队伍编号	MC2212325
题号	A

基于大规模指纹图像的快速检索算法的设计及实现

摘要

问题一中，分三步构建了指纹图像检索模型框架。通过对相同指纹不同细节点的对比，发现指纹细节点存在**坐标偏移但结构相似**的特征（见文内图 ）。首先，采用**聚类**的方法，得到每条指纹**细节点坐标**(x,y)、**方向**θ的类中心点。在此基础上，采用最短路径的算法，得出**坐标**(x,y)、**方向**θ**类中心的最短距离与其他特征指标。**然后，**二次聚类**得到与目标指纹相似的一类数据。并且按照相似程度从高到底排序，得到**目标指纹的“同一”小规模子集**。最后，根据题目要求分析算法的**时间复杂度、空间复杂度**，并计算得出相应结果。

问题二中，要求采用问题一所提出的检索方法，将 TZ_同指.txt 中全部 500 个匹配对子（合计 1000 行）作为查询图像逐一在 TZ_异指.txt 与 TZ_同指.txt 组成的数据集中进行检索，过滤掉数据集 80%、90%、95%、97%的图像后，在剩下的小规模子集中将目标指纹进行检索，并计算指纹匹配的穿透率。本文首先对每一个指纹样本内的细节点进行 **C 均值聚类**，并找出每一类别的类中心；其次采用**禁忌算法（TS 算法）**寻找每条指纹数据中类中心点之间的最短路径，并计算出类中心点的每一条连线的**长度**，从而**构造出新的指纹特征**，接下来对于所有样本数据（11000 条数据）进行 **K 均值聚类**，并找出每一类别的类中心，计算每个目标指纹与类中心之间的距离，判断目标指纹所属的类别，实现**初次匹配**，再计算目标指纹与所属类别内每一条指纹之间的**欧式距离及 Pearson 相关系数**并分别排序，寻找符合过滤比例的指纹，实现**二次匹配**，从而找到目标指纹的检索库，通过计算穿透率，**评估**欧式距离及 Pearson 相关系数各自筛选出的指纹检索库的**有效性**，找到性能优良的指纹检索方法。

问题三中，要求采用和问题二相同的方法，针对 TZ_同指 200_乱序后 Data.txt 数据文件在 TZ_异指.txt 和 TZ_同指 200_乱序后_Data.txt 两个文件所组成的数据集中进行检索，并给出检索结果。因此，本文首先采用 **C 均值算法**对数据集中每一条指纹数据的细节点进行聚类分析，找出每一枚指纹中每一类的类中心，并采用**禁忌算法（TS 算法）**寻找类中心点之间的**最短路径**，接下来计算出类中心点之间**每一条连线的长度**，以此**构造出新的指纹特征**，然后**采用 K 均值聚类**对数

据集的所有指纹进行聚类，并**判断目标指纹所属的类别**，实现目标指纹的**初次匹配**，再计算目标指纹与所属类别内每一枚指纹之间的**欧式距离**，筛选出符合过滤水平的指纹检索库，实现目标指纹的**二次匹配**。

问题四中，对本文构建的指纹快速检索模型进行**评价**，并针对 97%以上过滤水平提出**优化方案**。首先是模型评价，本文具有构造指纹**样本特征变量**、*KMeans* 算法**分类快**、预防“同一”指纹样本丢失的优点，也存在考虑不够周全、算法选择单一的问题。然后是提出优化方案，结合本文模型框架以及研究文献，以快速高效搜索为目标，从**检索精度、检索时间和内存占用**三个角度提出了改进策略。

关键词： 指纹检索；C 均值聚类；最短路径；禁忌算法；K 均值聚类

目录

1 问题重述

1.1 问题背景

21 世纪以来，随着信息化程度的日渐普及，计算机模式识别及图像处理等相关技术不断进步，使得生物识别技术的应用范围越来越广泛。在所有可用于识别的生物特征中，指纹是应用范围最为广泛的生物特征。指纹具有唯一性、永久性、可靠性高及易采集性等优良特性，已经成为个人身份认证最有效的手段之一，是具有法律效力的生物特征[1]。

指纹识别系统的主要功能体现在两个方面：验证和辨识。一般在国际上最常见的验证主要是将指纹的细节点记为三元存储格式(x,y,θ)，与已经记录的指纹进行一对一的匹配，从而确定身份的过程。辨识则是指将采集的指纹与已经入库的指纹逐一对比，在其中找出能匹配所采集的指纹的过程。随着社会中指纹识别系统的不断应用，用于识别人员身份的指纹数据库规模在迅速上升，居民身份证的指纹库甚至达到了亿人级别，这对指纹图像检索而言，是一项前所未有的挑战。

虽然现阶段的指纹识别技术日益成熟，优秀的指纹识别算法不断被提出，但这些算法在识别速度和准确性方面仍然存在一些问题，不能满足某些高要求的应用场合。尤其是在大规模的指纹库的分类检索、低质量指纹的处理等领域，指纹识别系统的发展仍然受限。因此，如果能够进一步优化指纹识别算法，在上述领域取得一定的突破，使得指纹识别速度和准确性有明显的提升，将会极大地拓展指纹识别的应用领域，同时带动相关学科的发展。

1.2 问题提出

围绕大规模指纹图像检索的模型与实现，本文依次解决如下问题：

问题一：指纹快速检索方法与模型框架

指纹库中数据量过大，如何利用指纹图像的细节点特征，对指纹样本数据进行快速的检索。并且，在这过程中避免筛除掉“同一”指纹的机制。同时，给出图像检索模型框架及实现方法，对检索方法的时间复杂度、空间复杂度进行分析与计算。

问题二：指纹快速检索方法的验证

针对问题 1 中所提到图像检索模型框架及实现方法，将数据文件中提供的具有“同一”关系的全部指纹对子（TZ_同指.txt）作为查询图像在指纹数据集中进行验证。在分别过滤掉数据集 80%、90%、95%、97%的图像后，将 TZ_同指.txt 中 500 个匹配对子的数据逐一在剩下的小规模子集中进行检索，并计算穿透率（该问题所需数据集为 TZ_异指.txt 和 TZ_同指.txt 两个文件中 10500 枚指纹，即 11000

条数据）。

问题三：指纹快速检索方法的运用

针对 TZ_同指 200_乱序后_Data.txt 数据文件采用和问题 2 相同的方式进行检索，分别保存过滤掉 90%，95%，97%的指纹图像（TZ_异指.txt）后，给出 200 枚图像（TZ_同指 200_乱序后_Data.txt）的检索结果。

问题四：指纹快速检索方法的评价与优化

利用问题 2 的数据验证、优选出最佳的检索方法，并且评价你们所考虑过的模型及技术路线的优缺点。当最高的筛选量为 97%，检索精度、检索时间及内存占用等方面有什么更好的改进策略或者会尝试什么新的检索方法？

2 问题分析

2.1 问题一的分析

首先需要对**指纹细节点的特征**进行识别，找出每枚指纹自己的特征点。而这个特征点可能是相似的坐标、方向、或者指纹结构，所以我们采用**聚类**的方法，将相似指纹聚类在一起。聚类的类别越大，就越能避免“同一”指纹的丢失。指纹采样时坐标、方向会发生偏移，但是**指纹的结构基本不会发生变化**。基于此，我们可以计算出指纹**类中心的最短距离**，从而得到每个指纹**独有的特征结构**。然后，将计算出的最短距离等**指标二次聚类**，得到与目标指纹**相似的一类指纹**。将其按照相似程度从高到低**排序**，就能按照不同的过滤程度得到最终的**“同一”小规模子集**。最终，分析算法的**时间复杂度、空间复杂度**，并计算得出相应结果。

2.2 问题二的分析

该问题要求采用问题 1 所提出的检索方法，将 TZ_同指.txt 中全部 500 个匹配对子（合计 1000 行）数据逐一在 TZ_异指.txt 与 TZ_同指.txt 组成的数据集中进行检索，并计算在不同过滤水平下，指纹匹配的穿透率。由于赛题中仅提供了每一枚指纹图像的 ID、所提取到的细节点个数及每个细节点的三元信息（x 轴像素坐标、y 轴像素坐标及细节点方向），（x, y）和 θ 能够反应每一个细节点的不同信息，且每一条指纹数据的细节点个数不一致，导致**每一条指纹数据的长度不相等**，此外，数据集所提供的可用指标较少。因此，为了解决上述问题，本文首先**采用 C 均值聚类**对每一个指纹细节点的像素坐标（x, y）及角度 θ **分别进行聚类**（每一个指纹聚类数目相同），找出每个指纹样本中每一个类别的类中心；由于每一枚指纹内部**细节点之间的最短路径**相对固定，因此细节点之间的**最短路径**能够**大致**反映出**指纹的内部走势**，因此，在每个指纹内部采用**最短路径算法**，寻

找连接细节点类中心的**最佳路线**，并计算每一条指纹数据中**相互连接的类中心点之间的距离**，通过类中心之间的距离**构建出新的指纹样本特征**；最后基于所构建的指标，采用 **K 均值聚类初次判断出**所需匹配的指纹所属类别，再计算出所需匹配的指纹与其所属类别内每个指纹之间的**欧式距离或 Pearson 相关系数**，从而过滤掉一定比例的样本，并在留下的少量样本中，对所需匹配的指纹进行检索，采用穿透率对该方法的性能进行评估。

