

人工智能实验报告

实验名称： 聚类 K-Means算法

|  |  |  |  |
| --- | --- | --- | --- |
| 学生学号 | 202110120401 | 202110120420 | 202110120408 |
| 学生姓名 | 刘广润 | 赵威 | 喻慧超 |
| 分工 | 贝叶斯网络实现,测试算法性能，撰写报告 | 算法速度测试，绘制算法执行时间表并进行对比，撰写报告 | 优化算法、撰写报告 |

。

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **实验评价结果** | | | | |
| **占比** | 10% | 40% | 50% | 总分 |
| **评分标准** | 报告的规范性。报告中的术语、格式、图表、数据、公式、标注及参考文献是否符合规范要求。 | 报告的严谨性。结构是否严谨，论述的层次是否清晰，逻辑是否合理，语言是否准确。 | 实验的充分性。实验是否完备，数据是否合理，是否有创新性成果或独立见解。 |
| 评分 |  |  |  |  |

报告提交时间： 2023 年 12 月 23 日

# 试验任务

现在有一批鸢尾花的数据，共包含150个样本，每个样本有四个属性， Sepal Length （花萼长度），Sepal Width（花萼宽度），Petal Length （花瓣长度），Petal （花瓣宽度） Width。同时，每个样本所属类别也已经标出，一共有3个类别：Iris Setosa（山鸢尾）、Iris Versicolour（杂色鸢尾），以及Iris Virginica（维吉尼亚鸢尾）。

根据样本的属性数据将鸢尾花用K-Means 算法进行聚类，获得3个类别，并将每个样本分到一个类别中。然后将聚类所得的样本类别分布情况与原始数据中的样本类别分布情况进行对比，分析K-Means算法的性能。

# 实验平台

Windows操作系统，Python编译环境，sklearn，matplotlib等程序库。

# 实验原理

## K-Means算法概述

K-Means算法是一种广泛使用的聚类算法，它的目标是将相似的数据点分组到一起，形成聚类。这种方法基于特征空间中数据点的距离来进行分组，通常使用欧几里得距离来衡量。算法的核心思想是找到聚类中心（centroids），使得数据点到其最近聚类中心的距离之和最小。

## K-Means算法基本原理

### 初始化

算法开始时首先选择K个数据点作为初始聚类中心。由于K-Means算法试图找到最小化每个点与其最近聚类中心距离平方和的聚类中心，算法的最终结果很大程度上依赖于初始聚类中心的选择。不恰当的初始化可能导致不理想的聚类结果或收敛到局部最优解。

### 分配阶段

在K-Means聚类算法中，分配阶段是一个关键步骤，它直接影响着聚类的结果。在这个阶段，算法将每个数据点分配给最近的聚类中心，形成初始的聚类。

**分配阶段的步骤：**

1. 计算距离：对于数据集中的每个点，计算其与每个聚类中心的距离。这个距离通常是欧氏距离，虽然也可以使用其他类型的距离度量，如曼哈顿距离或余弦相似度。
2. 分配聚类：根据计算出的距离，将每个数据点分配给最近的聚类中心。具体来说，对于每个数据点，找出距离它最近的聚类中心，并将该点归入该聚类。

### 更新阶段

在K-Means聚类算法中，更新阶段紧随分配阶段之后，是算法的核心部分之一。在这一阶段，算法会更新每个聚类的中心点，以更好地反映当前分配给该聚类的数据点的“中心”或平均位置。

**更新阶段的步骤：**

1. 计算新的聚类中心：对于每个聚类，算法计算所有分配给该聚类的数据点的平均值，以此作为新的聚类中心。这通常涉及到计算每个维度上的均值。
2. 更新聚类中心：将每个聚类的中心点更新为上一步计算出的平均值。

### 迭代

在K-Means算法中，迭代过程是实现数据点有效聚类的核心。这个过程不断交替执行分配阶段和更新阶段，直至满足特定的停止准则。迭代的目的是逐渐调整聚类中心的位置，使得聚类内部的数据点尽可能相似（即内部距离最小化），同时使不同聚类之间尽可能区分（即聚类间距离最大化）。

**迭代步骤：**

1. 初始设置：首先进行聚类中心的初始化。
2. 分配阶段：将每个数据点分配给最近的聚类中心。
3. 更新阶段：更新每个聚类的中心点，使其成为该聚类内所有点的平均位置。
4. 重复：重复分配和更新步骤，直到满足停止准则。

