Geomatics Science and Technology 测绘科学技术, 2025, 13(2), 109-128
Published Online April 2025 in Hans. https://www.hanspub.org/journal/gst
https://doi.org/10.12677/gst.2025.132014

Hans汉斯

DBSCAN算法在优化农业灌区划分中的应用研究

毛新宇*，刘泓辰，郜婷婷，董欣月，彭　亮

华北水利水电大学地球科学与工程学院，河南　郑州

收稿日期：2025年3月28日；录用日期：2025年4月20日；发布日期：2025年4月29日

摘　要

随着农业现代化进程的不断推进，农业灌区划分的优化工作显得尤为重要。本研究运用一种基于密度分类的DBSCAN算法作为农业灌区划分的新方法，该方法结合当地灌溉井、水系、耕地土壤类型、土地利用现状等影响因素，并基于Visual Studio 2019软件与C#编程语言，编写DBSCAN算法代码及聚类评估代码，以灌溉井坐标信息为例进行聚类分析，得出当地农业灌区的划分结果。结果表明：① 对比传统划分方法需要考虑的八大方面，本研究采用的基于DBSCAN算法的灌区划分方法可以在保证准确率(92.86%)的前提下，仅考虑四个方面，即通过更少的工作量完成灌区划分的前期准备工作。② 基于C#语言的DBSCAN算法架构及聚类评估体系，可以有效反映不同要素对灌区划分结果的影响，即可以通过改变坐标点的含义达到侧重性划分的目的。③ 算法中的参数Eps和MinPts可以代表目标点的半径及密度限制，这为以后的新建、续建和维护灌区提供了广泛的适用性。研究成果对新增灌区建设及续建工程的规划有一定借鉴意义。

关键词

农业灌区，聚类分析，DBSCAN算法，SC-DB-V-Measure

The Application Research of DBSCAN Algorithm in Optimizing the Division of Agricultural Irrigation Areas

Xinyu Mao*, Hongchen Liu, Tingting Gao, Xinyue Dong, Liang Peng

College of Geosciences and Engineering, North China University of Water Resources and Electric Power, Zhengzhou Henan

Received: Mar. 28th, 2025; accepted: Apr. 20th, 2025; published: Apr. 29th, 2025

*第一作者。

文章引用: 毛新宇, 刘泓辰, 郜婷婷, 董欣月, 彭亮. DBSCAN 算法在优化农业灌区划分中的应用研究[J]. 测绘科学技术, 2025, 13(2): 109-128. DOI: 10.12677/gst.2025.132014

Abstract

As the process of agricultural modernization continues to advance, the optimization of agricultural irrigation area division has become particularly important. This study employs a density-based clustering algorithm known as DBSCAN as a new method for dividing agricultural irrigation areas. The method integrates local factors such as irrigation wells, water systems, soil types of cultivated land, and current land use status, and is based on the Visual Studio 2019 software and the C# programming language to write DBSCAN algorithm code and clustering evaluation code. Taking the coordinates of irrigation wells as an example, cluster analysis is conducted to obtain the division results of local agricultural irrigation areas. The results show: ① Compared to the eight aspects that need to be considered in traditional division methods, the DBSCAN algorithm-based irrigation area division method used in this study can, while ensuring accuracy (92.86%), only consider four minor aspects, that is, to complete the preliminary preparation work of irrigation area division with less workload. ② The DBSCAN algorithm architecture and clustering evaluation system based on the C# language can effectively reflect the impact of different factors on the division results of irrigation areas, that is, it is possible to achieve a focused division by changing the meaning of coordinate points. ③ The parameters Eps and MinPts in the algorithm can represent the radius and density limit of the target point, which provides broad applicability for the future construction, continuation, and maintenance of irrigation areas. The research findings have certain reference significance for the planning of new irrigation area construction and continuation projects.

Keywords

Agricultural Irrigation Areas, Cluster Analysis, DBSCAN Algorithm, SC-DB-V-Measure

1. 引言

随着农业生产方式的转型和农田水资源日益紧张，传统老旧灌区灌溉效益下降，已经难以适应现代农业的需求[1]-[3]。因此，灌区划分成为实现农业水资源高效配置的关键步骤。2022 年 9 月国务院常务会议审议通过了《关于扩大当前农业农村基础设施建设投资的工作方案》，方案要求抓紧实施一批大型灌区、中型灌区续建配套与现代化改造，新建一批大型灌区[4]。通过科学合理地将农田划分为不同的灌区，可以更精准地满足不同作物对水分的需求，避免因水资源浪费和不均匀利用而导致的生态环境压力[5]-[8]。

近年来诸多学者对 DBSCAN (Density-Based Spatial Clustering of Applications with Noise)算法及灌区进行研究。孙林等[9]提出一种基于优化初始聚类中心和轮廓系数的 K-means 聚类算法，能够选取最优的 K 和唯一的初始中心点，获得了更好的聚类结果，同时也提升了聚类算法的稳定性。王子龙等[10]提出一种基于距离和样本权重改进的 K-means 算法，在 UCI 数据集上进行测试，对比经典 K-means 算法、WK-means 算法、ZK-means 算法和 DCKmeans 算法，基于距离和权重改进的 K-means 算法的聚类效果更好。彭波等[11]对 DBSCAN 算法进行了优化研究，通过减少核心对象邻域查询计算的方法，提高了算法的时间效率，并将其应用于村镇网格化管理中，实现了数据记录的有效挖掘和辅助决策。许雪婷等[12]利用 DBSCAN 算法和地理探测器方法，分析了伊犁河谷住宿业的空间分布特征及其影响因素，发现其呈现整体分散与局部聚集的空间格局，并受景区、星级酒店、交通设施和人口等核心因素的影响。除此之外，

在灌区建设方面，范习超等[13]通过构建大型灌区节水水平评价指标体系，筛选出 23 个关键指标，对河南省 4 个大型灌区的节水水平进行了实证研究，为科学评估灌区节水建设提供了有效方法，并为灌区现代化改造提供了指导。杨振峰等[14]应用知识图谱可视化工具 VOSviewer 梳理灌区综合评价的指标体系，构建了宁夏贺兰县现代化生态灌区综合评价体系，并利用层次－模糊综合评价方法对灌区健康水平进行评价，为西北地区现代化生态灌区建设提供了参考。史良胜等[15]开发非完整先验物理机制下灌区水分、盐分、养分、污染物迁移转化以及作物生长和生态系统演化模型，实现灌区尺度上的水量、水质和生态的最优化管理。谢崇宝等[16]提出了现代化数字灌区建设主要业务应用结构体系，有效提升灌溉供水服务的安全性、公平性、可靠性和灵活性。魏新光等[17]通过分析辽宁省气象数据和玉米生长资料，对玉米需水量、灌溉需水量、水分盈亏指数等进行了时空分布研究，提出了辽宁省玉米种植区的灌溉区划方案，为区域农业用水管理提供了理论依据。

通过 DBSCAN 聚类算法，可以通过已知水井的位置数据作为划分灌区的有利条件之一。DBSCAN 算法需要用户指定 Eps (半径)和 MinPts (最小点数)参数在实际应用中需要根据具体数据集的特点和需求选择合适的参数和算法，而在灌区划分的应用中，Eps 可以代指单口水井的控制灌溉面积，MinPts 可以代指每亩或指定面积上灌溉水井的数量。判断聚类结果的好坏主要应用轮廓系数、DB 指数和 V-measure，可以与无聚类的原始数据作比较，得到更好的结果。基于 Visual Studio 2019 软件与 C#编程语言编写 DBSCAN 算法代码及聚类评估代码，带入灌溉井坐标信息进行聚类分析，根据水域、耕地、人口、土地利用现状，得出当地农业灌区的划分结果。

2. 原理与方法

2.1. DBSCAN 原理定义

DBSCAN 算法基于密度的聚类思想，通过定义核心点、边界点和噪声点，实现对高密度区域的聚类。其核心原理在于通过样本点的密度来确定簇的形成。DBSCAN 的核心概念包括“Eps”和“MinPts”，这两个参数决定了簇的形成条件。

DBSCAN 算法是一种基于密度的空间聚类算法，该算法可以划分具有一定密度条件下的任意形状的簇，能有效识别出数据集中噪声点和离群点。DBSCAN 算法的基本概念包括 2 个算法参数，即 ε 邻域与最少点数，3 种类别的点，即核心点、边界点与噪声点，4 种点的关系，即密度直达、密度可达、密度相连与非密度相连，各基本概念定义如表 1 所示，示意图如图 1 所示，算法流程图如图 2 所示。

