# 《模式识别》课程

# 实 验 报 告



**姓 名： 金家耀**

**专 业：**  人工智能

**学 号： 1193210320**

**江南大学人工智能与计算机学院**

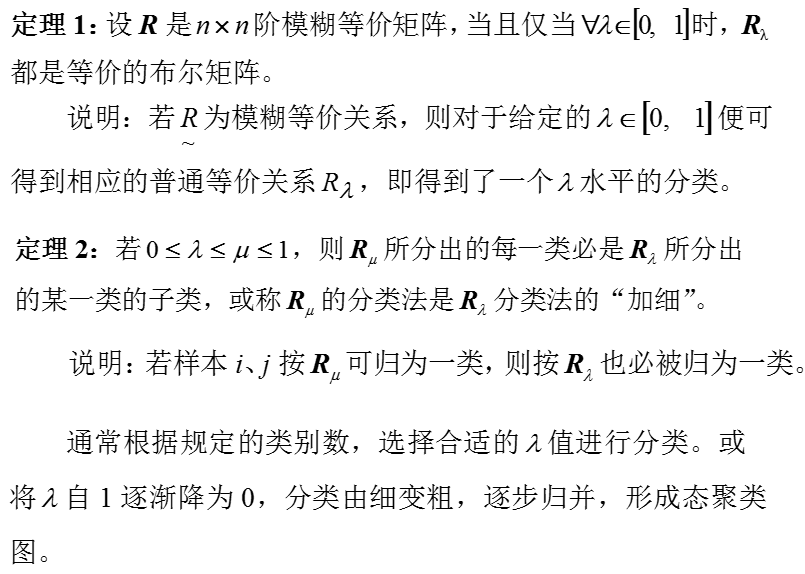
# 基于模糊等价关系的模糊聚类

**1实验目的**

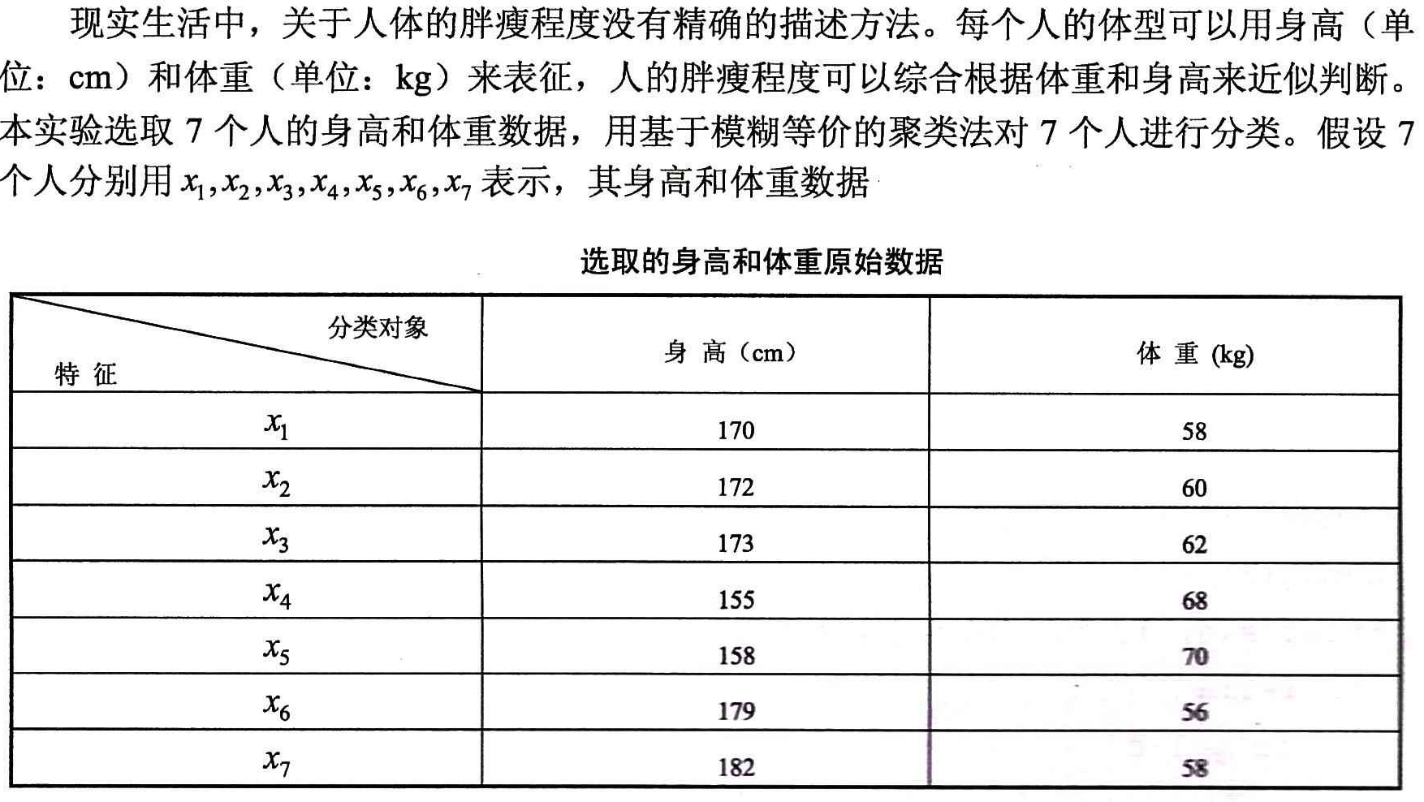
模糊模式识别是将模糊数学的一些概念和方法应用到模式识别领域而形成的一类识别方法。模糊模式识别以隶属度为基础，运用模糊数学中的“关系”概念和运算进行分类。基于模糊等价关系的模糊聚类算法是模糊模式识别中常用的非监督数据处理方法。本实验目的在于加深学生对基于模糊等价关系的模糊聚类原理的理解，掌握算法的实现过程，体会其在模式识别中的作用。

