Computer Science and Application 计算机科学与应用, 2024, 14(7), 114-120
Published Online July 2024 in Hans. https://www.hanspub.org/journal/csa
https://doi.org/10.12677/csa.2024.147169

Hans汉斯

基于强化学习的聚类算法

李佳辉，徐应涛，张　莹*

浙江师范大学数学科学学院，浙江　金华

收稿日期：2024年6月16日；录用日期：2024年7月19日；发布日期：2024年7月26日

摘　要

创新性地将强化学习技术引入聚类算法中，旨在解决传统聚类方法面临的两大难题：初始聚类中心选择的不确定性以及计算过程中欧氏距离划分样本导致的高时间复杂度。通过引入强化学习的奖惩机制，设计了一种基于“代理”Agent的行为选择策略，有效替代了传统的欧氏距离计算过程，从而消除了初始聚类中心对算法稳定性的潜在影响，并大幅提升了算法的收敛速度。提出了一种全新的基于强化学习的聚类算法，不仅在数学上严谨证明了其收敛性，而且在实际应用中展现了显著优势。通过数值实验验证，该算法在聚类准确率上较传统方法有明显提升，同时在算法性能上也表现出更加优越的特点，这一研究对于提升数据处理效率和准确性具有重要意义。

关键词

聚类算法，强化学习，贪婪策略，奖惩机制，强化信号，RLC算法

Clustering Algorithm Based on Reinforcement Learning

Jiahui Li, Yingtao Xu, Ying Zhang*

School of Mathematical Sciences, Zhejiang Normal University, Jinhua Zhejiang

Received: Jun. 16th, 2024; accepted: Jul. 19th, 2024; published: Jul. 26th, 2024

Abstract

Innovatively introducing reinforcement learning techniques into clustering algorithms, this research aims to address two major challenges faced by traditional clustering methods: the uncertainty in selecting initial cluster centers and the high time complexity caused by the Euclidean distance metric in sample classification. By introducing the reward-punishment mechanism of

*通讯作者。

文章引用: 李佳辉, 徐应涛, 张莹. 基于强化学习的聚类算法[J]. 计算机科学与应用, 2024, 14(7): 114-120.
DOI: 10.12677/csa.2024.147169

reinforcement learning, this paper designs a behavior selection strategy based on an "agent," effectively replacing the traditional Euclidean distance calculation process. This approach eliminates the potential impact of initial cluster centers on the stability of the algorithm and significantly improves the convergence speed. A novel clustering algorithm based on reinforcement learning is proposed, which not only rigorously proves its convergence mathematically but also demonstrates significant advantages in practical applications. Through numerical experiments, it is verified that the algorithm achieves significantly higher clustering accuracy compared to traditional methods, while also exhibiting superior algorithm performance. This research is of great significance for improving the efficiency and accuracy of data processing.

Keywords

Clustering Algorithm, Reinforcement Learning, Greedy Strategy, Reward and Punishment Mechanism, Strengthen the Signal, RLC Algorithm

1. 引言

随着科技的飞速发展，数学在多个领域的重要性日益凸显。尤其在当代社会，利用数学原理解决现实挑战至关重要。互联网产业虽带来便利[1]，但也产生海量冗余数据[2]。因此，数据挖掘技术成为处理大数据的关键工具[3]，通过算法从无序数据中提取有价值信息[4]。借助 AI 和机器学习，数据挖掘能深入处理、分析大数据，帮助发现数据模式、降低风险、制定策略[5]。其中，聚类分析因其高效和线性计算复杂度备受青睐[6]。数学在聚类分析等领域发挥关键作用，当前研究强调如何有效利用数据挖掘的见解。

2. 预备知识及问题分析

聚类分析是数据挖掘的核心，它将无标签数据分类，是无监督学习的重要部分[6]。现有聚类算法虽技术多样但原理相通[7]，旨在最大化类别间差异并最小化内部相似性。其中，基于划分的算法如 K-means，虽受欢迎但受限于初始中心和欧氏距离计算。