2.3 问题三的分析

该问题要求采用和问题 2 相同的方法，将 TZ_同指 200_乱序后 Data.txt 数据文件中的指纹逐一 TZ_异指.txt 和 TZ_同指 200_乱序后_Data.txt 两个文件所组成的数据集中进行检索，并给出检索结果。TZ_同指 200_乱序后 Data.txt 文件中只给出了 200 枚指纹的细节点个数及每个细节点的三元信息，但是每枚指纹所提取的**细节点个数不同**，即每一行数据的长度不同，为了解决上述问题，此处首先对于每一条指纹样本内的细节点分别按照像素坐标（x, y）和方向θ进行 **C 均值聚类**，并找出每一类别的**类中心**，采用**禁忌算法（TS 算法）**寻找每一条指纹内部类中心之间的**最短路径**，基于类中心之间的连线计算出**连线的长度**，从而**构造出新的指纹特征**；最后采用 **K 均值聚类**进行**初次匹配和二次匹配，**过滤掉一定比例的样本之后，筛选出与所需匹配的指纹高度相似的指纹检索库。

2.4 问题四的分析

本题首先需要对构建的指纹快速检索方法模型及技术路线进行评价，需要从指标构建、模型算法、模型能力验证等方面进行仔细分析，找出其具备的优缺点。其次，问题 2 和问题 3 中涉及的最高指纹过滤比例为 97%，需要思考当进一步提高过滤比例时，如何制定模型的优化方案或者可以采用什么样的新检索方法，应该从检索精度、检索时间及内存占用等方面入手。

3 模型假设

假设 1：题目中给出的指纹数据真实可靠，不存在采集的人工误差

假设 2：同指指纹的采集数据至少存在一对匹配细节点

假设 3：每对匹配细节点具有相似结构且误差较小

假设 4：聚类后的类中心能够表示手指指纹的特征值

假设 5：细节点坐标、方向的类中心最短路径达到最优

假设 6：同指指纹细节点的位移差、方向差近似线性相关

4 符号说明

符号	说明
x	指纹细节点的横坐标
y	指纹细节点的纵坐标
θ	指纹细节点的方向
K	聚类数量
n	检测样本的指纹个数
$f_i(n)$	待检测目标的循环工作量函数
ε	误差阈值
c_j	第 j 个聚类中心
U_j	隶属度矩阵

5 指纹快速检索方法与模型框架

5.1 问题描述

问题一要求对大规模指纹检索，并且满足以下三个条件：（1）在进行检索过程中应该尽量避免删除掉同一指纹；（2）要给出完整、清晰的指纹图像检索模型框架和实现方法；（3）在该模型和方法的基础之下，计算出检索所需要耗费的时间复杂度、空间复杂度。我们从以上三点分别对问题展开解答。

5.2 指纹特征检索

从第一个条件中我们可以看出，为避免删掉同一指纹，我们首先需要做的便是对指纹样本的细节点进行提取。而在现有文献中，对指纹样本的细节点的提取主要是将指纹的特征分为全局特征、局部特征，从这些特征中进行匹配。全局特征通常是指，指纹中心区所形成的独特结构的指纹，包含：纹路图案、脊线、谷线、模式区、核心点、三角点等。局部特征是指指纹中的细节点，包括指纹纹线上的中断、分叉、端点。

图 5.1　指纹的端点及分叉点

上述方法主要适用于指纹图片上点的检索，但是在本题中我们不再需要从图像中提取出细节点。已知所有细节点的坐标(x,y)、方向θ，求解指纹样本的特征。因此，我们在提出相应假设的前提下，基于前人全局特征筛选的研究方法，并进行相应的方法创新。

首先选取 500 个指纹比对好后的数据画图，初步观察数据之间的一一对应关系。如下图所示：

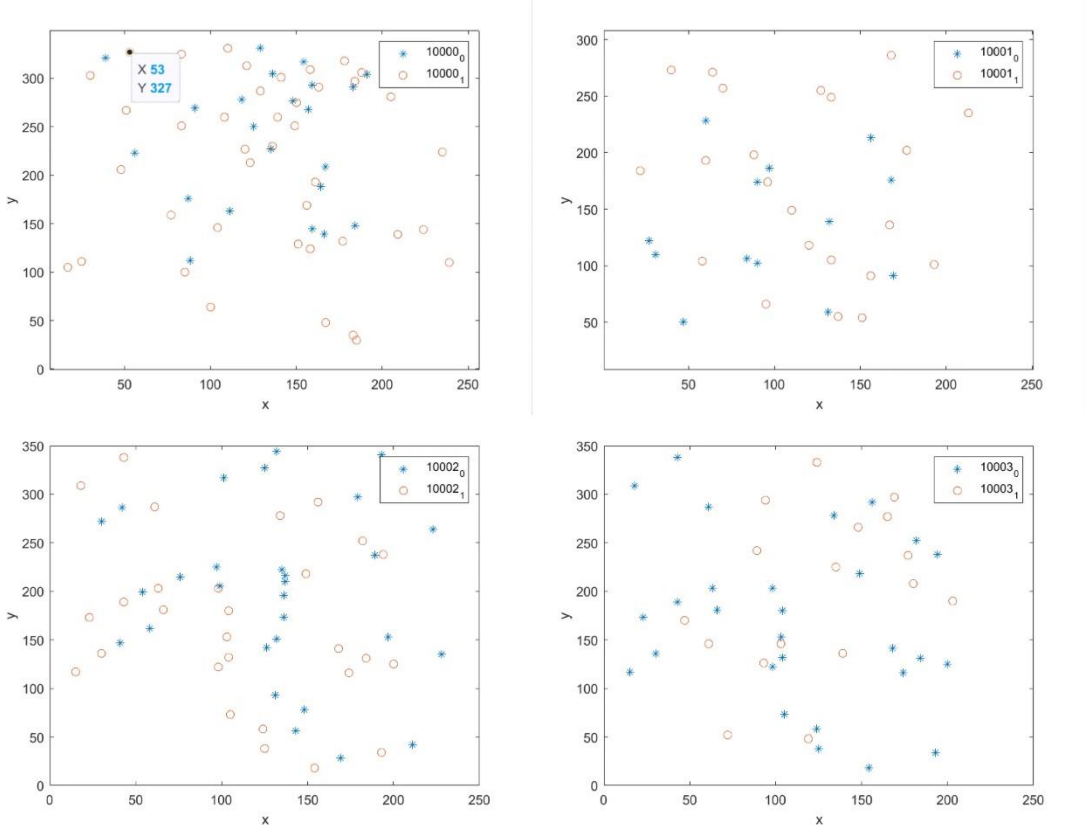

图 5.2　同个指纹两组数据对比

从上图中可以看出，10000_0 与 10000_1 之间除边缘的值以外，所取的样本点基本一致，图像吻合。10001_0 与 10001_1 之间出入较大，主要是由于指纹位置的偏离，导致较少指纹之间存在吻合情况。10002_0 与 10002_1 之间有较少吻合得点，但是可以看出指纹数据的基本趋势一致。10003_0 与 10003_1 之间也是如此。

对同样的指纹取样，样本的坐标可能一致或存在偏差，样本细节点的结构基本一致。因此，在对指纹的特征点进行提取时，不能仅考虑坐标的一致性，应该考虑数据之间的结构性问题。所以我们采取聚类的方法，将相似位置、相似结构的指纹坐标点聚类在一起，以类中心为依据，判断是否为同一指纹。

同时，在聚类分析中聚类结果的好坏会受到聚类数目的直接影响。因此聚类数目的选择是决定聚类质量的关键因素之一，也是聚类有效性分析的主要任务。如果聚类数目较少时，则可能出现下图所示的问题：

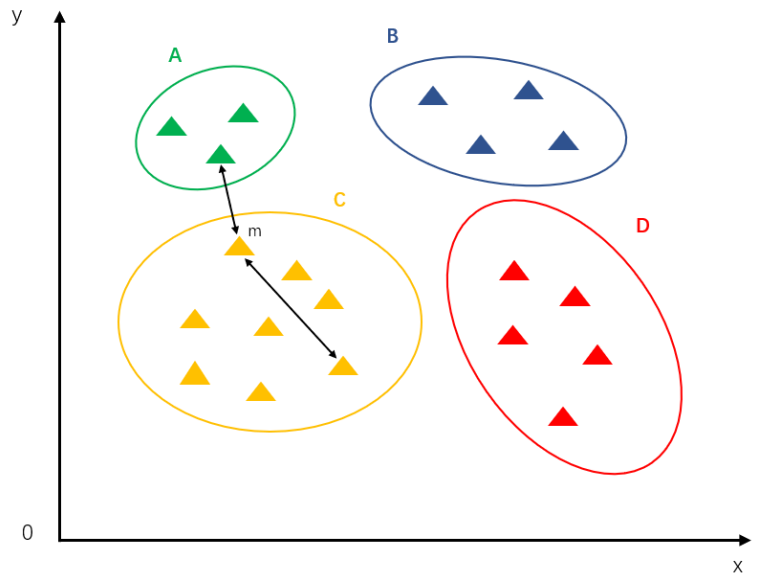

图 5.3　聚类分析图

如上图所示，A、B、C、D 分别表示所聚成的四类，点 m 为属于 C 类的某一点，但点 m 距离 A 类边缘上一点的距离可能比 C 类中下半部分更近，若直接选择 C 类中的样本作为 m 点的相似样本而过滤掉其他类别样本，非常容易导致“同一”指纹误删。总之，当聚类数目较少时，每一类别内部差异相对较大，为了避免聚类数目较少时发生上述问题，本文选择较多的聚类数目进行分析。

