型研究

常添

哈尔滨工业大学

2022111699@stu.hit.edu.cn

2025 年 2 月 4 日

## 摘要

在无监督学习任务中，数据质量问题（如错误值、缺失值与噪声）常显著影响聚类算法的性能。针对这一挑战，本文提出了一种协同优化方法，将数据清洗策略与聚类算法的组合纳入统一框架。通过构建综合得分体系，对多种清洗-聚类组合在不同数据集特征下的表现和适配性进行系统量化和广泛评估；同时，基于先验数据特征，采用多标签学习技术实现从“数据特征”到“优选方案子空间”的映射，显著减少搜索空间；在此基础上，我们设计了一种自动化管线优化模型，在保证聚类质量的前提下有效提升了时间效率。实验结果表明，本文方法在多个公开数据集上均表现出较高的聚类精度与显著的效率提升，验证了其在数据清洗与聚类协同优化中的实用性与鲁棒性。本研究为应对大规模、高噪声数据场景下的聚类问题提供了理论支持和实际解决方案。

## 1 引言

在许多实际应用场景中（例如医疗、金融以及工业物联网等领域），数据经常面临缺失值、错误值与噪声等质量问题 [?, ?]。与分类或回归等有监督学习任务相比，聚类作为一种无监督学习方法对于数据分布的依赖性更为强烈 [?]，一旦数据中存在过多噪声或不确定性，就容易导致簇结构与真实分布出现显著偏离 [?]。这类偏离不仅影响聚类本身的准确性，也会对后续的模式挖掘和决策支持造成不可忽视的干扰。可见，在下游聚类任务中，数据质量对于结果的影响往往举足轻重。

为了减少噪声干扰与纠正错误数据，研究者们提出了多种数据清洗策略，包括缺失值填充、异常值检测与剔除、错误值纠正等 [?]。其核心目标在于“修复或减少数据中的噪声与错误”，以期在随后的分析过程中尽量保留准确可靠的分布结构。然而，清洗操作并不必然带来正向收益。有研究指出，过度严格的清洗可能修改或删除原本正确的数据 [?]; 过于简单的填充策略则会扭曲原有分布特征，反而可能削弱聚类算法对关键信息的捕捉能力 [?]。在少量噪声的情形下，去除异常点确实能使 K-Means 算法获得更稳定的聚类效果，但若这些“异常点”恰恰是某些簇的重要特征，则对基于密度的聚类方法（如 DBSCAN 算法）反而不利。由此可见，清洗方法与聚类算法之间存在紧密关联，如果仅从清洗或聚类单方面着手，往往难以兼顾双方需求。

现有研究通常在两条技术路线下分别进行：其一，机器学习视角侧重于改进聚类算法自身（例如 K-Means 的变体、层次聚类或密度聚类等） [?, ?, ?]; 其二，数据质量视角主要针对如何提升数据完整性与准确度 [?, ?]。然而，不同清洗策略会对数据分布带来不同程度的修正，而不同聚类算法又对噪声、缺失率等有着各自的敏感度，若只聚焦其中一端，难以获得全局最优的方案。因此，“清洗策略 + 聚类算法 + 超参数”一体的管线协同优化方案逐渐受到关注，但该管线的搜索空间常呈指数级增长，依赖人工穷举或简单试验往往难以在可接受的时间范围内完成。

为此，我们尝试在“数据质量（数据清洗理论）”与“自动化机器学习（AutoML）”两大领域之间建立桥梁，引入自动化聚类模型（基于 AutoML 的思路）来缩减庞大的管线搜索开销并兼顾聚类质量。当前大部分 AutoML 研究主要集中在有监督学习任务（如分类和回归）的模型选择与超参数优化上，对无监督学习——尤其是“聚类 + 清洗”这种协同自动化的探索仍较为有限 [?, ?]。因此，本研究将多种清洗方法、聚类算法及其参数一并纳入搜索空间，通过学习模型捕捉“数据特征与优选方案组合”之间的映射关系。当面对新的数据集及其特征时，系统能够自动推荐若干优选的清洗-聚类-参数组合，既能减少冗余探索，又能提升聚类效果的稳定性和准确性。

在无监督学习场景下，协同优化的自动化框架具有显著的潜力。聚类算法在产业界和学术界有着广泛的应用场景，而大多数真实世界的的数据通常都存在一定的质量问题。因此，如果能够实现对数据质量的适配清洗与高效聚类的有机结合，将大大拓宽聚类技术的应用范围，促进其在更多领域的落地。针对这一尚未得到充分探索的方向，本研究提出了一个面向

数据质量的清洗与聚类协同优化的自动化模型。这一模型不仅为学术研究提供了新的思路，也有望在实际应用中发挥重要作用。

**贡献。**我们针对“数据清洗与下游聚类协同优化”这一交叉研究领域，围绕算法性能与自动化效率，总结如下方面的贡献：

### 1. 系统评估多种清洗-聚类组合的有效性 with 局限性

我们基于 40 个具备多元质量问题的公开数据集，深入研究了 3 种清洗策略与 6 种聚类算法的交互关系，并针对不同错误率、噪声水平及数据规模的多场景进行实验测试。通过大量实证结果，量化了清洗方法与聚类算法之间的适配度，并揭示了特定组合在极端环境（如高维度、高错误率数据）下可能产生的极端现象及风险。这些结论为后续的清洗-聚类管线设计提供了可操作的参考依据，丰富了现有文献在无监督学习场景下对数据质量处理方法的系统性比较。

### 2. 提出基于管线思维的清洗与聚类协同优化框架

我们将“数据清洗策略 + 聚类算法 + 超参数”视作一个整体管线（Pipeline），并结合实验结果总结出在不同场景下的优先组合与适配性建议。该框架帮助研究者在设计无监督学习流程时，能同时考虑数据质量和算法性能，避免了只聚焦某一端而造成的局限性。

