山东外事职业大学
SHANDONG VOCATIONAL UNIVERSITY OF FOREIGN AFFAIRS

本科毕业论文

聚类算法的研究与应用

姓　名：	曹 * *
学　号：	11929****
学　院：	信息与控制工程学院
专业班级：	软件工程 19**
指导教师：	刘法胜 教授
	徐玮辰 助教

2023 年 5 月 18 日

聚类算法的研究与应用

Research and Application of Clustering Algorithms

曹 * *

C * * * * * *

2023 年 5 月 18 日

独创性声明

本人声明：所呈交毕业论文，是在导师指导下独立进行研究所取得的研究成果。论文除文中已经注明引用的内容外，不包含任何其他集体或个人已经发表或在网上发表的内容。

特此声明。

学生签名：　　时间：　2023 年　5 月　18 日

摘 要

聚类算法是一种无监督的机器学习算法，该算法可以把数据划分为不同的簇，每个簇内的数据具有较高的相似度，同时不同簇的数据会有比较大的差异。本文介绍了常用的几种聚类方法，包括 K-means、DBSCAN 等，从理论上分析了这些方法的优缺点和适用场景。系统性地介绍和总结了聚类算法的原理、方法和应用。分析了不同聚类算法的优点和缺点，以及在实际应用中的应用情况，为实际应用提供了参考。探讨了聚类算法的评价指标，为聚类结果的准确性和可解释性提供保障。

本文使用 DBSCAN 算法对我国城镇化数据和 GDP 数据进行了聚类分析，使用枚举法通过观察聚类效果与比较轮廓系数，确定算法的输入参数。最终聚类效果较好。

最后，对聚类算法的研究进行了总结和讨论，指出聚类算法在实际应用中的潜在问题以及未来研究方向。未来的研究方向可从深度学习、大数据等方面展开，以进一步提高聚类算法的效率和准确度。

关键词：聚类算法;数据标准化;轮廓系数;DBSCAN 算法;python;城镇化

Abstract

Clustering algorithm is an unsupervised machine learning algorithm that can divide data into different clusters, with high similarity within each cluster, and significant differences in data across different clusters. This article introduces several commonly used clustering methods, including K-means, DBSCAN, etc., and theoretically analyzes the advantages,disadvantages, and applicable scenarios of these methods. Systematically introduce and summarize the principles, methods, and applications of clustering algorithms. Analyzed the advantages and disadvantages of different clustering algorithms, as well as their practical applications, providing reference for practical applications. Explored the evaluation indicators of clustering algorithms, providing guarantees for the accuracy and interpretability of clustering results.

This article uses the DBSCAN algorithm to cluster and analyze urbanization and GDP data in China. The enumeration method is used to determine the input parameters of the algorithm by observing the clustering effect and comparing the contour coefficients. The final clustering effect is good.

Finally, a summary and discussion were conducted on the research on clustering algorithms, pointing out the potential problems and future research directions of clustering algorithms in practical applications. Future research directions can be explored in areas such as deep learning and big data to further improve the efficiency and accuracy of clustering algorithms.

Key words： Clustering algorithm; Data standardization; Profile coefficient; DBSCAN algorithm; python; Urbanization

目 录

第 1 章 绪论

1.1 研究目的与意义

随着社会经济的发展和科技的进步，数据的产生和存储呈指数级增长，如何高效地对数据进行分类和聚类成为了一项热门的研究课题。聚类算法是数据挖掘领域的重要算法之一，可以将数据集中相似的数据点分成一组，不同组之间的数据点则具有较大的差异性[1]。传统的聚类算法如 K-Means 聚类算法和层次聚类算法存在着一些问题，比如对数据的分布情况敏感、聚类中心不易确定等，因此需要不断优化和改进[2]。本文将对聚类算法的基础知识、综述、融合、优化以及在实际问题中的应用进行分析和研究，旨在深入探讨聚类算法的优化和改进方向，提高聚类算法在实际应用中的效果。聚类相关方法在各个领域都有着广泛的应用，如金融领域中对客户进行分类、医学领域中对病人进行分类、社交网络领域中对用户进行分类等。同时，聚类算法的发展也面临着很多问题，如算法复杂度的提高、数据集规模的增大、聚类结果的可解释性、聚类中心点的选择等问题[3]。因此，对聚类算法进行深入研究具有重要的理论和实践意义。