Table 1. The basic concept of DBSCAN
表 1. DBSCAN 基本概念

名称	概念
ε 邻域	给定对象 X，其半径为 ε 内的区域称为该对象的 ε 邻域
最少点数	核心点邻域内包含的最小点数量 MinPts
核心点	给定对象 ε 邻域内的样本点数不小于 MinPts，则称该对象为核心点
边界点	在核心点的邻域内的非核心点
噪声点	非核心点与边界点的其他点
密度直达	P 为核心点，Q 在 P 的 ε 邻域内，那么称 P 到 Q 密度直达
密度可达	存在一对象链 $P_1,P_2,\ldots,P_n$，若满足任意 P_i 到 P_{i+1} 密度直达，则 P_0 到 P_n 密度可达
密度相连	若存在核心点 S，使得 S 到 P 和 Q 都密度可达，则 P 和 Q 密度相连
非密度相连	若两个点不属于密度相连关系，则两个点非密度相连

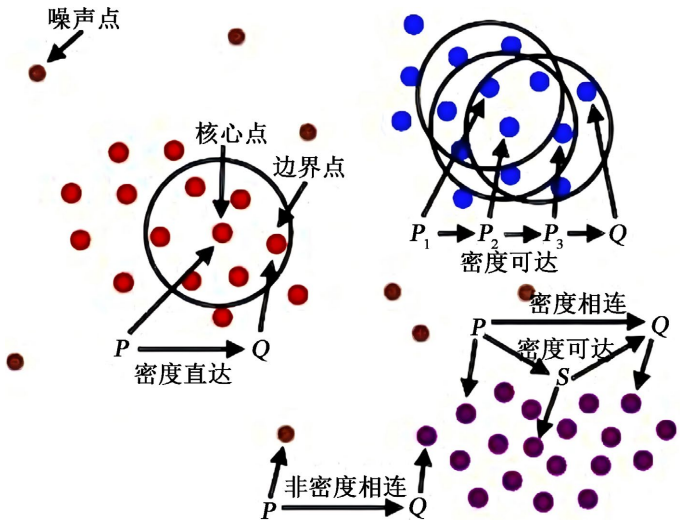

Figure 1. DBSCAN algorithm clustering schematic diagram
图 1. DBSCAN 算法聚类示意图

DBSCAN 算法对于 K-Means 无法发现形状不规则的类簇的缺陷进行了弥补，且可以自动确定类簇个数。DBSCAN 算法不断地搜索核心对象，根据对象间的关系，不断进行簇的生长，最后将具有关系的对象聚类成一个簇。

2.2. DBSCAN 算法步骤

DBSCAN 算法步骤如下：

(1) 给定样本集 $P=\{p_1,p_2,p_3,\cdots,p_n\}$，从中取出一点 p_i，判断 p_i 是否可以成为核心点。即对于一个给定点 p_i，如果在 p_i 半径 Eps 范围内(即 ε 邻域)至少有 MinPts 个点，定义为以 p_i 为核心，以 Eps 为半径的 d 维超球体区域，则这个点 p_i 是一个核心点。数学表达见公式(1)：

$$N_{\text{Eps}}(p)=\{q\in D\mid \text{dist}(p,q)\le \text{Eps}\} \tag{1}$$

式中，$D\subseteq R^d$ 为 d 维实空间上的数据集，$\text{dist}(p,q)$ 表示 D 中的 2 个对象 p 和 q 之间的距离，通常使用欧氏距离。数学表达见公式(2)：

$$\text{dist}(p,q)=\sqrt{\sum_{x=1}^{n}(p_x-q_x)^2} \tag{2}$$

式中 p_x 和 q_x 分别为点 p 和点 q 的第 x 个维度。

若 p_i 不满足核心点定义，则继续从 P 中取点。若 p_i 满足核心点定义，则以 p_i 为核心点形成簇 C。

(2) 从 p_i 的邻域内选取其他点 p_j，判断 p_j 能否作为核心点继续扩张该簇。若 p_j 不满足核心点定义，则 p_j 定义为边界点，不再进一步处理。若 p_j 满足核心点定义，则将 p_j 邻域内的点加入簇 C。数学表达见公式(3)：

$$C=C\cup N_{\in}(p_j) \tag{3}$$

式中 $N_{\in}(p_j)$ 为核心点 p_j 的邻域。

同时再从 C 继续选点进行核心点判断和簇的扩张，直至簇 C 中的点已经全部遍历完成，无法继续扩张为止，完成簇 C 的聚类。

(3) 将簇 C 中的所有点从样本集 P 中去除。数学表达式见公式(4)：

$$P = P \setminus C \tag{4}$$

重复上述步骤，直至样本集 P 中所有点经过判断与聚类，剩余在样本集中的点，如果它们既不满足核心点条件，也没有被任何簇包含，则定义为噪声点。数学表达式见公式(5)、(6)：

$$\left|N_{\epsilon}\left(p_i\right)\right| < \text{MinPts} \tag{5}$$

$$p_i \notin C \tag{6}$$

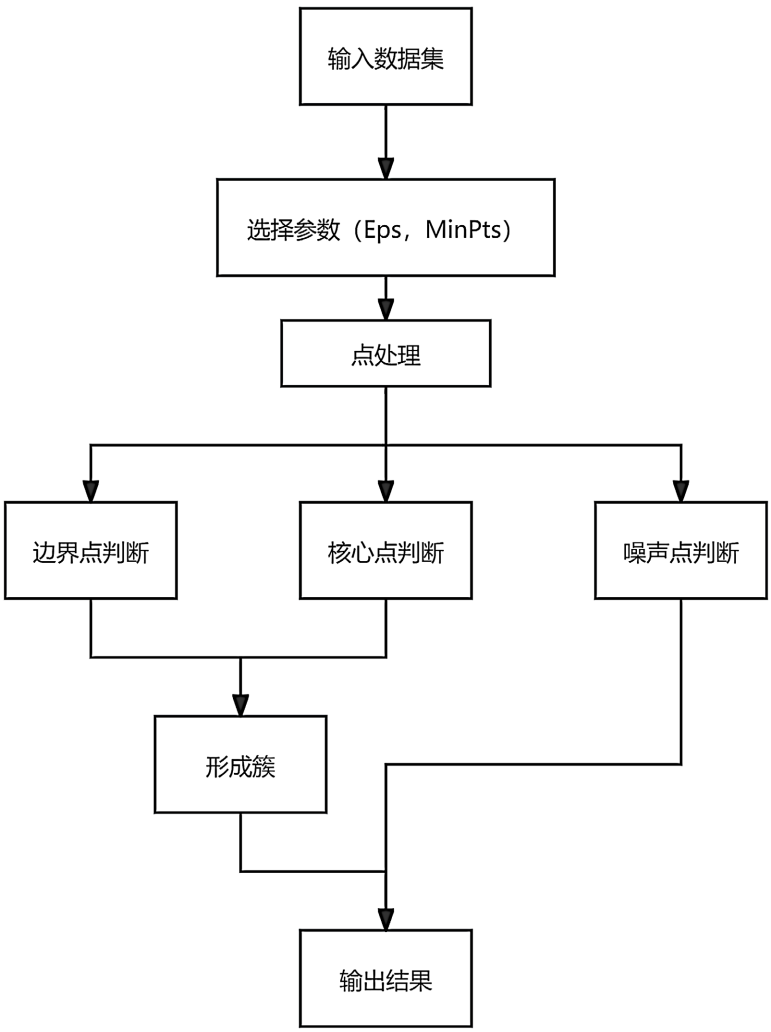

Figure 2. DBSCAN algorithm flowchart
图 2. DBSCAN 算法流程图

2.3. DBSCAN 应用

在农业领域，DBSCAN 算法需要指定 Eps 和 MinPts 参数在实际应用中需要根据具体数据集的特点和需求选择合适的参数和算法，而在灌区划分的应用中，Eps 可以代指单口水井的控制灌溉面积，MinPts 可以代指每亩或指定面积上灌溉水井的数量。通过对参数 Eps 和 MinPts 的调整实现对灌溉区域的初步划分，使相似灌溉特征的区域形成簇。

3. 研究区域、数据来源与处理

3.1. 研究区域

研究区域为某市总面积 2369 平方公里，南北最长距离约 69 千米，东西最长距离约 67 千米。地处南阳盆地，“山少岗多平原广”为其地貌特点，地势西北高、东南低，自西向东呈缓慢倾斜状。属亚热带季风型大陆性气候，受季风转换影响，四季更迭分明，温暖湿润。辖 28 个乡镇(街、区)、626 个行政村(社区)，123.25 万人，253 万亩耕地。研究涉及灌溉井共 83 口，水库 17 座，大小河流共 29 条，主要河流为刁河、湍河、严陵河、赵河。采集包括乡镇中心、道路、水系、行政区划、地形、土壤类型分布、土地利用状况、耕地土壤类型分布、产值、数字高程模型、全域气温降水日照情况、灌溉水井信息等在内的农业相关数据，以构建具有多维度信息的数据库。

3.2. 数据来源

数据获取时间截至 2022 年末。主要分为两部分：第一部分是灌溉井空间数据等，数据来源某市数字化城市课题数据，剔除不符合的类目后共筛选出 83 口灌溉井经纬度坐标，并获得水系分布图、土地利用现状图及耕地土壤分布图。第二部分为基础数据，其中，行政区划数据来源于自然资源部标准地图服务系统(http://bzdt.ch.mnr.gov.cn/)，审图号为 GS(2022)4306；人口数据来源于第七次全国人口普查；水库分布现状及灌区划分现状来源于某市水利局官网(http://www.dengzhou.gov.cn//dzsslj/index.htm)。