### 3. 构建并验证了自动化管线优化模型，显著提升效率与性能

在深入理解清洗与聚类之间协同关系的基础上，我们进一步设计了一种自动化管线模型：该模型能够根据数据集特征快速筛选可能的最优清洗-聚类组合，大幅削减搜索空间。与传统手动调参或穷举策略相比，自动化模型在效率指标上展现出显著优势，同时在多数数据集上保持了与完整搜索接近的聚类效果。我们以损失率（Loss Rate）和综合加速比（Acceleration Ratio）等指标量化了该模型在平衡聚类质量与运行时间方面的成效，为未来在大规模和多样化数据场景下的应用提供了可迁移的实践路径。

## 2 相关工作

当前针对数据清洗模型与聚类算法的研究，主要集中在以下三个方向：**(1) 机器学习视角下的聚类算法改进**；**(2) 数据质量视角下的清洗策略优化**；**(3) 无监督场景中的自动化机器学习探索**。本节将对相关工作进行梳理，并指出与本研究的区别。

### 2.1 机器学习视角：聚类算法的改进

从机器学习的角度，研究者更多关注聚类算法本身的改进和变体设计。例如，K-Means 在初始中心选择、迭代更新策略等方面衍生了多种增强方法，以提升收敛速度或降低局部最优的风险 [?, ?]；层次聚类与图聚类则在相似度度量和聚类结构可视化方面提出了新的思路 [?]。不过，这些研究通常默认数据预处理已经完成，或在简单清洗之后才进行聚类算法的优化，往往没有深入讨论不同清洗策略对数据分布和聚类效果的潜在影响。

### 2.2 数据质量视角：清洗策略及其影响

从数据质量视角出发，学者们在缺失值填充、异常值检测与错误值纠正等方面做了大量探索 [?, ?, ?]。常见方法包括基于统计或回归模型的缺失值插补，基于密度或距离的异常值检测等。然而，不同清洗策略对下游聚类效果的具体影响仍缺乏系统验证 [?, ?]，大多数工作只选择单一或少数聚类算法进行测试，对不同聚类方法在噪声和缺失率方面的差异需求尚未形成全面比较。

### 2.3 自动化机器学习：无监督场景的探索

AutoML 在有监督学习（分类与回归）方面已经取得了显著进展，如自动模型选择和超参数搜索等 [?, ?]。然而，对于无监督学习，尤其是“聚类 + 清洗”的联合自动化仍处于相对初步的探索阶段。现有少数工作关注自动确定聚类数目或部分预处理自动化，但尚未建立起“清洗策略 + 聚类算法 + 超参数”的综合管线搜索体系 [?, ?]。

### 2.4 与本研究的差异

综合上述三个方向，可以发现现有研究在“清洗-聚类”协同优化方面尚存在以下不足：

- **缺乏清洗与聚类交互影响的系统性评估：**当前研究多侧重于独立优化聚类算法或改进清洗策略，缺乏对两者交互影响的系统性分析，特别是在多场景、多数据特征下的验证。
- **缺乏统一的自动化管线优化框架：**现有方法通常通过手动配置或依赖单一预处理方案来调整清洗与聚类过程，缺少能够动态适配不同数据特征的端到端管线式优化框架。
- **高效搜索机制的缺失：**面对清洗策略、聚类算法及其超参数构成的庞大搜索空间，现有方法在效率和性能平衡方面仍存短板，难以满足实际应用中快速与高质量结果的需求。

针对上述问题，本文提出了一种自动化管线优化模型，将清洗策略、聚类算法及超参数统一纳入动态优化框架。通过结合数据特征和实验分析，我们的模型能够高效捕捉最优组合方案，并在多种场景中验证了其性能优势和适配性。与现有研究相比，本研究的自动化管线框架从全局角度实现了数据清洗与聚类算法的协同优化，为无监督学习提供了新的研究方向与实践路径。

### 3 问题和模型定义

在各种实际应用（如医疗、金融以及工业物联网等）中，数据由于错误值、缺失值及噪声等问题而呈现出多元的质量特征。如果仅依赖单一的预处理或聚类方式，往往难以兼顾不同场景下的精度需求与时间成本。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，本节将依次介绍相关概念、系统化的评价方法，以及应的映射模型，最后给出核心研究问题的形式化描述。

#### 3.1 核心概念与变量定义

令  $D$  表示待处理的数据集，其中可能同时存在错误值（Error）、缺失值（Missing）以及噪声（Noise）等质量问题。为了刻画这些问题与数据规模的差异，定义**特征向量**

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

其中  $\text{ErrorRate}(D)$ 、 $\text{MissingRate}(D)$  与  $\text{NoiseRate}(D)$  分别表示数据集中错误值、缺失值以及噪声的相对比例， $m$  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横向对比，也为后续的映射模型提供了可学习的输入特征。

在数据清洗与聚类算法的设定中，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）， $\mathcal{H}$  为聚类算法的集合（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  $\theta$  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

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

在实际应用中， $\Omega$  的规模常随数据清洗或聚类算法的多样性呈现出指数级增长，因此对其进行穷举评估常常带来巨大的时间与计算负担。

#### 3.2 评价系统与最优方案

为衡量任意策略  $\omega \in \Omega$  在数据集  $D$  上的聚类质量，通常采用若干无监督指标加以综合。本文主要采用 Davie-Bouldin (DB) 指数与轮廓系数 (Silhouette) 两类典型指标，并将它们线性组合为**综合得分**。

首先，对于算法划分出的  $K$  个簇，DB 指数通过考察簇内紧凑度与簇间分离度来衡量聚类效果，具体定义为

$$DB(D, \omega) = \frac{1}{K} \sum_{i=1}^K \max_{j \neq i} \left( \frac{S_i + S_j}{d_{ij}} \right), \quad (3.4)$$