1.2 国内外研究现状

在国内外，聚类算法一直是数据挖掘领域的研究热点和难点之一。近年来，随着数据规模和种类的不断增加，聚类算法的研究也日趋广泛和深入。聚类算法在国内外均是数据挖掘领域的重要研究课题。研究人员一直在探索各种新的聚类算法、方法和应用场景，以提高聚类效率和准确性，并且在不断改进和优化现有的聚类算法，使其更好地适应不断增长的数据量和种类。

1.2.1 国外研究现状

聚类算法的研究始于 20 世纪 50 年代，最早提出的是层次聚类算法。此后，聚类算法一直是机器学习、数据挖掘等领域的重要研究课题之一。近年来，随着深度学习技术的兴起，深度聚类（Deep Clustering）也成为了研究热点之一。同时，研究人员对聚类算法在大数据处理、自然语言处理、图像分析等方面的应用和改进也进行了广泛研究。

1.2.2 国内研究现状

在国内，聚类算法的研究始于上世纪 80 年代末和 90 年代初。国内聚类算法的研究较为活跃，并在不同领域取得了一定的进展，如在航空、气象、遥感图像处理、数据挖掘、网络安全等领域得到了广泛应用。同时，我国研究人员在层次聚类、K-Means 聚类、谱聚

类等算法上也进行了深入研究，并提出了不少改进和优化方案，如基于深度学习的聚类、基于 PSO 优化的 K-Means 聚类等。此外，国内研究人员也关注聚类算法在大数据处理、社交网络分析、计算机视觉等方面的应用和改进。

1.3 本文主要研究内容

本研究将以聚类算法及其应用为主要研究对象，探讨聚类算法的基本概念、具体方法和实际应用，并分析不同聚类算法的优点以及缺点。其中，将着重研究以下内容：

聚类算法的基本原理和分类方法。

聚类算法的评价指标，如聚类质量、聚类效率等。

聚类典型算法，如 K-Means、DBSCAN 算法。

聚类算法在具体领域的应用。

第 2 章 聚类算法基础

2.1 聚类定义

聚类算法是一种通过相似性度量将数据集划分成若干个组（簇）的统计学方法。聚类算法能够把相似数据点分到同一簇中，使得每簇内的数据的相似程度较高。这提供了一种有助于为大数据集提供直观性归纳的方法[4]。聚类算法根据簇建立的规则、簇建立的起点数量和簇的形状以及基于数据对象之间的距离进行分类。聚类算法的目的是把数据集中的数据点分成多个相似的子集。聚类算法可以应用于许多领域，例如数据挖掘、图像分割、自然语言处理等。聚类算法可以用于探索数据的结构和模式，从而为后续分析和决策制定提供基础和参考。聚类算法有许多种，例如 Kmeans、DBSCAN、层次聚类、密度聚类等。每种算法都有其自身的优点和局限性，并且适用于不同类型和不同形状的数据[5]。

聚类算法是一种非常通用且广泛应用的数据挖掘技术。不同的聚类算法可以有不同的簇判定规则、相似性度量和优化策略，每种算法都有着其特点和适用范围，用户应该根据自己的需求和数据集的特点选择合适的算法。

2.2 聚类算法的分类

聚类算法有多种分类[5]，具体如下：

（1）层次聚类方法是基于数据的相似性和距离构建聚类层次结构。常见的层次聚类方法包括凝聚层次聚类和分裂层次聚类。

（2）划分聚类方法，通常采用迭代式的贪心算法来寻找全局最优解，其中比较典型的算法是 k-means 和 k-medoids 算法，都采用距离度量和质心点作为聚类的中心点来进行划分。

（3）网格聚类方法是将数据点放入 2D 或 3D 网格中，然后计算每个网格领域的密度和聚类性质，最终形成多个簇。该方法相对简单快速，但对于密度不均匀或包含离群值的数据集容易失效。

