Computer Science and Application 计算机科学与应用, 2025, 15(10), 163-175
Published Online October 2025 in Hans. https://www.hanspub.org/journal/csa
https://doi.org/10.12677/csa.2025.1510258

Hans汉斯

一种面向增量数据的重叠聚类算法研究

李　峰[1]，韩祝华[2*]，许美玲[3]

[1]河北金融学院河北省金融科技应用重点实验室，河北　保定
[2]河北金融学院统计与数据科学学院，河北　保定
[3]河北金融学院信息与人工智能学院，河北　保定

收稿日期：2025年9月12日；录用日期：2025年10月15日；发布日期：2025年10月27日

摘　要

数据的在线聚类问题分为完全在线聚类和半在线聚类两种。而增量数据的聚类属于半在线聚类，是在已有数据聚类的基础上进行聚类。另外，由于大多数的聚类算法只会把一个数据点分配给唯一聚类，但很多现实环境的数据集都包含交叉重复信息，而重叠聚类允许一个数据点属于多个聚类。传统的重叠聚类方法overlapping k-means (OKM)作为对k-means的扩展，存在聚类效率低和对聚类中心初始化敏感的问题。为解决以上问题，本文提出一种面向增量数据的重叠k-means混合模型，其主要思想是利用KHM的输出对OKM方法的聚类中心进行初始化。该模型首先应用k调和均值算法与重叠k均值算法来对已有数据进行重叠聚类；然后迭代使用OKM算法对增量数据进行聚类，即k-harmonic means & iterative overlapping k-means混合算法(KHM-IOKM)，算法时间复杂度为$O(kn)$。实验结果显示：在仿真数据集和UCI真实数据集上与传统的OKM和KHM算法进行比较，该方法能够有效降低聚类中心初始化敏感的问题，并且相比于传统的聚类方法能获得更好的聚类效果。

关键词

重叠聚类，调和均值，隶属函数，增量数据，K-Means

Study on an Overlapping Clustering Algorithm for Incremental Data

Feng Li[1], Zhuhua Han[2*], Meiling Xu[3]

[1]Financial Technology Application Key Laboratory of Hebei Provincial, Hebei Finance University, Baoding Hebei
[2]School of Statistics and Data Science, Hebei Finance University, Baoding Hebei
[3]School of Information and Artificial Intelligence, Hebei Finance University, Baoding Hebei

Received: September 12, 2025; accepted: October 15, 2025; published: October 27, 2025

[*]通讯作者。

文章引用: 李峰, 韩祝华, 许美玲. 一种面向增量数据的重叠聚类算法研究[J]. 计算机科学与应用, 2025, 15(10): 163-175. DOI: 10.12677/csa.2025.1510258

Abstract

The online clustering problem of data is divided into two types: full online clustering and semi-online clustering. The clustering of incremental data belongs to semi-online clustering, which is based on existing data clustering. In addition, most clustering algorithms produce exclusive clusters, meaning that each data point can just belong to one cluster. In fact, a great many real-world datasets have inherently overlapping information, while the overlapping clustering methods allow one data point to belong to more than one cluster. The traditional overlapping clustering method is overlapping k-means (OKM). As an extension of the k-means algorithm, it has the problems of low clustering efficiency and sensitivity to the selection of the initial clustering center. In order to solve the above problems, in this study, we propose an overlapping k-means hybrid model for incremental data, whose main idea is to use the output of KHM to initialize the clustering center of the OKM method. The model first uses k-harmonic mean algorithm and overlapping k-means algorithm to overlap clustering existing data; then iteratively uses OKM algorithm to cluster incremental data, that is k-harmonic means & iterative overlapping k-means hybrid algorithm (KHM-IOKM), the time complexity is $O(kn)$. By comparing with the traditional OKM and KHM algorithms on synthetic dataset and UCI real dataset, the experimental results demonstrate that the proposed method can effectively reduce the sensitivity of clustering center initialization and obtain better clustering than traditional clustering methods.

Keywords

Overlapping Clustering, Harmonic Mean, Membership Function, Incremental Data, K-Means

1. 引言

在非监督学习领域，最基本的和被广泛研究的优化模型之一就是 k-means 聚类算法。在这个问题里，定义了欧几里得空间中一个包含 n 个点(或向量)的集合 V 。目的是把这些数据点(或向量)划分到 k 个聚类簇 $(s_1,\cdots,s_k)$ 中，并使下列目标函数最小：

$$Q=\sum_{j=1}^{k}\sum_{v\in S_j}\left\|v-c_j\right\|_2^2 \tag{1}$$

其中， c_j 是聚类 s_j 的中心点，记作：

$$c_j=\frac{\sum_{v\in S_j}v}{\left|S_j\right|}.$$

一个在线的 k-means 算法必须在算法运行过程中，把接收到的数据分配到相应的聚类中。在这样的在线环境下，一个先前未知数量的数据点一个接一个地按照任意顺序到达。当在线算法接收到某个数据点时，会根据聚类标准把这个数据点分配到已经存在的某个聚类中并及时更新相关聚类中心。在线机器学习的情况下，数据的标签必须被在线分配。在线 k-means 聚类算法的重要性在整个机器学习领域已经得到了广泛的认可[1]-[3]。例如对于信息检索问题，Charikar 等[4]研究了增量 k-centers 的问题。他们认

为，在现实世界的实践过程中，聚类问题往往都需要在线处理。例如，在给用户推荐新闻的时候，就需要避免推荐给用户已经阅读的同类型新闻。但是对于在线购物来讲，推荐系统就需要把用户所关注的同类型商品推荐给用户。当然，对于推荐系统而言，它需要对接收到的数据(一条新的新闻或者一件新上架的商品)及时进行聚类分析。