其中  $S_i$  为第  $i$  个簇的平均离散度， $d_{ij}$  为簇  $i$  与簇  $j$  的中心距离，数值越低表示聚类效果越理想。

其次，轮廓系数 (Silhouette Coefficient) 量化了每个样本点  $x$  在其所属簇内的凝聚度以及跟最近邻簇的分离度，定义为：

$$\text{Sil}(x) = \frac{b(x) - a(x)}{\max\{a(x), b(x)\}}, \quad (3.5)$$

其中  $a(x)$  表示  $x$  与同簇内其他样本的平均距离,  $b(x)$  表示  $x$  到最近邻簇内样本的平均距离。总体轮廓系数为所有样本轮廓系数的平均值:

$$\text{Sil}(D, \omega) = \frac{1}{n} \sum_{x \in D} \text{Sil}(x), \quad (3.6)$$

其中  $n$  为数据集  $D$  中样本的总数。平均轮廓系数越高, 表示聚类划分越合理。

结合 DB 指数与轮廓系数, 可定义**综合得分**, 以量化清洗-聚类策略的整体效果:

$$S(D, \omega) = \alpha \cdot [-DB(D, \omega)] + \beta \cdot \text{Sil}(D, \omega), \quad (3.7)$$

其中  $\alpha, \beta > 0$  为可调权重,  $[-DB(D, \omega)]$  项强调 DB 指数越低越好, 便于与轮廓系数在同一方向上加和。通过综合得分  $S(D, \omega)$ , 本文能够系统评估不同清洗-聚类策略组合在特定数据集上的性能, 并为优选方案提供直观依据。

当我们仅从聚类精度角度出发, 给定  $D$  的最优策略可表示为

$$\omega^*(D) = \arg \max_{\omega \in \Omega} S(D, \omega). \quad (3.8)$$

然而, 若要在全量空间  $\Omega$  上评估每个策略  $\omega$ , 往往需要花费大量计算时间。为此, 我们将定义一个**优化子空间**  $\Omega'(D) \subseteq \Omega$ , 仅在其中执行评估, 以缓解时间消耗。假设评估单个策略的耗时记为  $T(D, \omega)$ , 则

$$T_{\text{original}}(D) = \sum_{\omega \in \Omega} T(D, \omega) \quad \text{与} \quad T_{\text{reduced}}(D) = \sum_{\omega \in \Omega'(D)} T(D, \omega) \quad (3.9)$$

分别表示完整搜索与缩减搜索的总耗时。若  $\Omega'(D)$  能在**保证聚类质量**的情况下大幅减少评估开销, 即可平衡精度与效率。为度量这两方面的综合效果, 引入**损失率与综合加速比**。

### 损失率与综合加速比

若令  $\bar{S}(\Omega)$  表示在完整搜索  $\Omega$  上得到的平均得分,  $\bar{S}(\Omega'(D))$  表示在优选子空间  $\Omega'(D)$  中的平均得分, 则可定义

$$\eta(D) = 1 - \frac{\bar{S}(\Omega'(D))}{\bar{S}(\Omega)}, \quad (3.10)$$

作为**损失率**, 反映缩减搜索后对聚类质量的平均影响; 值越接近 0, 表示压缩空间后性能损失越小。此外, 记

$$\mathcal{A}(D) = (1 - \eta(D)) \times \frac{T_{\text{original}}(D)}{T_{\text{reduced}}(D)}, \quad (3.11)$$

作为**综合加速比**, 数值越大意味着在质量损失可控的前提下获得了更显著的搜索加速效果。

### 3.3 从数据特征到优选方案的映射

在应用场景中, 不同数据集  $D$  往往拥有差异明显的特征向量  $\mathbf{x}(D)$ , 如错误率、缺失率、噪声率、样本规模等 (参考式 (3.1))。这些特征会显著影响“数据清洗 + 聚类算法”组合的表现, 使得某些策略更契合特定类型的数据分布。若能提前根据  $\mathbf{x}(D)$  预测哪些组合最可能获得较高综合分数  $S(D, \omega)$ , 就可减少对大规模搜索空间  $\Omega$  的穷举评估, 进而大幅缩减时间成本。

为此, 本文引入一个映射函数

$$G: \mathbf{x}(D) \mapsto \Omega'(D), \quad (3.12)$$

其中  $\Omega'(D) \subseteq \Omega$  为一个规模较小的**优选子空间**。通过在先验数据集上积累“数据特征—策略表现”的关联信息, 便可利用机器学习或统计方法 (例如多标签分类) 训练出映射  $G$ , 使其在新数据集上快速推荐表现优良的候选组合, 从而避免对完整空间  $\Omega$  的重复尝试。后续章节将结合实例场景和模型设计, 详细说明如何构建并验证该映射。

### 3.4 核心问题的形式化定义

在介绍了数据集特征、清洗-聚类策略与映射函数等概念后, 现对本文的核心研究任务做形式化总结。具体而言, 本研究聚焦以下三个关键问题:

(1) **定量评估与比较不同清洗-聚类组合的协同表现** 在初始搜索空间  $\Omega$  中, 每个策略  $\omega$  都可根据综合得分  $S(D, \omega)$  进行排名。如何通过该排名定量分析与比较不同组合在**适配性和协同优化**方面的表现, 进而识别出在给定数据集  $D$  上效果最为突出的组合? 这一过程需要关注数据质量特征对综合得分的影响机理, 并结合最优策略  $\omega^*(D)$  或排名靠前的若干策略作综合评判。

(2) **寻找并建立数据特征到优选方案集合的映射模型** 当数据集特征差异较大时, 最优或近优的组合往往随之发生变化, 难以通过固定规则一概而论。如何**基于数据特征  $\mathbf{x}(D)$  来自动或半自动地推荐一个优选子空间  $\Omega'(D)$** , 从而在不显著损失聚类质量的情况下减少搜索规模? 为解决这一问题, 需要构建映射函数