5.3 图像检索模型与实现方法

第一，考虑到我们聚类的目的为的是找到指纹细节点的类中心，目前主要有四种主流的聚类方法。（1）k-means。k-means 是划分方法中较经典的聚类算法之一。由于该算法的效率高，所以在对大规模数据进行聚类时被广泛应用。目前，许多算法均围绕着该算法进行扩展和改进。（2）层次聚类算法。根据层次分解的顺序是自底向上的还是自上向下的，是先将每个对象作为一个簇，然后合并这些原子簇为越来越大的簇，直到所有对象都在一个簇中，或者某个终结条件被满足。（3）SOM 聚类算法。SOM 神经网络假设在输入对象中存在一些拓扑结构或顺序，可以实现从输入空间(n 维)到输出平面(2 维)的降维映射，其映射具有拓扑特征保持性质。（4）FCM 聚类算法，也称模糊 C 均值聚类。为克服非此即彼的分类缺点，出现了以模糊集合论为数学基础的聚类分析。FCM 算法是一种以隶属度来确定每个数据点属于某个聚类程度的算法。

这些聚类方法中能计算出类中心的只有 KMeans 和 C 均值聚类，而两者各有优缺点。但是 **C 均值聚类**能够计算出每个样本对所有类的隶属度，更好的识别指纹细节点的结构特征。

指纹采样点(x,y,θ)数据都能进行 C 均值聚类，但是方向θ指细节点的角度并不会影响坐标(x,y)的位置。在此基础上，我们对坐标(x,y)、方向θ分别进行

C 均值聚类，剔除大部分不相似的指纹，得出大约 2500 条基本具有相似特征的数据。

第二，指纹样本的坐标(x,y)、方向θ会由于外界因素改变发生变化，所以通过 C 均值聚类出的特征指标也会随之变化，而唯一不变的是类中心之间的最短路径。每个指纹都有不一样的类中心，所以类中心的最短路径不一样。反之亦然。计算多源最短路径的算法主要有 Floyd 算法、TSP 问题的多种算法。Floyd 算法是一种用于寻找给定的加权图中顶点间最短路径的算法，计算所有点到所有点的最短路径。TSP 问题也称为旅行商问题，是一个经典的组合优化问题。经典的 TSP 可以描述为：一个商品推销员要去若干个城市推销商品，该推销员从一个城市出发，需要经过所有城市后，回到出发地。应如何选择行进路线，以使总的行程最短。从图论的角度来看，该问题实质是在一个带权完全无向图中，找一个权值最小的 Hamilton 回路。解决 TSP 问题，可以使用贪心算法、动态规划、模拟退火、禁忌搜索、LKH 算法以及 Concorde 求解器算法。

在对指纹类中心点最短路径求解中，TSP 问题的多种算法是远远优于 Floyd 算法。本文我们选取**禁忌搜索算法解决 TSP 问题**，求解指纹坐标(x,y)、方向θ的类中心点最短距离。

第三，构建指标，进行二次匹配。基于禁忌搜索算法，计算出的最优路径线段长度的最大值、最小值、标准差、均值、总值指标。与第二步计算出的最短距离结合，搭建指标。接着采用 K 均值聚类，处理大规模的数据，得到二次匹配的结果。最后，对该类数据进行排序，得到最终“同一”指纹的数据范围。

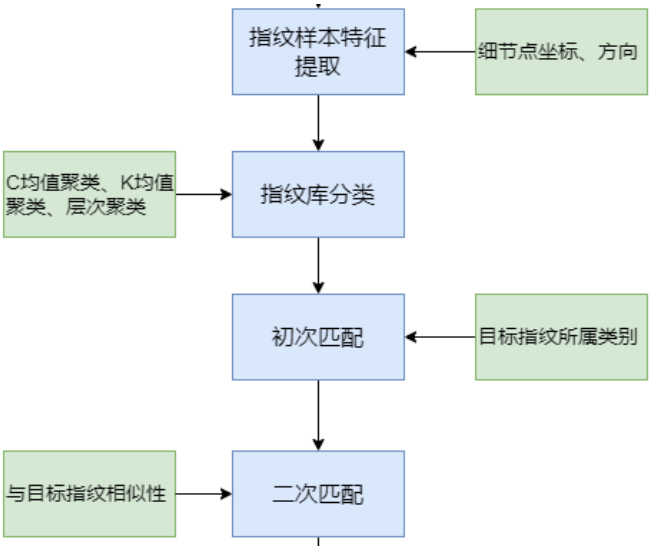

图 5.4　模型构建基本流程图

5.4 算法复杂度

算法的时间、空间复杂度与算法运行过程的实际运算次数相关，由于待检测目标不同则所需循环次数也不同。本文首先对各自循环体中每次最终循环次数进

行统计，即在进行匹配开始到寻找目标集的循环次。

对于时间复杂度，令单个待检测目标的循环工作量为 $f_i(n)$，其中在算法中若出现多个循环体并列分布各自的循环次数为 n_1、n_2，则 $f_i(n)=n_1+n_2$。若出现多个循环体内部镶嵌次数为 n_3、n_4，则 $f_i(n)=n_3\times n_4$。综上所述，$fi(n)=\sum f_{1i}(n)+\sum f_{2i}(n)$。

对于空间复杂度，令单个待检测目标的循环工作量单元格量 $M(n)$，其中 $M(n)=\sum m_j(n)$。其中，$m_j(n)$ 表示各变量的字符个数。

其中，禁忌搜索算法的时间复杂度主要包括产生初始解、生成邻域、判断禁忌表和解除禁忌等操作，表中分别列出每步操作的时间复杂度。

表 5.1　禁忌搜索算法操作

操作名称	时间复杂度
产生初始解	n^2
生成领域	C_n^2（定义它的领域映射为 2-opt）
判断禁忌表	$n\times l$
解除禁忌表	$n\times l$

禁忌搜索算法时间复杂度为 $O\left(Max\_GEN\times\left(n^2+n+l\right)\right)$。

6 指纹快速检索方法的验证

6.1 问题描述

问题二要求采用问题一所提出的检索方法，将 TZ_同指.txt 中全部 500 个匹配对子（合计 1000 行）数据逐一在 TZ_异指.txt 与 TZ_同指.txt 组成的数据集中进行检索，并计算在不同过滤水平下，指纹匹配的穿透率。由于赛题中仅提供了每一枚指纹图像的 ID、所提取到的细节点个数及每个细节点的三元信息，且每一条指纹数据的细节点个数不一致，此外，数据集所提供的可用指标较少。因此，本文首先采用 C 均值聚类在每一个指纹内部进行聚类（每一个指纹聚类数目相同），找出每一个类别的类中心；接下来采用最短路径算法，寻找每一枚指纹内部类中心之间的最短路径，并通过相互连接的类中心之间的距离构造出新的指纹特征；最后采用 K 均值聚类进行指纹匹配分析，采用穿透率对该方法的性能进行评估。该问题所用流程图如下所示：

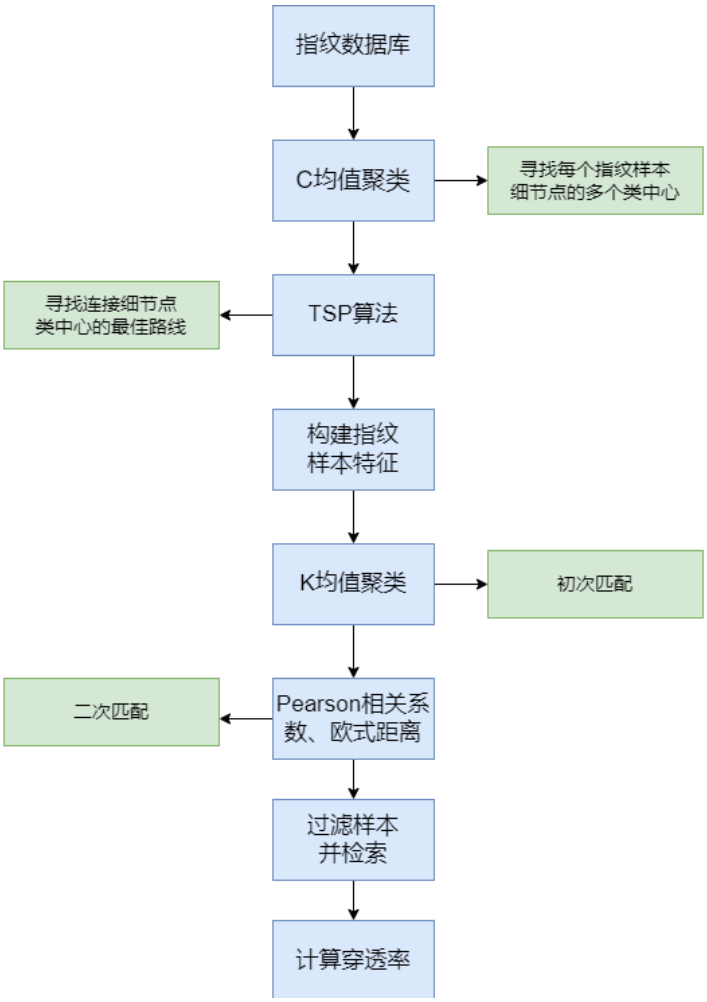

图 6.1　问题二流程图

6.2 指标构建

6.2.1 C 均值聚类

聚类分析是无监督模式识别的一个重要分支，在模式分类，图像处理和模糊规则处理等众多领域拥有广泛的应用。聚类分析将没有类别标记的样本按照某种准则划分为若干子集，使相似的样本尽可能归于一类，而把不相似的样本划分到不同的类中。