目前，大多数的聚类算法都属于单一聚类算法，即每一个样本数据只能属于唯一的一个聚类。但是现实世界的很多数据集都有内在的重复信息，而重叠聚类方法则允许一个样本数据属于多个聚类(如图 1 所示)。在众多的重叠聚类算法当中，最常见的方法就是 overlapping k-means (OKM)，这种方法是对传统 k-means 算法的扩展，会产生重叠聚类[5]。一些最近对 OKM 重叠聚类算法进行扩展的方法包括了由 Cleuziou 提出的 overlapping k-medoid (OKMED)和 weighted overlapping k-means (WOKM)，由 N’Cir 等人提出 kernel overlapping k-means (KOKM)和 parametrized overlapping k-means (POKM)。OKMED 方法围绕聚类的中心进行数据的聚集，是通过对 k-medoid 方法的扩展来确定重叠的聚类。WOKM 方法是对加权 k-means 方法的概括，并且通过分配权重的形式来考虑每个数据点在聚类中的重要性。核重叠 k-means 方法(KOKM) [6]采取一种不同的方法，其目的是使用一个核函数来确定数据的非线性划分。KOKM 方法首先在传统的 k-means 算法中通过欧几里得距离的核化来实现，然后就是在高维空间执行所有的聚类步骤[7]。最后，参数化 OKM 方法即通过使用一个参数 α 来调整重叠量从而增加 OKM 方法的灵活性。对于更多的关于重叠算法的细节以及 OKM 算法的延伸，可以进一步参考文献[8] [9]。实际上所讨论的众多方法的本质都是对 k-means 方法的扩展延伸，不仅聚类效率低，其最大的局限性就是在初始化聚类中心的过程中非常敏感。因此，如何解决聚类对初始化中心的敏感问题就显得尤为必要。

为了解决这一问题，本文提出了一种面向增量数据的重叠 k-means 混合模型，其主要思想是利用 KHM 的输出来初始化 OKM 方法的聚类中心。该模型首先应用 k 调和均值算法与重叠 k 均值算法来对已有数据进行重叠聚类；然后再迭代使用 OKM 算法对增量数据进行聚类，即 k-harmonic means & iterative overlapping k-means 混合算法(KHM-IOKM)。通过在仿真数据集和 UCI 真实数据集上与传统的 OKM 和 KHM 算法进行比较，实验结果表明：该方法能够有效降低聚类中心初始化敏感的问题，并且相比于传统的单一聚类方法能获得更好的聚类效果。

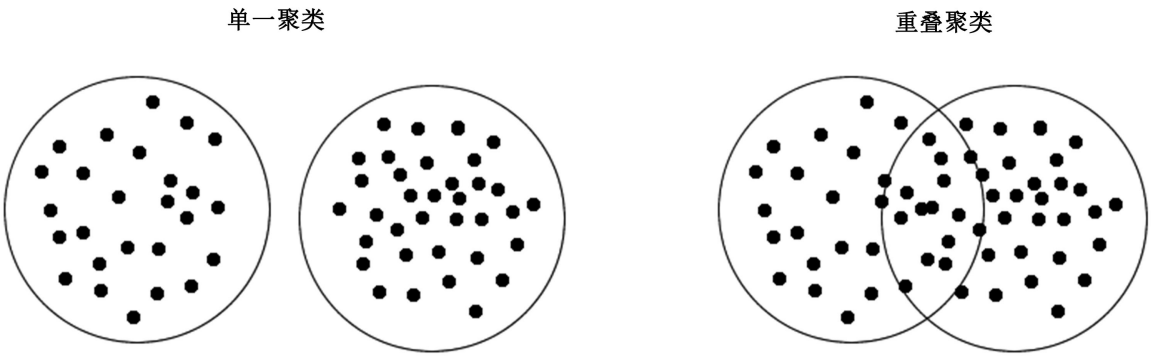


Figure 1. Schematic diagram of two different types of clustering
图 1. 两种不同聚类的示意图

2. 相关工作

本节首先给出 KHM-IOKM 混合重叠聚类算法执行的总体框架图，如图 2 所示。表 1 展示了文中所用的数学符号。2.1 和 2.2 分别详细介绍了传统的 KHM 算法和本文提出的 OKM 算法。

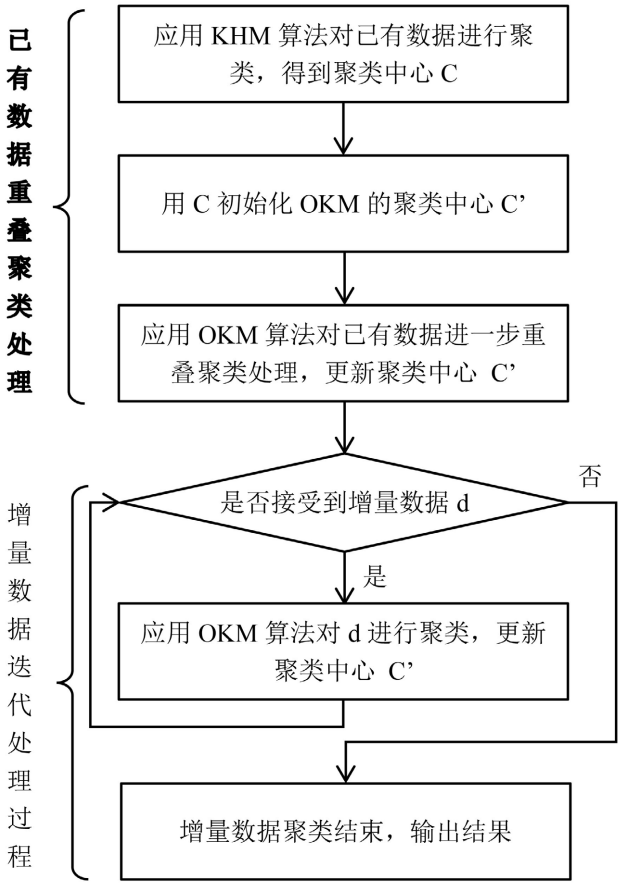


Figure 2. Overall framework of the KHM-IOKM
图 2. KHM-IOKM 总体框架

Table 1. Mathematical notations
表 1. 符号表示说明

符号	描述
V	训练数据集
$\boldsymbol{D}$	增量数据集
v_i	当前接收到的数据(多维向量)
S_i	第 i 个聚类的集合
c_j	第 j 个聚类的中心
δ_i	包含数据 v_i 的聚类数量
Q'	OKM 算法的目标函数
Q''	KHM 算法的目标函数
$\rho_{i,j}$	数据 v_i 的隶属度
$\phi(v_i)$	v_i 所属聚类的平均中心
$w(v_i)$	数据 v_i 的权重
$C(v_i)$	包含了数据 v_i 的所有聚类中心集合
$m(c_j \mid v_i)$	数据 v_i 到聚类中心 c_j 的隶属函数