**停止准则：**

1. 聚类中心不再变化：如果连续两次迭代中所有聚类中心的位置保持不变，则可以认为算法已经收敛。
2. 达到最大迭代次数：设置一个最大迭代次数以防止无限循环。当迭代次数达到这个预设值时，即使聚类中心仍在变化，也停止迭代。
3. 目标函数下降不足：如果目标函数（如聚类内方差之和）的下降量小于某个阈值，也可以停止迭代。

## K-Means算法收敛性

K-Means算法的收敛性是指在经过若干次迭代后，聚类中心和数据点的分配稳定下来的特性。这个算法通常能够相对快速地收敛，但它往往收敛到局部最优解而不是全局最优解，这意味着最终的聚类结果在很大程度上取决于初始聚类中心的选择。由于这种依赖性，不同的初始聚类中心可能导致算法收敛到不同的局部最优解。因此，选择一个好的初始化策略，比如K-Means++方法，可以显著提高找到更好局部最优解的概率。

数据的分布特性也对算法的收敛性有重要影响。对于一些特定类型的数据分布，比如非凸形状或者大小极不均匀的聚类，K-Means可能会面临收敛困难或者收敛到无意义的解。此外，在高维空间中，由于“维度的诅咒”，距离度量可能变得不那么有效，这也可能影响聚类过程和收敛性。

为了有效监控和管理K-Means算法的收敛过程，通常会设置一个最大迭代次数来避免无限循环。此外，可以通过监测连续迭代中聚类中心的变化情况来判断算法是否已经收敛。如果聚类中心在连续几次迭代后变化很小或者不再变化，就可以认为算法已经收敛。在实际应用中，还可以设置一个阈值来判断聚类中心变化的显著性，以此作为算法停止的依据。这样的收敛判定机制可以确保算法在达到稳定状态后及时停止，从而提高效率和结果的可靠性。

## K-Means算法优化

为了提高算法的稳定性和减少对初始值的敏感性，可以采用如K-Means++这样的聚类中心初始化方法，它通过一种特定的策略来选择初始聚类中心，以提高找到更好聚类的概率。

## K-Means应用场景

K-Means算法因其简单性和效率，在许多领域都有广泛应用。它主要用于将数据根据特征分组成若干个聚类，每个聚类中的数据项在某种程度上是相似的。在市场细分中，K-Means可用于识别具有相似购买行为或偏好的客户群体，助力企业更精准地定位市场和设计营销策略。在社交网络分析中，它能够帮助识别具有类似兴趣和互动模式的用户群组，为社交媒体营销提供支持。在生物信息学中，K-Means用于分析基因表达数据，帮助科学家探索不同基因的功能和关联。图像分割也是K-Means的一个重要应用，通过对像素点进行聚类，可以将图像中不同颜色或纹理的区域区分开来。此外，K-Means在文档聚类、财务分析、天文数据分析、供应链管理、教育领域和医疗健康数据分析中也有广泛应用，例如用于识别具有相似特征的文档、股票、天体、仓库位置、学生学习模式或患者病症。这些应用体现了K-Means在处理各种数据聚类问题时的通用性和实用性。

# 实验步骤

## 加载数据集

### 数据集特征：

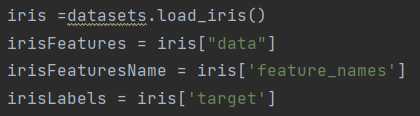
鸢尾花数据集是机器学习中常用的数据集之一，包含150个样本，每个样本有四个特征：花萼长度、花萼宽度、花瓣长度和花瓣宽度。

### 数据结构

这些样本分为三个类别，每类50个样本：山鸢尾（Iris Setosa）、杂色鸢尾（Iris Versicolour）和维吉尼亚鸢尾（Iris Virginica）。