强化学习通过智能体与环境交互优化策略，为聚类算法提供新思路[8]。它指导智能体选择聚类中心，减少对初始值的依赖，并引入更复杂的相似度度量提升准确性。提出基于强化学习的聚类算法 RLC，旨在解决传统算法的初始中心选择和计算复杂度问题。RLC 通过奖惩机制智能选择聚类中心，优化样本空间划分，并采用高效相似度度量降低计算复杂度。这一创新方法不仅提升了性能，还为处理复杂数据集提供新方案。RLC 通过构建 Q 表存储知识，避免随机选择初始中心的问题。然而，该算法研究尚有限，其收敛性、准确率和性能需进一步验证，以推动聚类算法的研究与应用。

在深入 RLC 算法之前，首先梳理了相关的预备知识，包括极限的相关性质、Q-Learning 算法以及学习自动机算法，为后文研究奠定理论基础。

引理 1(极限的相关性质)极限具有以下如式(1)所示的性质：

$$\lim_{n\to\infty}\frac{1}{2^n}=0 \tag{1}$$

Q-Learning 算法是一种经典的强化学习技术，用于解决马尔可夫决策过程中的最优决策问题。该算法通过构建 Q 表来估计每个状态——动作对 Q (s_i, a_i)的长期期望收益，智能体“Agent”根据这些 Q 值在未知环境中进行决策。Q-Learning 算法无需事先了解环境模型，而是通过不断试错和从经验中学习来优化行为策略。其独特的离策略特性使得智能体能够灵活选择动作，既追求当前的最大收益，又保留探索新策略的可能性。因此，Q-Learning 算法被广泛应用，如机器人导航、游戏策略优化以及自动驾驶等领域。Q-Learning 算法通过不断学习和迭代，最终构建出一个二维的 Q 表(Q-table)，其中横轴代表不同的状态(state, s)，纵轴代表可执行的动作(action, a)。这个 Q 表的核心在于存储了智能体在不同状态下采取不同动作所能获得的期望奖励值，也被称为 Q 值。通过这些信息，智能体可以高效地在未知环境中进行决策，从而最大化长期累积的奖励。Q-Learning 是一种针对单智能体的强化学习算法，其核心在于通过不断试错学习来优化决策过程。表 1 所描述了在一个特定环境中，智能体在不同状态下选择不同动作所获得的奖励情况，为智能体后续的决策提供了关键依据。其中 s_n 代表 Agent 所处的状态，a_k 代表 Agent 可选择的行为，q (s_n, a_k)代表 Agent 在状态 s_n 下选择行为 a_k 后所得到的反馈奖励期望值。

Table 1. Q represents the meaning table
表 1. Q 表示意表

S	a_1	a_2	...	a_k
s_1	q (s_1, a_1)	q (s_1, a_2)	...	q (s_1, a_k)
s_2	q (s_2, a_1)	q (s_2, a_2)	...	q (s_2, a_k)
...	...	...	...	...
s_n	q (s_n, a_1)	q (s_n, a_2)	...	q (s_n, a_k)

学习自动机算法，亦称为多智能体算法[9]，是强化学习的重要分支，在概率空间中通过与环境交互，根据强化信号调整动作选择概率，以逼近最优决策。算法分为有限(FALA)和无限(CALA)动作学习自动机，分别适用于组合和连续数值优化问题。FALA 以其试错学习方式，在处理非线性及不确定性优化问题时表现出色，为复杂环境下的决策优化提供高效解决方案。重点介绍 FALA 算法，其核心特性由四元组<α, β, R, U>描述，其中 α 为行为动作，β 为反馈信号，R 为累积概率集合，满足概率和为 1 的约束条件，如下公式(2)。