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

并在先验数据集上学习和验证该函数的可靠性。

(3) **建立自动化模型以在有限时间内找到接近最优的方案, 并尽量提高综合加速比** 若直接对  $\Omega$  进行完整搜索, 将面临极高的时间成本。因而, 需要设计一个**自动化流程**: 在给定  $\mathbf{x}(D)$  之后, 仅在子空间  $\Omega'(D)$  进行快速评估, 找到

$$\hat{\omega}(D) = \arg \max_{\omega \in \Omega'(D)} S(D, \omega). \quad (3.13)$$

同时, 引入**损失率  $\eta(D)$  和综合加速比  $\mathcal{A}(D)$**  (参考式 (3.10) 与 (3.11)), 量化压缩搜索空间带来的精度损失与效率提升。在**有限时间内**实现较低损失率与较高加速比, 为实际部署提供可行的协同优化方案。

基于上述三方面的研究重点, 本文将从实验与算法设计两个角度展开: 首先在多个具有噪声、缺失或错误值的问题数据集中, 系统验证和评估不同清洗-聚类组合的表现和适配性; 然后以多标签学习或自动化搜索策略为基础, 搭建映射机制与自动化管线, 力图在有限时间内逼近最优聚类效果并兼顾运行效率。后续章节将对具体的模型结构、实验步骤及结果进行详细阐述。

## 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, 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。该方法旨在通过离线阶段积累的先验知识, 缩减在线搜索空间, 从而在**较短时间内找到接近最优**的清洗-聚类组合并兼顾评估效率。以下是本章节所定义的符号与描述:

表 1: 符号与描述

符号	描述
$D_{\text{train}}$	先验数据集 (训练集), 用于离线评估和学习先验知识
$D_{\text{test}}$	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
$\ell$	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

## 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**  $D_{\text{train}}$ ：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  $\Omega$  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**  $D_{\text{test}}$ ：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，若对每个先验数据集  $D^{(i)}$  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

在离线阶段，除了得到各数据集  $D^{(i)}$  的 Top-K 策略外，还要提取其特征向量  $\mathbf{x}(D^{(i)})$ 。通过**多标签学习**的方法，可将“数据特征”与“优选策略集合”关联起来，从而在面对新数据集  $D_{\text{test}}$  时，根据其特征向量  $\mathbf{x}(D_{\text{test}})$  预测出最优或近优的策略子空间  $\Omega'(D_{\text{test}})$ 。

**标签空间与多标签构造** 在离线阶段，为了构建从数据特征到优选方案的映射模型，需引入**标签空间**的概念。首先，将所有先验数据集中出现过的“优选策略”记录下来，表示为：

$$\{\omega^{(1)}, \omega^{(2)}, \dots, \omega^{(m)}\},$$

并为每个优选策略  $\omega^{(j)}$  赋予唯一标签  $\ell_{\omega^{(j)}}$ ，从而形成一个离散的**标签空间**：

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

对于某个先验数据集  $D^{(i)}$ ，其 Top-K 组合  $\mathbf{M}^{(i)}$ （式 (4.1)）中每个策略都可视为一个“正”标签。这些标签的集合定义为：

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

结合数据集的特征向量  $\mathbf{x}(D^{(i)})$ ，可构造出多标签训练样本：

$$(\mathbf{x}(D^{(i)}), \mathbf{L}^{(i)}).$$

最终，所有先验数据集的多标签样本汇总成多标签训练集  $\mathcal{M}$ ：

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

**分类器训练与映射生成** 在完成多标签训练集  $\mathcal{M}$  的构造后，下一步是利用该训练集对多标签分类器  $\mathcal{F}$  进行训练。分类器的目标是学习数据特征  $\mathbf{x}(D)$  与优选策略标签  $\mathcal{L}$  之间的关联关系。

具体而言，分类器  $\mathcal{F}$  的输出为每个标签  $\ell_{\omega^{(j)}}$  的置信度  $q^{(j)} \in [0, 1]$ 。对任意给定的新数据集  $D_{\text{test}}$ ，输入其特征向量  $\mathbf{x}(D_{\text{test}})$  后，分类器将返回以下形式的预测结果：

$$\mathcal{F}(\mathbf{x}(D_{\text{test}})) = \{(\ell_{\omega^{(1)}}, q^{(1)}), (\ell_{\omega^{(2)}}, q^{(2)}), \dots, (\ell_{\omega^{(m)}}, q^{(m)})\}, \quad (4.5)$$

其中  $q^{(j)}$  表示数据集  $D_{\text{test}}$  在优选策略  $\omega^{(j)}$  下的置信度。

为减少评估成本，仅选取置信度最高的  $r$  个标签，构成优选标签集合：

$$\mathbf{L}' = \{\ell_{\omega^{(j)}} \mid q^{(j)} \text{ 属于前 } r \text{ 大值}\}. \quad (4.6)$$

将这些标签映射回对应的清洗-聚类策略，得到优化后的优选子空间：

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

此时，优选子空间  $\Omega'(D_{\text{test}})$  通常远小于原始搜索空间  $\Omega$ ，从而在减少计算成本的同时，保持较高的聚类质量。最终，该映射过程可表示为：

$$G(\mathbf{x}(D)) = \Omega'(D). \quad (4.8)$$

## 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。基于这一映射，本节将进一步探讨其在**自动化聚类优化流程**中的应用，重点分析如何利用该知识在新数据上高效筛选清洗-聚类组合，从而减少搜索空间并提高优化效率。