3.3. 数据处理

对采集到的农业数据进行整理，其中对于行政区划、灌溉水井信息导入 ArcGIS 10.8 软件获得灌溉水井分布情况信息(图 3)和某市数字正射影像图(Digital Orthophoto Map)数据(图 4)，并用 AutoCAD 2021

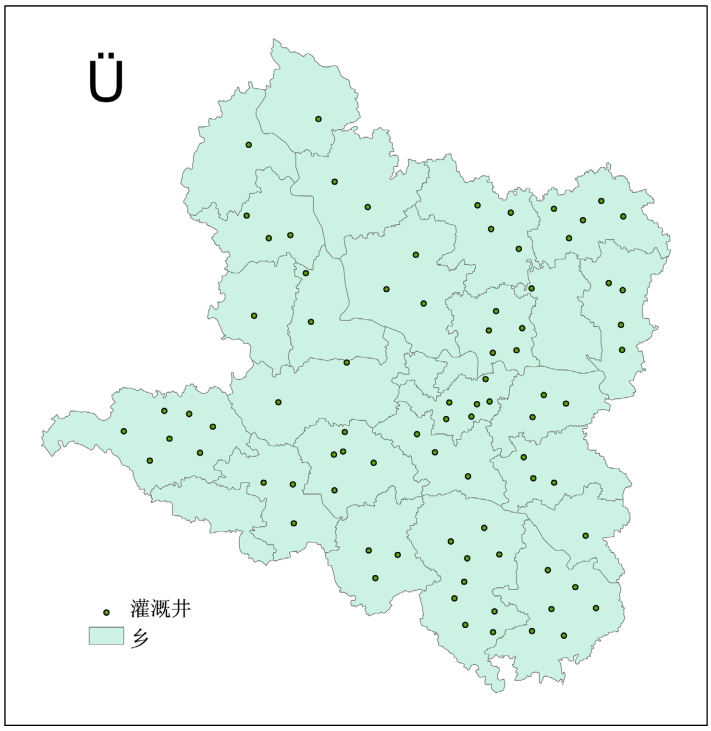

Figure 3. Irrigation wells overall distribution
图 3. 灌溉井整体分布

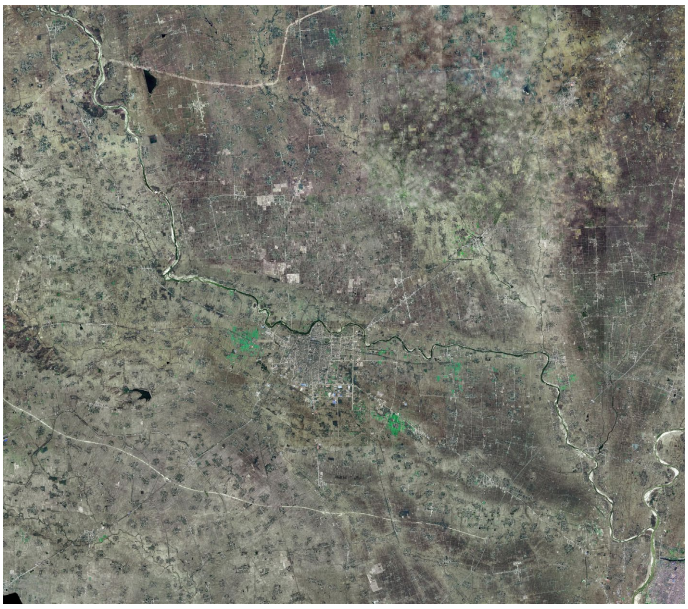

Figure 4. Digital orthophoto map distribution of irrigation wells
图 4. 某市数字正射影像图

软件标注获得灌溉水井的相对坐标信息。对数据进行可视化后发现，灌溉井在空间上呈现整体分散、局部集聚的分布格局。其中，在中部有明显的聚集区，其余地区分布则相对分散。

4. 聚类评估与算法的实现

4.1. 聚类评估

为确保 DBSCAN 算法的可靠性和稳定性，需要使用多个评估指标对聚类结果进行验证。常见的评估指标分为内部指标和外部指标。

其中，内部指标包括轮廓系数、DB 指数、V-measure、类内距离的紧密度；外部指标包括调整兰德指数。

4.1.1. 轮廓系数(Silhouette Coefficient)

轮廓系数是一种用于衡量聚类结果紧密性和分离性的指标。对于每个样本，轮廓系数考虑了该样本与其簇内样本的相似度(a)和与最近簇内其他簇的样本的相似度(b)。轮廓系数的计算方式见公式(7)：

$$S = \frac{b - a}{\max(a, b)} \tag{7}$$

式中，a 为样本到同簇其他点的平均距离，b 为样本到不同簇中所有点的平均距离。

轮廓系数的取值范围在[−1, 1]之间。一个较高的轮廓系数表示样本与其簇内的点相似度高且与其他簇的样本相似度低，表明聚类结果较为合理。反之，轮廓系数较低可能表示聚类结果不够明显。

在实践中，轮廓系数越接近 1，表示聚类效果越好；越接近−1，表示聚类效果越差。当轮廓系数接近 0 时，说明样本在两个相邻簇的边界上。

4.1.2. DB 指数(Davies-Bouldin Index)

DB 指数衡量了簇的紧密度和分离度。具体而言，DB 指数计算每个簇与最近的簇之间的平均相似度，然后取这个值的最大值。DB 指数的计算方式见公式(8)：

$$\mathrm{DB}=\frac{1}{k}\sum_{i=1}^{k}\max_{j\neq 1}\left(\frac{\mathrm{avg\_dissim}\left(C_i\right)+\mathrm{avg\_dissim}\left(C_j\right)}{\mathrm{distance}\left(C_i,C_j\right)}\right) \tag{8}$$

式中，k 为簇的个数，$\mathrm{avg\_dissim}\left(C_i\right)$ 为簇 C_i 内所有点与簇中心的平均距离，$\mathrm{distance}\left(C_i,C_j\right)$ 为簇中心 C_i 和 C_j 之间的距离。

DB 指数的值越小越好，表示簇的紧密度高且簇之间分离度好。

4.1.3. V-Measure

V-measure 是一种用于评估聚类算法效果的指标[18]，同时考虑了聚类的完整性(homogeneity)和分离性(completeness)。V-measure 的计算方式见公式(9)：

$$V=2\times\frac{\mathrm{homogeneity}\times\mathrm{completeness}}{\mathrm{homogeneity}+\mathrm{completeness}} \tag{9}$$

式中，homogeneity 衡量了每个簇只包含单一类别的程度，completeness 衡量了同一类别的所有样本是否都被分配到了同一个簇。

V-measure 的取值范围在[0, 1]之间，值越接近 1 表示聚类效果越好。当 homogeneity 和 completeness 都很高时，V-measure 也会更高。

4.1.4. 类内距离的紧密度

类内距离的紧密度指定义为同一簇内所有点对之间的平均距离，较小的值表示聚类更加紧密。数学表达如公式(10)所示：

$$D_{intra}=\frac{1}{\left|C_k\right|^2}\sum_{i,j\in C_k}d\left(x_i,x_j\right) \tag{10}$$

其中，C_k 为簇 k，$d\left(x_i,x_j\right)$ 为点 x_i 和点 x_j 之间的距离。

4.1.5. 调整兰德指数

调整兰德指数(ARI)是用来衡量聚类结果与真实标签的相似性，见公式(11)：

$$\mathrm{ARI}=\frac{\mathrm{RI}-\mathrm{E}\left[\mathrm{RI}\right]}{\max\left(\mathrm{RI}\right)-\mathrm{E}\left[\mathrm{RI}\right]} \tag{11}$$

其中，RI 为 Rand 指数，$\mathrm{E}\left[\mathrm{RI}\right]$ 为随机聚类时兰德指数的期望值，$\max\left(\mathrm{RI}\right)$ 为最佳情况下兰德指数的最大值，即两个聚类完全一致时的 Rand 指数。

调整兰德直属的范围是[−1, 1]，ARI = 1 表示两个聚类完全一致；ARI = 0 表示两个聚类的相似性等同于随机分配的情况；ARI < 0 表示两个聚类之间的相似性低于随机聚类(即聚类结果差异大于随机情况下的差异)。

4.2. 算法实现

使用 Visual Studio 2019 软件，基于 C#编程语言，编写 DBSCAN 算法与聚类评估方法核心代码，其中评估算法中加入噪声点清洗功能。算法主要进行六项工作：