在众多模糊聚类算法中，模糊 C-均值聚类算法（Fuzzy C-Means Algorithm，即 FCM）应用较广泛且成功，它通过优化目标函数得到每个样本点对所有类中心的隶属度，从而决定样本点的类属以达到自动对样本数据进行分类的目的。该算法主要包括三个关键参数：固定数量的集群、每个群集一个质心、每个数据点属于最接近质心对应的簇[2]。

（1）目标函数

模糊 C-均值聚类通过最小化目标函数来得到聚类中心。目标函数本质上是各个点到各个类的欧氏距离的和（误差平方和）。聚类的过程就是最小化目标函

数的过程，通过反复的迭代运算，逐步降低目标函数的误差值，当目标函数收敛时，可得到最终的聚类结果。其目标函数如下所示：

$$J_m = \sum_{i=1}^{N}\sum_{j=1}^{C} u_{ij}^{m} \left\| x_i - c_j \right\|^2, 1 \le m < \infty \tag{6.1}$$

其中，m 为聚类的簇数（类数），N 为样本数，C 为聚类中心数。c_j 表示第 j 个聚类中心，和样本特征维数相同，x_i 表示第 i 个样本，u_{ij} 表示样本 x_i 对聚类中心 c_j 的隶属度（即 x_i 属于 c_j 的概率）。$\|*\|$ 可以是任意表示数据相似性（距离）的度量最常见的是欧几里得范数（又称欧式范数，L2 范数）：

$$d = \|x\|_2 = \sqrt{\sum_i x_i^2} \tag{6.2}$$

（2）隶属度矩阵 U_{ij} 和簇中心 C_{ij}

隶属度矩阵应当是 $N*C$ 矩阵，隶属度矩阵表示的是每个样本点属于每个类的程度，对于单个样本 x_i，它对于每个簇的隶属度之和为 1。每个样本在哪个类的隶属度最大则归为哪个类，越接近于 1 表示隶属度越高，反之越低。

求每组的聚类中心 c_i，使得目标函数最小（因为目标函数与欧几里德距离有关，目标函数达到最小时，欧式距离最短，相似度最高），这保证了组内相似度最高，组间相似度最低的聚类原则。

$$u_{ij} = \frac{1}{\sum_{k=1}^{C}\left(\frac{\left\| x_i - c_j \right\|}{\left\| x_i - c_k \right\|}\right)^{\frac{2}{m-1}}} \tag{6.3}$$

$$c_j = \frac{\sum_{i=1}^{N} u_{ij}^{m} \cdot x_i}{\sum_{i=1}^{N} u_{ij}^{m}} \tag{6.4}$$

（3）终止条件

$$max_{ij}\left\{\left| u_{ij}^{(t+1)} - u_{ij}^{(t)} \right|\right\} < \varepsilon \tag{6.5}$$

其中，t 是迭代步数，ε 是一个很小的常数表示误差阈值，也就是说迭代更新 u_{ij} 和 c_j 直到前后两次隶属度最大变化值不超过误差阈值。即继续迭代下去，隶属程度也不会发生较大变化，认为隶属度不再变化，达到了比较优（局部最优或全局最优）状态，该过程最终收敛于 J_m 的局部极小值点或鞍点。

FCM 算法的步骤如下：

①选择类别的数目 C，选择合适的 m，初始化由隶属度函数确定的矩阵 U_0（随机值[0,1]之间初始化）；

②计算聚类的中心值C_j；

③计算新的隶属度矩阵U_j；

④比较U_j和U_{j+1}，如果二者的变化小于某个阈值，则停止算法，否则转向②。

6.2.2 聚类结果分析

由于聚类分析会按照某一准则使相似的样本尽可能归于一类，而把不相似的样本划分到不同的类中。因此，聚类能够在一定程度上反映出每一类中各个对象之间的共同特征。在本题中，聚类能够反映出每一枚指纹的各个类别中的大致特征，通过总结每一类别的特征可得出该枚指纹的多项总体特征，可以简化指纹分析，使得指纹特征更为集中；且通过将各条指纹数据聚为相同的类别，能够使得每一枚指纹的数据长度一致。

每一条指纹数据中包含所提取的细节点的像素坐标及方向，像素坐标（x, y）决定每个细节点所在的位置，而方向θ决定指纹的大致走势，二者反映出不同的信息。在聚类时，可能会出现同一类别内的细节点虽然像素坐标较为靠近，而方向杂乱无章的情况，因此本文分别基于像素坐标（x, y）和方向θ进行聚类分析。

本文大部分指纹采集的细节点在 20-40 个，因此本文选择将每一条指纹数据分别聚为 8、10、12 类进行分析，对于采集点个数少于聚类数目的指纹数据的处理方法是：将每一个点看做一个类中心，例如某一指纹的细节点个数为 10，而聚类数目为 12，则 10 个细节点各自为 1 类，此时便有 10 类，剩余的 2 类则为空值。下图展示了 10000_0（左）和 10000_1（右）指纹内部细节点按照（x, y）进行聚类（聚类数目=8）的结果：

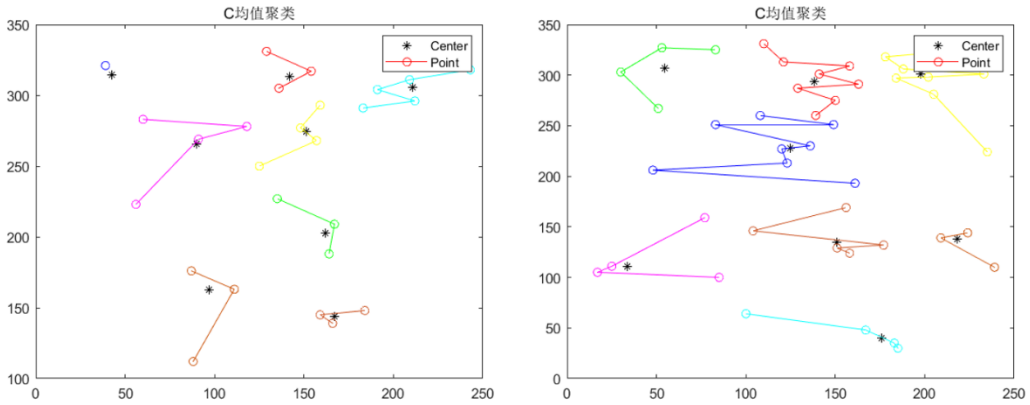

图 6.2　10000_0（左）和 10000_1（右）指纹内部细节点的聚类结果

如上图所示，同一种颜色的线条所连接的点表示同一类别，而*则表示该类别的类中心。通过上图可发现，虽然同一枚指纹两次采集的细节点个数不同，但每一类别所处的位置大致相同，表明进行两次采集时，同一枚指纹内部的细节点具有较大的相似性，进行聚类有助于简化分析，且不会改变指纹内部的大致结构。

此外，采用方向θ进行聚类时，也具有相同的结果，此处不再赘述。

6.3 寻找指纹内部结构

6.3.1 TSP 问题

TSP 问题（Traveling Salesman Problem，旅行商问题）是典型的 NP 完全问题，即在最坏情况下的时间复杂度随着问题规模的增大按指数方式增长，到目前为止还未找到一个多项式时间的有效算法。

TSP 问题可描述为：已知n个城市相互之间的距离，某一旅行商从某个城市出发访问每个城市有且仅有一次，最后回到出发城市，如何安排才能使其所走路线距离最短[3]。

简言之，就是寻找一条最短的遍历n个城市的路径，或者说搜索自然子集$X=\{1,2,\cdots,n\}$（X的元素表示对n个城市的编号）的一个排列$\pi(X)=\{V_1,V_2,\cdots,V_n\}$，使得

$$T_d=\sum_{i=1}^{n+1}d(V_i,V_{i+1})+d(V_n,V_1) \tag{6.6}$$

取得最小值，其中$d(V_i,V_{i+1})$表示城市V_i到城市V_{i+1}的距离。

TSP 问题的实质可以抽象为：在一个带权重的完全无向图中，找到一个权值综合最小的哈密顿回路。TSP 问题是一个经典的最短路径问题，其独特之处在于最短路径包含了回来的路径。

6.3.2 禁忌算法

贪心算法、模拟退火、禁忌搜索、LKH 算法等是解决 TSP 问题的常用算法，在进行算法之间的对比后，本文采用禁忌搜索算法求解 TSP 问题。

禁忌搜索算法（Tabu Serach，简称 TS）是一种全局性邻域搜索算法，模拟人类具有记忆功能的寻优特性，它通过领域搜索机制和相应的禁忌准则来避免迂回搜索，并通过破禁水平来释放被一些禁忌的优良状态，从而保证多样化，实现全局最优。