2.1. 传统的 OKM 算法

k-means 聚类算法的目的，就是使引言中所提到的目标函数最小，如公式(2)所示：

$$Q=\sum_{j=1}^{k}\sum_{v\in S_j}\left\|v-c_j\right\|_2^2 \tag{2}$$

v 是一个观察到的多维向量，其中 $S=\{S_1,\cdots,S_K\}$ 表示包含了 k 个聚类簇的集合，并且各个聚类簇满足 $S_i\cap S_j=\varnothing$，$i\neq j$，$ij\in[1,k]$。其中 $C=\{c_1,\cdots,c_k\}$ 是聚类的中心点集合。

Overlapping k-means (OKM)方法是在使用 k-means 算法时，允许出现重叠的聚类，也就是说把上述提到的 $S_i\cap S_j=\varnothing$ 条件去掉，即 OKM 最优解的目标函数是：

$$Q'=\sum_{i=1}^{n}\left\|v_i-\phi(v_i)\right\|_2^2 \tag{3}$$

其中，$\phi(v_i)$ 是向量 v_i 所属聚类的平均中心[8]，记作：

$$\phi(v_i)=\frac{\sum_{c_j\in c(v_i)}c_j}{\left|C(v_i)\right|} \tag{4}$$

其中，$c_j\in C(v_i)$ 是指包含了数据 v_i 的所有聚类中心 c_j 的集合，根据文献[8]可计算聚类中心 c_j：

$$c_j=\frac{1}{\sum_{v_i\in S_j}\frac{1}{\delta_i^2}}\sum_{v_i\in S_j}\frac{1}{\delta_i^2}\cdot\left(\delta_i\times v_i-\sum_{c_i\in C(v_i)/c_j}c_i\right) \tag{5}$$

其中，δ_i 是包含向量 v_i 的聚类数量，记作 $\delta_i=\left|C(v_i)\right|$。

2.2. KHM 算法

因为传统的重叠聚类算法是基于 k-means 的聚类方法，它对异常值比较敏感。更具体来说，当数据集包含异常值的时候，他们有可能被初始化为 k-means 算法的中心值，这就会导致 k-means 算法不能收敛到最好的聚类效果[10]。文献[11]提出的 k-harmonic means (KHM)算法通过最小化聚类中心的所有点的调和均值来解决此问题。调和均值会基于每个数据点与每个聚类中心的接近度赋予一个权重。该权重被认为是在数据集中识别每个点的聚类的重要指标。KHM 算法的最优解目标函数记为：

$$Q''=\sum_{i=1}^{n}\frac{k}{\sum_{j=1}^{k}\frac{1}{\left\|v_i-c_j\right\|^p}} \tag{6}$$

其中，p 是一个自由参数(一般 $p\geq 2$)，调和平均值是：

$$\frac{k}{\sum_{j=1}^{k}\frac{1}{\left\|v_i-c_j\right\|^p}}$$

为了计算调和平均值，首先计算聚类的中心 c_j，记作：

$$c_j=\frac{\sum_{i=1}^{n}m(c_j\mid v_i)w(v_i)v_i}{\sum_{i=1}^{n}m(c_j\mid v_i)w(v_i)} \tag{7}$$

其中，$m(c_j\mid v_i)$ 是数据点 v_i 到聚类中心 c_j 的隶属函数，记作：

$$m(c_j\mid v_i)=\frac{\left\|v_i-c_j\right\|^{-p-2}}{\sum_{j=1}^{k}\left\|v_i-c_j\right\|^{-p-2}} \tag{8}$$

公式中的 $w(v_i)$ 表示数据点 v_i 的权重，记作：

$$w(v_i)=\frac{\sum_{j=1}^{k}\left\|v_i-c_j\right\|^{-p-2}}{\left(\sum_{j=1}^{k}\left\|v_i-c_j\right\|^{-p}\right)^2} \tag{9}$$

3. KHM-IOKM 算法

3.1. 算法思想

提出的 KHM-IOKM 方法是一种解决增量数据重叠聚类问题的半在线算法，其目的是对新接收到的一系列无序增量数据在先前聚类的结果上进行重叠聚类处理。KHM-IOKM 算法的核心思想是：在处理增量数据集聚类之前，需要首先利用 KHM 算法对已有数据进行聚类处理，从而得到聚类中心集；再用这个中心集去初始化 OKM 方法的聚类中心并进行重叠聚类得到新的聚类中心；最后再对每个增量数据通过迭代使用 OKM 算法来进行聚类并更新相关的聚类中心，直到整个增量数据处理完毕为止。(具体实现过程参考后续的 3.2 节)。

3.2. 算法实现

KHM-IOKM 算法提高了传统 OKM 算法的性能，并降低了对初始聚类中心的敏感度。该算法具体包含五个主要步骤：

第一步：通过 KHM 方法对已有数据进行初步的聚类处理，确定聚类的中心。

第二步：使用第一步得到的聚类中心对 OKM 算法的聚类中心进行初始化。

第三步：在已有数据集上执行一次 OKM 重叠聚类处理，更新聚类中心。

第四步：确定已有数据集中的每个数据点所属的聚类。因为在重叠聚类情况下，数据点可以属于多个聚类，所以不能采用普通的距离测量法确定数据点的聚类，这里我们引入一个聚类隶属函数：

$$m(c_j\mid v_i)=\frac{\left\|v_i-c_j\right\|^2}{\sum_{j=1}^{k}\left\|v_i-c_j\right\|^2} \tag{10}$$

其中，式(10)的取值范围在[0, 1]之间，函数值越小，表明数据点 v_i 与中心点 c_j 的距离越近，隶属程度越高。为了确定隶属度 $\rho_{i,j}$ 的取值，引入平均距离公式：

$$d_{i,j}=\frac{\sum_{j=1}^{k}\left\|v_i-c_j\right\|^2}{k} \tag{11}$$

只要 $m(c_j\mid v_i)\le\rho_{i,j}$，我们就认为数据点 v_i 属于中心点为 c_j 的聚类。其中：

$$\rho_{i,j}=\frac{d_{i,j}}{\sum_{j=1}^{k}\left\|v_i-c_j\right\|^2}=\frac{1}{k} \tag{12}$$