4.2.1. 数据结构定义

① 定义点(Point)类，包含坐标(X，Y)、簇标识符(ClusterId)和真实标签(TrueLabel)。

② 实现计算两点之间欧几里得距离的方法(DistanceTo)。

具体算法实现：

```
class Point:
    def __init__(self, x, y, trueLabel):
        self.X = x
        self.Y = y
        self.ClusterId = -1
        self.TrueLabel = trueLabel

    def DistanceTo(self, other):
        return sqrt((self.X - other.X)**2 + (self.Y - other.Y)**2)
```

4.2.2. 算法初始化

实例化 DBSCAN 类，传入点集合、邻域半径(_eps)和最小点数(_minPts)。

具体算法实现：

```
class DBSCAN:
    def __init__(self, points, eps, minPts):
        self._points = points
        self._eps = eps
        self._minPts = minPts
```

4.2.3. 簇的形成

(1) 遍历每个点，对于未分类的点(ClusterId == −1)，尝试扩展簇。

(2) 使用 ExpandCluster 方法，首先获得点的邻域内所有点(GetNeigubors)，如果邻域内点数小于 MinPts，则将该点标记为噪声。

(3) 如果邻域内点数满足条件，则将这些点分配到当前簇，并逐一从邻域列表中移除，继续扩展簇。

具体算法实现：

```
def Run(self):
        clusterId = 0
        for point in self._points:
            if point.ClusterId == -1:
                if self.ExpandCluster(point, clusterId):
                    clusterId += 1

    def ExpandCluster(self, point, clusterId):
        seeds = self.GetNeighbors(point)
        if len(seeds) < self._minPts:
            point.ClusterId = 0
            return False

        for seed in seeds:
            seed.ClusterId = clusterId

        seeds.remove(point)

        while seeds:
            currentPoint = seeds[0]
            neighbors = self.GetNeighbors(currentPoint)
            if len(neighbors) >= self._minPts:
                for neighbor in neighbors:
                    if neighbor.ClusterId == -1 or neighbor.ClusterId == 0:
                        if neighbor.ClusterId == -1:
                            seeds.append(neighbor)
                        neighbor.ClusterId = clusterId
            seeds.remove(currentPoint)
        return True
```

```
def GetNeighbors(self, point):
    neighbors = []
    for p in self._points:
        if point.DistanceTo(p) <= self._eps:
            neighbors.append(p)
    return neighbors
```

4.2.4. 簇扩展

(1) 在 ExpandCluster 方法中，对于每个邻域内的点，如果它还未被分类(ClusterId == −1)，将其加入到当前簇，并将其邻域内的点加入待处理队列。

(2) 重复上述过程，直到队列为空。

4.2.5. 簇的输出

通过 PrintClusters 方法输出每个簇的点。

具体算法实现：

```
def PrintClusters(self):
    clusters = self._points.groupby(p.ClusterId).orderby(g => g.Key).toList()
    for cluster in clusters:
        print(f"Cluster {cluster.Key}:")
        for point in cluster:
            print(f"({point.X}, {point.Y})")
```

4.2.6. 性能评估

实现轮廓系数(Calculate Silhouette Coefficient)、戴维斯－布尔丁指数(Calculate Davies-Bouldin Index)、和 V-度量(Calculate VMeasure)的计算方法，以评估聚类结果的质量。

具体算法实现：

```
def CalculateSilhouetteCoefficient(self):
        # Implementation details omitted for brevity
        pass

    def CalculateDaviesBouldinIndex(self):
        # Implementation details omitted for brevity
        pass

    def CalculateVMeasure(self):
        # Implementation details omitted for brevity
        pass
```

4.3. 聚类评估结果

为了系统性地选择 Eps 和 MinPts，我们可以采用网格搜索和 K 折交叉验方法。

网格搜索(Grid Search)是指通过在一个参数范围内进行网格化搜索，逐步遍历 Eps 和 MinPts 的组合，然后根据交叉验证的结果选择最佳参数组合。

网格搜索的过程可以通过优化目标函数来表达，即公式(12)。

$$\arg\min_{\mathrm{Eps},\mathrm{MinPts}}\left(\sum_{i=1}^{K}\mathscr{L}\left(\hat{y}_i,y_i\right)\right) \tag{12}$$

其中，$\mathscr{L}\left(\hat{y}_i,y_i\right)$表示聚类标签的误差(如交叉熵或其他损失函数)，$\hat{y}_i$为预测的标签，$y_i$为真实标签。

同理，可以应用为参数敏感性分析。通过改变 Eps 和 MinPts 的值，观察聚类结果的变化。敏感性分析的目标是检测算法在不同参数下对聚类结果的响应度。

假设我们有一组不同的参数组合$\{(\text{Eps}_1,\text{MinPts}_1),(\text{Eps}_2,\text{MinPts}_2),\cdots\}$，定义一个聚类稳定性度量如公式(15)所示：

$$\text{Stability}(\text{Eps},\text{MinPts})=1-\frac{1}{N}\sum_{i=1}^{N}\left(\frac{\left|C_i^{(\text{Eps},\text{MinPts})}\cap C_i^{(\text{best})}\right|}{\left|C_i^{(\text{Eps},\text{MinPts})}\cup C_i^{(\text{best})}\right|}\right) \tag{15}$$

其中，$C_i^{(\text{Eps},\text{MinPts})}$为使用特定 Eps 和 MinPts 参数得到的第 i 个聚类，$C_i^{(\text{best})}$为最优参数下的第 i 个聚类。

K 折交叉验证(K-fold Cross Validation)是指将数据集划分为 K 个子集，每次用 K-1 个子集进行训练，用剩余的一个子集进行验证，评估聚类性能。通过验证不同参数组合下的平均聚类性能，选择最优参数。

对不同邻域半径 ε 和不同 MinPts 分别带入算法进行计算，得到不同参数下的空间聚类结果及对应轮廓参数、DB 指数、V-measure、类内距离的紧密度、兰德指数(表 2)。另外，进行 MinPts 的选取时应从 2 开始选取，否则评价将不具备意义。

Table 2. Clustering evaluation results of irrigation wells
表 2. 灌溉井聚类评估结果

Eps	MinPts	集群	轮廓系数	DB 指数	V-measure	类内距离的紧密度	兰德指数
1	2	1	NaN	NaN	0	0	NaN
2	2	4	0.606076421760657	0.36234764018707	0.367201007439961	0.676533123331511	0
2	3	1	NaN	NaN	0	0	NaN
3	2	15	0.488219093158278	0.484517506703335	0.902839812370029	0.351539945197723	0
3	3	8	0.544297219884216	0.542184123331462	0.646437674270116	0.494656132665236	0
3	4	1	NaN	NaN	0	0	NaN

4.4. DBSCAN 算法参数选择的优化

4.4.1. Eps 的选择

Eps 表示一个点被视为核心点的最大距离阈值。选择 Eps 的一个有效方法是通过 K 距离图来确定。首先，我们为每个点计算其到第 K 个最近邻的距离，然后根据距离的变化趋势来选择合适的 Eps。

给定一个数据点 x_i 和其 K 个最近邻 $x_{i1},x_{i2},\cdots,x_{iK}$，Eps 的计算见公式(13)：

$$d(x_i,x_{ik})\ \text{ for } k=1,2,\cdots,K \tag{13}$$

其中，$d(x_i,x_{ik})$表示点 x_i 和点 x_j 之间的距离，即欧氏距离见公式(14)：

$$d(x_i,x_{ik})=\sqrt{\left(x_i^1-x_j^1\right)^2+\left(x_i^2-x_j^2\right)^2+\cdots+\left(x_i^n-x_j^n\right)^2} \tag{14}$$

基于 K 距离排序，选择 Eps 为距离急剧变化点对应的距离。具体而言，Eps 应选择为 K 距离图中距离急剧增大的点的值。

应用 C#与 Visual Studio 2019 平台，编写 K 距离代码并带入坐标数据进行 K 距离图的生成，如图 5 所示。

从图中可知，当索引值 = 3 时，K 距离曲线出现了一个巨大的变化趋势，即 Eps = 3。

4.4.2. MinPts 的选择

MinPts 表示一个点被视为核心点所需的最小邻居数。结合 3.3.1 中 Eps 值的选取，对照表 2，综合考

虑轮廓系数较小、DB 指数较小、V-measure 较大，在 Eps = 3 的条件下选取类内距离的紧密度最小的组合，即 Eps = 3，MinPts = 2。这种选择方式可以确保密度足够高，能够形成聚类。