**2实验原理**

对于模糊等价关系，可以用模糊等价矩阵的截矩阵直接进行模式分类。对模糊相似关系，必须由相应的模糊相似矩阵生成模糊等价矩阵，然后对生成的等价矩阵利用截矩阵的办法分类。



**3实验内容**



本实验采用上述数据，利用基于模糊等价关系的模糊聚类算法进行分类实验。

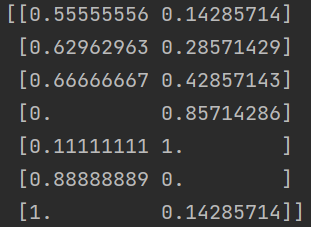
**4实验要求**

1. 建立模糊关系（正规化后，计算*rij*时可采用欧式距离法），生成相应的模糊矩阵；
2. 利用所得的模糊矩阵生成模糊等价矩阵；
3. 选取不同的λ值，对数据进行分类，观察并比较结果。

**5实验代码和结果**

**5.1 正规化**

# 输入数据  
data = np.array([  
 [170, 58],  
 [172, 60],  
 [173, 62],  
 [155, 68],  
 [158, 70],  
 [179, 56],  
 [182, 58]  
])  
  
# 正规化数据  
normalized\_data = (data - data.min(axis=0)) / (data.max(axis=0) - data.min(axis=0))  
print(normalized\_data)



使用最大最小值归一化到[0, 1]，用以消除不同量纲之间的差异，为后续计算距离做准备。

**5.2 使用欧式距离计算模糊矩阵**

# 计算模糊关系矩阵  
def calculate\_fuzzy\_relation\_matrix(data, c):  
 n = len(data)  
 fuzzy\_relation\_matrix = np.zeros((n, n))  
 for i in range(n):  
 for j in range(n):  
 fuzzy\_relation\_matrix[i, j] = np.linalg.norm(data[i] - data[j])  
 max\_distance = np.max(fuzzy\_relation\_matrix)  
 print(max\_distance)  
 fuzzy\_relation\_matrix = (max\_distance - fuzzy\_relation\_matrix) / max\_distance  
 return fuzzy\_relation\_matrix

