标题示例：AutoML for Clustering

常添
所属单位
email@example.com

2025 年 1 月 18 日

摘要

摘要内容的示例。

1 引言

引言内容的示例。

2 聚类问题定义与优化目标

以下是本章节所定义的符号与描述：

表 1: 符号与描述 1

符号	描述
D	数据集的通用表示，包含特定场景下的数据
$\mathbf{x}(D)$	数据集 D 的特征向量，包含质量与规模信息
$\mathcal{C}$	数据清洗方法的集合
$\mathcal{H}$	聚类算法的集合
$\mathcal{P}$	聚类算法的超参数空间
Ω	聚类策略的初始搜索空间
ω	聚类策略的策略组合
$S(D,\omega)$	聚类策略 ω 在数据集 D 上的综合得分
$T_{\text{original}}(D)$	在初始搜索空间上评估的耗时
$T_{\text{reduced}}(D)$	在优化后搜索空间上评估的耗时
$\eta(D)$	损失率，表示优化后综合得分的平均下降比例
$\mathcal{A}(D)$	综合加速比，表示搜索效率的综合提升程度

在本文中，我们将数据集记为 D，其特征向量为 $\mathbf{x}(D)$。给定数据清洗方法集合 $\mathcal{C}$、聚类算法集合 $\mathcal{H}$ 以及超参数空间 $\mathcal{P}$，定义**聚类策略**为三元组合：

$$\omega = (c, h, \boldsymbol{\theta}), \quad c \in \mathcal{C},\ h \in \mathcal{H},\ \boldsymbol{\theta} \in \mathcal{P}. \tag{2.1}$$

所有可行组合构成**初始搜索空间**：

$$\Omega = \mathcal{C} \times \mathcal{H} \times \mathcal{P}. \tag{2.2}$$

当 Ω 规模非常大时，若**无法遍历**整个空间，可以采取**随机采样**或**分层采样**等方法，从 Ω 中选取若干代表性策略 ω 用于计算并估计综合得分，以平衡搜索精度与时间成本。

2.1 数据集特征

在真实世界的聚类任务中，数据集往往同时面临多种质量问题（错误值、缺失值、噪声）。为便于对不同数据集进行横向对比与后续建模，本研究对每个数据集 D 抽取如下**特征向量**：

$$\mathbf{x}(D) = \big(\text{ErrorRate}(D),\ \text{MissingRate}(D),\ \text{NoiseRate}(D),\ m,\ n\big), \tag{2.3}$$

其中：

- $\text{ErrorRate}(D)$：错误值占总单元的比例；
- $\text{MissingRate}(D)$：缺失值占总单元的比例；
- $\text{NoiseRate}(D)$：噪声或离群点占总单元的比例；
- m：特征维度（属性数量）；
- n：记录条数（样本规模）。

2.2 聚类评价指标

给定数据集 D 与聚类策略 ω，我们选用以下评价指标：

Davies-Bouldin (DB) Score 衡量簇内紧凑度与簇间分离度，值越低越好：

$$DB(D,\omega) = \frac{1}{K}\sum_{i=1}^{K}\max_{j\neq i}\left(\frac{S_i + S_j}{d_{ij}}\right), \tag{2.4}$$

其中 K 为聚类数，S_i 表示第 i 个簇的平均离散度，d_{ij} 表示簇间中心距离。

Silhouette Score (轮廓系数) 衡量每个样本在所属簇的凝聚力与最近簇的分离度，值越高越好：

$$\text{Sil}(x) = \frac{b(x) - a(x)}{\max\{a(x),\, b(x)\}}, \tag{2.5}$$

其中 $a(x)$ 为 x 到同簇其他样本的平均距离，$b(x)$ 为 x 到最近簇的平均距离。

综合得分 本研究将二者线性组合得到：

$$S(D,\omega) = \alpha \cdot \big(-DB(D,\omega)\big) + \beta \cdot \text{Sil}(D,\omega), \tag{2.6}$$

其中 $\alpha, \beta > 0$ 为加权系数。

2.3 优化目标与衡量指标

当对搜索空间 Ω 做全面或抽样评估后，我们希望找到：

$$\omega^*(D) = \arg\max_{\omega\,\in\,\Omega}\ S(D,\omega), \tag{2.7}$$

但在 Ω 很大时，穷尽搜索会带来极高的时间成本。因此核心目标是：**在不显著牺牲聚类质量的前提下，尽可能减少实际运行时间**。为此，我们定义了以下两个指标：

2.3.1 损失率

记 $S(D,\omega)$ 为策略 ω 在 D 上的综合得分；令 $\Omega'(D)$ 表示优选子空间。定义**损失率**：

$$\eta(D) = 1 - \frac{\frac{1}{|\Omega'(D)|}\sum_{\omega\in\Omega'(D)} S(D,\omega)}{\frac{1}{|\Omega|}\sum_{\omega\in\Omega} S(D,\omega)}. \tag{2.8}$$

损失率 $\eta(D)\in[0,1]$，越接近 0 表示优化后的平均聚类质量越接近完整搜索。

2.3.2 综合加速比

评估方案 ω 的时间耗时记为 $T(\omega,D)$。若在原始空间 Ω 上做全量搜索，时间为

$$T_{\text{original}}(D) = \sum_{\omega\,\in\,\Omega} T(\omega,D). \tag{2.9}$$

在优选子空间 $\Omega'(D)$ 上搜索时为

$$T_{\text{reduced}}(D) = \sum_{\omega\,\in\,\Omega'(D)} T(\omega,D). \tag{2.10}$$

综合加速比定义为：

$$\mathcal{A}(D) = \big(1-\eta(D)\big)\frac{T_{\text{original}}(D)}{T_{\text{reduced}}(D)} \tag{2.11}$$