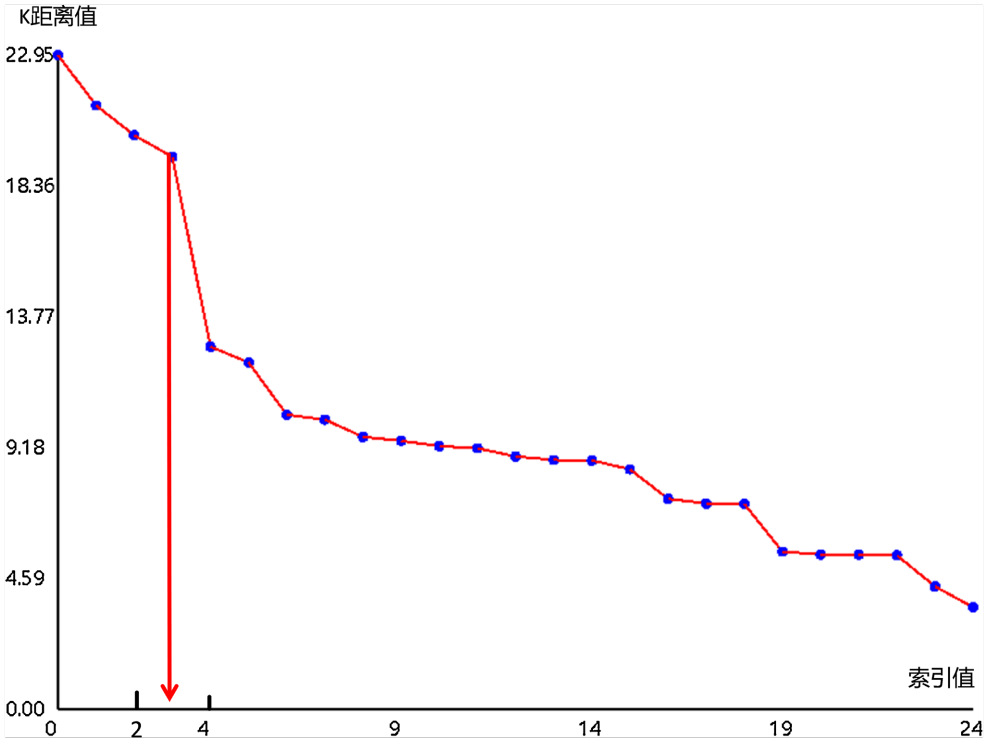

Figure 5. K-distance graph
图 5. K 距离图

4.5. 空间聚类结果

基于 C#语言编程，将灌溉井坐标数据导入，对算法进行初步实验，根据 3.3 所述，选择邻域半径 Eps = 3，簇的最小点选择 MinPts = 2 为最佳参数，输出灌溉井的空间聚类集群。从聚类 0 开始计算，共分为 15 个集群，如表 3 所示，并利用 ArcGIS 10.8 软件实现可视化，如图 6 所示。灌溉井空间上主要集中在中部，其他县(市)分布较为稀疏。从聚类算法输出结果来看，灌溉井数量超过 5 个的有 3 个，其中以聚类 9 的数量居多。

4.6. 灌区划分完善

根据某市水系分布图、土地利用现状分布图、耕地土壤分布图、道路区划图，如图 7、8、9 所示，结合灌溉井聚类结果，首先考虑本次灌区划分的大致原则并对灌区进行如下完善：

4.6.1. 完善原则

(1) 划分应均匀分布，且以水库河流为首要依托。当水库河流无法与集群簇综合分析时，水库河流可以作为唯一因素。计算可依据欧几里得距离公式，见公式(10)：

$$D = \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2} \tag{10}$$

式中，(x_1, y_1) 和 (x_2, y_2) 分别为集群簇和水库或河流的坐标。

(2) 划分应使得灌区位于耕地及人口相对集中区，对于松散分布的人口与耕地，应将灌区划分到相对

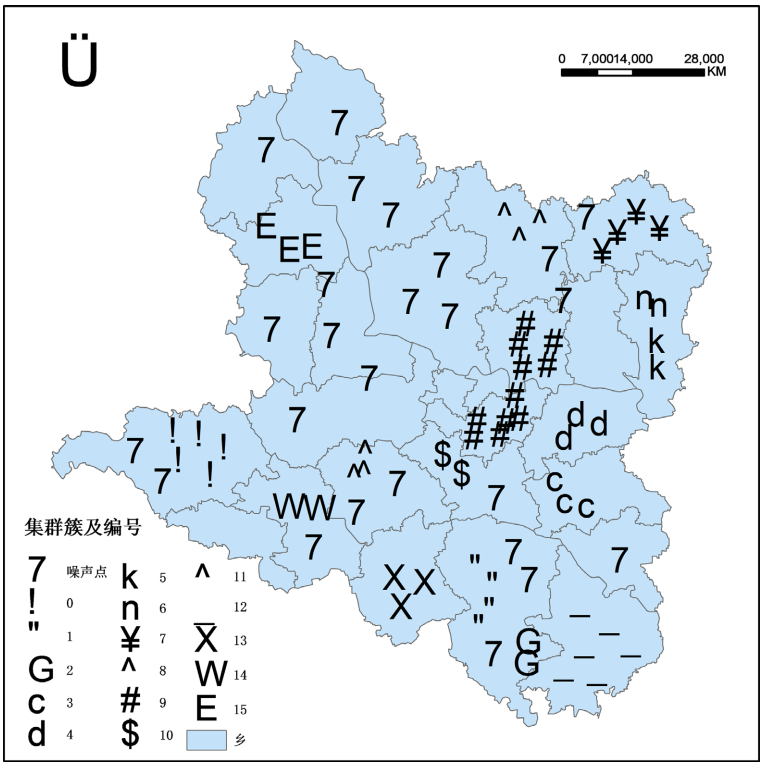

Figure 6. Clustered distribution of irrigation wells
图 6. 灌溉井聚类分布

Table 3. Clustering results of irrigation wells
表 3. 灌溉井聚类结果

聚类	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	噪声点
数量/个	5	3	2	3	3	2	2	4	3	11	2	3	6	3	2	3	24

中心位置。计算可依据：

人口和耕地密度计算公式，见公式(11)：

$$\text{Density} = \frac{\text{Population or Cultivated Land Area}}{\text{Area}} \tag{11}$$

式中，Population or Cultivated Land Area 为人口或耕地总数，Area 为区域面积。

中心公式，见公式(12)：

$$\text{Center} = \left(\frac{\sum_{i=1}^{N} q_i x_i}{\sum_{i=1}^{N} q_i}, \frac{\sum_{i=1}^{N} q_i y_i}{\sum_{i=1}^{N} q_i} \right) \tag{12}$$

式中，q_i 为人口或耕地面积，x_i 和 y_i 为第 i 个点的坐标。

(3) 整体划分应综合考虑集群簇与水系的位置关系、耕地及人口、土地利用状况、耕地土壤分布状况。综合考虑涉及因素，计算可依据综合评价公式，见公式(13)：

$$S = w_1 \cdot D + w_2 \cdot C + w_3 \cdot L + w_4 \cdot So \tag{13}$$

式中，D 为欧几里得距离指数，C 为人口或耕地中心指数，L 为土地利用指数，So 为土壤分布指数，w_1，

w_2， w_3， w_4为相应权重，且满足$\sum_{i=1}^{N} w_i = 1$。

4.6.2. 完善步骤

(1) 水库与集群簇：水库 A 与集群簇 0 为灌区 1、水库 B 与集群簇 15 为灌区 2、水库 D 与集群簇 11 为灌区 3、水库 E 与集群簇 12 为灌区 4。

(2) 河流与集群簇：河流 6、7、8 与集群簇 8 为灌区 5；河流 5 与集群簇 0 为灌区 6；河流 2 与集群簇 10 为灌区 7；河流 3、4 与集群簇 5、6 为灌区 8；河流 9 与集群簇 13 为灌区 9；河流 10 与集群簇 12 为灌区 10；河流 11 与集群簇 3、4 为灌区 11。

(3) 水库：水库 C 为灌区 12、水库 F 为灌区 13、水库 G 为灌区 14。

(4) 河流：河流 1 为灌区 15。

(5) 综合考虑：综合土地利用现状分布(图 8)、耕地土壤分布(图 9)及上述分析，取$w_1 = 0.4$，$w_2 = 0.2$，$w_3 = 0.2$，$w_4 = 0.2$。得到最终划分结果为水库 A 与集群簇 0 为灌区 I、合并灌区 2、13 为灌区 II、合并灌区 12、5 为灌区 III、河流 3、4 与集群簇 5、6 为灌区 IV、河流 1 为灌区 V、水库 D 与集群簇 11 为灌区 VI、河流 5 与集群簇 0 为灌区 VII、水库 G 为灌区 VIII、河流 9 与集群簇 13 为灌区 IX、河流 2 与集群簇 10 为灌区 X、水库 E 与集群簇 12 为灌区 XI、11 与集群簇 3、4 为灌区 XII，结果如图 10 所示。

将基于 DBSCAN 的农业灌区划分方法拓展到更多地理信息数据，如本地水系分布、土壤利用现状、耕地土壤分布数据等，以提高划分的全面性和准确性。考虑多源数据融合的方法，以更全面地描述农业区域的特征。