（4）密度聚类方法将聚类定义为数据集中高密度的区域，在密度高的区域内定位簇中心，并把相邻的高密度区域的边界点归为同一个簇。

（5）模型聚类方法是采用概率模型建模的方法，通过拟合概率分布模型来刻画数据集内的聚类结构，包括高斯混合模型、贝叶斯聚类模型等。模型聚类方法会考虑更多的

统计性质，但模型的拟合过程较为复杂，需要考虑模型选择、超参数调整等问题。

2.3 聚类评价

聚类评价是用来衡量聚类算法效果的一种指标体系，通过对聚类结果进行评估和对比，以确定聚类算法的优劣性。内部评价是在不使用真实标签或类别标签的情况下对聚类结果进行评估。聚类算法通常会在数据集上运行数次以生成基于不同初始值的聚类结果，然后使用内部评价度量选择最优的聚类算法和参数。在内部评价中，最常用的是聚类分布的一些重要特征，例如聚类类间的分离度（距离）和聚类内部的紧密度（方差）。外部评价是由已知标签的真实数据集进行评价，已知的标签可以是人类指定的，也可以基于已有的分类信息。一般情况下，外部评价提供了更有效的聚类算法选择和性能度量。常用的外部指标包括误分类率、精度、查准率和查全率等。内部评价和外部评价各有优点和缺陷。内部评价可以提供对聚类结果生成的指导，但对于真实数据可能不太准确。外部评价所需数据对比较锚定（即包括真实分类的数据），但可能受到样本偏差的影响[6]。

2.3.1 内部评价

内部评价[6]指的是在不使用真实标签或类别标签的情况下对聚类结果进行评估，通常用于确定聚类算法的优化参数或评估聚类算法之间的优劣性。常用的内部评价指标有：

轮廓系数（Silhouette Coefficient）：衡量聚类质量的一种方式，其取值范围在-1 到 1 之间。

DB 指数（Davies-Bouldin Index）：利用聚类中心的距离和类内离散程度，来度量聚类算法的性能。

CH 指数（Calinski-Harabasz Index）：基于聚类间的方差和聚类内部的方差比值来度量聚类质量。

2.3.2 外部评价

外部评价[6]通常使用已知标签的真实类别信息进行评估，以此来度量聚类算法的性能。通常用于检查聚类算法的误差率和错误分类数，外部评价指标有：

Purity 纯度：计算聚类结果中每个簇中出现频率最高的真实类别，将所有簇的这个值相加作为评价指标。

NMI (Normalized Mutual Information) 归一化互信息：计算聚类结果和真实类别之间的信息量，并将其标准化，使其范围在 0 到 1 之间。

第 3 章 常用聚类算法

3.1 K-Means 聚类算法

K-Means 聚类算法是一种迭代的分区聚类算法。该算法需要预设聚类中心点的个数 k，随机分配 k 个对象为初始聚类中心点[7]。然后，将所有其他对象分配到最近的聚类中心，并重新计算聚类中心的位置，迭代以上操作直到达到收敛条件。在 K-means 算法中，簇内的数据点的方差越小，意味着簇内的数据点越相似[8]。因为 K-means 算法经常被用于数据压缩和图像处理中对数据的聚类分析，所以该算法的聚类中心通常被用作数据的描述符[9]。K-means 算法的缺点是需要事先确定簇的数量 K，对于无法确定 K 的情况，就需要使用基于密度的聚类算法。K-Means 聚类算法的优点是计算速度快，易于实现，但其结果取决于初始随机选取的簇中心点和聚类数量的选择[10]。

代码示例：

```
def kmeans(X, k, max_iters=100):
    centroids = X[np.random.choice(len(X), k, replace=False)]
    for i in range(max_iters):
        distances = np.sqrt(((X - centroids[:, np.newaxis])**2).sum(axis=2))
        cluster_ids = np.argmin(distances, axis=0)
        new_centroids = np.array([X[cluster_ids == j].mean(axis=0)
                                  for j in range(k)])
        if np.allclose(centroids, new_centroids):
            break
        centroids = new_centroids
    return centroids, cluster_ids
```