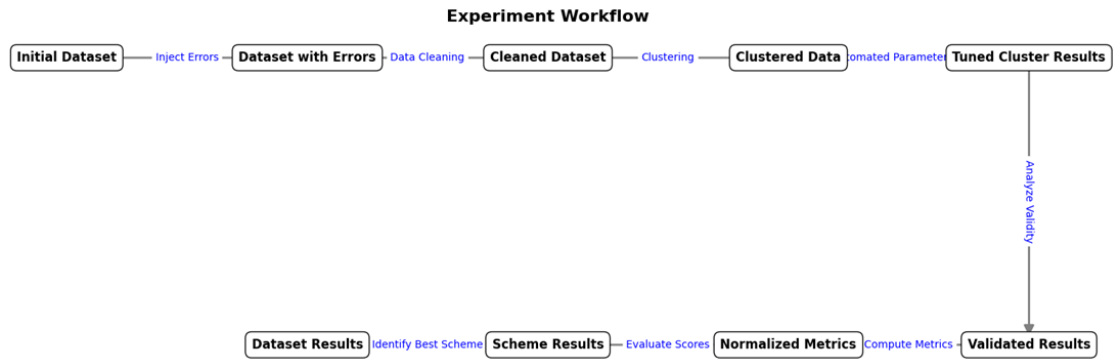


图 1: 自动化聚类优化流程示意图

如图 1 所示，自动化优化流程主要包括**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  $D_{\text{train}}$ ，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  $G(\mathbf{x}(D))$ 。
2. **测试阶段（在线优化）**：面对新的数据集  $D_{\text{test}}$ ，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  $\Omega$  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。

在接下来的小节中，我们将详细介绍两个阶段的具体实现，并给出关键算法的伪代码。

### 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  $D_{\text{train}}$  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。