4.7. 传统划分方法

传统灌区划分方法主要依据水源、地形、土壤、气候等因素，结合农业生产需要和社会经济条件，进行综合考虑。以下是传统灌区划分的几个主要方法和步骤：

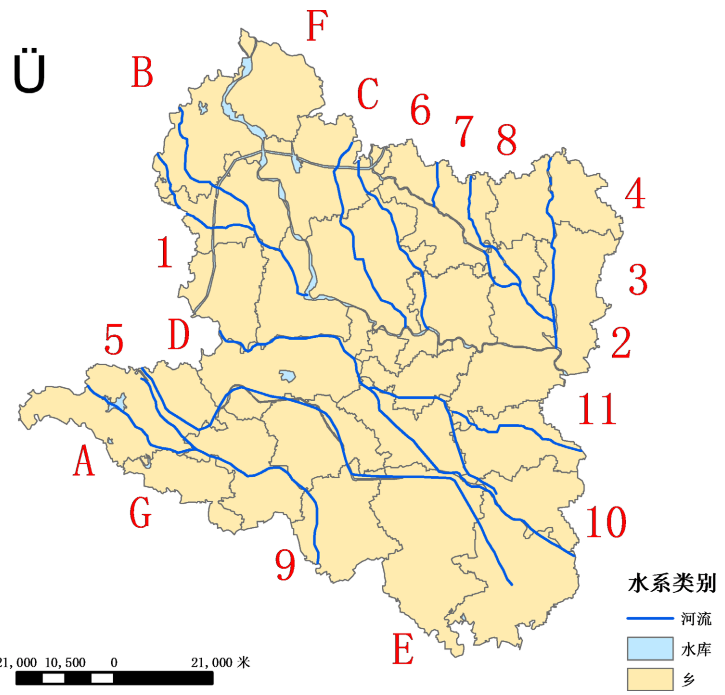

Figure 7. Water system distribution map
图 7. 水系分布图

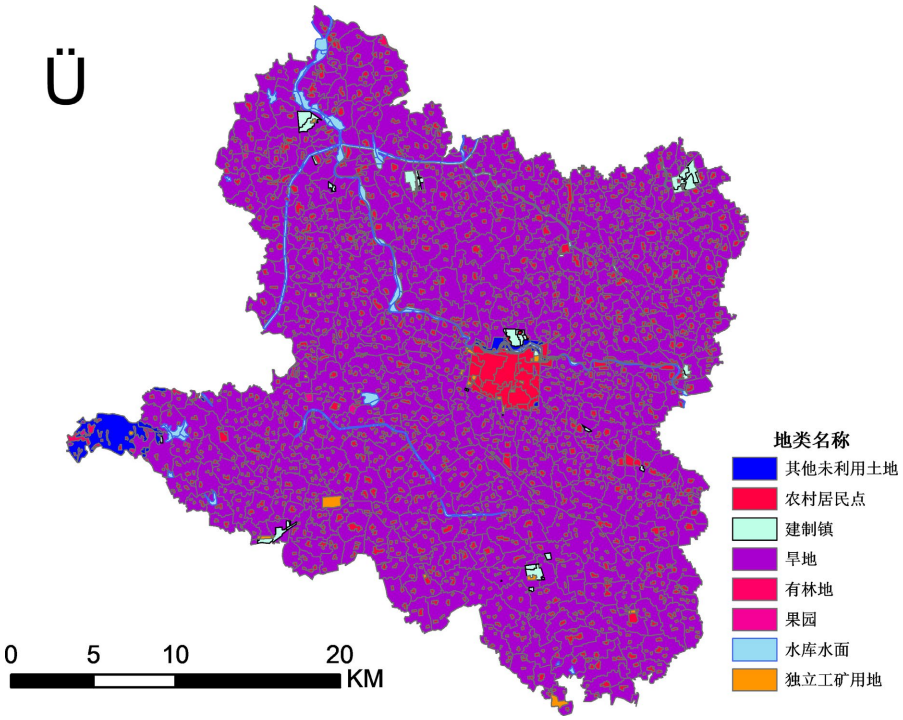

Figure 8. Current status distribution map of land use
图 8. 土地利用现状分布图

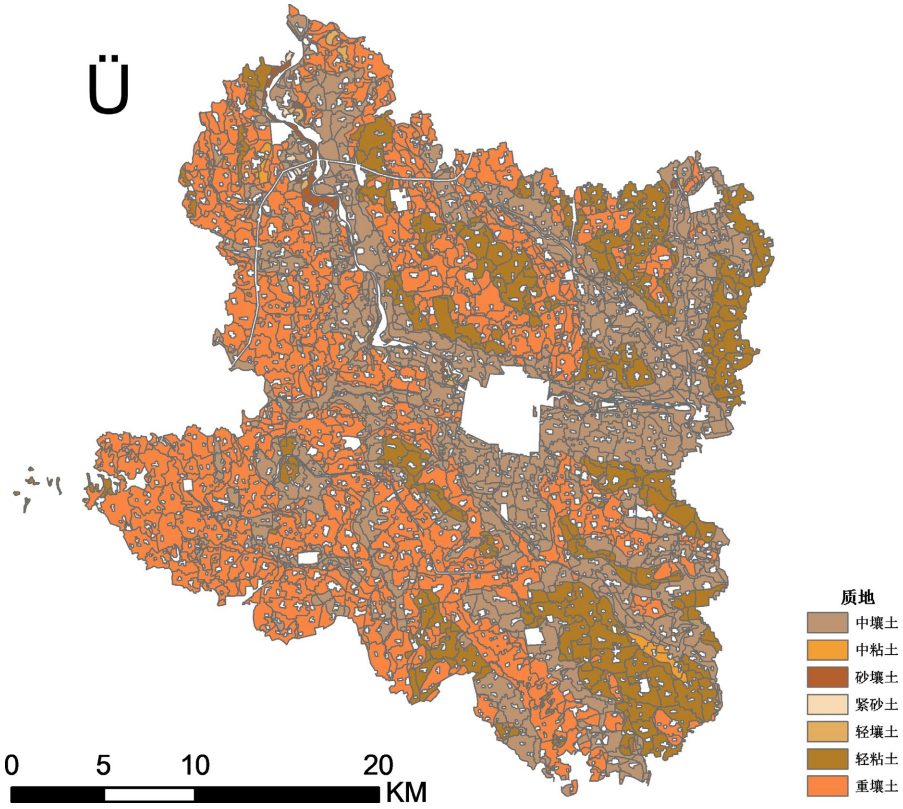

Figure 9. Soil distribution map of cultivated land
图 9. 耕地土壤分布图

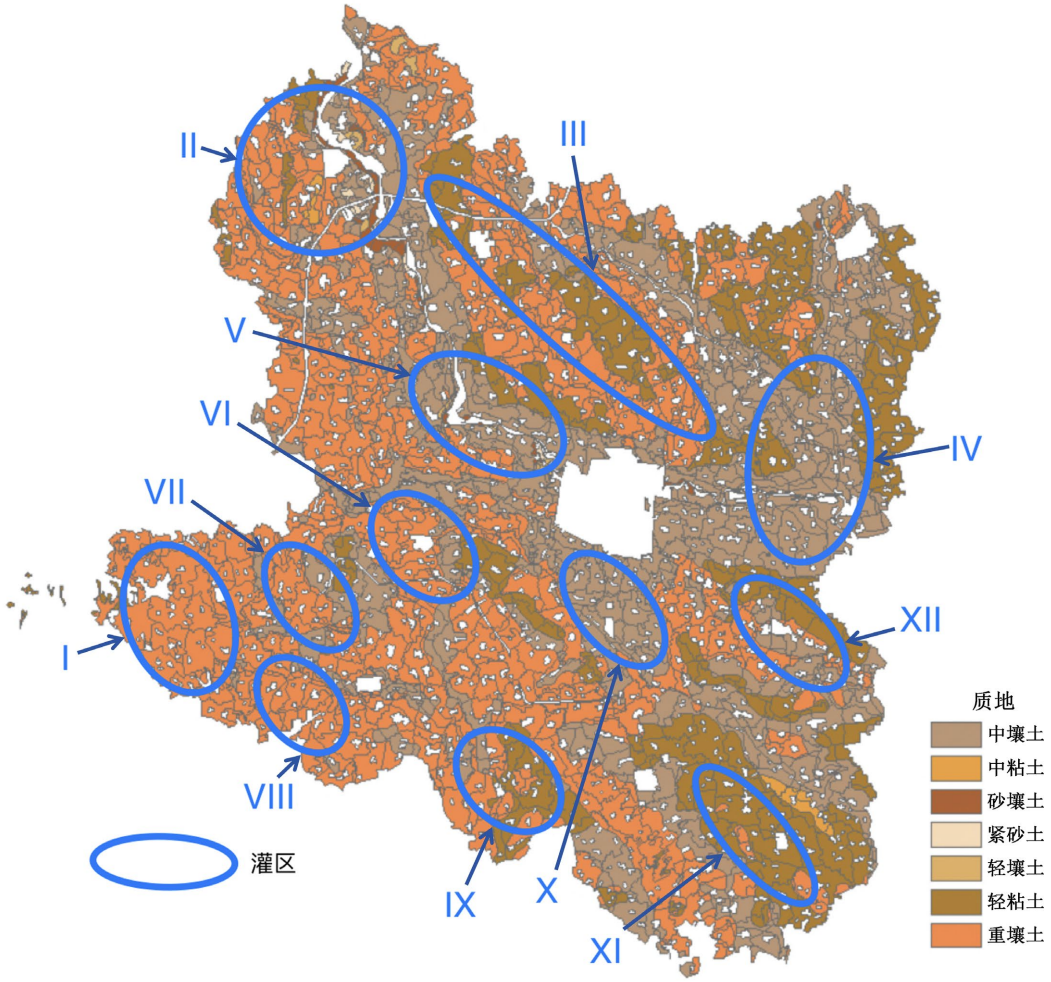

Figure 10. Irrigation area distribution map
图 10. 灌区分布图

(1) 水源分析

灌区划分首先需要确定水源，了解水源的类型(如河流、湖泊、水库、地下水等)、水量、水质和季节变化情况。确保灌区能够获得足够的水资源供给是灌区划分的首要条件。

(2) 地形地貌分析

地形地貌直接影响灌溉系统的设计和水流分布。灌区划分过程中需要对地形进行详细测绘，了解区域的高低起伏、坡度等情况。通常选择地势平坦、水流自然汇聚的区域作为灌区。

(3) 土壤条件分析

土壤类型、质地和肥力影响灌区的生产能力和灌溉需求。对土壤进行采样和分析，确定土壤的渗透性、保水能力和适宜种植的作物类型，是灌区划分的重要环节。

(4) 气候条件分析

气候因素如降雨量、蒸发量、温度和风速等都会影响灌溉需求和灌区划分。通过气象数据分析，评估区域的水资源供需平衡情况，为灌区规划提供依据。

(5) 农业生产需求

根据农业生产的需求，选择适宜种植的作物和相应的灌溉方式，合理规划灌区。考虑到作物的生长