### 数据加载

使用sklearn.datasets.load\_iris()函数加载数据。



## 实现K-Means算法

选择K值：确定聚类数量K为3，对应三种鸢尾花。

初始化聚类中心：可以随机选择或采用K-Means++方法初始化聚类中心。

分配数据点：根据距离度量，将每个数据点分配到最近的聚类中心。

更新聚类中心：计算每个聚类的均值并更新为新的聚类中心。

迭代过程：重复分配和更新步骤，直到聚类中心稳定或达到最大迭代次数。

## 观察聚类结果

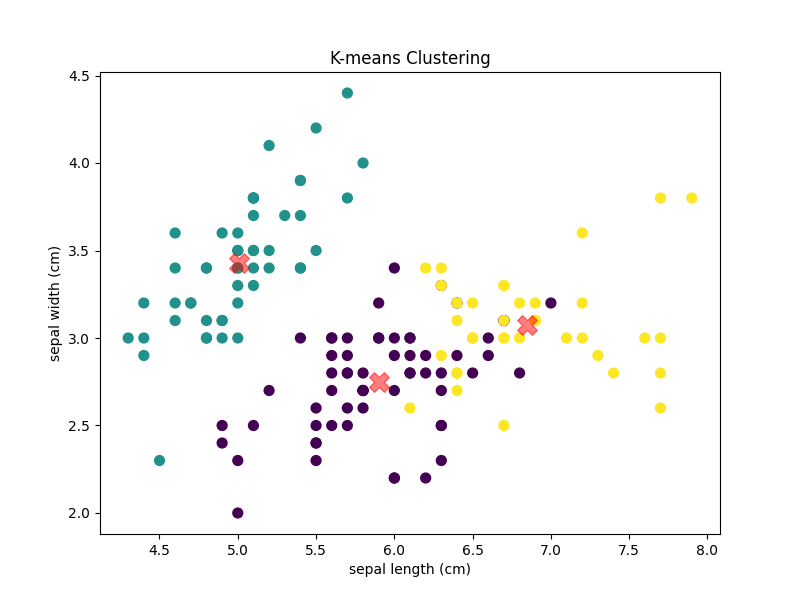
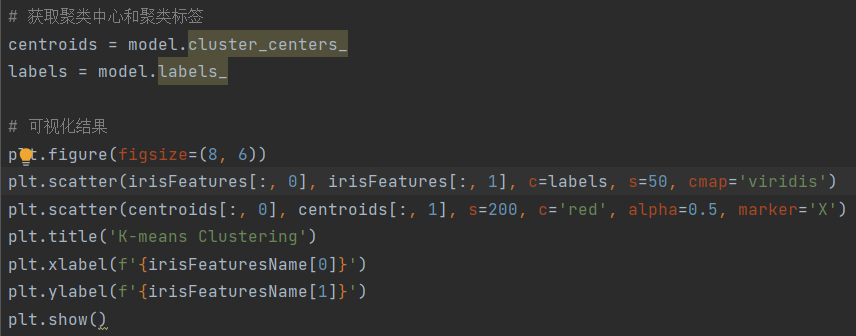