3.2 DBSCAN 聚类算法

DBSCAN 聚类算法是另一种基于密度的聚类算法。DBSCAN 将基于密度的样本分为三个类型：核心密度、边界密度和噪声样本[11]。该算法基于两个参数，即 ε-邻域半径和最小密度阈值，来定义样本的密度[12]。该算法将核心密度点和其密度可达的边界点划分到一个簇中，并将噪声样本点放到单独的簇中。DBSCAN 算法的优点是对任意形状的簇能够进行分组，对噪声能够具有很好的容错性[13]。可以自动发现凸和非凸簇，并且不需要预先规定聚类数量，但其结果受噪声点比较敏感。该算法的缺点是需要知道 ε 和 Minpts 的值，这些值一般需要通过试错、根据业务经验来寻找最优值[14]。同时，在处理高维数据时效果会降低，需要利用 PCA 等方法先进行降维处理[15]。

代码示例：

```
def DBSCAN(X, eps, min_samples):
    nbrs = NearestNeighbors(n_neighbors=min_samples, algorithm='ball_tree').fit(X)
    distances, indices = nbrs.kneighbors(X)
    labels = np.zeros(X.shape[0], dtype=int)
    C = 0
    for i in range(X.shape[0]):
        if not labels[i] == 0:
            continue
        neighbors = indices[i, :]
        if len(neighbors) >= min_samples:
            labels[i] = C
            for j in neighbors:
                if not labels[j] == 0:
                    continue
                labels[j] = C
                sub_neighbors = indices[j, :]
                if len(sub_neighbors) >= min_samples:
                    for k in sub_neighbors:
                        if not labels[k] == 0:
                            continue
                        labels[k] = C
        if labels[i] == 0:
            labels[i] = -1
        C += 1
    return labels
```

3.3 K-Means 与 DBSCAN 的对比

3.3.1. Kmeans 算法优缺点

Kmeans[16]是一种基于距离的聚类算法，该算法通过将数据点分配给离其最近的质心来找到聚类。

该算法的优点有：易于实现和解释，计算速度快，对大数据集有良好的可扩展性。由于该算法将每个数据点都分配到一个簇中，因此对于非常规形状的数据集也可以有很好的效果。

然而，Kmeans 的缺点也是显著的，比如：对于任意聚类数量的选择非常敏感；对于非凸场景的数据集，Kmeans 可能无法得到很好的聚类结果。

3.3.2. DBSCAN 算法优缺点

DBSCAN[17]是一种基于密度的聚类算法。DBSCAN 的优点有：能够处理簇中存在的噪音点；不需要为聚类数量进行设置。缺点：对噪声点和边缘点敏感；对于密度相差较大的数据集，必须手动调整参数。

3.3.3 应用场景

Kmeans 算法通常适用于需要特定聚类数量并且数据集是凸的情况。例如，市场分析和客户细分等领域中，通常需要确定一定数量的客户类别；

DBSCAN 算法适用于更泛化的场景，可能不确定簇数或簇大小，或者簇的形状可能非凸形状。例如，诊断医学中，对于疾病群体可能没有先验知识，也没有对应的标签。

第 4 章 基于 DBSCAN 的聚类应用

4.1 数据收集与处理

4.1.1 数据收集

数据集为我国 32 省 2022 年 GDP 数据与 2021 年城镇化率数据。

4.1.2 数据处理 Z-score 标准化处理

标准化的目的是将数值范围不同的变量转换为具有相同数值范围的标准化变量，以便于后续聚类算法的计算。Z-score 标准化处理是根据每个数据点距离该数据集的平均值的偏差程度来对数据进行处理，把数据变为标准正态分布。经过这样处理之后，所有数据都将分布在以 0 为中心，标准差为 1 的正态分布中，便于进行统计分析[18]。

将变量'GDP'和'Urbanization_rate'标准化成均值为 0，标准差为 1 的形式。

4.2 DBSCAN 聚类分析

利用 DBSCAN 算法对标准化后的建模变量进行聚类。DBSCAN 算法的优点是可以把噪声点单独识别出来，不会被归到任何一个簇群中去。

4.2.1 参数选取

在 DBSCAN 算法中，eps 控制着邻域的范围，min_samples 为范围内最小点数[19]。

确定 DBSCAN 算法参数的常用方法如下：

（1）根据数据集特征直接选择ε半径

这种方法比较直观简单，根据数据的特点，可视化观察数据分布情况，然后选择合适的ε半径。但是这种方法也有一定的局限性，适用于数据点分布比较密集、距离比较接近的情况，不适用于数据分布比较稀疏的情况。