TS 算法的基本思想是：首先给定一个当前解（初始解）和一种邻域，然后在当前解的邻域中确定若干候选解；若最佳候选解对应的目标值优于全局最优解，则忽略其禁忌特性，用其替代当前解和全局最优解，并将相应的对象加入禁忌表，同时修改晋级表中各对象的任期；若不存在上述候选解，则在候选解中选择非禁忌的最佳状态为新的当前解，而无视它与当前解的优劣性，同时将相应的对象加入禁忌表，并修改禁忌表中各对象的任期，重复上述搜索过程，直到满足停止条件为止[4]。其流程如下所示：

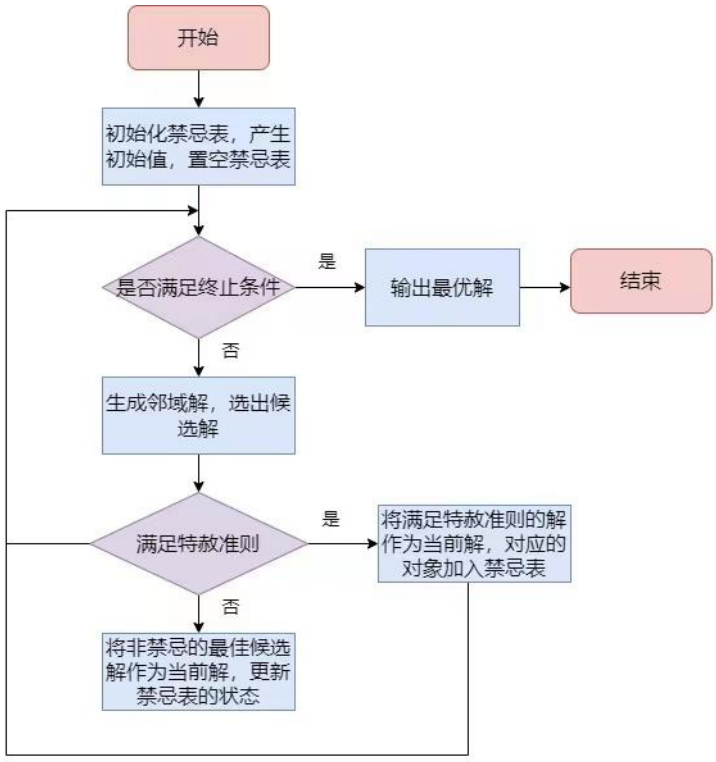

图 6.3　TS 算法流程图

该流程的特点如下：

1、在搜索过程中可以接受劣解，得到全局最优解的概率增大。

2、新解不是随机产生，选取优良解的概率大于其他解。

3、是一种局部搜索能力很强的全局迭代寻优算法。

6.3.3 结果分析

在采集指纹时，由于手指按压图像采集设备的角度、轻重及位置各不相同，因此会指纹图像会发生平移、旋转或局部变形，使得特征点的绝对位置信息有很大变化，但是相邻点间的距离和相对角度等不会有太大改变[5]，而直接计算每一枚指纹所有采集点之间的距离和相对角度，会导致数据非常多，且会再次产生每一条指纹数据的长度不一致的问题，因此此处采用每一枚指纹各个类中心之间的距离和角度来进行代替，研究指纹的相关特点。

此处首先对于上述每条指纹的聚类结果，采用最短路径算法，寻找出每条指纹数据中连接各个类中心的最短路径，同一枚指纹的最短路径也不会有太大改变，且能够大致勾勒出每一枚指纹最短路径的走向。

10000_0（左）和 10000_1（右）指纹按照（x, y）聚类（聚类数目=8）找出类中心之后，采用 TS 算法所寻找到各个类中心之间的最佳路径如下图所示：

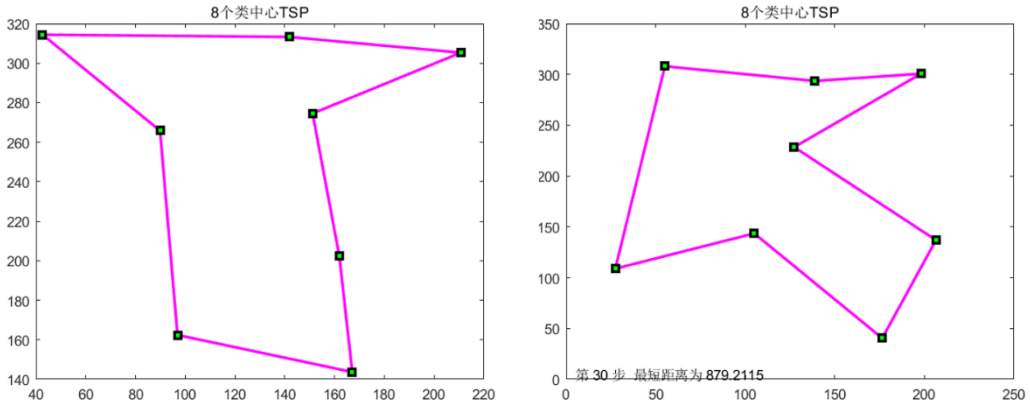

图 6.4　10000_0（左）和 10000_1（右）指纹内部类中心之间的最佳路径

通过观察上述最短路径，可以发现每一枚指纹的最短路径都是各个类中心首尾顺次相连。两幅图的大致形状一致，方向有所差别，可能是由于指纹采集时发生了一定的偏移所造成的，因此，可以认为**同一枚指纹的最佳路径具有较大的相似性**，即通过对比每一枚指纹的最佳路径可以在一定程度上进行指纹间的匹配。

为了研究路径之间的关系，此处分别计算出每一条指纹内部相连的两个类中心点之间的距离，即图中**每一条连线的长度**。而按照方向 θ 进行聚类并寻找类中心之间的最佳路径时，也具有同样的效果。

当聚类数目为 8、10、12 时，进行同样的处理，并对所求得的距离及方向差进行描述性统计分析，即计算其总值、平均值、最大值、最小值、标准差，这些指标基本反映了指纹的特点，因此**将上述指标作为本文新构建的指标**。其中部分结果如下表所示：

表 6.1　指纹细节点（x, y）坐标的类中心 TSP 结果统计量（K=8）

指纹 ID	采集个数	总值	最小值	最大值	平均值	标准差
10000_0	26	611.56	58.97	103.60	76.45	16.07
10000_1	44	875.46	59.81	197.31	109.43	46.68
10001_0	13	543.16	49.75	84.37	67.90	11.49
10001_1	24	678.73	53.20	121.25	84.84	29.03
10002_0	31	770.13	67.27	118.18	96.27	17.42
10002_1	28	760.46	57.06	127.98	95.06	24.41
10003_0	17	652.27	47.05	116.02	81.53	26.25
10003_1	28	803.67	54.68	173.78	100.46	36.11
10004_0	29	620.89	39.89	113.52	77.61	26.35
10004_1	28	596.28	52.35	107.56	74.54	21.56

表 6.2　指纹细节点（x, y）坐标的类中心 TSP 结果统计量（K=10）

指纹 ID	采集个数	总值	最小值	最大值	平均值	标准差
10000_0	26	710.37	56.37	92.20	71.04	12.35
10000_1	44	912.17	50.46	191.10	91.22	48.91
10001_0	13	644.73	38.89	116.16	64.47	20.61
10001_1	24	723.19	40.53	105.78	72.32	21.06
10002_0	31	849.09	26.32	116.42	84.91	25.07
10002_1	28	865.52	49.66	141.04	86.55	29.70
10003_0	17	750.89	47.14	92.80	75.09	18.18
10003_1	28	901.28	42.02	125.75	90.13	28.32
10004_0	29	733.03	48.43	109.68	73.30	19.40
10004_1	28	694.19	40.73	124.39	69.42	31.01

表 6.3　指纹细节点（x, y）坐标的类中心 TSP 结果统计量（K=12）

指纹 ID	采集个数	总值	最小值	最大值	平均值	标准差
10000_0	26	827.34	44.31	101.44	68.95	18.82
10000_1	44	971.44	32.61	150.11	80.95	33.39
10001_0	13	654.40	12.65	97.17	54.53	24.43
10001_1	24	776.85	28.64	103.24	64.74	20.67
10002_0	31	918.45	24.66	123.29	76.54	30.71
10002_1	28	928.71	23.20	131.94	77.39	34.20
10003_0	17	747.98	21.95	91.32	62.33	23.95
10003_1	28	1061.30	51.49	158.18	88.44	31.31
10004_0	29	712.38	19.24	111.94	59.37	24.36
10004_1	28	715.55	26.54	87.74	59.63	16.21

上述三个表中分别展示了在聚类数目为 8、10、12 下前 5 枚同指指纹类中心之间距离的最短路径（总值）、平均值、最大值、最小值及标准差，可以发现，不同指纹类中心点之间距离的总值、平均值等特征均相差较大，而同一枚指纹两次采集所得数据的类中心点之间距离的总值、平均值则相差较小。

此外，随着聚类数目的增加，同一枚指纹两次采集所得到的数据（例如编号为 10000_0 和 10000_1）的最短路径的差距在减小，均值也在逐渐接近，标准差也有所减小，说明随着聚类数目的增加，每一枚指纹的细节点在进行分类时更加细化，每一类别内细节点具有更多的共同特征，类中心之间的距离可能也在减小；而总体来看，每一条指纹数据的标准差均较大，其主要原因在于每一条指纹数据类中心距离的最小值与最大值之间差距较大，且聚类数目少，因此每个线段（即