### 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  $D_{\text{test}}$  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 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  $\Omega$ ;  
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  $\Omega$ ) do  
    计算  $S(D^{(i)}, \omega)$ ;  
    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  $\mathcal{F}$ 
```

---

### 4.3 小结

本节系统介绍了自动化聚类方法的流程，通过离线学习和在线推断，实现了对大规模搜索空间的有效缩减，同时在保证精度的情况下显著提升了效率。

## 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（尤其是第 3.4 小节）展开实验与结果分析。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，并且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 个公开数据集，这些数据集在规模、维度及数据缺陷的分布特征上存在显著差异。基于第 3.1 小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。这些统计信息不仅帮助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更精细的对照依据。表 2 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) 聚类算法及其参数。

**数据清洗策略** 我们在第 2 节介绍了若干常用数据清洗方法。本次实验选取了以下三种最具代表性的：

- **Mode 填补**：该方法用于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**：作为模拟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和复杂的场景。



---

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

---

**Input:** 测试数据集  $D_{\text{test}}$ ;  
多标签分类器  $\mathcal{F}$ ;  
搜索空间  $\Omega$ ;  
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  $\mathbf{L}'_{\text{top}}$ ;  
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**  $\hat{\omega}$

---

表 2: 实验中部分数据集的规模与质量问题概览

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%

- **GroundTruth (GT):** 此方法仅用于对照实验，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 3 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。

表 3: 选定聚类算法的特点比较

算法	聚类方式	算法参数	时间复杂度	参数敏感度
K-Means	质心	$k$ (簇数)	$O(nkT)$	高
DBSCAN	密度	$\varepsilon$ (邻域半径), MinPts (最小点数)	$O(n \log n)$	低
OPTICS	密度	MinPts (最小点数), $\xi$ (相对密度变化阈值)	$O(n \log n)$	低
层次聚类 (HC)	距离	$k$ (簇数), Linkage (链接方式), Metric (距离度量)	$O(n^2)$	高
GMM	高斯混合模型	$k$ (高斯分布数), CovType (协方差类型)	$O(nkT)$	高
AffinityPropagation	消息传递	Damping (阻尼系数), Preference (偏好值)	$O(n^2)$	中

从表 3 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，且参数选择影响显著。

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 实验流程步骤

为保证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优化。

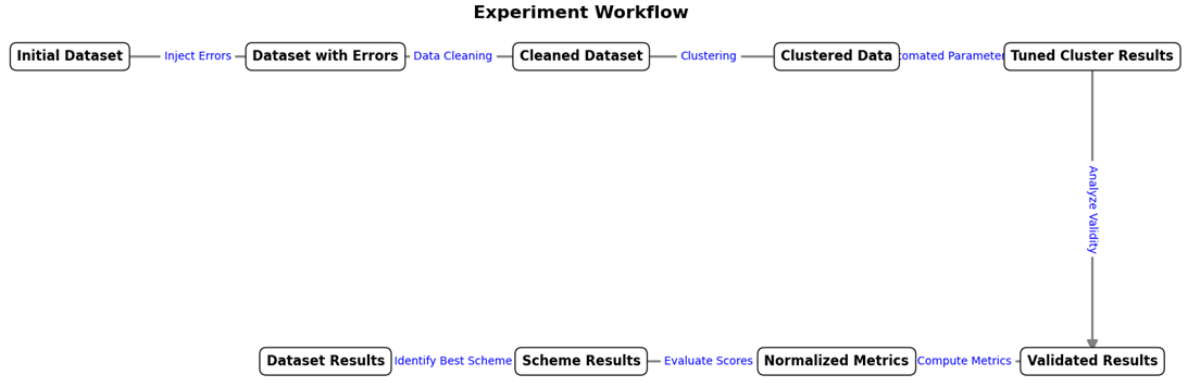


图 2: 实验流程示意图

1. **特征统计与数据预处理**: 计算每个数据集的错误率、缺失率及噪声水平, 生成数据特征向量  $\mathbf{x}(D)$ 。对文本字段进行必要的编码转换, 确保数据格式适用于后续聚类分析。
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, 记录簇数、运行时间及综合得分  $S(D, \omega)$  (见式 (3.7))。
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, 该模型用于高效推荐清洗-聚类方案, 并与全量搜索结果对比, 评估损失率  $\eta(D)$  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。

在接下来的实验部分, 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  $\mathcal{A}(D)$ , 评估其在大规模数据场景下的可行性。