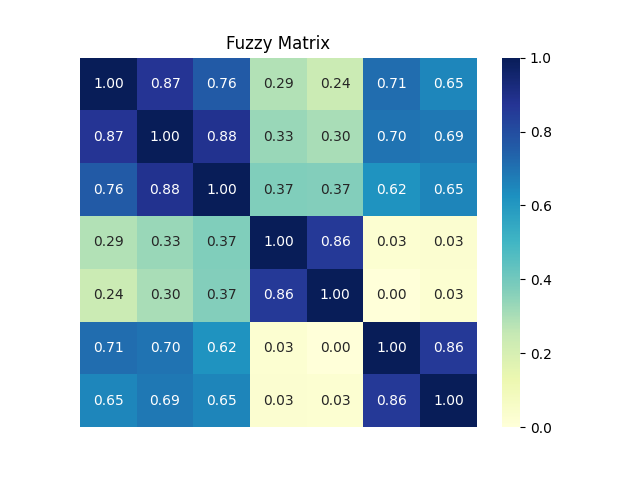


图5-1 模糊关系矩阵

如图5-1所示，为使用欧式距离计算而来的模糊关系矩阵。

**5.3 计算闭包即模糊等价矩阵**

def calculate\_equal\_matrix(R):  
 a = R.shape[0]  
  
 # 创建一个空的 C 矩阵  
 C = np.zeros((a, a))  
  
 while True:  
 for i in range(a):  
 for k in range(a):  
 C[i, k] = np.max(np.min([R[i, :], R[:, k].T], axis=0))  
 if np.array\_equal(R, C):  
 print('模糊等价矩阵：C=\n{}'.format(C))  
 break  
 else:  
 R = C  
 return C

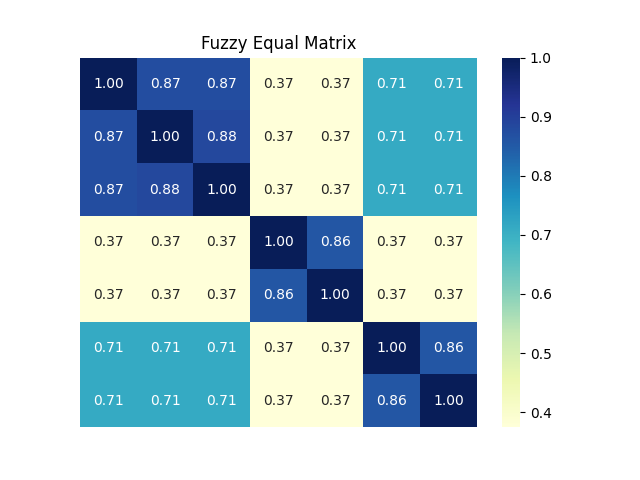


图5-2 模糊等价矩阵

如图5-2所示，为模糊等价矩阵，使用 次闭包，使得。使得同时满足对称性、自反性、传递性的模糊等价矩阵。

**5.4 计算λ截断矩阵并计算聚类**

def lambdaCutMatrix(B, lam):  
 return (B >= lam).astype(int)

def getClusters(B):  
 def find(x):  
 if x == pre[x]:  
 return x  
 else:  
 return find(pre[x])  
  
 a = B.shape[0]  
 pre = [i for i in range(a)]  
 for i in range(a):  
 for j in range(a):  
 if B[i, j] == 1:  
 fx = find(i)  
 fy = find(j)  
 if fx != fy:  
 pre[fy] = fx  
  
 clusters = {}  
 for i in range(a):  
 f = find(i)  
 clusters.setdefault(f, [])  
 clusters[f].append(i)  
  
 for i, value in enumerate(clusters.values()):  
 print(f'Cluster {i + 1}: {value}')