两个类中心之间的连线）的长度之间差异较大。

但是仍然有少数同一枚指纹的两次采集数据中，类中心距离的各个指标相差较大，例如 10003_0 和 10003_1 的最短路径、平均值、最大值相差均较大，一方面可能是由于采集点个数不同导致每一条指纹数据在聚类时差异较大所造成的；另一方面可能是采集点位置相差较远，二者的较为接近的信息较少所造成的。

表 6.4　指纹细节点方向θ的类中心 TSP 结果统计量（K=8）

指纹 ID	采集个数	总值	最小值	最大值	平均值	标准差
10000_0	26	671.97	26.41	141.26	84.00	42.91
10000_1	44	668.61	23.47	212.27	83.58	61.01
10001_0	13	671.57	32.39	158.77	83.95	44.11
10001_1	24	652.31	19.19	156.20	81.54	49.99
10002_0	31	519.88	17.54	227.31	64.98	67.91
10002_1	28	520.38	24.13	112.71	65.05	32.95
10003_0	17	662.72	8.00	167.35	82.84	73.69
10003_1	28	675.70	19.54	161.22	84.46	64.30
10004_0	29	611.17	12.75	262.80	76.40	92.23
10004_1	28	632.42	17.57	259.88	79.05	100.98

表 6.5　指纹细节点方向θ的类中心 TSP 结果统计量（K=10）

指纹 ID	采集个数	总值	最小值	最大值	平均值	标准差
10000_0	26	672.18	11.85	170.55	67.22	54.44
10000_1	44	669.35	14.81	185.16	66.94	46.83
10001_0	13	671.98	17.90	144.70	67.20	38.24
10001_1	24	653.13	9.73	154.48	65.31	54.69
10002_0	31	512.81	4.67	134.46	51.28	41.97
10002_1	28	522.03	15.88	109.02	52.20	29.33
10003_0	17	664.15	4.50	163.96	66.42	73.67
10003_1	28	675.63	4.09	167.74	67.56	59.82
10004_0	29	645.57	13.21	208.91	64.56	72.80
10004_1	28	647.37	15.13	252.23	64.74	91.59

表 6.6　指纹细节点方向θ的类中心 TSP 结果统计量（K=12）

指纹 ID	采集个数	总值	最小值	最大值	平均值	标准差
10000_0	26	672.27	7.26	145.97	56.02	38.25

10000_1	44	670.91	15.56	110.76	55.91	33.54
10001_0	13	672.00	7.00	270.00	56.00	71.93
10001_1	24	653.27	9.06	102.33	54.44	31.79
10002_0	31	522.75	11.51	138.75	43.56	35.11
10002_1	28	523.40	7.49	150.06	43.62	43.22
10003_0	17	668.68	1.00	172.70	55.72	70.28
10003_1	28	682.43	9.38	202.51	56.87	57.67
10004_0	29	646.55	12.29	268.38	53.88	78.21
10004_1	28	647.44	11.53	202.27	53.95	62.91

上表可以发现，在不同聚类数目下，每一条指纹数据按照细节点角度进行聚类后，不同指纹的类中心点之间角度差的总值、标准差等相差较大，而同一枚指纹两次采集所得的类中心点之间角度差的最短路径非常接近，最小值也较为接近，但最大值相差较大，可能是由于采集点个数不同及采集位置相差较远所造成的，故平均值相差也比较大。

对比按照细节点的像素坐标及角度分别进行聚类的结果，可以发现，同一枚指纹两次采集数据中角度的类中心点之间的角度差的总和、平均值的差距比像素坐标更小，可能与指纹本身的方向变化范围较小且指纹的角度略小有关。

TZ_异指.txt 与 TZ_同指.txt 两个文件的数据进行了相同的数据处理，此处不再赘述，至此构建了 **30 个特征**用于后文分析。

6.4 指纹匹配

6.4.1 K 均值聚类

K 均值聚类（*KMeans*）算法是无监督聚类算法的代表之一，主要作用是将相似的样本自动归到一个类别中[6]。其基本原理如下：

假定给定数据样本 X，包含了 n 个对象 $X=\{X_1,X_2,X_3,\cdots,X_n\}$，其中每个对象都具有 m 个维度的属性。*KMeans* 算法的目标是将 n 个对象依据对象间的相似性聚集到指定的 k 个类簇中，每个对象属于且仅属于一个其到类簇中心最近的类簇中。对于 *KMeans*，首先需要初始化 k 个聚类中心 $\{C_1,C_2,C_3,\cdots,C_k\},1<k\le n$，然后计算每一个对象到每一个聚类中心的欧氏距离：

$$dis(X_i,C_j)=\sqrt{\sum_{t=1}^{m}(X_{it}-C_{jt})^2} \tag{6.7}$$

其中，X_i 表示第 i 个对象 $1\le i\le n$，C_j 表示第 j 个聚类中心 $1\le j\le k$，X_{it} 表示第 i 个对象的第 t 个属性 $1\le t\le m$，C_{jt} 表示第 j 个聚类中心的第 t 个属性。

依次比较每一个对象到每一个聚类中心的距离，将对象分配到距离最近的聚

类中心的类簇中，得到k个类簇$\{S_1, S_2, S_3, \cdots, S_k\}$。

KMeans 算法用中心定义了类簇的原型，类簇中心就是类簇内所有对象在各个维度的均值，其计算公式如下：

$$C_l = \frac{\sum_{X_i \in S_l} X_i}{|S_l|} \tag{6.8}$$

其中，C_l表示第l个聚类的中心$1 \le l \le k$，$|S_l|$表示第l个类簇中对象的个数，X_i表示第l个类簇中第i个对象$1 \le i \le |S_l|$。

6.4.2 聚类结果分析

为了将 TZ_同指.txt 中 500 枚指纹（共 1000 条数据）逐一在 TZ_异指.txt 与 TZ_同指.txt 组成的 11000 条数据中进行检索，在过滤掉数据集中 80%、90%、95%、97%的图像后，计算仍含有“同一”指纹匹配对子的数量占总指纹“同一”匹配对子数量的比例，本文选择分两步进行指纹的检索：

（1）**初次匹配**：基于前文所构建的 30 个指标，采用 *KMeans* 算法将 11000 条数据进行聚类，寻找出每一类别的类中心，计算并比较所需匹配的指纹与每一个类中心之间的距离，并将距离由小至大依次排列，寻找出与所需匹配的指纹距离最近的类中心，该类中心所在的类别就是指纹所属的类别。当所属类别的样本数量小于要求的保留的样本数量（例如所属类别只有 300 条数据，而要求只过滤 80%的样本，即保留 550 条（11000*20%=550）样本）时，则寻找与所匹配指纹距离排名第二的类中心，依次类推，直至几个类中心所在类别的样本数量达到所要求保留的样本量（例如为前文所述的 550 条样本）。

（2）**二次匹配**：为了过滤掉一定比例的指纹图像，进一步将所需匹配的指纹在留下的样本库中进行匹配，此处采用两种方法进行处理：

方法一：依据初匹配阶段所给出的指纹所在的类别，依次计算出所需匹配的指纹与该类别中每一枚指纹之间的**欧式距离**，并将距离进行**升序排列**，选择出**符合过滤要求**的样本，将所需匹配的指纹在选择出的样本中进行检索，寻找是否有匹配样本，并计算穿透率。例如过滤掉 80%的样本，留下指纹库中 20%的样本，即合计 2200 条样本（11000*20%=2200），此处需要计算出所需匹配的指纹与所属类别内各个指纹之间的欧式距离，并将距离进行升序排序，选择出距离最近的前 2200 条样本，将所需匹配的指纹在 2200 条样本中进行检索，寻找是否有匹配样本，从而计算整体穿透率（其余过滤水平下，指纹的检索方法类似）。此外，如前文 6.1.2 所述，由于聚类数目对于聚类结果具有重要影响，因此此处仍然选择不同聚类数目进行分析，通过对比不同聚类数目下的穿透率水平，选择最佳聚类数目。

方法二：**假设指纹之间存在线性相关关系**，此处计算所需匹配指纹细节点的

类中心与其所属类别中每一枚指纹细节点类中心之间的 **pearson 相关系数**，并进行降序排列，筛选出符合过滤比例的少量样本，再将所需匹配的指纹进行检索，计算穿透率。例如，在 80%的过滤水平下，则筛选出 pearson 相关系数最高的前 2200 条样本，将指纹进行检索后，计算穿透率。为了和方法一形成对比，此处使用上述所寻找的最佳聚类数目进行聚类分析。

依据上述方法，所计算出不同聚类数目及不同过滤水平下所匹配指纹的穿透率如下表所示：

表 6.7　不同过滤水平下所匹配指纹的穿透率

编号	聚类数目	过滤水平	穿透率	平均耗时	编号	聚类数目	过滤水平	穿透率	平均耗时
1	5	80%	0.507	4.19	2	10	80%	0.500	4.21
		90%	0.331				90%	0.333	
		95%	0.233				95%	0.234	
		97%	0.156				97%	0.150	
3	15	80%	0.493	4.10	4	20	80%	0.501	4.22
		90%	0.332				90%	0.334	
		95%	0.232				95%	0.228	
		97%	0.157				97%	0.155	
5	25	80%	0.495	4.98	6	15	80%	0.481	1.16
		90%	0.330				90%	0.317	
		95%	0.220				95%	0.200	
		97%	0.154				97%	0.141	

上表中**编号 1-5** 表示采用**欧氏距离**所筛选出的符合过滤水平的样本的穿透率，**编号 6** 表示采用 **pearson 相关系数**所筛选的符合过滤水平的样本的穿透率。其中平均耗时是指将一条指纹在筛选后的指纹样本库中进行检索所花费的时间。