这两个实验从数据清洗与聚类的协同优化到自动化优化的高效性两个层面展开分析, 以验证所提出方法的实际可行性。

### 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, 以回答第 3.4 小节提出的核心问题: “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  $\beta$  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%，以避免对整体结果分析的干扰。

### 5.3.2 实验结果与分析

本节围绕不同数据集的聚类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：

(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**，其得分是所有清洗-聚类组合中距离 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。

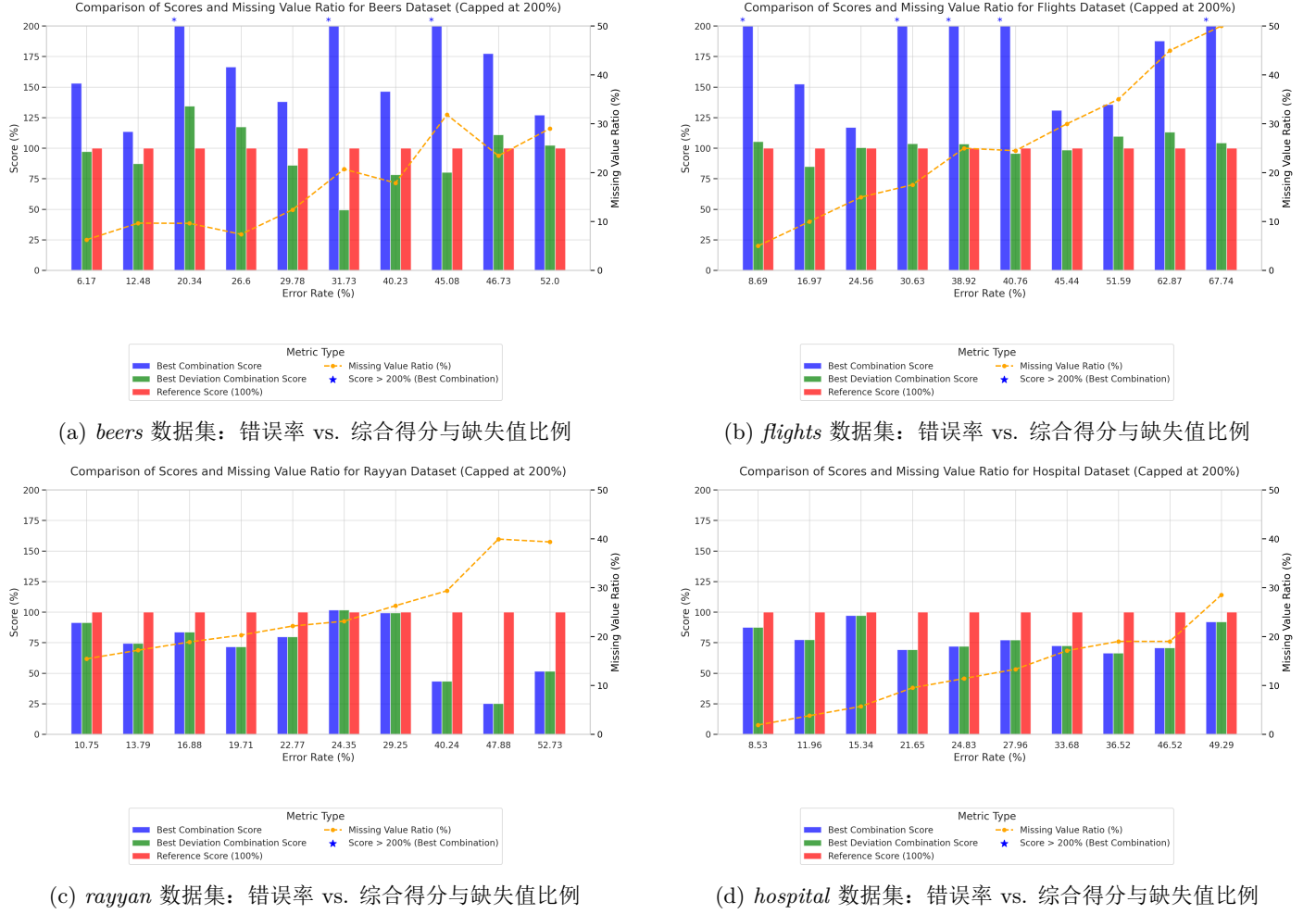


图 3: 不同数据集在不同错误率条件下的得分与缺失值比例变化

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 4a 至 4d 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。

基于表 4a 至表 4d 所示的各数据集 (*beers*、*flights*、*hospital*、*rayyan*) 在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”往往不一致

在 *beers* 和 *flights* 数据集中，若数据规模较大（分别 2410 与 2376 条）且数值型特征占主导，*mode* + HC 或 *mode* + DBSCAN 有时会出现高达 200%~300%、甚至超过 1000% 的“爆分”情形。这说明在特定内部指标（Davies-Bouldin

(a) <i>beers</i> 数据集						(b) <i>flights</i> 数据集					
Name	Error (%)	C <sub>1</sub>	S <sub>1</sub> (%)	C <sub>2</sub>	S <sub>2</sub> (%)	Name	Error (%)	C <sub>1</sub>	S <sub>1</sub> (%)	C <sub>2</sub>	S <sub>2</sub> (%)
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 <sub>1</sub>	S <sub>1</sub> (%)	C <sub>2</sub>	S <sub>2</sub> (%)	Name	Error (%)	C <sub>1</sub>	S <sub>1</sub> (%)	C <sub>2</sub>	S <sub>2</sub> (%)
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 4: 不同错误率下各数据集的最佳组合 (C<sub>1</sub>) 与最接近基准组合 (C<sub>2</sub>) 及得分

与 Silhouette) 上, 简单填补方式 (mode) 若恰好搭配了合适的超参数 (如 DBSCAN 的邻域半径或 HC 的距离度量), 可能导致极端聚类结果。然而, 这些高分往往与基准结构偏离较大, 原因可能在于: DBSCAN 对噪声和缺失值极度敏感, 或 HC 选择了与理想簇数相差较远的分割方式。此时, “最接近基准” 的组合通常是 Raha-Baran + HC 或 mode + KMeans, 它们的绝对分值虽不及 “爆分” 方案, 但其聚类结构更贴近 GroundTruth。

## 2. 不同数据集的 “极端高分” 分布并不均衡

在 *flights* (7 个数值型特征) 数据集中, mode + DBSCAN 出现了 1800%~2500% 的极端情况, 表明对中等规模、低维且噪声较明显的的数据, 基于密度的聚类对填补方式和超参数的敏感度尤为突出。而在 *hospital* (1000 条、20 维) 与 *rayyan* (1000 条、11 维) 这类更高维或含部分语义规则的数据中, 聚类分数通常较为温和, 极少越过 200%。这说明当数据在特征和类型上具有更多语义约束或上下文逻辑时, Raha-Baran 可能在修复后为层次聚类等算法保留更合理的分布结构, 减少了极端划分的发生。

## 3. Raha-Baran + HC 在中低错误率时表现稳健

在 *hospital* 和 *rayyan* 数据集中, 当错误率低于 25% 时, Raha-Baran + HC 常能同时获得 “最高分” 和 “最贴近基准” 这两项佳绩。对于具备一定语义约束、特征更丰富 (20 维或 11 维) 且规模中等 (1000 条) 的数据而言, Raha-Baran 的精细化修复显著降低了噪声干扰, 而 HC 的层次聚类方式在更优质的数据环境下更容易逼近基准结构。即使当错误率升至 40% 以上, Raha-Baran + HC 的得分也只是有一定波动, 但总体仍较为稳定, 显示在高维场景 (如 *hospital*) 中, 采用上下文修复结合分层聚类的策略仍优于其他组合。

综合而言, 在 “最佳组合” 与 “最接近基准” 不匹配的情况下, 需结合具体应用需求来平衡分数绝对值与聚类结构的合理性。此外, 不同数据集的特征维度、噪声成分与语义约束都可能影响聚类的极端表现, 使用精细化清洗 (如 Raha-Baran) 往往有助于在高维或语义复杂的环境中维持更稳定的聚类结果。

**(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。

## 结果分析 (基于方法与错误率变化的整体趋势)

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:

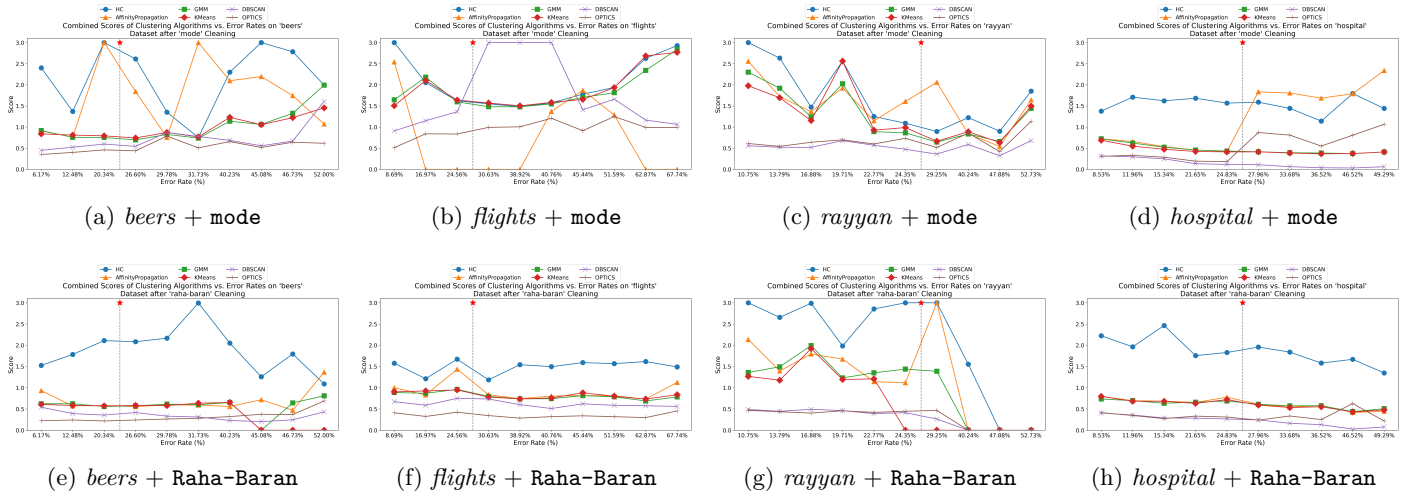


图 4: 不同数据集在不同错误率条件下的聚类算法综合得分变化趋势（第一行: mode 清洗, 第二行: Raha-Baran 清洗）

### 1. 错误率升高更容易诱发“极端爆分”或“收敛失败”

在 *mode* 清洗下, 我们常见到某些算法（如 HC、DBSCAN、AP）在中高错误率时突然冲至 3.00 的最高得分, 或在极少数情形直接收敛到 0.00。前者多出现于对数据特征敏感的算法（HC/DBSCAN 等）, 因其内部指标可能在局部最优下得到极端划分; 后者则可归因于聚类过程中的不收敛或过度聚合（如某一簇占据全部样本）。因此, 错误率越高, 越容易出现算法失真或偏离真实结构的情况。

### 2. 更精细的清洗（raha-baran）虽减少“爆分”, 但在极端情形仍可能降至 0.0

与 *mode* 相比, *raha-baran* 通过语义/规则等方式提升了数据质量, 显著降低了类似 200%~300% 甚至上千的得分飙升现象。这表明, 噪声修复的精细程度能抑制部分算法在错误率攀升时的剧烈震荡。然而, 在个别高错误率场景下, 部分算法（如 KMeans、GMM 或 AP）可能仍直接收敛失败, 导致得分归零, 说明当数据噪声过多, 且超参数不匹配时, 算法本身依旧无法获取稳定簇结构。

### 3. 层次聚类（HC）与密度聚类（DBSCAN）在错误率提升时波动尤其明显

观察所有数据集可发现, HC 常在中高错误率段出现数值极端（3.00 或 0.00）, 反映其对层次拆分策略的依赖: 当错误率影响到簇数或距离度量时, 就会引发大幅波动。DBSCAN 则因其对“邻域半径”和“最小点数”等超参数极度敏感, 导致在某些数据集中（如 *flights*）数值瞬间跃升至顶点或跌至低点。从实际应用看, 若噪声/错误率较高, DBSCAN 与 HC 应格外关注超参数调优, 否则结果极易失真。

### 4. KMeans、GMM、OPTICS 较少出现极端值, 但对数据结构依赖性仍较大

在绝大多数情形下, KMeans 和 GMM 的分数更趋于 0.5~2.0 的区间, 极端爆分现象较为少见。但当错误率超过 40% 左右时, 这两种算法若未能捕捉到正确的簇形态, 也会出现快速下滑或无法收敛的迹象。OPTICS 由于与 DBSCAN 同属基于密度的聚类方法, 表面看波动相对较小, 但其得分也较低, 说明对噪声或错误值的处理效果并不理想。

### 5. “mode”与“raha-baran”的差异在高错误率下尤为突出

当错误率处于低（<10%）或中等水平（10%~25%）时, 两种清洗方法仅在极少数场景产生分数巨大差异; 一旦错误率升至 30% 甚至 40% 以上, *mode* 清洗更倾向于激发少数算法的极端爆分或直接失败, 而 *raha-baran* 则整体波动较小, 偶尔出现 0.0 的收敛异常, 但相比之下大部分算法处于中等分数水平。这显示出精细化清洗在高错误率场景具有更明显的“稳健”优势。

综上所述, 随着错误率的不断攀升, 数据质量对聚类算法的影响呈显著放大趋势。使用更精细的清洗方法（*raha-baran*）可以减少极端结果的发生概率, 但也无法完全避免算法在“超高噪声”时的崩溃风险。若对聚类结构的连贯性和解释性要求较高, 建议优先选择 KMeans、GMM 或在 HC 中通过恰当的链接度量与距离阈值进行调优; 而若希望对簇形态有更灵活的捕捉, 应结合数据特征与超参数, 慎重使用 DBSCAN 和 OPTICS。

## 5.4 自动化聚类模型评估实验

为了验证本文在第 4 节提出的自动化管线模型, 能否在缩减搜索空间的同时维持较高的聚类质量, 本节将基于先验数据学习多标签分类器, 并在测试数据集上与全量搜索做对比, 重点衡量下述指标:

- 损失率  $\eta(D)$ : 子空间搜索与全量搜索的平均得分之比;
- 综合加速比  $\mathcal{A}(D)$ : 在可控损失下, 能带来多少倍的搜索时间缩短。

### 5.4.1 实验设置

在离线阶段, 我们对前面第 5.3 节中的大规模对比结果进行汇总, 将各数据集的特征向量  $\mathbf{x}(D)$  与在其上表现最好的 Top- $K$  个清洗-聚类组合映射为多标签训练样本 (更多细节见第 4 节)。随后, 我们在新的测试数据集上仅对分类器推荐的子空间进行评估, 记录最终得分并与全量搜索比较。

### 5.4.2 结果与分析

表 5 展示了部分数据集在自动化管线下的损失率与加速比。可见, 多标签学习能够显著减少无效组合的尝试, 将评估时间平均缩短 5~10 倍, 而聚类得分仅有 2%~3% 的下降。

表 5: 自动化聚类管线与全量搜索对比示例

数据集	原总耗时 (s)	自动化耗时 (s)	加速比	损失率 (%)
Beers	10.5	2.4	4.4	3.1
Flights	18.2	2.1	8.7	2.8
Hospital	25.0	3.2	7.8	1.9
...其余省略 ...				

从这些结果可以得出, 对于多数存在噪声/缺失的问题数据集, 自动化管线在牺牲少量聚类质量的前提下, 实现了显著的搜索加速, 证实了第 4 节提出的方法在实际应用中的可行性与鲁棒性。

### 5.5 实验总结

通过本章的多阶段实验, 我们对前文第 ?? 小节中提出的两大问题做出了实证回答:

- 对于“清洗策略 + 聚类算法”的大规模对比: 18 种组合在 40 个数据集上的表现差异明显, 具有“高方差”“多峰”特征。无法靠简单规则单独选出“通用最优”, 但可以总结若干适合特定场景的搭配, 如 Raha-Baran + HC 等。
- 对于“自动化聚类模型”的评估: 利用多标签学习策略, 仅需评估极少数候选组合即可获得接近全量搜索的聚类分数, 显著降低了运行时间, 在损失率与加速比之间达成了更好的平衡。这证明了在多场景、多数据质量问题下, 自动化管线具有较好的适用性。

综上所述, 本章实验不仅验证了“数据清洗与聚类协同优化”的价值, 也通过对自动化方法的测试展现了该思路在大规模数据场景下的实际可行性。下一章中我们将针对算法细节和可能的局限性进行更深入的讨论, 并给出进一步的改进方向。