重复第三步和第四步，直到公式(3)收敛于一个稳定的值，循环结束。

第五步：接收在线测试数据(增量数据集)，再次迭代利用 OKM 算法对无序的增量数据进行逐一聚类处理，并更新相关聚类的中心。直到接收增量数据完毕，算法运行结束。

整个 KHM-IOKM 模型的实现主要包括以下三大算法，具体描述如下：

算法 1：KHM 聚类算法

Step 1：输入参数和数据的初始化工作。

Input：k #聚类数量

p #距离的指数参数，一般取值 $p \geq 2$

Initialize：V #训练数据集

$centers = \{c_1, \cdots, c_k\}$ #聚类中心集合

Step 2：根据数据集中的每个样本，计算他们的隶属函数和权重，从而更新聚类中心。

For c in $centers$:

For v in V:

根据式(8)计算 x 的隶属函数

根据式(9)计算 x 的权重

根据式(7)计算聚类中心 c

Step 3：根据式(6)求解聚类最优解。

Step 4：重复执行 Step 2 和 Step 3，直到式(6)聚类的最优解收敛到固定的值为止。

Step 5：输出 KHM 算法的最优解，并得到最新的聚类中心。

算法 2：OKM 重叠聚类算法

Step 1：输入参数和数据的初始化工作。

Input：k #聚类数量

Initialize：V #训练数据集

Initialize： $centers = \{c_1, \cdots, c_k\}$ #用 KHM 算法得到的 $centers$ 进行初始化

Initialize： $C_j, j = \{1, \cdots, k\}$ #用 KHM 算法得到的聚类结果初始化

Initialize： $Q' = 0$ #聚类成本

Step 2：根据数据集中的每个样本，计算他们的隶属度，从而更新聚类中心。

For c in $centers$:

For v in V:

根据式(5)计算聚类中心 c

输出聚类中心

Step 3：对样本集数据进行聚类处理，并计算聚类的最优解。

For v in V:

根据式(12)计算样本数据的隶属度 $\rho_{i,j}$

根据式(10)计算隶属函数值 $m\left(c_j \mid v_i\right)$

If ($m\left(c_j \mid v_i\right) \leq \rho_{i,j}$):

对样本数据 v 进行聚类处理

根据式(4)计算包含样本数据 v 的所有聚类中心均值

$Q' = Q' + \left\| x - \phi(x) \right\|_2^2$ #OKM 的聚类成本

Step 4：重新初始化 $Q' = 0$，并重复执行 Step 2 和 Step 3，直到式(3)聚类的最优解收敛到固定的值为止。

Step 5：输出 OKM 算法的最优解，并得到最新的重叠聚类中心。

算法 3：增量数据的聚类算法

Step 1：接收增量数据和完成初始化工作。

Receive：D #接收到的增量数据集

Initialize： $centers = \{c_1, \cdots, c_k\}$ #用前面的混合算法 KHM-IOKM 得到的 $centers$ 进行初始化

Initialize： $C=\{C_1,\cdots C_k\}$ #前面的混合算法 KHM-IOKM 得到的各个聚类簇所包含样本集

Step 2：根据增量数据集中的每个样本，依次计算他们的隶属度，并及时更新受影响的聚类中心。

For d in D:

For c in *centers*:

根据式(12)计算样本数据的隶属度 $\rho_{i,j}$

根据式(10)计算隶属函数值 $m\left(c_j \mid d_i\right)$

If ($m\left(c_j \mid v_i\right) \leq \rho_{i,j}$):

$C_j \leftarrow C_j \cup d_i$ #聚类处理

根据式(5)更新相关聚类中心

3.3. 时间复杂度

KHM-IOKM 算法实现经历五步，其中前四步是第五步增量数据聚类的基础。因此，文中提出的这种混合算法的时间复杂度由两部分组成，分别是对已有数据的聚类和对新增数据的聚类处理。设定已有的数据集为 V，对已有数据的聚类处理主要包含：通过 KHM 方法在已有数据集上确定初始聚类的中心点，其时间复杂度为 $O\left(k|V|\right)$；通过 OKM 方法更新每一个聚类中心，其时间复杂度为 $O\left(k|V|\right)$；划分每个数据到相应的聚类中，其时间复杂度为 $O\left(|V|\right)$。对于一个新增数据，需要计算其到每一个聚类中心的距离，执行时间为 $O(k)$，k 为聚类簇个数。确定所属聚类之后，更新数据点所在聚类的中心，最坏情况下需要遍历整个数据集，执行时间为 $O\left(|V|+1\right)$，$|V|$ 为整个数据集规模。所以 KHM-IOKM 算法的总时间复杂度为 $O\left(k|V|\right)$。

4. 实验测试及性能分析

为了验证文中提出的 KHM-IOKM 算法的有效性、可扩展性和聚类的高效性，我们分别使用了 3 组仿真数据和 6 组真实数据进行实验测试。每组数据都会随机抽取一定量的样本作为增量数据集。在对 KHM-IOKM 算法进行效率测试的同时，还和传统重叠聚类算法 OKM 的运行效率进行了对比，实验过程中 KHM-IOKM 算法和 OKM 算法分别进行了 10 次迭代处理。同时，为了验证在这些具有重叠交叉信息的数据集上进行重叠聚类要比进行单一聚类效果更好，还和单一聚类算法 KHM 进行了比较。

所有的算法均采用 Python 3.6 编码实现，整个实验在 Spyder 集成开发环境下进行测试，实验平台配置为双核 i7 处理器，8 G 内存，64 位 Windows 10 操作系统。

4.1. 实验数据

为了更好地验证混合重叠算法的性能，3 组仿真数据集均是模拟金融领域的数据特征，基本信息描述如表 2 所示。

Table 2. Summary of synthetic datasets
表 2. 仿真数据集描述

Dataset	Data Set Characteristics	Attribute Characteristics	Missing Values	Area	Instances	Features
Data1	Univariate	Integer, Real	No	Finance	2000	5
Data2	Univariate	Integer, Real	No	Finance	1000	6
Data3	Univariate	Integer, Real	No	Finance	1000	9