（2）基于距离分布的方法

通过计算不同距离阈值下数据点和邻居的个数的对数图像(即 k-距离曲线)来确定最佳的ε半径以及最小点数。该方法首先需要计算每个数据点到第 k 个邻居的距离 k-dist，然后将这些距离从小到大排序，画出 k-距离曲线。通过分析 k-距离曲线，选择上升的拐点位置作为ε半径。同时，通过拐点附近的 k 值，可以确定最小点数[20]。

（3）基于平均密度的方法

该方法需要计算每个样本点的密度值和平均密度分布。首先，确定ε半径，计算每个数据点的 eps 邻域内的数据点个数，即密度值。接着，计算所有样本点的密度值的平均值

和标准差。最后，选择平均密度分布的中心位置作为ε半径，并且选择平均密度附近的一个合适的密度值作为最小点数[21]。

（4）基于网格结构的方法

该方法首先在空间上划分网格，然后在不同的ε半径下观察数据点在每个网格中的个数变化情况。在这个过程中，可以调整ε半径、网格大小等参数来选择最佳的聚类结果。当ε半径增加时，每个网格中数据点的个数也会增加，当单个网格中数据点个数达到或超过最小点数时，这些数据点就可以被视为一个簇，最终确定ε半径和最小点数[22]。

本文采用枚举参数组合的方法来确定参数，具体为通过双重 for 循环在给定范围内遍历两个重要参数的组合情况，同时记录簇的个数、异常点的个数等聚类情况。

eps	min_samples	n_clusters	outlines	stats
0.151	2	3	25	[2 2 2]
0.451	5	3	12	[7 7 5]
0.501	2	3	5	[22 2 2]
0.551	2	3	4	[23 2 2]
0.601	2	3	4	[23 2 2]
0.651	2	3	4	[23 2 2]
0.701	2	3	4	[23 2 2]
0.751	2	3	4	[23 2 2]
0.801	2	3	4	[23 2 2]
0.851	2	3	3	[24 2 2]
0.901	2	3	1	[25 3 2]
0.951	2	3	1	[25 3 2]

图 4-1 部分参数组合情况

通过多次实验观察后在组合中选取了两组聚类效果比较好的参数，最后比较各自的轮廓系数，将组合 1 确定为输入参数。

组合 1：eps=0.951 min_samples=2 轮廓系数：0.46894328043814537

组合 2：eps=0.801 min_samples=2 轮廓系数：0.4502337314683505

最终输入参数：邻域半径 eps=0.951 最小样本数 min_samples=2

4.3.2 DBSCAN 聚类

首先，使用 Pandas 读取了 Excel 文件 21 城市化.xlsx，选择需要聚类的特征（GDP 和 Urbanization_rate'）作为预测变量。然后，通过 preprocessing 库进行数据标准化（z-score 归一化），并使用 DBSCAN 算法进行聚类（eps=0.951, min_samples=2）。聚类结果存储在数据集的“dbscan_label”列中。

接下来，使用 sklearn 库中的 metrics 函数计算聚类结果的轮廓系数，并打印输出结果。最后，使用 matplotlib 和 seaborn 库对聚类结果进行可视化（以 GDP 和 Urbanization_rate' 为坐标轴）。

4.3 结果分析

使用 Seaborn 绘制出聚类效果的散点图，并将省份的名称指定为数据点的标签。图形的横纵坐标分别为“GDP”和“Urbanization_rate”，点的颜色和形状表示所属的簇。同时，计算并打印出所有数据点的轮廓系数，该系数可以用于评估聚类的效果好坏。

