深 圳 大 学 实 验 报 告

实验课程名称：最优化方法

实验项目名称：**k-Means 聚类实验**

学院：计算机与软件学院　专业：软件工程（腾班）

报告人：黄亮铭　学号：**2022155028**　班级：腾班

同组人：无

指导教师：李炎然

实验时间：**2024 年 10 月 8 日-2024 年 10 月 24 日**

实验报告提交时间：**2024 年 10 月 24 日**

教务处制

实验报告包含内容

一、 实验目的与要求

1. 熟练掌握 k-Means 方法对手写数字图像进行分类；
2. 用 Matlab 编写代码，熟悉其画图工具，进行实验，并验证结果；
3. 锻炼数学描述能力，提高报告的叙述能力。

二、 问题

手写数字图像数据分类问题：文件train_images.mat包含大小为28×28的手写数字图像，共60000张；文件train_labels.mat是其对应的数字标签。文件数据的具体读写和数据格式，请参考附件DataRead.m文件。实验要求对手写数字图像进行聚类，并讨论其性能：

(MNIST DATABASE下载网址: http://yann.lecun.com/exdb/mnist/)

(1) 对train_images.mat的前100张手写数字图像进行聚类，共10类；

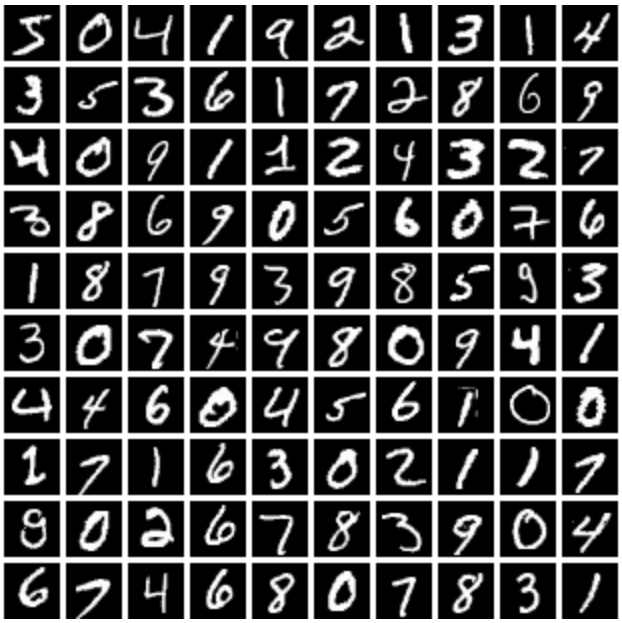


图1. 文件前100张手写数字图像

(2) 对train_images.mat的前1000张手写数字图像进行聚类，共10类；

(3) 根据实际情况，讨论k-Means能对多少张手写图像进行聚类，性能如何？

三、模型建立及求解

解决问题思路，模型建立、性能分析，存在问题等方面进行阐述；代码不要放在报告里面，可以作为附件提交！

3.1 解决问题思路

本次实验要求解决手写数字图像数据分类问题。为了更改地解决这个问题，我将问题拆分成如下步骤：选择数据集、算法选取与模型建立、模型性能分析、存在的问题以及优化方法。

每个步骤需要完成的具体任务如下：

1. **选择数据集：**我们选取 MNIST 数据集。MNIST 数据集是机器学习领域中一个经典数据集，由 60000 个训练样本和 10000 个测试样本组成，每个样本都是一张 28 * 28 像素的灰度手写数字图片。本次实验使用到 MNIST 的手写体的图像数据（train_images.mat）和图像数据对应的标签（train_labels.mat）。
2. **算法选取与模型建立：**我们选取经典的机器学习算法 K-means 建立模型，对数据进行聚类分析。
3. **模型性能分析：**我们从两个方面评估模型的性能：1）模型的训练时间；2）模型的正确率。对于 1），我使用迭代次数来代替。因为在算法时间复杂度相同的情况下，影响因素最大的当属迭代次数，其余影响均为常数级影响，可以忽略。对于 2），因为实际标签值和聚类标签值不一匹配，所以我会统计每类中出现最多的实际标签值作为该类的预测标签，并根据如下公式计算正确率 $正确率 = \frac{预测标签数量}{总标签数量}$。
4. **存在问题：**通过对上述步骤的分析，我们可能得出一些问题。然后我们会尝试解决上述步骤中存在的问题。
5. **优化方法：**针对 4 中提出的问题，如果能找到解决办法，将会在该模块中提出。