图 1 K=3



# 实验结果分析

## 研究K值的影响

### 实验结果

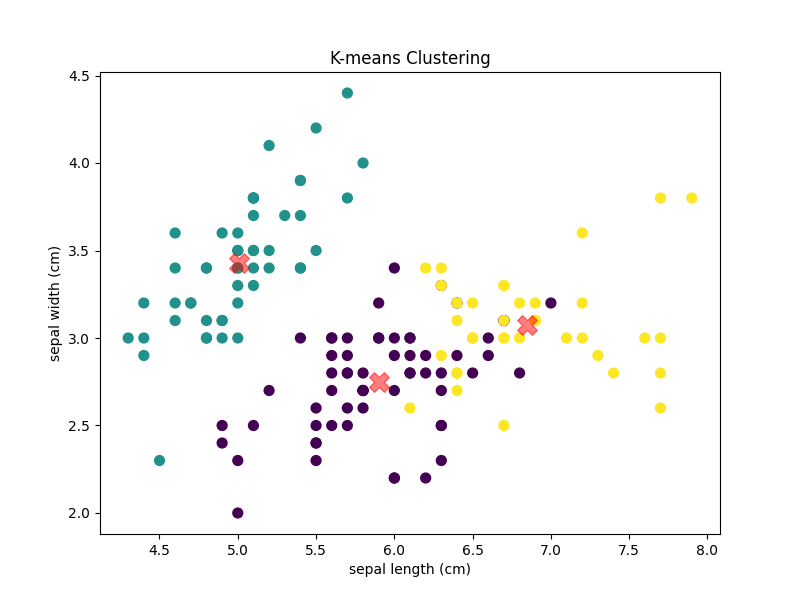


图 2 K=3

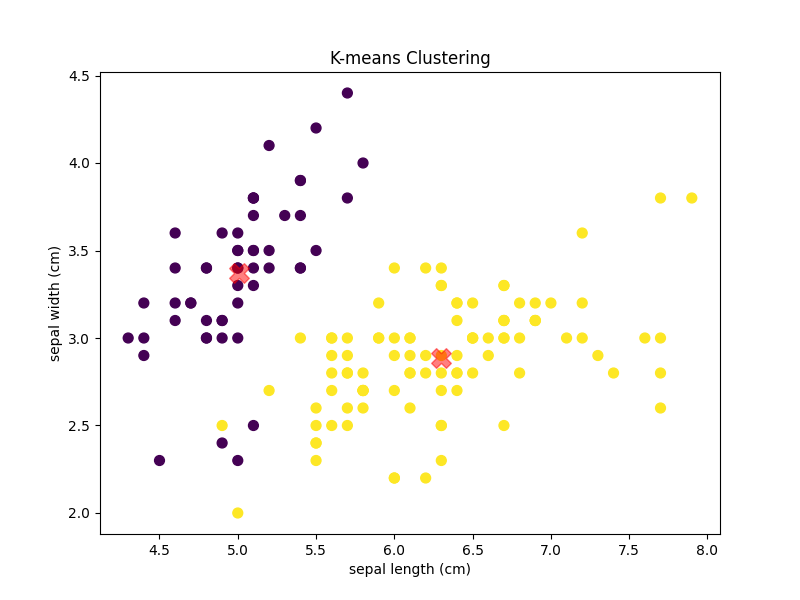


图 3 K=2

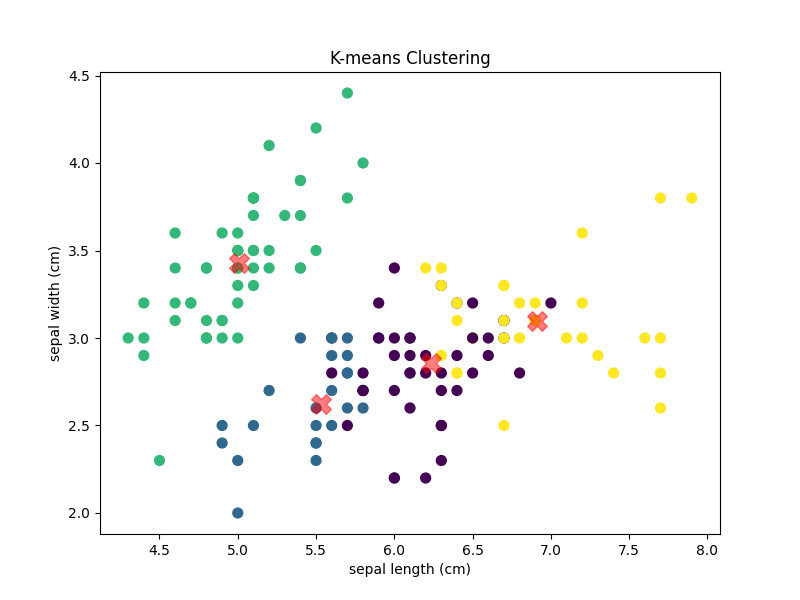
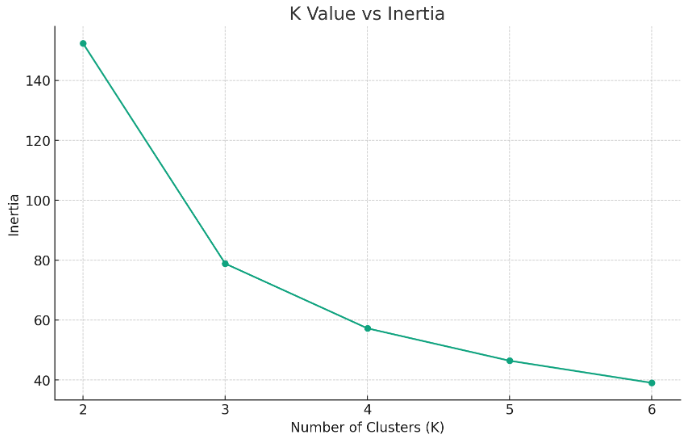