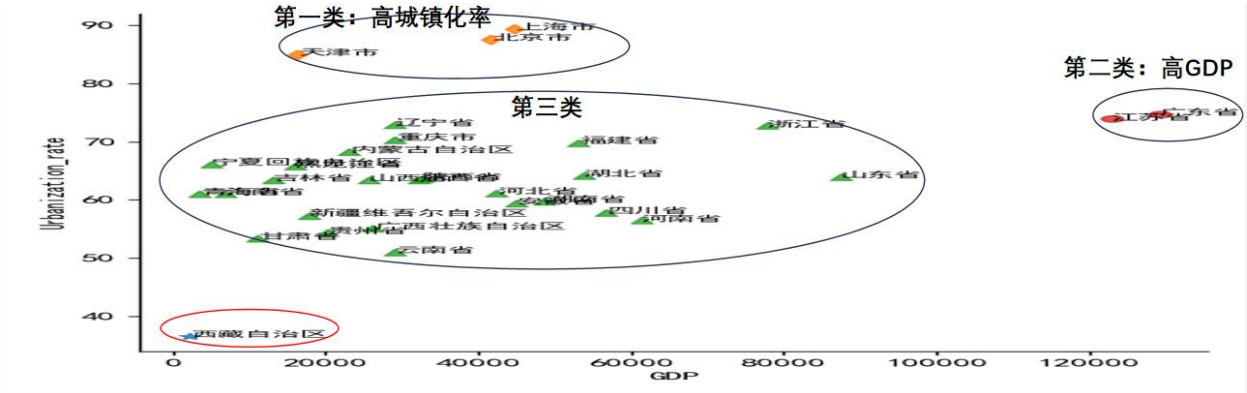

图 4-2 DBSCAN 聚类结果图

由图 4-1 可以看出，数据集被分为了三类，第一类的特征是高城镇化率，包括北京市、上海市和天津市。第二类的特征为高 GDP，有江苏省和广东省。第三类为大部分省份，特征比较集中，但同时也有两个点的 GDP 较高，为浙江省和山东省。数据集中有一个异常点为西藏自治区，GDP 和城镇化率较低。最终轮廓系数为 0.46894328043814537，数据集被分为三类，轮廓系数较高，聚类效果较好。

第 5 章 总结与展望

5.1 总结

论文首先阐述了聚类算法研究的目的与意义，以聚类算法为主要研究对象，探讨聚类算法的概念、原理、方法和应用，对各种聚类算法的相关理论进行了学习研究，对典型聚类算法，如 K-Means、DBSCAN 进行了具体介绍，并分析了不同聚类算法的优点和缺点。同时运用 DBSCAN 算法对具体数据集进行了实际聚类应用。在实际应用中，采用枚举参数组合的方法确定参数，最终轮廓系数较高，聚类效果较好。

5.2 展望

论文中存在部分要完善的地方：

（1）对聚类算法的概念、基本原理的认识还不够深入

（2）对聚类算法在不同领域的应用、具体评价指标的研究不够完善

（3）对于典型聚类算法的研究不够深入

（4）对 DBSCAN 算法参数的选取方法研究不足、存在参数不精确等情况

聚类算法在各个领域中具有广泛的应用，在未来的研究中，可以进一步探索改进聚类算法来提高其表现和可用性。例如，可以将深度学习技术引入聚类算法中来提高准确率和速度，或者开发新的聚类评估指标来更好地评估算法的性能。

同时，可以将聚类算法与其他算法结合使用，例如分类算法、回归算法等，以建立更为精确的模型。最后，可以通过数据可视化的方法将聚类结果更直观地展示出来，方便更直观地理解和使用聚类算法。

参考文献

[1]孟增辉. 聚类算法研究[D]. 河北大学.

[2]蔡元萃, 陈立潮. 聚类算法研究综述[J]. 科技情报开发与经济, 2007, 17(1):145-146.

[3]聂跃光, 陈立潮, 陈湖. 基于密度的空间聚类算法研究[J]. 计算机技术与发展, 2008, 18(8):4.

[4]陈良维. 数据挖掘中聚类算法研究[J]. 微计算机信息, 2006(07X):3.

[5]周涛, 陆惠玲. 数据挖掘中聚类算法研究进展[J]. 计算机工程与应用, 2012, 48(12):100-111.

[6]李莹. 聚类结果评价方法与聚类知识提取技术的研究[D]. 南京航空航天大学.