$$\sum_{i=1}^{K} r_i = 1 \tag{2}$$

鉴于离散行为学习自动机(FALA)与聚类算法在动作选择和相似度计算上的契合度，提出利用 FALA 优化聚类算法，以提升聚类效果和效率。

3. RLC 算法

当前，强化学习与聚类算法的结合是优化聚类性能的研究热点。已有研究如王玉荣等[10]利用学习自动机算法优化模糊 FCM 聚类的参数，而 Yuanfeng Yang 等[11]则通过引入强化学习的奖惩机制优化隶属度，提出新的对抗学习聚类算法。提出一种基于强化学习的聚类算法(RLC)，结合离散行为学习自动机(FALA)的多智能体协同优化思想与 K-means 算法，以提升聚类效果和算法性能。RLC 算法通过引入强化学习思想，使智能体能自适应地调整聚类行为，优化聚类过程。实验表明，RLC 算法在模型性能和聚类结果上均优于主流聚类算法，为聚类分析提供了新工具。该算法将聚类任务形式化为马尔科夫决策过程，利用累积奖励指导聚类，并通过引入 ε-贪婪策略平衡探索和利用，通过监测平均类内距离变化发送强化

信号，优化聚类效果。首先给出 RLC 算法的流程图，如下表 2 所示。

Table 2. RLC algorithm process steps
表 2. RLC 算法流程步骤

算法：RLC 算法
输入：数据集 Data_ori、聚类数 K、最大迭代次数数 Max_iter、平均类内距离变化阈值 Stop；输出：样本所在类簇的标签 Label。 步骤 1：前两次迭代，Agent 通过 K-means++算法选择行为输出，同时计算各类别的平均类内距离； 步骤 2：根据前两次迭代的结果，比较平均类内距离的变化，并向 Agent 发送强化信号，更新 Q 表； 步骤 3：从第三次迭代开始 Agent 根据 ε-贪婪策略选择行为输出，计算类别的平均类内距离； 步骤 4：比较平均类内距离的变化，向 Agent 发送强化信号，更新 Q 表； 步骤 5：算法依次迭代下去，直至平均类内距离变化小于阈值 Stop 或迭代数达到 Max_iter，停止运行； 步骤 6：输出所有样本的类标签。

RLC 算法在初始化阶段需设定初值和参数。在前两次迭代中，由于 Q 表尚未更新，所有 Agent 均采用 K-means++策略选择行为，并计算各类别平均类内距离存入向量。自第三次迭代起，Agent 依据 ε-贪婪策略选择动作并输出，随后重新计算并保存平均类内距离。算法比较当前与上一次迭代的平均类内距离，据此发出强化信号。Agent 接收信号后更新 Q 表，此过程反复进行，直至平均类内距离变化低于阈值 Stop 或达到最大迭代次数 Max_iter。各 Agent 选择累积奖励最大的行为，将选择相同行为的 Agent 对应样本归入同一类别并赋予类标签，输出结果。

3.1. 贪婪系数的选取

RLC 算法通过迭代收敛，使 Q 表中每个行为的累积奖励值稳定，即每行仅有一个行为的累积奖励值为 1。算法执行中，引入 ε-贪婪策略，Agent 以 ε 的概率选择最大累积奖励行为实现“利用”，以 $1-\varepsilon$ 的概率随机选择实现“探索”。根据环境反馈调整累积奖励值，优化聚类效果。实验利用 Matlab 对 iris 数据集进行聚类，比较不同贪婪系数(0.5、0.6、0.7、0.8、0.9)下的准确率，发现贪婪系数设为 0.9 时，算法准确率最高。因此，设定 ε-贪婪策略的 ε 为 0.9，显著提升收敛性能，实现高效聚类分析。

3.2. 平均类内距离的构建

在构建强化信号时，该信号是 Agent 在选择行为后从环境获得的反馈，分为奖励和惩罚信号。传统聚类算法依赖聚类中心变化判断迭代停止，但聚类中心不能全面反映类别内部样本的紧密程度和分布。因此，RLC 算法采用平均类内距离作为强化信号的关键指标，以反映类别内样本的紧密程度。平均类内距离定义为同类别样本到聚类中心距离的均值，量化类别的紧凑程度，与样本紧凑程度呈反比。通过计算平均类内距离，可精确度量类别的紧凑性，公式如下(3)所示。

$$Adis\left(c_j\right)=\frac{1}{\left|c_j\right|}\sum_{i=1}^{\left|c_j\right|}\sqrt{\sum_{m=1}^{M}\left(x_{im}-c_{jm}\right)^2} \tag{3}$$