仿真数据 Data1 由 Python 中 pandas 数据分析包模拟生成具有 5 个特征值的 1000 条数据作为已有数据，然后继续随机生成 1000 条数据作为增量数据集。

仿真数据 Data2 由 Python 中 pandas 数据分析包模拟生成具有 6 个特征值的 1000 条信用卡用户申请数据，为了便于聚类处理，特征值都是单一的数值型数据，并随机抽取 300 条数据作为增量数据集(及测试集)。

仿真数据 Data3 由 Python 中 pandas 数据分析包模拟生成具有 9 个特征值的 1000 条信用卡消费数据，随机抽取 300 条数据作为增量数据集(及测试集)。

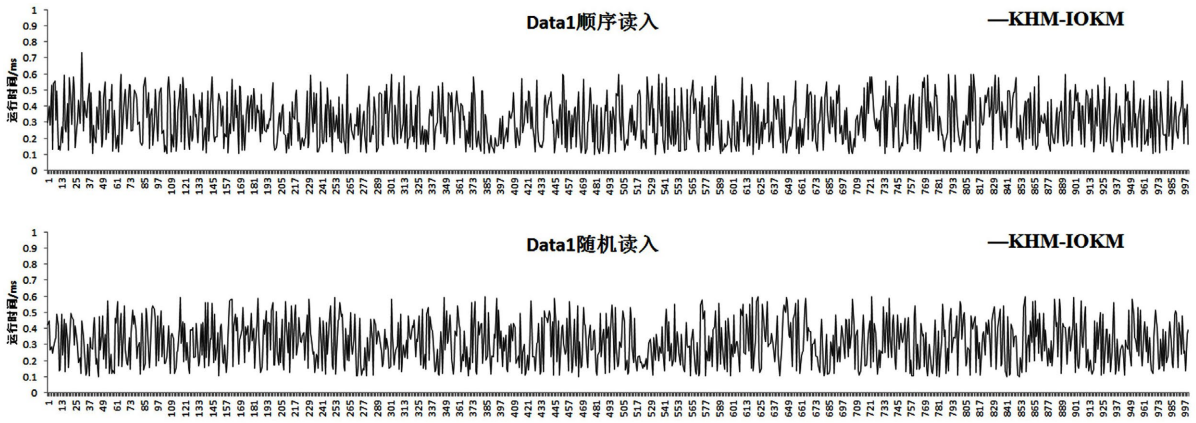
真实数据一共包含了不同领域的 6 组数据集，均来自 UCI Machine Learning Repository 网站，具体访问地址为 http://archive.ics.uci.edu/。6 组真实数据集的基本信息描述如表 3 所示。

Table 3. Summary of UCI real datasets
表 3. UCI 真实数据集描述

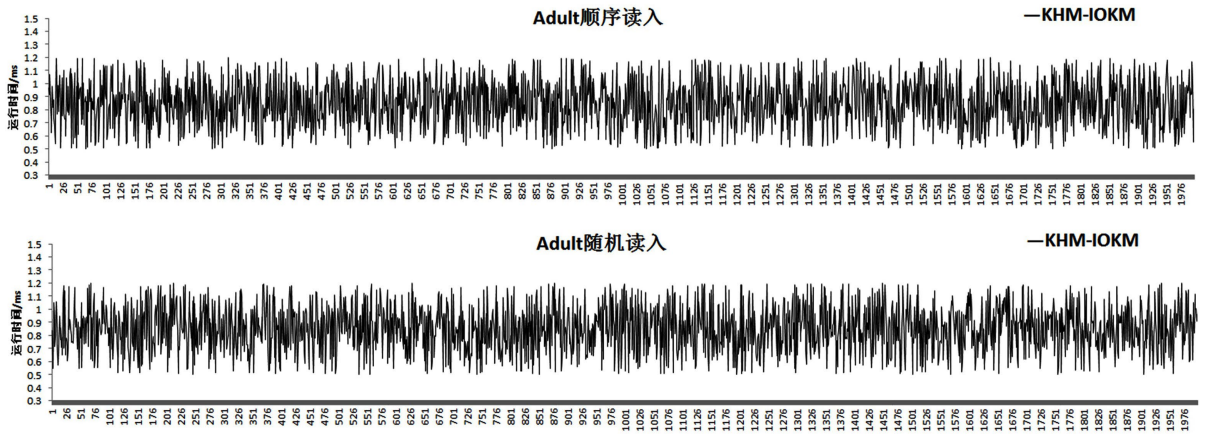
Dataset	Data Set Characteristics	Attribute Characteristics	Missing Values	Area	Instances	Features
Breast Cancer Wisconsin	Multivariate	Real	N/A	Life	699	10
Heart Disease Cleveland	Multivariate	Categorical, Integer, Real	Yes	Life	303	13
Heart Disease-Hungarian	Multivariate	Categorical, Integer, Real	Yes	Life	294	13
Wine Quality-white	Multivariate	Real	N/A	Business	4898	11
Adult	Multivariate	Categorical, Integer	Yes	Social	48,842	14
Skin Segmentation	Univariate	Real	N/A	Computer	245,057	3

4.2. KHM-IOKM 的效率测试

KHM-IOKM 混合算法分别在仿真数据集 Data1 和真实数据集 Adult 上做了增量数据的聚类效率测试，测试的方法是：算法按一定的速率分别按顺序和随机两种方式读取增量数据集。在测试实验中，设定 Data1 的聚类簇 $k = 7$，已有数据规模 $n = 1000$，增量数据的规模 $m = 1000$；Adult 的聚类簇 $k = 2$，已有数据规模 $n = 46{,}842$，增量数据的规模 $m = 2000$。聚类的运行时间如图 3 所示，图 3(a)为 Data1 增量数据的两种读取方式运行时间，图 3(b)为 Adult 增量数据的两种读取方式运行时间。从实验分析结果来看，增量数据的读取方式并不影响聚类的效率，其聚类运行时间比较稳定，只有极个别的耗时较高，符合算法复杂度的分析，同时也验证了 KHM-IOKM 具有较高的稳定性和可伸缩性。