3.2 选择数据集

实验提供的手写体数据集文件以 mat 格式存储，matlab 可以通过双击文件导入或者通过 load 命令导入。

导入完成后，我们需要测试数据是否能正确使用。为此，我们使用实验提供的读取图像数据并显示的代码进行验证。验证结果如下图所示，说明数据可以被正确使用。

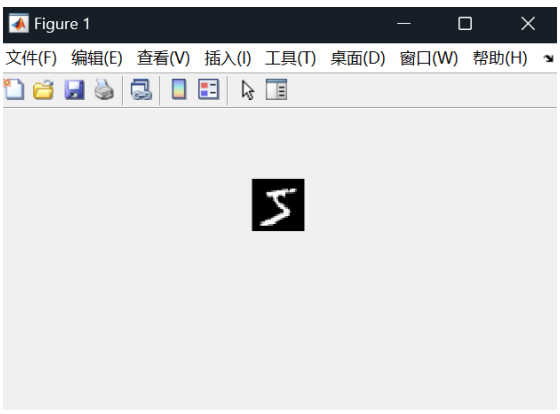


图 1：验证结果

在选择并导入数据集之后，为了方便后续操作，我们还需要对导入的数据进行预处理。直接导入的数据为三维张量（如图 2 所示），第一维和第二维代表每一张图像的长宽像素，第三维代码图像数据的总数量。我们需要将一张图像拉成一维的向量（如图 3 所示）。

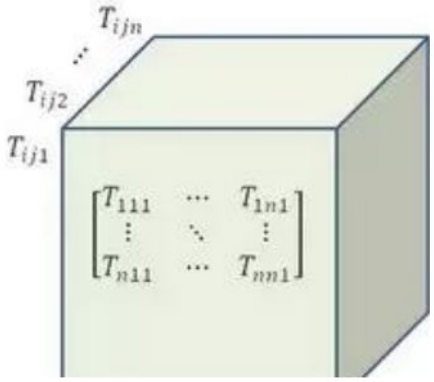


图 2：三维张量

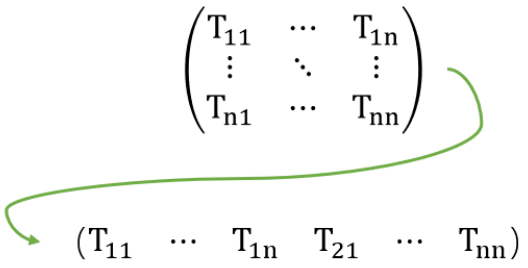


图 3：变换示例

3.3 算法选取与模型建立

我们选择 K-means 聚类算法建立模型，对问题进行求解。

3.3.1K-means 聚类算法原理

K-means 聚类是一种无监督学习的聚类算法，用于将数据集中的样本分成 K 个类别。它的原理是将样本分配给 K 个类别，使得每个样本都属于与其最近的类别中心点所代表的类别。然后，计算每个类别的新的中心点，重复这个过程直到类别的中心点不再发生变化或达到预定的迭代次数。

上述表达可以转化为下面的优化问题求解：给定样本集$D = \{x_1, x_2, \ldots\ldots, x_n\}$，K-means 算法针对聚类所得类划分$C = \{C_1, C_2, \ldots\ldots, C_k\}$最小化平方误差

$$E = \sum_{i=1}^{k} \sum_{x \in C_i} ||x - \mu_i||_2^2$$

其中$\mu_i = \frac{1}{|C_I|}\sum_{i\in C_i} x$ *是类*c_i*的*均值向量。

3.3.2K-means 聚类算法流程

K-means 的算法流程与 K-means 聚类算法原理中的文字描述基本一致，具体的步骤如下所示。

① 确定聚类的类别数量 K 以及训练样本数量。
② 从数据集从读取设定的训练样本数量的数据。
③ 从读取的数据中随机选取 K 个样本作为初始聚类中心。
④ 对数据集中的每个样本x_i分别计算它到 K 个聚类中心的距离，并将其分配到距离最短的聚类中心对应的类中。
⑤ 对于每个类，重新计算聚类中心$\mu_i = \frac{1}{|C_I|}\sum_{i\in C_i} x$。
⑥ 重复步骤④和步骤⑤，直到聚类中心变化小于预设的阈值或者迭代次数超过设定值。