$\mathrm{Adis}\left(c_j\right)$ 代表第 j 个类别的平均类内距离，其中 $1\le \mathrm{j}\le \mathrm{K}$，$\left|c_j\right|$ 代表第 j 个类别中样本的总数，M 代表输入的样本数据的维度，x_{im} 代表样本 i 的第 m 个属性值，c_{jm} 代表聚类中心 j 的第 m 个属性。

平均类内距离作为量化类别紧密程度的指标，既融合了聚类中心信息，又显著降低了计算复杂度，尤其在算法迭代中表现突出。因此，RLC 算法选用平均类内距离作为强化信号的关键指标，以准确反映聚类质量。

3.3. Q 表的建立

由表 2 可知，在 RLC 算法中，N 个 Agent 与数据集中的 N 个样本形成一一映射关系。同时输入的聚类数 K 与离散行为集中的行为一一对应。设 r_{ij} 为第 i 个 Agent 执行第 j 个行为的累积奖励值，其中各行为的累积奖励值之和满足归一化条件。在一次迭代过程中，Agent 会根据环境反馈更新其所有行为的累积奖励值。最终，算法将使得某一行为的累积奖励值收敛至 1，而其余行为的累积奖励值收敛至 0。

在 RLC 算法的运行过程中，各 Agent 在贪婪策略的指导下，从 Q 表的离散动作集中选择行为。初始化时，所有 Agent 选择各行为的 Q 值均设为 1/K，其中 K 为离散动作总数。首次迭代时，Agent 通过 K-means++算法选择行为，并将选择相同行为的 Agent 划分至同一类别，赋予相应的类标签。此时，由于算法尚未计算平均类内距离，故环境无法给出强化信号。自第二次迭代起，Agent 再次通过 K-means++算法选择行为并完成类别划分后，计算各类别的平均类内距离 $\mathrm{Adist}\left(c_j\right)$，$1 \le j \le \mathrm{K}$，其中 t 为迭代次数。此时，算法得以根据平均类内距离的变化给出强化信号，Agent 据此更新 Q 表。在算法运行过程中，环境提供的反馈信号中，“1”代表奖励信号，而“0”代表惩罚信号。通过这些信号，Agent 不断优化其选择行为，直至算法收敛。当 RLC 算法推进至第三次迭代时，Q 表已基于前两次迭代的反馈信号进行过一次更新，其内容已不再是初始的均匀分布 1/K。因此，Agent 现在能够依据 ε-贪婪策略在“利用”已知信息和“探索”新可能行为之间取得平衡，从而更有效地选择行为。总体而言，在 RLC 算法的运行过程中，各个 Agent 的“探索”与“利用”行为相互交织、互为制约，并持续更新 Q 表以趋于收敛。

若 $\mathrm{Agent_i}$ 接收到的强化信号为奖励信号，则增大 Q 表中该 Agent 行为 k 的累积奖励值，同时减小其余行为的累积奖励值，若 $\mathrm{Agent_i}$ 接收到的强化信号为惩罚信号，则减小 Q 表中该 Agent 行为 k 的累积奖励值，增大其余行为的累积奖励值，Q 表的累积奖励，的更新公式如下式(4)和式(5)所示。

$$r_k\left(iter\right)=\begin{cases} r_k\left(iter-1\right)+0.5\left(1-r_k\left(iter-1\right)\right), & k=i \\ 0.5r_k\left(iter-1\right), & k\neq i \end{cases} \tag{4}$$

$$r_k\left(iter\right)=\begin{cases} 0.5r_k\left(iter-1\right), & k=i \\ r_k\left(iter-1\right)+\dfrac{1}{K-1}\left(0.5\left(r_k\left(iter-1\right)\right)\right), & k\neq i \end{cases} \tag{5}$$

式(4)给出 $\mathrm{Agent_i}$ 在接收到环境所反馈的奖励信号后，Q 表中该 Agent 的各行为累积奖励值的更新公式，式(5)给出 $\mathrm{Agent_i}$ 在接收到环境所反馈的惩罚信号后，Q 表中该 Agent 的各行为累积奖励值的更新公式。$r_k\left(iter\right)$ 和 $r_k\left(iter-1\right)$ 分别代表第 $iter$ 次和第 $iter-1$ 次迭代结束后，Q 表中 $\mathrm{Agent_i}$ 行为 k 的累积奖励值，其中 $1 \le i$，$\mathrm{k} \le \mathrm{K}$。