为了直观表示不同聚类数目及不同过滤水平下所匹配指纹的穿透率，将上表进行可视化分析，如下图所示：

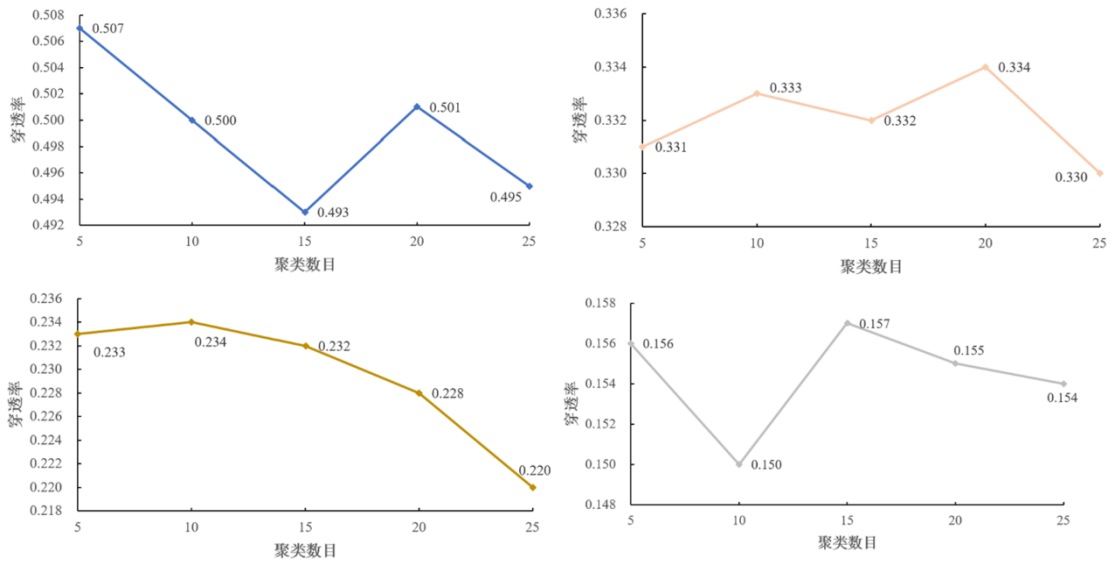

图 6.5　不同过滤水平下所匹配指纹的穿透率图

结合上表和上图（图中左上角表示 80%的过滤水平，右上角表示 90%的过滤水平，左下角表示 95%的过滤水平，右下角表示 97%的过滤水平），可以发现在同一聚类数目的不同过滤水平下，通过指纹检索所计算的穿透率不同，随着过滤水平的增加，穿透率在明显下降，是由于图像检索过程筛选掉的指纹图像越多，则在筛选后保留下来“同一”关系指纹图像的可能性相对变小所造成的。

通过上图可以发现，在同一过滤水平的不同聚类数目下，指纹匹配的穿透率也不相同，主要是由于聚类结果不同所导致的。在实际应用中，往往更侧重于在较高的过滤水平下进行高效率的指纹检索，当**聚类数目为 15** 时，在 97%的过滤水平下，**其穿透率最高**，且 80%、90%、95%过滤水平下的穿透率与其余聚类数目之间相差较小，耗费时间较短；聚类数目为 20 时，在 97%的过滤水平下，其穿透率比聚类数目为 15 的穿透率低 0.002，其余过滤水平的穿透率表现良好，且耗费时间最短；而聚类数目为 25 时，不仅耗时较长，且穿透率低，因此聚类数目不能选取过大；聚类数目为 5 时，虽然穿透率较高，但耗时略长，且对于本题中大容量的数据，每一类别内各个指纹细节点之间可能存在较大的差距，当进行指纹检索时会造成结果不准确等问题；当聚类数目为 10 时，耗时和穿透率均表现一般。因此，综合而言，在现有的聚类数目下，聚类数目为 15 时，指纹检索效果比较好。

采用皮尔孙相关系数进行分析时，所选择的聚类数目为 15，虽然其耗时较短，但在不同过滤水平下的指纹检索的穿透率均较低，表现差。基于算法本身而言，pearson 相关系数是内置函数，而采用欧氏距离进行分析时需要进行循环，故两种计算方式所耗费时间差距较大。

综上所述，欧式距离进行样本筛选时虽然耗时久，但穿透率高；而 pearson 相关系数则恰好相反，采用欧氏距离进行样本筛选可以在确保穿透率的水平下，

进一步优化代码，减少指纹检索所耗费的时间。因此，本文选择的最佳聚类数目为 15，即采用 K 均值聚类将所给出的 11000 条样本数据聚为 15 类，并计算所匹配指纹与每一类中心的距离，判断指纹所属类别，再计算所匹配指纹与该类别中每一条指纹数据间的欧氏距离，按欧氏距离将该类别中的指纹数据进行升序排列，筛选出符合过滤比例的指纹样本库，从而计算穿透率。

7 指纹快速检索方法的运用

7.1 问题描述

问题 3 要求针对 TZ_同指 200_乱序后 Data.txt 数据文件采用和问题 2 相同的方式进行检索，并给出检索结果（所需数据集的构成为 TZ_异指.txt 和 TZ_同指 200_乱序后_Data.txt 两个文件，合计 10400 条指纹数据）。TZ_同指 200_乱序后 Data.txt 文件中只给出了 200 枚指纹的细节点个数及每个细节点的三元信息，由于每枚指纹所提取的细节点个数不同，所以每一行数据的长度不同，且每个细节点的三元信息所提供的可用信息是较少的，因此此处仍然对于每一条指纹样本内的细节点分别按照像素坐标（x, y）和方向θ进行 **C 均值聚类**，找出每一类别的类中心，并采用最佳路径寻找类中心之间的**最短路径**，从而**构造出新的指纹特征**，最后采用 **K 均值聚类**过滤掉一定比例的样本之后，筛选出与所需匹配的指纹高度相似的指纹库。

7.2 模型建立

7.2.1 C 均值聚类

本题中首先采用 C 均值聚类算法对每一条指纹内部细节点分别按照像素坐标（x, y）及方向θ进行聚类分析，并找出每一类别的类中心（为了便于描述，下文的类中心包含按照（x, y）及θ分别找到的类中心），此处聚类数目和前文保持一致，即对每一条指纹数据均进行三次聚类，聚类数目分别为 8、10、12。

当细节点个数少于聚类数目时，则将该指纹的每个细节点作为一个类中心，例如某一枚指纹的细节点个数为 9，聚类数目为 10，则 9 个细节点就是一个类中心，而缺失的另一类则为空值。

7.2.2 构建指标

（1）寻找指纹内部结构

在同一枚指纹内部虽然细节点的坐标、方向会发生一定的偏差，但细节点之间的最短路径是相对固定的，且最短路径能够反映出指纹的内部结构及大致走势。但由于每个指纹的细节点个数不同，难以进行指纹间的比对，因此此处选择用类

中心之间的最短路径来近似代替（每一枚指纹的类中心个数是一致的）。

本文采用禁忌算法（TS 算法）求解每枚指纹类中心之间的最短路径（连接各个类中心之间的线段的最短距离的总值），并绘制出最佳路线图（即连接各个类中心点的图），所绘制的最佳路线图便能够在一定程度上反映出指纹的内部结构。

（2）构建指标

在上述所绘制的最佳路线图中，计算出每一枚指纹类中心点之间所有连线的距离，并输出每一枚指纹这些距离的总和（即最短路径）、平均值、最小值、最大值和标准差，从而便构建了指纹样本的新特征，且每一条样本的数据长度一致，便于进一步分析。

7.2.3 指纹匹配

（1）初次匹配

基于上述所构建的指标，将 TZ_异指.txt 和 TZ_同指 200_乱序后_Data.txt 中共计 10400 条指纹数据采用 K 均值方法进行聚类，所选择的聚类数目为 15（与前文 6.4.2 所述的最佳聚类数目一致），并找出每一类别的类中心。

当利用 TZ_同指 200_乱序后_Data.txt 中的某条指纹在本题的指纹库里进行检索时，首先利用该指纹所构建的特征，计算该指纹与每一类别的类中心之间的欧式距离，并将所计算的距离由小至大依次排列。在一定的过滤水平下，寻找距离该指纹最近的一个或多个类中心所在的类别。

（2）二次匹配

在上述所选择的类别中，计算所需匹配的指纹与类别中所有指纹之间的欧式距离，并进行升序排列，选择达到过滤要求的样本量作为所需匹配指纹的检索库。

例如，在 90%的过滤水平下，需要保留 1040 条（10400*10%=1040）样本数据，而距离所需匹配的指纹最近的类中心所在的类别仅有 600 条数据，则需要寻找与所需匹配的指纹之间的距离排名为第二的类中心所在的类别，若该类别中有 700 条数据，则计算所需匹配的指纹与该类别中每一指纹之间的欧式距离并进行升序排列，选择排名前 440 的样本，与前面的 600 条样本共同组成该指纹的检索库。

同理，若在 90%的过滤水平下，距离所需匹配的指纹最近的类中心所在的类别有 1200 条数据，则计算所需匹配的指纹与该类别中每一指纹之间的欧式距离并进行升序排列，选择排名前 1040 的样本作为该指纹的检索库。