周期和需水特性，确定灌溉频率和灌溉量。

(6) 社会经济条件

灌区划分还需考虑社会经济条件，包括人口分布、经济发展水平、农民的生产习惯和技术水平等。确保灌区规划能够满足当地农民的需求，并有助于提高农业生产效率和农民收入。

(7) 灌溉系统设计

在上述因素综合分析的基础上，进行灌溉系统的设计，包括渠道布置、水源调度、田间工程等。确保灌溉系统能够高效、稳定地运行，实现水资源的合理利用和有效分配。

(8) 环境影响评价

灌区划分还需进行环境影响评价，评估灌溉活动对生态环境的影响，确保灌区规划符合可持续发展的原则，保护区域内的生态环境。

综上所述，传统灌区划分方法涉及多方面的综合考虑，通过科学分析和合理规划，确保灌区的高效运行和农业生产的稳定发展。传统划分方法的某市灌区分布如图 11 所示。

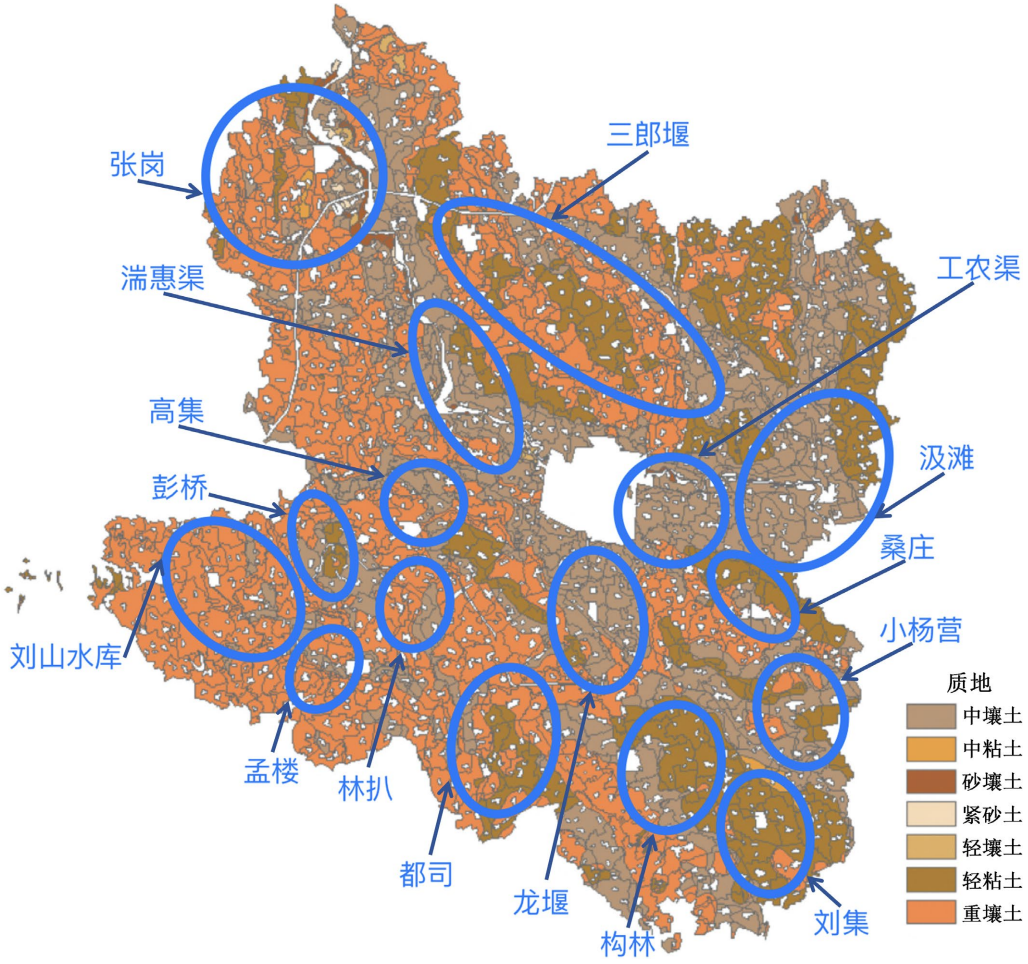

Figure 11. Traditional irrigation area distribution map
图 11. 传统灌区分布图

考虑到传统方法涉及数据庞杂和获取难度较高，在轻量化数据灌区划分中可以考虑基于 DBSCAN 算法的灌区划分方法。

4.8. 对比与分析

4.8.1. 灌区对比

对比基于传统方法的灌区划分得到的某市灌区划分，可以看出基于 DBSCAN 算法的灌区划分方法得到的灌区划分结果与传统方法得到的划分结果基本类似，结果如图 12 所示。

对比结果显示：

(1) 传统灌区分布为 16 处灌区，分别为张岗、湍惠渠、高集、彭桥、刘山水库、孟楼、林扒、都司、龙堰、构林、刘集、小杨营、桑庄、汲滩、工农渠、三郎堰。

(2) 基于 DBSCAN 算法的灌区分布为 12 处灌区，即灌区 I 至灌区 XII。

(3) 对比传统灌区划分，基于 DBSCAN 算法的灌区划分方法缺少 4 处灌区划分，分别为：小杨营、构林、林扒、工农渠。但通过对比图可发现灌区 VI 包含构林及刘集，故缺少的灌区为小杨营、林扒、工农渠三处。