3.4. 收敛性证明

定理 1：如上表 2 所示的 RLC 算法中 Q 表经过多次迭代之后，最终每个 Agent 的行为将会形成以下的形式：

1) 其中一个行为的累积奖励值收敛至 1；

2) 其余行为的累积奖励值则收敛至 0。

证明由式(2)所示的 $\sum_{i=1}^{K} r_i = 1$，$1 \le i \le \mathrm{N}$ 的约束条件，其中 K 代表有限行为动作集合中 Agent 选择的动作数量，可知若 2)成立，则 1)一定也成立。则证明 2)。以 $\mathrm{Agent_1}$ 为例，其余 $\mathrm{Agent_i}$ 同理得证。

Q 表中累积奖励的更新公式(4)与累积惩罚更新公式(5)可知当奖励信号 $k \neq i$ 与惩罚信号 $k = i$ 时，奖励值为上一次迭代所产生的奖励值的 1/2，则由引理 1，

$$\lim_{N\to\infty}\frac{c}{2^N}=c\lim_{N\to\infty}\frac{1}{2^N}=0$$

其中 c 为任一常数，则 2)得证。

证明成立。

4. 试验设计与分析

本节对比了 K-means (KM)、K-means++ (KM++)、FCM 以及提出的 RLC 算法的性能，使用准确率作为主要评价指标。实验基于 iris、KEEL、enterprise 等公共数据库的 10 个带标签数据集，通过量化各算法在准确率上的表现，评估了它们在聚类效果上的优劣。实验数据集信息及 Q 表初始化详见电子附录，为后续性能分析提供支持。

Table 3. Accuracy results of cluster algorithm experiment
表 3. 聚类算法实验准确率结果表

数据集	编号	K-means 准确率	K-means++准确率	FCM 准确率	RLC 准确率
0enterprise_nk	D1	0.8663	0.9151	0.9215	0.9321
1iris	D2	0.9017	0.9692	0.9212	0.9513
2zsweather1	D3	0.8726	0.9026	0.9326	0.9152
3jhweather3	D4	0.9052	0.8827	0.8923	0.9652
4sensor0	D5	0.7923	0.8724	0.8826	0.8923
5vegetable_vir	D6	0.7723	0.7326	0.7523	0.7982
6provision012	D7	0.7513	0.7966	0.8152	0.8235
7degree	D8	0.6623	0.6525	0.6895	0.6826
8bid_set	D9	0.6922	0.7523	0.6925	0.8236
9med_refer	D10	0.6529	0.6328	0.6513	0.6613

Table 4. Experimental dataset and Q-table initial values table
表 4. 实验数据集及 Q 表初始化值表

数据集	编号	样本数量	维度	类别数(行为数)	Q 表初值
0enterprise_nk	D1	53	9	3	1/3
1iris	D2	150	4	3	1/3
2zsweather1	D3	180	6	19	1/19
3jhweather3	D4	198	6	18	1/18
4sensor0	D5	223	4	2	1/2
5vegetable_vir	D6	233	2	4	1/4
6provision012	D7	268	2	5	1/5
7degree	D8	295	5	3	1/3
8bid_set	D9	369	3	5	1/5
9med_refer	D10	658	3448	11	1/11

经过全面测试 10 个数据集，本实验统计了各算法运行 100 次的平均准确率，并将准确率扩大三倍以百分比形式展示。表 3 详细对比了 RLC 算法与 K-means、K-means++及 FCM 在准确率方面的数据，直观

展现了各算法在聚类性能上的优劣，表 4 为实验数据集及 Q 表初始化值表。实验结果有力支持了对各算法性能的深入分析，显示 RLC 算法在 7 个数据集中准确率更高，验证了其可行性。

5. 总结

在起始部分，首先探讨聚类算法面临核心挑战，旨在寻找更为精准和高效的解决方案。第二节中，聚焦于强化学习的基本原理，为基于强化学习的聚类算法构建奠定基石。还介绍 Q-Learning 算法和学习自动机算法，特别是以离散行为集为核心的 FALA 算法，这些理论知识的积累为后续算法的提出奠定了坚实的基础。