3.3.3 计算正确率

统计每类中出现最多的实际标签值作为该类的预测标签，并根据如下公式计算正确率*正确率* $= \frac{预测标签数量}{总标签数量}$。

3.3.4 小结

实现 K-means 聚类算法和正确率的计算，以及实现 3.1 中提到的对原始数据预处理，我们就可以建立模型并对实验中提出的问题进行求解。

3.4 求解问题

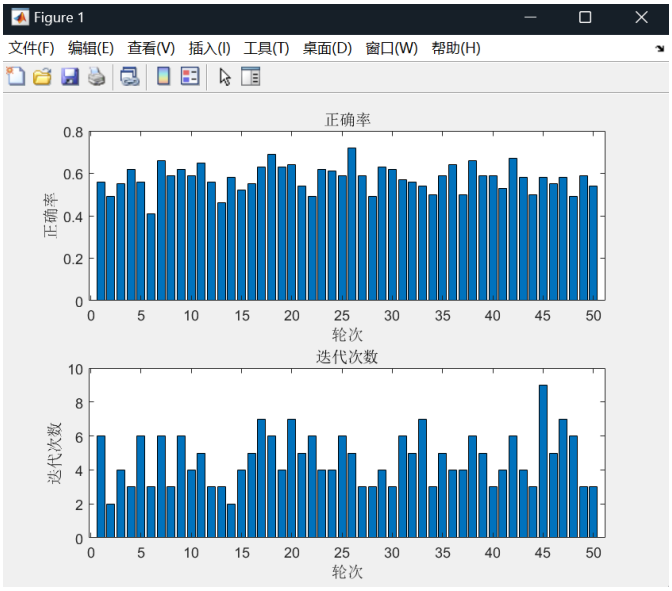
在这个部分，我们需要解决实验提出的前两个问题，即：1）对 train_images.mat 的前 100 张手写数字图像进行聚类，共 10 类；2）对 train_images.mat 的前 1000 张手写数字图像进行聚类，共 10 类。对于这两个问题，我们只需要修改读取的样本数量即可完成。

3.4.1 对 train_images.mat 的前 100 张手写数字图像进行聚类

方案：为了保证结果的准确性，我们对算法进行 50 次循环测试，对结果取平均值。

结果：50 次循环测试结果为：平均迭代次数为 19.72 次，平均正确率为 0.552。

将结果可视化后如下图所示。



4：问题 1 结果可视化

取前 5 次测试结果生成表格如下图所示。

测试轮次	1	2	3	4	5	均值
迭代次数	3	8	4	6	5	5.2
正确率	0.62	0.57	0.53	0.61	0.61	0.588

图 5：前 5 次测试结果

3.4.2 对 train_images.mat 的前 1000 张手写数字图像进行聚类

方案：为了保证结果的准确性，我们对算法进行 50 次循环测试，对结果取平均值。

结果：50 次循环测试结果为：平均迭代次数为 19.72 次，平均正确率为 0.552。

将结果可视化后如下图所示。

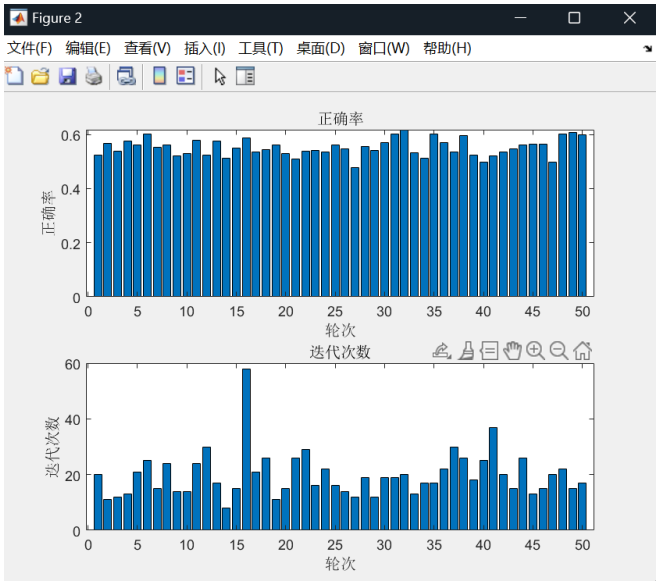


图 6：问题 2 结果可视化

取前 5 次测试结果生成表格如下图所示。

测试轮次	1	2	3	4	5	均值
迭代次数	20	11	12	13	21	15.4
正确率	0.52	0.56	0.54	0.58	0.56	0.554

图 7：前 5 次测试结果

3.5 性能分析

由 3.4 的运行结果可知，K-means 聚类算法在手写数字识别上应用的效果并不是很好，正确率在 50%-60%之间浮动。并且在样本容量为 1000 时，算法的迭代次数的浮动太大，在 10 到 60 之间浮动，说明算法性能不稳定。