以综合衡量**聚类质量损失**与**评估时间降低**的平衡性。

3 先验数据与映射构建

为提升聚类策略搜索的效率，我们可将数据集划分为先验数据（离线学习）与测试数据（在线应用），并通过多标签学习构建“数据特征 $\rightarrow$ 优选子空间”的映射。以下是本章节所定义的符号与描述：

表 2: 符号与描述 2

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

3.1 先验数据集与测试数据集

- **先验数据集** D_{train}：包含若干历史数据集 $\{D^{(1)},D^{(2)},\dots\}$，可在上面对 Ω 进行大范围或抽样评估，形成“先验知识”。
- **测试数据集** D_{test}：实际部署场景下的新数据集。目标是**利用先验知识**，减少搜索规模并**降低评估时间**。

在先验数据集 $D^{(i)}$ 上，遍历或采样若干 $\omega\in\Omega$，计算综合得分 $S(D^{(i)},\omega)$；选取**评分最高**的 K 个组合构成 **Top-K 方案矩阵**

$$\mathbf{M}^{(i)} = \begin{pmatrix} c_1 & h_1 & \boldsymbol{\theta}_1 & S_1 \\ \vdots & \vdots & \vdots & \vdots \\ c_K & h_K & \boldsymbol{\theta}_K & S_K \end{pmatrix}, \tag{3.1}$$

其中第 j 行的策略可记为 $\omega_j^{(i)} = (c_j, h_j, \boldsymbol{\theta}_j)$，得分为 S_j。行从上到下按 S_j 降序排列。

3.2 基于多标签学习的映射策略

当每个先验数据集 $D^{(i)}$ 可对应多个**优选方案**时，本研究采用**多标签学习**来构建分类器 $\mathcal{F}$，并基于该分类器得到映射函数 G。下文说明标签空间定义、数据构造与预测流程。

3.2.1 标签空间与多标签分配

将所有出现过的优选策略记为

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

为每个优选策略 $\omega^{(j)}$ 分配唯一标签 $\ell_{\omega^{(j)}}$，形成离散**标签空间**：

$$\mathcal{L} = \{\ell_{\omega^{(1)}},\ell_{\omega^{(2)}},\dots,\ell_{\omega^{(m)}}\}. \tag{3.2}$$

若先验数据集 $D^{(i)}$ 的 Top-K 组合为

$$\mathbf{M}^{(i)} = \{\omega_1^{(i)},\omega_2^{(i)},\dots,\omega_K^{(i)}\},$$

则其**多标签**集合为

$$\mathbf{L}^{(i)} = \{\ell_{\omega_1^{(i)}},\ell_{\omega_2^{(i)}},\dots,\ell_{\omega_K^{(i)}}\}. \tag{3.3}$$

可见标签 ℓ_ω 与聚类策略 ω 是**一一对应**的，以便后续分类器的输出可映射回具体策略。

3.2.2 训练数据与多标签分类器

训练数据构造 将每个先验数据集 $D^{(i)}$ 视为一条多标签样本：

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

汇总所有先验数据集，得到训练集

$$\mathcal{M} = \{(\mathbf{x}(D^{(1)}),\mathbf{L}^{(1)}),\dots,(\mathbf{x}(D^{(N)}),\mathbf{L}^{(N)})\}. \tag{3.4}$$

模型训练 记 $\mathcal{F}$ 为多标签分类器，它输出对于每个标签 $\ell_{\omega^{(j)}}$ 的概率 $q^{(j)}\in[0,1]$。可根据具体需求使用 Binary Relevance、ML-kNN、神经网络等多标签算法。

3.2.3 预测与映射函数 G

在测试阶段，给定新数据集 D_{test} 的特征 $\mathbf{x}(D_{\text{test}})$，可得到：

$$\mathcal{F}\big(\mathbf{x}(D_{\text{test}})\big) = \{(\ell_{\omega^{(1)}},q^{(1)}),\dots,(\ell_{\omega^{(m)}},q^{(m)})\}, \tag{3.5}$$

从中选取概率最高的 r 个标签：

$$\mathbf{L}' = \{\ell_{\omega^{(j)}} \mid q^{(j)} \text{ 属于前} r \text{ 大}\}, \tag{3.6}$$

然后再映射回相应的策略，得到优化后的搜索空间：

$$\Omega'(D_{\text{test}}) = \{\omega^{(j)} \mid \ell_{\omega^{(j)}}\in\mathbf{L}'\}. \tag{3.7}$$

基于上述预测过程，可以将“多标签分类器”的输出转化为“映射函数”：

$$G\big(\mathbf{x}(D)\big) = \Omega'(D). \tag{3.8}$$

在测试阶段，只需在 $\Omega'(D)$ 内对相对少量的组合做聚类评估，从而显著降低评估成本并提升实际搜索速度。

4 自动化聚类优化流程

4.1 流程概念图（占位）

基于上述思想，本文提出的自动化聚类优化方法主要分为训练阶段和测试阶段，下面是这两个阶段算法的伪代码实现。

4.2 训练阶段

Algorithm 1: 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 $\mathcal{F}$
$\mathcal{M} \leftarrow \texttt{GenerateTrainingData}(D_{\text{train}}, \Omega, K)$;
$\mathcal{F} \leftarrow \texttt{TrainClassifier}(\mathcal{M})$;
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 $\mathbf{M}^{(i)} = \{\omega_1^{(i)}, \dots, \omega_K^{(i)}\}$ 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 `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 测试阶段

在此基础上，根据损失率 $\eta(D_{\text{test}})$ 与综合加速比 $\mathcal{A}(D_{\text{test}})$ 即可评估优化后的时间效率与聚类质量损失情况。

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 $\hat{\omega}$