在第三节中，详细阐述所提出的基于强化学习的聚类算法(RLC)，并对其收敛性进行了严格的数学证明。构建二维 Q 表，通过其横纵坐标对应离散行为集和样本空间中的代理 Agent，实现了策略的选择与更新。利用 ε-贪婪策略，Agent 在 Q 表中平衡对已知知识的利用和对环境的探索。采用计算复杂度较低的平均类内距离作为强化信号，该信号综合考虑了聚类中心的关键因素。环境根据此信号向每个 Agent 提供奖励或惩罚，进而更新 Q 表。通过 iris 鸢尾花数据集的初步实验，确定了贪婪系数 ε 为 0.9 时算法表现最佳。最后，经过严格的收敛性证明，验证 RLC 算法的可行性和有效性。

在第四节中，设计了实验方案，将提出的 RLC 算法与 K-means 算法、K-means++算法和 FCM 算法进行对比分析。选用十个公开的、带有标签的数据集，以确保实验结果的全面性和公正性。在实验中，每种算法都运行了 100 次，每次运行都将聚类结果与真实标签进行对比，计算出准确率。最终，取这 100 次运行的平均准确率作为各聚类算法的性能指标。通过对比分析，得出结论：在大量的实验运行中，RLC 算法展现出了更高的准确率，从而充分证明了其在聚类任务中的卓越性能。

参考文献

[1] Hashem, I.A.T., Yaqoob, I., Anuar, N.B., Mokhtar, S., Gani, A. and Ullah Khan, S. (2015) The Rise of “Big Data” on Cloud Computing: Review and Open Research Issues. *Information Systems*, **47**, 98-115. https://doi.org/10.1016/j.is.2014.07.006

[2] Shah, G., Shah, A. and Shah, M. (2019) Panacea of Challenges in Real-World Application of Big Data Analytics in Healthcare Sector. *Journal of Data, Information and Management*, **1**, 107-116. https://doi.org/10.1007/s42488-019-00010-1

[3] 尹刚, 王涛, 刘冰珣, 等. 面向开源生态的软件数据挖掘技术研究综述[J]. 软件学报, 2018, 29(8): 2258-2271.

[4] 李战怀, 于戈, 杨晓春. 人工智能赋能的数据管理、分析与系统专刊前言[J]. 软件学报, 2020, 31(3): 597-599.

[5] Cireşan, D., Meier, U., Masci, J. and Schmidhuber, J. (2012) Multi-Column Deep Neural Network for Traffic Sign Classification. *Neural Networks*, **32**, 333-338. https://doi.org/10.1016/j.neunet.2012.02.023

[6] Xu, D. and Tian, Y. (2015) A Comprehensive Survey of Clustering Algorithms. *Annals of Data Science*, **2**, 165-193. https://doi.org/10.1007/s40745-015-0040-1

[7] Cao, H., Jia, L., Si, G. and Zhang, Y. (2013) A Clustering-Analysis-Based Membership Functions Formation Method for Fuzzy Controller of Ball Mill Pulverizing System. *Journal of Process Control*, **23**, 34-43. https://doi.org/10.1016/j.jprocont.2012.10.011

[8] 朱光宇, 张德颂. 基于强化学习的遗传算法求解一种新的钻削路径优化问题[J]. 控制与决策, 2024, 39(2): 697-704.

[9] 隋丽蓉, 高曙, 何伟. 基于多智能体深度强化学习的船舶协同避碰策略[J]. 控制与决策, 2023, 38(5): 1395-1402.

[10] 王玉荣, 万秋兰, 陈昊. 基于模糊聚类和学习自动机的多目标无功优化[J]. 电网技术, 2012, 36(7): 224-230.

[11] Yang, Y., Cui, Z., Jian, W., *et al*. (2012) Fuzzy C-Means Clustering and Opposition-Based Reinforcement Learning for Traffic Congestion Identification. *Journal of Information & Computational Science*, **9**, 2441-2450.