图 4 K=4

### 分析

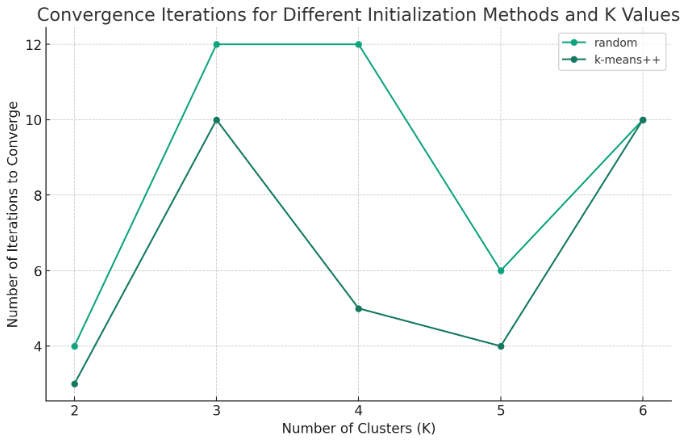


通过对不同K值的K-Means聚类结果进行分析，可以看到K值与簇内误差平方和（Inertia）之间的关系图。这里的Inertia指的是聚类内部数据点与其聚类中心的距离之和，通常用来衡量聚类效果的好坏。

**从图中可以观察到以下几点：**

1. Inertia随K值增加而减少：随着聚类数量K的增加，每个聚类包含的数据点减少，因此每个数据点到其聚类中心的距离也减少，导致总的Inertia下降。
2. “肘部”方法：理想的K值通常是Inertia下降速率突然变缓的点，即所谓的“肘点”。在这个点之后，增加更多的聚类中心会导致Inertia的降低幅度减少，这意味着每增加一个聚类中心所带来的信息量相对较少。
3. 选择最佳K值：根据图表，可以尝试识别这个“肘点”，以确定最佳的聚类数量。对于鸢尾花数据集，根据数据分布的实际情况，原本有3个类别，但最佳的K值可能因初始化方法和数据特性而有所不同。

## 研究初始聚类中心的影响



图表展示了使用随机初始化和K-Means++初始化方法在不同K值下K-Means算法收敛所需的迭代次数。通过对比这两种初始化方法，我们可以观察到它们对算法收敛性的影响。

**从图表中可以看出：**

1. 收敛迭代次数差异：随着K值的变化，两种初始化方法所需的迭代次数也相应变化。这提供了一个直观的视图来比较不同初始化方法在不同聚类数量下的收敛速度。
2. 初始化方法的差异：不同的初始化方法可能会导致不同的收敛迭代次数。在某些K值下，这种差异可能更为明显。
3. 收敛速度的影响：图表中每种初始化方法所需的迭代次数反映了不同初始化策略对算法收敛速度的影响。

通过这种分析，我们可以更好地理解不同初始化方法对K-Means算法收敛性的影响，从而为实际应用中的聚类算法选择提供依据。收敛迭代次数作为衡量算法效率的一个指标，较低的值通常表示更快的收敛速度。

# 实验结论

在本次实验中，我们首先加载了鸢尾花数据集，并观察了其特征，包括数据集的结构和样本特征。接着，我们实现了K-Means算法，运行该算法并观察了聚类结果。在此基础上，我们深入研究了初始聚类中心的设置对K-Means算法收敛性的影响，以及参数K对聚类结果的影响。最后，我们对聚类算法的优缺点进行了分析。

**实验结果表明：**

## 数据集特征对聚类结果的影响

鸢尾花数据集包含150个样本，每个样本有四个特征。这些特征对于识别不同种类的鸢尾花至关重要，算法能够有效地根据这些特征将样本分为不同的群组。

## K-Means算法的有效性

实验中实现的K-Means算法能够有效地对数据进行聚类。通过调整参数K，我们可以控制聚类的数量，观察到不同K值下聚类结果的变化。

## 初始化方法的影响

初始聚类中心的选择对算法的收敛速度和最终聚类结果有显著影响。相较于随机初始化，K-Means++初始化方法在大多数情况下提供了更快的收敛速度和更高的聚类质量。

## 参数K的重要性

K值的选择对聚类结果有决定性影响。不适当的K值可能导致聚类结果的不准确或过度细分。

## 聚类算法的优缺点

1. Means算法简单、易于实现，对大数据集有良好的处理效率。然而，它也存在一些局限性，如对异常值敏感，可能收敛到局部最优解，且需要预先指定聚类数K。

综上所述，本次实验不仅展示了K-Means算法在实际数据集上的应用，也突显了选择合适的初始化方法和K值对于获得高质量聚类结果的重要性。此外，我们也认识到了K-Means算法在处理特定类型数据时的局限性，这为未来的研究和应用提供了有价值的见解。