（3）结果分析

依照上述方法，将 TZ_同指 200_乱序后_Data.txt 的指纹逐一在 TZ_异指.txt 和 TZ_同指 200_乱序后_Data.txt 组成的 10400 条指纹数据中进行检索。在 90%、95%、97%的过滤水平下，通过计算所需匹配的指纹与其所属类别中每一枚指纹

间的欧式距离，并按距离由小至大依次排列，分别筛选出排名前 10%、5%、3%（此处的比例为占总样本量的比例）的样本即为每一条指纹进行检索的指纹库。

对于同一枚指纹而言，在不同过滤水平下所筛选出来的排名靠前的一定数量（比如前 100 枚）指纹是一致的，因此此处仅展示某一过滤水平下不同指纹进行检索时，欧式距离排名前五的指纹数据库，部分结果如下表所示：

表 7.1　部分指纹检索的数据库

指纹 ID	指纹 1	指纹 2	指纹 3	指纹 4	指纹 5
A1	A1	0870_0	A350	A393	A239
A2	A2	A393	1550_0	A350	0870_0
A3	A3	A102	6538_0	1691_0	4215_0
A4	A4	5532_0	1373_0	8536_0	9478_0
A5	A5	3847_0	7136_0	3143_0	1591_0

上表展示了 90%的过滤水平下，编号 ID 为 A1-A5 的指纹经过检索后，所筛选出的与目标指纹之间的欧式距离排名为前五的高度相似的指纹，可以发现排在第一位的是目标指纹本身，符合实际。其余过滤水平下的所筛选出的高度相似的指纹类似，此处不再赘述。

8 指纹快速检索方法的评价与优化

8.1 问题描述

该题要求根据前文问题 2 中对构建的指纹快速检索方法模型及技术路线的检索能力验证结果，客观评价本文模型的优缺点，并需要针对 97%以上的筛选量，在检索精度、检索时间及内存占用等方面提出优化策略。

8.2 模型优缺点

8.2.1 模型优点

（1）本文模型尝试使用指纹细节点构造指纹样本特征变量。对细节点使用 C 均值聚类找出类中心并使用基于禁忌算法的 TSP 思想计算最短路径，在一定程度上描述细节点间的关系，建立指纹样本特征变量。

（2）*KMeans* 算法具有运算速度快的优势，能够借助指纹样本特征变量对指纹数据库进行快速分类，从而将具有“同一”性的指纹样本归为同类，从而减少一一比对指纹的工作量。

（3）本文在初次匹配过程中考虑到，聚类数量过少时仅保留目标指纹所属类别样本的做法，可能会引起“同一”指纹样本丢失的情况，通过选择较多聚类数量、合并多个相似类别样本的方法，在一定程度上避免了该问题。

8.2.2 模型缺点

（1）指纹样本特征变量的构造考虑得还不够周全，这些变量反映的指纹样本“独有”特征信息较少，最终导致了穿透率较低的情况。

（2）二次匹配过程中，指纹样本间相似性的评价方法过于简单，欠缺匹配样本点的寻找过程，尽管计算速度快，但降低了穿透率。

（3）数据库聚类、最短路径计算等方面的算法选择单一。

8.3 模型优化

该题提出需要考虑 97%以上筛选量的情况，考虑到本文模型存在的优缺点，再结合穿透率对模型能力的检验结果可知，如果想要在更高过滤水平下实现高穿透率、高检索效率，那么必须对模型进行改进。模型的改进通常分为检索精度、检索时间和内存占用三个角度，本文将逐一进行分析。

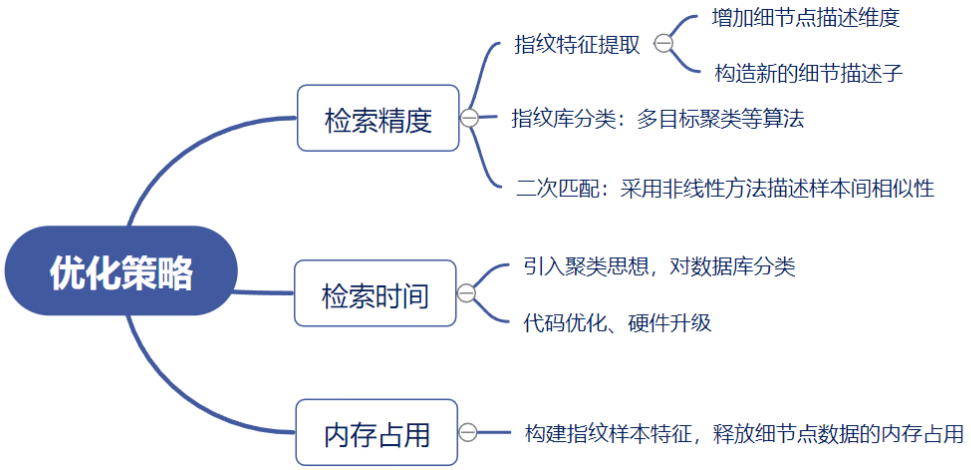

图 8.1　模型优化策略

8.3.1 检索精度优化

检索精度的优化指的是在更高过滤水平下实现高穿透率，想要达到这一目的，关键问题在于准确提取指纹特征。本题提供的指纹细节点属于局部特征描述，查阅文献发现，无论是传统的三角形结构[7]、低阶 Delaunay 三元组[8]、细节点四元组[9]等指纹特征检索算法，还是近些年的兴起指纹检索过滤算法，都能达成很高的成功率。这些算法具有一个共同点——基于更多细节点特征（如尺度 δ 、辅助点方向 θ_i 等）构造描述子，从而对得到不同指纹各自的“独有”特征，从而进行精确识别。因此，想要在更高过滤水平下实现高穿透率，一方面可以进一步扩展细节点特征维度，以更多信息进行两两匹配，另一方面可以改进现有描述子构建方法，对指纹的“独有”特征进行准确描述。

除了准确提取指纹特征可以实现高穿透率外，对于指纹数据过滤方法的改进

也能达成目的。以本文的模型方法为例，可以在指纹数据库分类方法进行突破，本文采用的 *KMeans* 聚类方法具有其局限性，不能对多目标任务进行聚类，而现在已有研究者[10]对多目标聚类算法进行了研究，可考虑将这些研究成果运用于指纹数据库的分类。此外，本文模型中的二次匹配步骤同样存在改进空间，鉴于时间限制，仅尝试了欧式距离和 Pearson 相关系数的效果，这两个指标只反映了样本间的距离和线性关系，还可以尝试非线性地描述样本关系等方法。

8.3.2 检索时间优化

检索时间的优化指的是在更高过滤水平下实现指纹检索的高效率。尽管现阶段的主流算法能较精确地识别指纹图像，但随着指纹数据库的不断扩大，这些算法消耗的时间也在不断增加，对指纹数据库的分类势在必行。指纹数据库分类的优势在于能大大减少两两比对的工作量，通过排名样本间相似度和设置过滤阈值，能迅速锁定具有“同一”性特征的可能样本，将大量比对工作缩减到阈值范围内，具有非常重要的现实意义。聚类方法具有非常优秀的性能，可分为有监督和无监督的聚类，其高效的运算逻辑和稳健的分类结果能很好适应实际需求，现已广泛运用于样本分类工作中。因此，先要提高指纹检索识别的工作效率，除了代码层面的优化以及硬件能力升级外，还可以将聚类思想融入进来。

8.3.3 内存占用优化

内存占用的优化指的是减少检索工作过程中的临时数据调用量。通常，在指纹识别工作中需要在数据库中提取指纹图像与目标指纹图像进行一一比对，如果一次性提取的指纹图像过多，难免需要占用更多的内存资源，即使提取的是指纹细节点特征，但因为细节点数量较多的缘故，同样会造成高内存占用。但是，若仅提取指纹样本的特征数据，那么临时数据调用量将会大大降低，带来更多内存释放。可见，构造指纹样本的“独有”特征具有非常重要的现实意义。

参考文献

[1] 袁宝玺. 超大规模指纹库的索引结构和检索方法[D].北京邮电大学,2013.
[2] 徐金东,赵甜雨,冯国政,欧世峰.基于上下文模糊 C 均值聚类的图像分割算法[J].电子与信息学报,2021,43(07):2079-2086.
[3] 唐文秀.基于改进禁忌搜索算法求解 TSP 问题[J].科学技术创新,2022(04):154-157.
[4] 陈展,公建宁,刘媛媛,徐京邦.基于禁忌搜索的多 AGV 系统路径优化算法[J].计算机工程与应用,2021,57(10):273-278.
[5] 廖阔,杨万麟.点模式指纹匹配算法研究与实现[J].电子科技大学学报,2004(02):154-157.
[6] 杨金,陈林,周强,陈建勋.基于数据的 K 均值理论及其应用[J].南京师大学报(自然科学版),2021,44(02):10-17.
[7] Eberhart R C, Kennedy J. A new optimizer using particle swarm theory[C]. Proceedings of 6th Symp of Micro Machine and Human Science,1995:29-43.
[8] Dunn J C. A fuzzy relative of the ISODATA process and its use in detecting compact well-separated clusters[J]. Journal of Cybernetics,1973(3):32-57.
[9] Bezdek J C. Pattern recognition with fuzzy objective function algorithms[M]. New York: Plenum Press,1981.
[10] 朱书伟. 基于群体智能的多目标聚类算法研究[D].江南大学,2016.