(a) Sequential and random on Data1



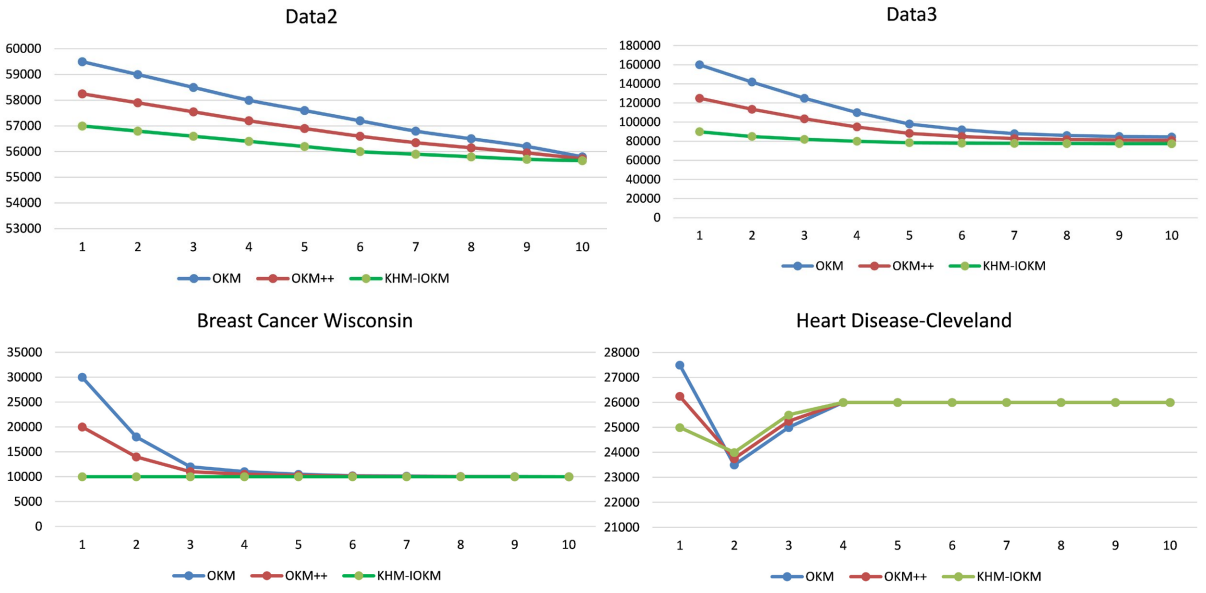
(b) Sequential and random on Adult

Figure 3. Clustering time of KHM-IOKM on Data1 and Adult
图 3. KHM-IOKM 在 Data1 和 Adult 上的运行时间

4.3. KHM-IOKM 的性能比较

4.3.1. 与重叠聚类模型的比较

本小节在 Data2、Data3 和 6 组真实数据集上比较了 KHM-IOKM、K-mean++初始化聚类中心后的重叠聚类(OKM++)和 OKM 的重叠聚类性能。测试结果如图 4 所示。KHM-IOKM 算法对增量数据的聚类处理是在 KHM 聚类的基础上进行重叠聚类处理，克服了聚类中心初始化带来的敏感性问题，所以最优解的收敛速度很快，性能很高。OKM++的性能表现低于 KHM-IOKM，但是优于 OKM，而传统的 OKM 算法的初始聚类中心是随机选取，对聚类的性能有很大的影响，算法的目标函数值一开始都很大，只有经过不断的迭代更新聚类中心，才逐渐收敛于一个比较稳定的值，所以其最优解的收敛速度慢，性能较低。因为 KHM-IOKM 算法是对传统重叠聚类算法 OKM 的改进，所以其性能表现出了较好的优越性。