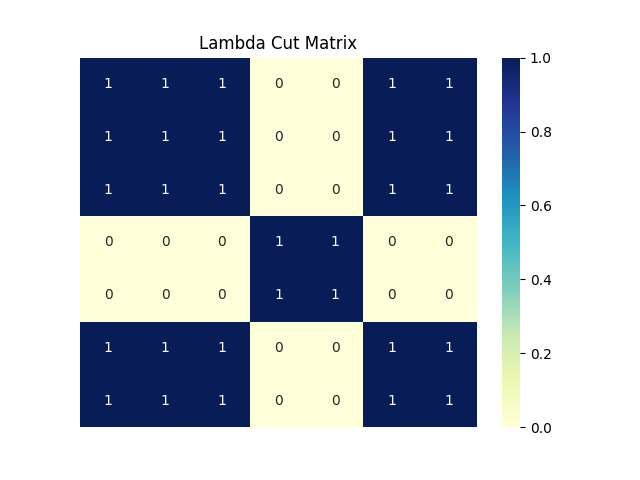


图5-3 截断矩阵

设置 生成阶段矩阵，并使用并查集搜索相同的集合，得到如下结果：



**5.5 生成聚类树**

# 生成聚类树谱图  
agens = AGENS(method='average', pre\_distance=C)  
linkage\_matrix = agens.start()[1]  
plt.figure(figsize=(10, 5))

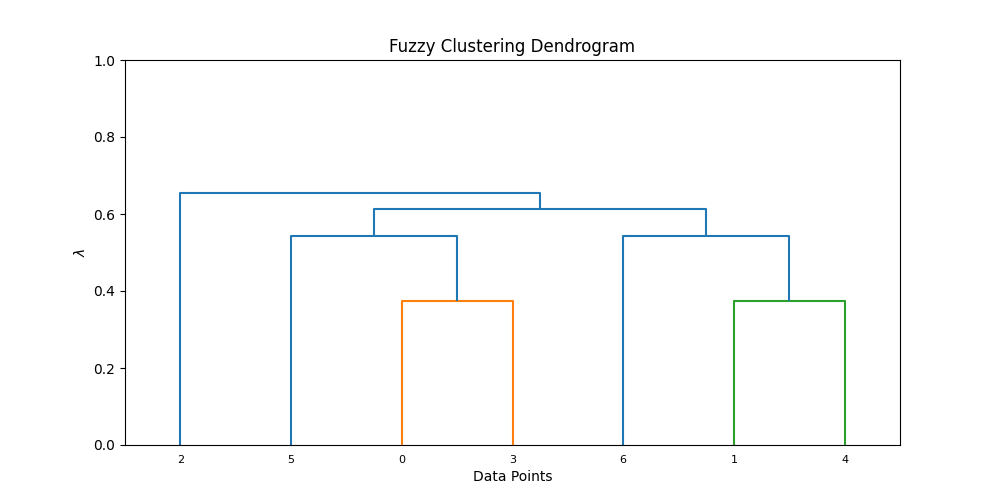


图5-4 聚类树可视化

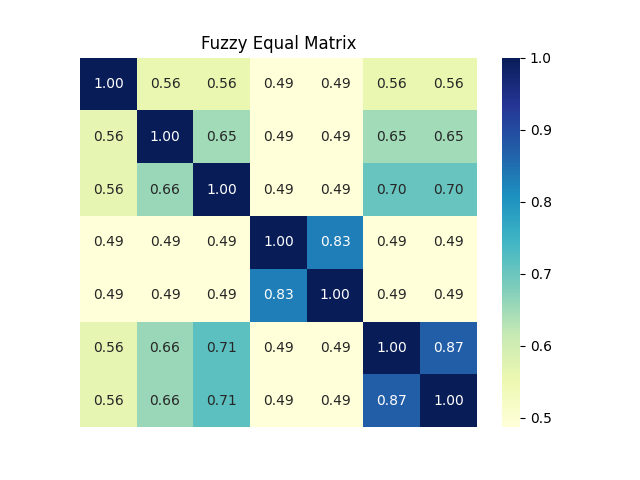
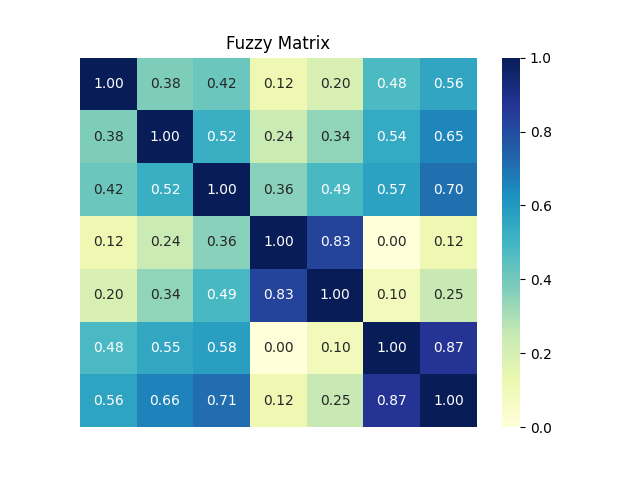
为了更好地可视化不同 的聚类结果，这里使用了模式识别第二次实验的层次聚类自己手写的代码。如图5-4所示，为欧式距离度量下不同 时的聚类结果。