[7]周爱武, 于亚飞. K-Means 聚类算法的研究[J]. 计算机技术与发展, 2011, 21(2):4.

[8]韩晓红, 胡彧. K-means 聚类算法的研究[J]. 太原理工大学学报, 2009(3):4.

[9]张建辉. K-means 聚类算法研究及应用[D]. 武汉理工大学, 2007.

[10]王千, 王成, 冯振元,等. K-means 聚类算法研究综述[J]. 电子设计工程, 2012, 20(7):4.

[11]吴雪琴. DBSCAN 算法研究[J]. 软件导刊, 2007(4):2.

[12]孙凌燕. 基于密度的聚类算法研究[D]. 中北大学.

[13]安计勇, 韩海英, 侯效礼. 一种改进的 DBscan 聚类算法[J]. 微电子学与计算机, 2015, 32(7):4.

[14]吴善杰, 王新. 基于 AGA-DBSCAN 优化的 RBF 神经网络构造煤厚度预测方法[J]. 计算机科学, 2021, 48(7):8.

[15]冯少荣, 肖文俊. DBSCAN 聚类算法的研究与改进[J]. 中国矿业大学学报, 2008, 37(1):7.

[16]吴夙慧, 成颖, 郑彦宁,等. K-means 算法研究综述[J]. 现代图书情报技术, 2011, 27(5):28-35.

[17]宫蕊, 舒红平, 郭远远. 基于 DBSCAN 的密度聚类算法的研究[C]// 中国信息技术与应用学术论坛. 2008.

[18]苍宏宇, 谭宗颖. 国内外信息检索研究热点分析——基于 Z-Score 标准化的词频[J]. 图书馆建设, 2009(1):6.

[19]李宗林, 罗可. DBSCAN 算法中参数的自适应确定[J]. 计算机工程与应用, 2016.

[20]高新华, 王超, 顾晓清,等. 基于 DBSCAN 聚类算法的疏散星团 NGC 188 的 3 维运

动学成员判定[J]. 天文学报, 2017, 58(5):8.

[21]胡博磊. 基于累积平均密度的聚类算法研究[D]. 湖南大学, 2013.

[22]张枫, 邱保志. 基于网格的高效 DBSCAN 算法[J]. 计算机工程与应用, 2007, 43(17):4.

附录：

DBSCAN 实验代码

```
import pandas as pd
import matplotlib.pyplot as plt
from sklearn import preprocessing, __all__, metrics
from sklearn import cluster
import numpy as np
import seaborn as sns
pd.set_option('display.max_columns', None)
pd.set_option('display.max_rows', None)
plt.rcParams['font.sans-serif'] = ['SimHei']
plt.rcParams['axes.unicode_minus'] = False
Province = pd.read_excel(r'D:\pythonProject\21 城市化.xlsx')
predictors = ['GDP', 'Urbanization_rate']
X = preprocessing.scale(Province[predictors])
X = pd.DataFrame(X)
dbscan = cluster.DBSCAN(eps=0.951, min_samples=2)
dbscan.fit(X)
Province['dbscan_label'] = dbscan.labels_
score=metrics.silhouette_score(X, Province['dbscan_label'])
print("score:")
print(score)
sns.lmplot(x='GDP', y='Urbanization_rate', hue='dbscan_label', data=Province,
markers=['*', 'd', '^', 'o'], fit_reg=False, legend=False)
for x, y, text in zip(Province.GDP, Province.Urbanization_rate, Province.Province):
plt.text(x + 0.1, y - 0.1, text, size=8)
plt.xlabel('GDP')
plt.ylabel('Urbanization_rate')
plt.show()
```

致 谢

在这里要感谢我的指导教师刘法胜教授，在整个研究过程中，他提供了极其宝贵的指导和帮助。他耐心地听取我的想法，并提供了重要的反馈和建议，使得我的研究更加严谨和完整。

此外，我还要感谢我的父母和家人，他们一直鼓励和支持我。他们在我面临困难和挫折时，给予了我精神上的支持和鼓励。

最后，我要感谢我的朋友们，在我学习和生活中给予了我很多的帮助和支持。他们在我需要帮助时，总是愿意伸出援手，帮助我度过难关。