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 1 月 26 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；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。“清洗策略 + 聚类算法 + 超参数”一体的管线协同优化也因此逐渐受到关注。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，这种协同优化的自动化框架具有显著的潜在价值：一方面，聚类在产业与学术界有着广泛的应用场景；另一方面，大部分真实世界的数据都或多或少存在质量问题，一旦实现了对多种数据特征的适配清洗与高效聚类结合，就能够在更多领域落地。本研究正是针对这一尚未被深入探索的方向，提出了面向数据质量的清洗与下游聚类协同优化的自动化模型，既可以为学术研究提供新的思路，也有望在实际应用中发挥切实的作用。

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-聚类组合的有效性与局限性**

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-聚类交互规律的基础上，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，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对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，本节将依次介绍关键概念、系统化的评价方法，以及相应的映射模型，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**

$$\mathbf{x}(D) \;=\; \big(\mathrm{ErrorRate}(D),\, \mathrm{MissingRate}(D),\, \mathrm{NoiseRate}(D),\, m,\, n\big), \tag{3.1}$$

其中 $\mathrm{ErrorRate}(D)$、$\mathrm{MissingRate}(D)$ 与 $\mathrm{NoiseRate}(D)$ 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 $\boldsymbol{\theta}$ 组合成**清洗-聚类策略**

$$\omega \;=\; \big(c,\, h,\, \boldsymbol{\theta}\big), \tag{3.2}$$

所有可行策略的笛卡尔积构成初始搜索空间

$$\Omega \;=\; \mathcal{C} \;\times\; \mathcal{H} \;\times\; \mathcal{P}. \tag{3.3}$$

在实际应用中，Ω 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

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

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 在所属簇内的凝聚力与与最近邻簇的分离度，定义为：

$$\mathrm{Sil}(x) \;=\; \frac{b(x) - a(x)}{\max\big\{a(x),\, b(x)\big\}}, \tag{3.5}$$

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：

$$\mathrm{Sil}(D,\omega) \;=\; \frac{1}{n}\sum_{x\in D}\mathrm{Sil}(x), \tag{3.6}$$

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：

$$S(D,\omega) \;=\; \alpha\cdot\big[-\,DB(D,\omega)\big] \;+\; \beta\cdot\mathrm{Sil}(D,\omega), \tag{3.7}$$

其中 $\alpha, \beta > 0$ 为可调权重，$\big[-DB(D,\omega)\big]$ 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

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

然而，若要在全量空间 Ω 上评估每个策略 ω，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

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

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 Ω 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

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

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。另外，记

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

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 (**??**)）。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 Ω 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

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

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 Ω 的重复尝试。后续章节将结合实证场景和模型设计，详细说明如何构建并验证该映射。

3.4 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 Ω 中，每个策略 ω 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、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 Ω 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

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

同时，引入**损失率** $\eta(D)$ 和**综合加速比** $\mathcal{A}(D)$（参考式 (**??**) 与 (**??**)），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，本节将在前文所述概念基础上，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_{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

4.1 先验数据与多标签映射策略

在实际应用中，往往可以从历史任务中获取大量已处理或部分标注的数据集，这些数据可视为**先验数据**（离线学习）。而当面对新任务时，需要快速完成聚类策略的优选与评估，该新数据集则记为**测试数据**（在线应用）。通过在先验数据上进行充分探索并存储“数据特征—策略表现”之间的关联，便可在测试数据上减少不必要的搜索开销。

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}：由若干历史数据集构成，记为 $\{D^{(1)}, D^{(2)}, \ldots, D^{(N)}\}$。此部分数据可在离线阶段（训练阶段）使用，对搜索空间 Ω 进行大范围或抽样评估，收集足量的策略得分信息。
- **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 $\omega \in \Omega$，便能计算各自的综合得分 $S(D^{(i)}, \omega)$。为高效记录在 $D^{(i)}$ 上表现最好的候选，我们定义一个 **Top-K 方案矩阵**（式 (**??**)），记为 $\mathbf{M}^{(i)}$，其中每一行是一个评分 S_j 较高的策略组合 $\omega_j^{(i)} = (c_j, h_j, \boldsymbol{\theta}_j)$。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。

$$\mathbf{M}^{(i)} = \begin{pmatrix} c_1 & h_1 & \boldsymbol{\theta}_1 & S_1 \\ \vdots & \vdots & \vdots & \vdots \\ c_K & h_K & \boldsymbol{\theta}_K & S_K \end{pmatrix}. \tag{4.1}$$

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，还可提取其特征向量 $\mathbf{x}(D^{(i)})$。通过**多标签学习**方法，便能将“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_{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，需要引入**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：

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

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**：

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

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

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

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：

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

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

$$\mathcal{M} = \big\{\big(\mathbf{x}(D^{(1)}), \mathbf{L}^{(1)}\big), \ldots, \big(\mathbf{x}(D^{(N)}), \mathbf{L}^{(N)}\big)\big\}. \tag{4.4}$$

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_{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：

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

其中 $q^{(j)}$ 表示数据集 D_{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：

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

通过这些标签，再映射回对应的清洗-聚类策略，得到优化后的**优选子空间**：

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

此时，优选子空间 $\Omega'(D_{\text{test}})$ 通常远小于原始搜索空间 Ω，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：

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

4.2 自动化聚类优化流程

完成离线阶段后，即可在部署过程中使用**自动化聚类优化**流程快速定位接近最优的清洗-聚类策略。该流程包括训练阶段和测试阶段两个环节。

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 **??** 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 $D_{\text{train}} = \{D^{(1)}, \ldots, D^{(N)}\}$；
搜索空间 Ω；
Top-K 大小 K。
Output: 多标签分类器 $\mathcal{F}$
$\mathcal{M} \leftarrow \texttt{GenerateTrainingData}(D_{\text{train}}, \Omega, K)$;
$\mathcal{F} \leftarrow \texttt{TrainClassifier}(\mathcal{M})$;
return $\mathcal{F}$

Function `GenerateTrainingData`(D_{train}, Ω, K)**:**
  $\mathcal{M} \leftarrow \emptyset$;
  for $i \leftarrow 1$ **to** $|D_{train}|$ **do**
    foreach $\omega \in \Omega$ *(或采样自 Ω)* **do**
      计算 $S(D^{(i)}, \omega)$;
    选出 Top-K 策略 $\mathbf{M}^{(i)} = \{\omega_1^{(i)}, \ldots, \omega_K^{(i)}\}$ 按得分降序;
    映射为多标签集合 $\mathbf{L}^{(i)} = \{\ell_{\omega_1^{(i)}}, \ldots, \ell_{\omega_K^{(i)}}\}$;
    $\mathcal{M} \leftarrow \mathcal{M} \cup \{(\mathbf{x}(D^{(i)}), \mathbf{L}^{(i)})\}$;
  return $\mathcal{M}$
Function `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 **??** 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**
  $q_\ell \leftarrow$ 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 $\omega \in \Omega'(D_{test})$ **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。