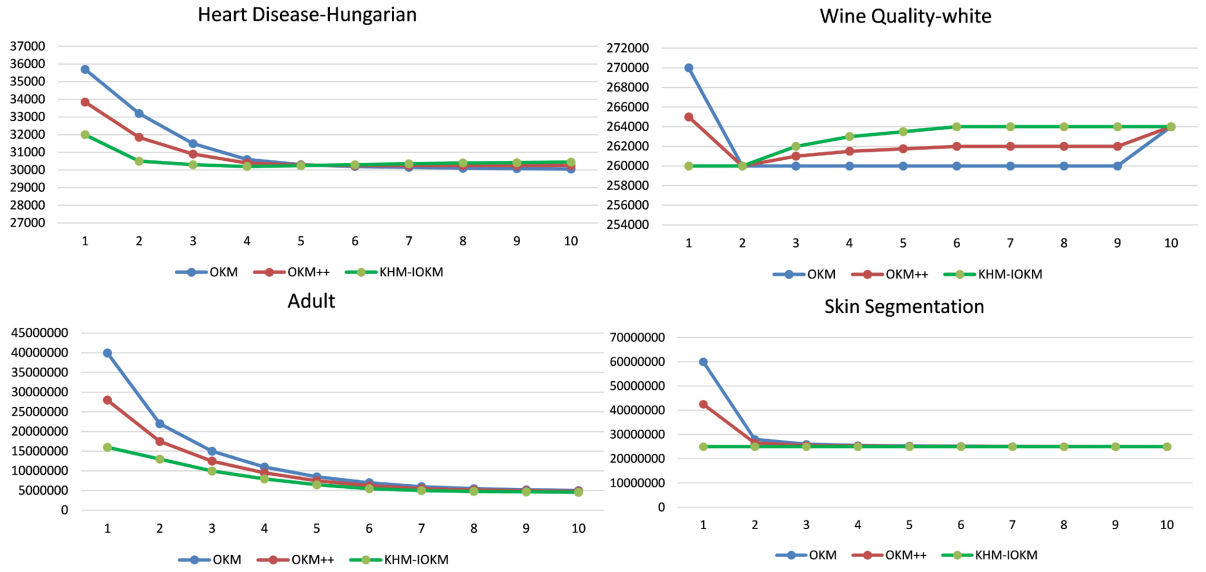
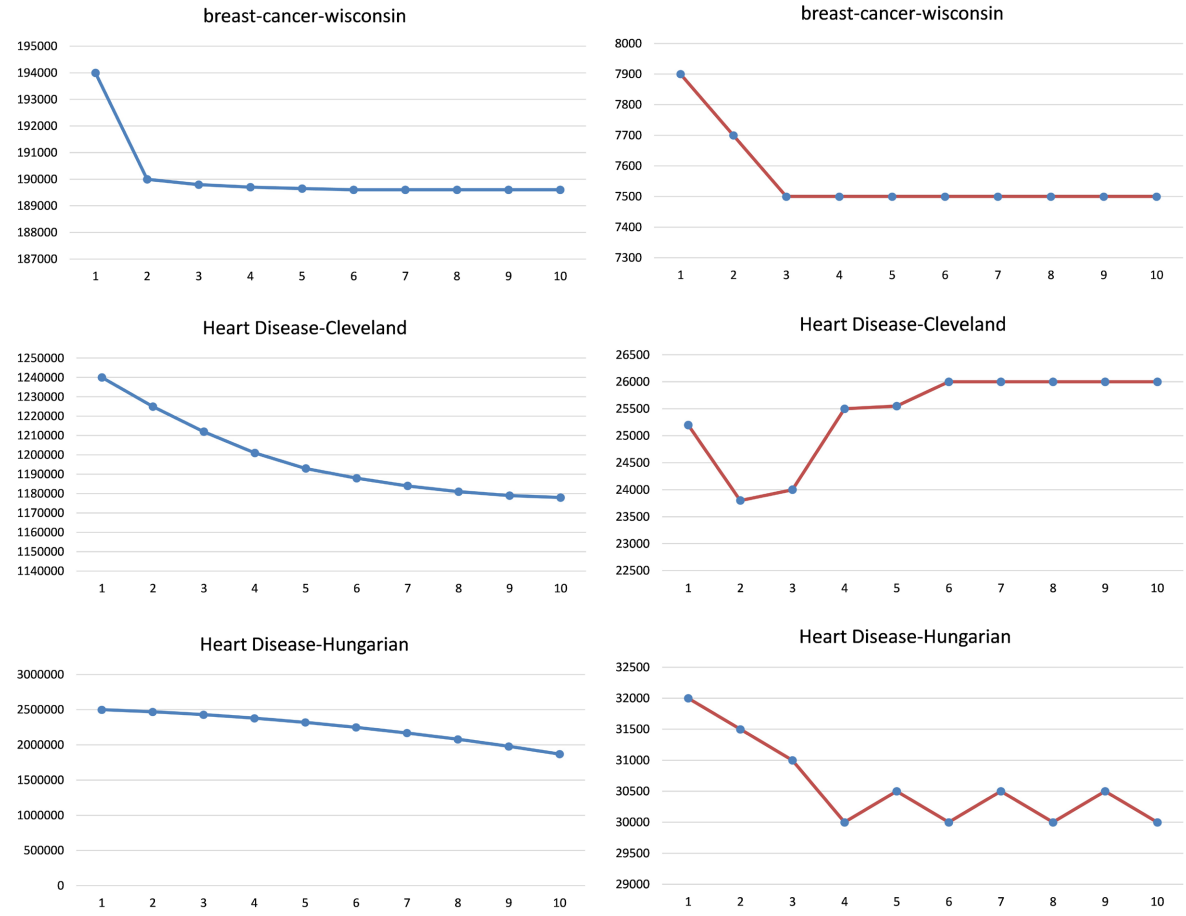


Figure 4. Comparison of objective function values of 10 iterations on 8 datasets by KHM-IOKM, OKM++, and OKM algorithms
图 4. KHM-IOKM、OKM++和 OKM 三种算法在 8 组数据集上进行 10 次迭代的目标函数值比较结果

4.3.2. 与单一聚类模型的比较



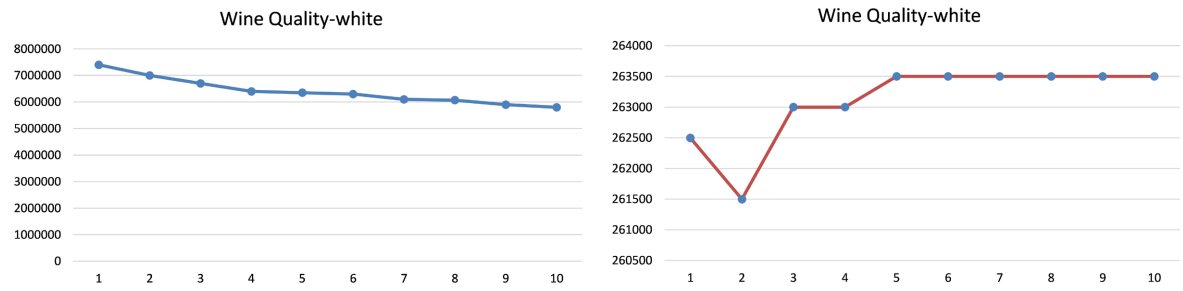


Figure 5. Comparison of objective function values of 10 iterations on 4 datasets by OKHM-OKM and KHM algorithms
图 5. OKHM-OKM 和 KHM 两种算法在 4 组数据集上进行 10 次迭代的目标函数值比较结果

本小节选择了 Breast Cancer Wisconsin、Heart Disease-Cleveland、Heart Disease-Hungarian 和 Wine Quality-white 共 4 组真实数据集，通过单一聚类算法 KHM 进行了聚类测试，并和 KHM-IOKM 算法进行了比较，由于两种算法在测试结果上不是一个数量级，为了更好地展示对比结果，分别进行了绘图，如图 5 所示。通过比较发现，在这 4 组数据集上，单一聚类的成本远远大于重叠聚类，这说明对于现实环境中存在重叠交叉信息的数据集进行重叠聚类处理能够大幅提高聚类的效果。