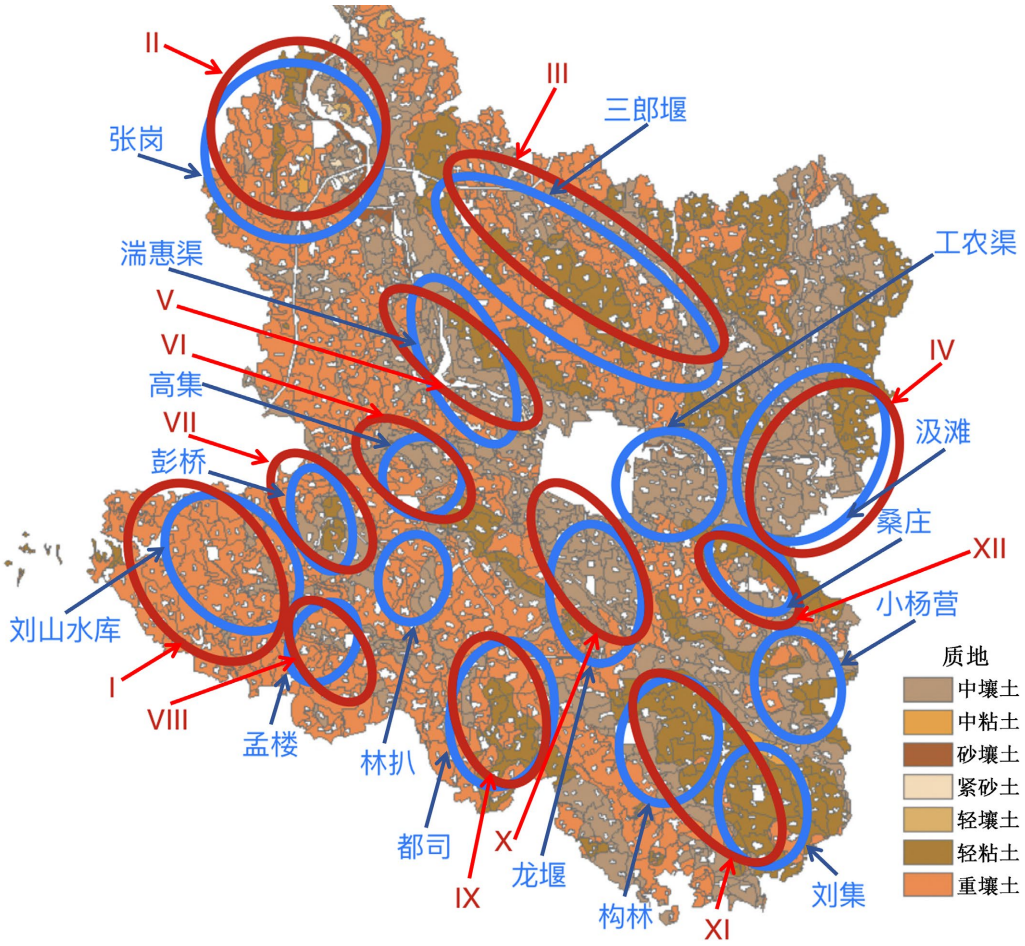

Figure 12. Irrigation area distribution comparison map
图 12. 灌区分布对比图

4.8.2. 灌区精确度分析

(1) 据某市水利局灌区划分数据，将桑庄、小杨营、刘集、构林、龙堰、都司、林扒、孟楼、彭桥、高集十处灌区合并为引丹灌区(大型)，更符合划分要求。即南部小型灌区分布密集，不利于管理，可降低

小杨营、林扒灌区的优先级。

(2) 由于本研究未考虑当地水利设施的分布情况，故未考虑到工农渠所在的工农渠灌区暨橡胶坝管理所(中型)对灌区划分的影响。

(3) 综上所述，剔除小杨营、林扒灌区的影响，基于 DBSCAN 算法的灌区划分方法对比已有灌区划分结果，定义灌区划分准确率($R_{灌区}$)计算方法见公式(14)：

$$R_{灌区}=\frac{N_{\mathrm{DBSCAN}}}{N_T-N_R}\times 100\% \tag{14}$$

式中，N_{DBSCAN} 为基于 DBSCAN 算法的灌区划分个数；N_T 为已有的传统灌区划分个数；N_R 为传统灌区划分中可以不考虑的灌区个数。

计算得灌区划分准确率为 92.86%。

(4) 在两种方法的对比中：

传统方法：

① 需要大量实地调查和数据分析，数据获取和处理的成本较高；② 受制于人工经验和主观判断，结果可能存在较大误差，不同专家可能得出不同的划分结果；③ 对环境变化的适应性较差，需要频繁更新数据和重新规划；通常通过经验和历史数据进行验证，难以进行大规模实验和测试。

基于 DBSCAN 算法的灌区划分方法：

① 可以自动识别数据中的聚类结构，尤其适合处理大规模、高维度数据，并且不需要预先指定聚类的数量；② 通过算法自动处理，可以显著减少数据处理的复杂度和成本；③ 基于密度的自动聚类方法，结果相对客观和稳定，不同次运行结果一致性较高，可以避免人为因素带来的误差；④ 计算速度快，能够在较短时间内完成灌区划分，尤其在处理大规模数据时具有明显的优势；⑤ 可以灵活处理数据的变化，算法对新数据的适应性强，能够快速调整划分结果。

5. 讨论

本文基于 DBSCAN 算法，通过综合运用影响灌区划分的部分原因如灌溉井、河流、水库、耕地土壤类型、土地利用现状，使用 ArcGIS 10.8 及 AutoCAD 2021 进行数据处理，利用 Visual Studio 2019 编写算法及聚类评估代码，并对灌区划分结果与已有传统灌区划分进行对比，得到以下结论：

(1) 对比传统划分方法需要考虑的水源、地形地貌、土壤条件、气候条件、农业生产需求、社会经济条件、灌溉系统设计、环境影响评价八个方面，本研究采用的基于 DBSCAN 算法的灌区划分方法可以在保证准确率(92.86%)的前提下，仅考虑灌溉井、水系、耕地土壤类型、土地利用现状四个方面，即通过更少的工作量完成灌区划分的前期准备工作。

(2) 基于 C#语言的 DBSCAN 算法架构及聚类评估体系，可以有效反映不同要素对灌区划分结果的影响，即可以通过改变坐标点的含义达到侧重性划分的目的。

(3) 算法中的参数 Eps 和 MinPts 可以代表目标点的半径及密度限制，这为以后的新建、续建和维护灌区提供了广泛的适应性。

尽管本研究尽可能地搜集并使用影响灌区划分的各种影响因素数据，但还是有很大一部分数据未能顺利获得，如灌溉水利用率、超采区分布、地下水位分布、化学物质含量等。本研究虽然对灌区划分方法进行算法融合计算分析，但仍存在一些不足：一是由于研究区域较大且聚类因素单一，难以得到各村镇真实的灌区分布，只能为灌区划分的前期准备工作提供参考，加之数据获取难度较大，难以获得十分精准的灌区划分结果。未来研究可尽量获取更丰富的研究数据，多方面地研究。二是因为当前数据的局限性，未对研究区域进行多因素聚类，如除了对灌溉井的聚类，还可以结合水利设施、城镇等进行综合分析。

参考文献

[1] 张运凤, 郭威, 徐建新, 等. 基于最严格水资源管理制度的大功引黄灌区的水资源优化配置[J]. 华北水利水电大学学报(自然科学版), 2015, 36(3): 28-32.

[2] 高占义. 我国灌区建设及管理技术发展成就与展望[J]. 水利学报, 2019, 50(1): 88-96.

[3] 倪文进. 大中型灌区现代化建设需处理好几个问题[J]. 中国水利, 2020(9): 6-7.

[4] 中华人民共和国农业农村部. 关于扩大当前农业农村基础设施建设投资的工作的方案[EB/OL]. https://www.gov.cn/xinwen/2022-10/11/content_5717368.htm, 2022-10-11.

[5] 鄂竟平. 坚定不移践行水利改革发展总基调加快推进水利治理体系和治理能力现代化——在 2020 年全国水利工作会议上的讲话[J]. 中国水利, 2020(2): 1-15.

[6] 张金慧, 王志强, 陈振. 补短板强监管促发展见实效奋力推进农村水利水电高质量发展——访水利部农村水利水电司司长陈明忠[J]. 中国水利, 2019(24): 23-24.

[7] 何雨田, 白美健, 张宝忠, 等. 灌区智慧化发展现状与问题分析[J]. 节水灌溉, 2023(4): 115-121.

[8] 曹冰雪, 李瑾, 冯献, 等. 我国智慧农业的发展现状、路径与对策建议[J]. 农业现代化研究, 2021, 42(5): 785-794.

[9] 孙林, 刘梦含, 徐久成. 基于优化初始聚类中心和轮廓系数的 K-Means 聚类算法[J]. 模糊系统与数学, 2022, 36(1): 47-65.

[10] 王子龙, 李进, 宋亚飞. 基于距离和权重改进的 K-Means 算法[J]. 计算机工程与应用, 2020, 56(23): 87-94.

[11] 彭波, 史春雷, 高万林. DBSCAN 算法优化及在村镇管理决策中的应用[J]. 农业机械学报, 2016, 47(10): 346-350.

[12] 徐雪婷, 孙浩捷, 宋长赢, 等. 伊犁河谷住宿业空间分布特征及影响因素研究[J]. 地域研究与开发, 2024, 43(2): 105-111.

[13] 范习超, 秦京涛, 徐磊, 等. 大型灌区节水水平评价指标体系构建与实证[J]. 农业工程学报, 2021, 37(20): 99-107.

[14] 杨振峰, 田军仓, 冯克鹏, 等. 基于多源数据的宁夏贺兰县现代化生态灌区健康评价[J]. 农业工程学报, 2021, 37(24): 307-316.

[15] 史良胜, 查元源, 胡小龙, 等. 智慧灌区的架构、理论和方法之初探[J]. 水利学报, 2020, 51(10): 1212-1222.

[16] 谢崇宝, 白静, 张武雄, 等. 现代化数字灌区建设主体功能及主要应用结构体系探讨[J]. 中国农村水利水电, 2024(4): 141-145+151.

[17] 魏新光, 王铁良, 李波, 等. 辽宁省玉米地水分盈亏时空分布特征及灌溉模式分区研究[J]. 农业工程学报, 2018, 34(23): 119-126.

[18] Rosenberg, A. and Hirschberg, J. (2007) V-Measure: A Conditional Entropy-Based External Cluster Evaluation Measure. *Proceedings of the* 2007 *Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning* (*EMNLP-CoNLL*), Prague, June 2007, 410-420.