**5.6 不同度量之下的聚类树**

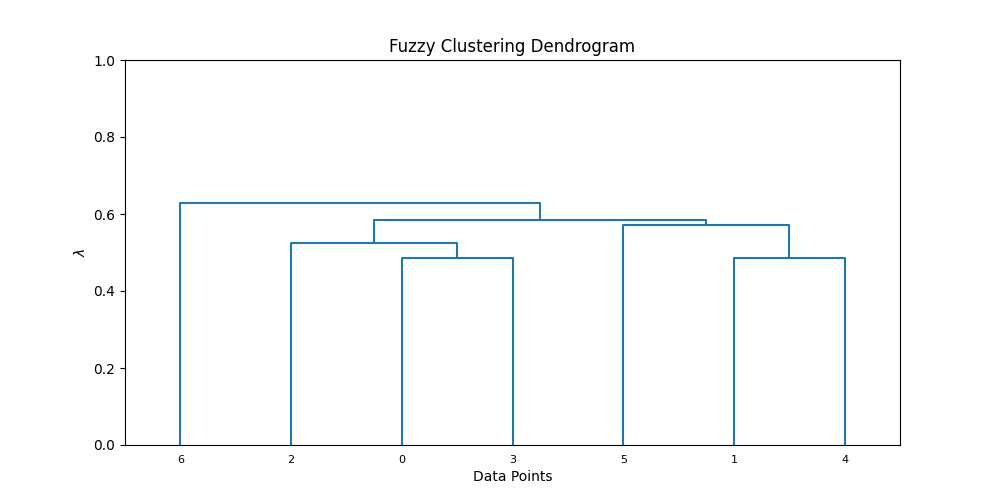
# 计算模糊关系矩阵  
def calculate\_fuzzy\_relation\_matrix(data, metric='euclidean'):  
 n = len(data)  
 fuzzy\_relation\_matrix = np.zeros((n, n))  
  
 for i in range(n):  
 for j in range(n):  
 if metric == 'euclidean':  
 fuzzy\_relation\_matrix[i, j] = np.linalg.norm(data[i] - data[j])  
 max\_distance = np.max(fuzzy\_relation\_matrix)  
 fuzzy\_relation\_matrix = (max\_distance - fuzzy\_relation\_matrix) / max\_distance  
 elif metric == 'dot\_product':  
 fuzzy\_relation\_matrix[i, j] = np.dot(data[i], data[j])  
 max\_distance = np.max(fuzzy\_relation\_matrix)  
 fuzzy\_relation\_matrix = fuzzy\_relation\_matrix / max\_distance

np.fill\_diagonal(fuzzy\_relation\_matrix, 1)  
 elif metric == 'min\_max':  
 min\_values = np.minimum(data[i], data[j])  
 max\_values = np.maximum(data[i], data[j])  
 fuzzy\_relation\_matrix[i, j] = np.sum(min\_values) / np.sum(max\_values)  
 else:  
 raise ValueError(f"Unsupported metric: {metric}")  
  