接下来我们解决实验中提出的第三个问题：根据实际情况，讨论 k-Means 能对多少张手写图像进行聚类，性能如何？

为了解决问题 3，我们进行如下设置：初始样本容量为 1000，步长为 1000，最终样本容量为 10000。对于每种样本容量，我们都进行 50 次循环测试，记录了迭代次数均值、正确率均值以及运行时间均值。

结果如下表所示。

样本容量	平均迭代次数	平均正确率	平均运行时间
1000	18.98	0.55	0.45
2000	27.58	0.58	1.21
3000	30.94	0.59	2.06
4000	38.9	0.56	2.98
5000	39.12	0.57	4.54
6000	42.94	0.56	4.94
7000	43.06	0.56	6.75
8000	46.12	0.58	7.02
9000	52.26	0.57	8.74
10000	54.52	0.58	10.11

图 8：性能测试结果

由图 8 可以看出，从正确率的角度来说，随着聚类的样本数的增加，K-means 算法的平均正确率仍然在 55%-60 之间波动，说明算法的正确率与样本数的关系较小。从运行时间的角度来说，随着聚类的样本数的增加，平均运行时间也不断增加。

3.6 存在问题

在预处理数据集时，为了方便后续处理，我们将二维的图像数据拉成了一维的向量，这会使得图像的空间信息丢失。这可能是正确率下降的原因之一。

K-means 算法假设各个簇的大小、形状和密度相似，如果数据集中的簇具有类似的分布特征，K-means 能够产生较好的聚类结果。数据集中的数字可能并不是均匀分布的，不同的数字可能出现频率不同。此外手写数字的形状有的即使是同一个数字，其区别也非常大。

K-means 算法的初始聚类中心为随机选择，这是造成算法性能（迭代次数和正确率）不稳定的核心原因。如果初始聚类中心之间非常接近，则算法性能会急剧下降；如果初始聚类中心之间的距离较大，则算法性能表现会比较好。

3.7 优化方法

为了解决 K-means 算法随机选取初始聚类中心造成的算法性能不稳定的问题，我们可以使用 K-means++解决该问题。

K-means++算法与 K-means 算法的主要区别在于初始化聚类中心。

K-means++算法的初始化聚类中心的流程为：

① 从训练样本中随机选择第一个点作为第一个聚类中心。

② 对于每个样本，计算它与当前已确定的聚类中心的距离，选择与已确定聚类中心距离最大的样本作为新的聚类中心。

③ 重复步骤②，直到选出 K 个聚类中心。

步骤②的问题可以看成是最小值最大问题。对于每一个样本，需要计算与已确定聚类中心的最近距离，然后再所有样本中找到最大的最近距离。

3.7.1 优化后的性能

我们进行如下设置：初始样本容量为 2000，步长为 2000，最终样本容量为 10000。对于每种样本容量，我们都进行 50 次循环测试，记录了迭代次数均值、正确率均值以及运行时间均值。

结果如下表所示。

样本容量	平均迭代次数	平均正确率	平均运行时间
2000	26.08	0.59	1.16
4000	31.94	0.58	2.77
6000	40.24	0.58	5.10
8000	40.82	0.58	6.80
10000	46.60	0.59	9.52

图 9：性能测试结果

由图 8 和图 9 对比可以看出，K-means++的正确率较 K-means 的正确率略微升高，并且更加稳定。同时，迭代次数（运行时间）更加稳定且较小，说明初始聚类中心选择策略的正确性。

四、小结（可含个人心得体会）

1. 在本次 K-means 聚类实验中，我们成功地对手写数字图像数据进行了分类。
2. 通过本次实验，我们不仅加深了对 K-means 聚类算法的理解，也提高了使用 Matlab 进行数据处理和算法实现的能力。
3. 我们通过实验发现，K-means 算法在手写数字识别上的正确率并不高，平均在 55%-60%之间。这可能与我们丢失了图像的空间信息有关，也可

能是因为手写数字的多样性和复杂性超出了 K-means 算法的处理能力。

4. 在性能分析中，我们观察到随着样本数量的增加，算法的迭代次数和运行时间也随之增加，但正确率并没有显著提升。这表明 K-means 算法在处理大规模数据时可能会遇到效率问题。

指导教师批阅意见： 成绩评定： 指导教师签字：李炎然 2024 年　10　月　　日
备注：

注：1、报告内的项目或内容设置，可根据实际情况加以调整和补充。

2、教师批改学生实验报告时间应在学生提交实验报告时间后 10 日内。