4.4. KHM 自由参数 p 的敏感性分析

KHM 算法中的自由参数 p ($p \geq 2$)是调控其聚类性能的核心超参数，其取值通过在目标函数、隶属度函数及权重函数中放大或缩小距离项，深刻影响着算法对异常值的鲁棒性、聚类边界的软硬程度以及收敛动力学行为。表 4 总结了参数 p 取不同值时的算法行为特性，为实践中的参数选择提供直观参考。参数 p 的敏感性可被概念化为一个连续的频谱：当 p 取值较小时(如 $p=2$)，算法行为趋近于传统 k-means，其对异常值较为敏感，但收敛速度快，且隶属度分布平滑，能较好地捕捉数据的重叠聚类结构；随着 p 值增大，距离的高次幂效应使得异常点的权重被显著抑制，算法鲁棒性极大增强，但同时聚类边界趋于硬化(隶属度分布尖锐)，优化 landscapes 变得更加复杂，收敛速度减缓且易陷入局部最优。因此，参数 p 的选择本质上是在鲁棒性、聚类粒度与计算效率之间进行权衡(Trade-off)。在实际应用中，需通过网格搜索等技术，依据具体数据集的噪声水平与聚类目标，确定最优的 p 值。

Table 4. Summary of the sensitivity analysis of KHM algorithm parameter p
表 4. KHM 算法参数 p 敏感性分析总结

算法特性	p 值较小(e.g., $p=2$)	p 值较大(e.g., $p=6$)
异常值敏感性	相对敏感，鲁棒性优于 k-means 但较弱	极其鲁棒，能有效抑制异常值影响
聚类边界	模糊/柔软，隶属度分布平滑，重叠聚类特性显著	清晰/坚硬，隶属度分布尖锐，近似硬聚类
收敛速度	通常较快，收敛行为更平顺	通常较慢，需要更多迭代次数
局部最优风险	相对较低	较高，优化 landscapes 更复杂
近似算法	行为接近但优于传统 k-means	行为接近其他鲁棒聚类变体

5. 结论

为了解决半在线重叠聚类过程中对聚类中心初始化敏感的问题，文中提出了一种改进的重叠聚类方法 KHM-IOKM，是在对已有数据进行高效聚类的基础上对增量数据进行处理。根据给出的算法描述，在完成增量数据集聚类之前，需要首先利用 KHM 算法对已有数据进行聚类处理从而得到聚类中心集，再用这个中心集去初始化 OKM 方法的聚类中心并进行重叠聚类得到新的聚类中心；然后对每个增量数据

通过迭代使用 OKM 算法来进行聚类并更新受影响的聚类中心。综合实验结果显示，算法具有高效的聚类性能，并且在处理不同类型的数据集上具有一定的可伸缩性，对增量数据的输入顺序不敏感，运行时间快并且性能稳定。尤其对于现实环境中含有重叠交叉信息的数据集(例如 Breast Cancer Wisconsin、Heart Disease-Cleveland 等数据集)，KHM-IOKM 混合重叠算法在聚类效果上要明显优于单一聚类算法。然而，目前的算法面临着聚类效果没有达到最佳的问题。下一步的工作将重点研究如何改进算法的聚类标准，从而提高 KHM-IOKM 算法的聚类效果，并将该算法应用到实时空间数据处理系统。

基金项目

河北省金融科技应用重点实验室课题(2025005)。

致　谢

感谢教育部《高等学校青年骨干教师国内访问学者项目》给本人提供在天津大学访学的机会，感谢天津大学智能与计算学部的博士生导师廖士中教授曾对我研究方向的指导和帮助。

参考文献

[1] Choromanska, A. and Monteleoni, C. (2012) AISTATS: Online Clustering with Experts. *Proceedings of the Fifteenth International Conference on Artificial Intelligence and Statistics*, La Palma, 2 May 2012, 227-235.

[2] Zhu, W., Yin, J. and Xie, Y. (2006) Arbitrary Shape Cluster Algorithm for Clustering Data Stream. *Journal of Software*, **17**, 379-387. https://doi.org/10.1360/jos170379

[3] 朱蔚恒, 印鉴, 谢益煌. 基于数据流的任意形状聚类算法[J]. 软件学报, 2006, 17(3): 379-387.

[4] Charikar, M., Chekuri, C., Feder, T. and Motwani, R. (1997) Incremental Clustering and Dynamic Information Retrieval. *Proceedings of the Twenty-Ninth Annual ACM Symposium on Theory of Computing*, El Paso, 4-6 May 1997, 626-635. https://doi.org/10.1145/258533.258657

[5] Coates, A., Ng, A.Y. and Lee, H. (2011) AISTATS: An Analysis of Single-Layer Networks in Unsupervised Feature Learning. *JMLR Proceedings*, **15**, 215-223.

[6] Cleuziou, G. (2010) Two Variants of the OKM for Overlapping Clustering. In: Guillet, F., Ritschard, G., Zighed, D.A. and Briand, H., Eds., *Studies in Computational Intelligence*, Springer, 149-166. https://doi.org/10.1007/978-3-642-00580-0_9

[7] Ben NCir, C. and Essoussi, N. (2010) KDIR: Kernel Overlapping K-Means for Clustering in Feature Space. *International Conference on Knowledge Discovery and Information Retrieval*, Valencia, 25-28 October 2010, 250-256.

[8] N'Cir, C.B. and Essoussi, N. (2012) Overlapping Patterns Recognition with Linear and Non-Linear Separations Using Positive Definite Kernels. *International Journal of Computer Applications*, **56**, 1-8. https://doi.org/10.5120/8916-2981

[9] Aroche-Villarruel, A.A., Carrasco-Ochoa, J.A., Martínez-Trinidad, J.F., Olvera-López, J.A. and Pérez-Suárez, A. (2014) Study of Overlapping Clustering Algorithms Based on Kmeans through Fbcubed Metric. In: Martínez-Trinidad, J.F., Carrasco-Ochoa, J.A., Olvera-Lopez, J.A., Salas-Rodríguez, J. and Suen, C.Y., Eds., *Lecture Notes in Computer Science*, Springer International Publishing, 112-121. https://doi.org/10.1007/978-3-319-07491-7_12

[10] Zhang, B. (2000) Generalized K-Harmonic Means. Technical Report, Hewlett-Packard Laboratoris.

[11] N'Cir, C.B., Cleuziou, G. and Essoussi, N. (2015) Overview of Overlapping Partitional Clustering Methods. In: Celebi, M., Ed., *Partitional Clustering Algorithms*, Springer International Publishing, 245-275. https://doi.org/10.1007/978-3-319-09259-1_8