 return fuzzy\_relation\_matrix

**5.6.1 数量积**

****

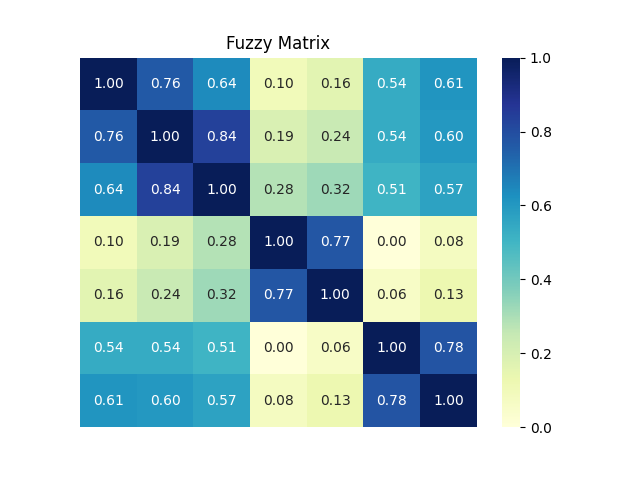
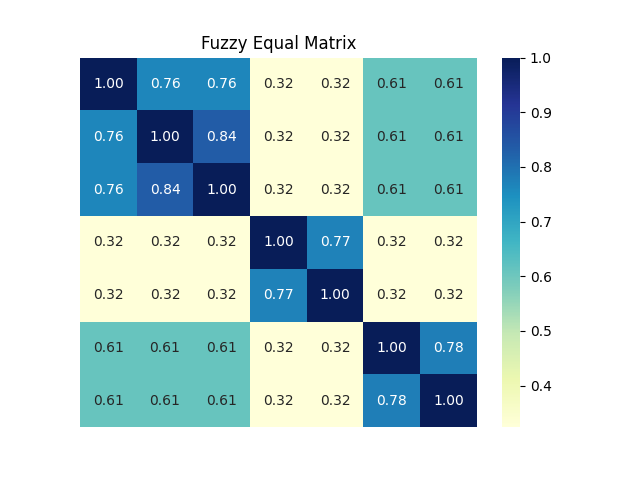
(a) 模糊关系矩阵 (b) 模糊等价矩阵

****

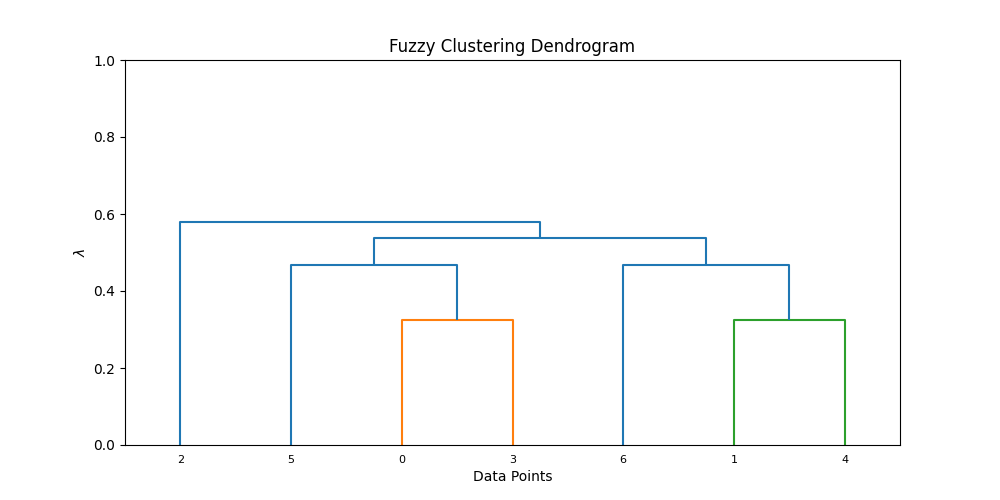
(c) 聚类树

图5-5 数量积度量下的模糊聚类

**5.6.2 最大最小值法**

** **

(a) 模糊关系矩阵 (b) 模糊等价矩阵

****

(c) 聚类树

图5-6 最大最小法度量下的模糊聚类

**6实验心得**

这次进行基于模糊等价关系的模糊聚类实验让我更深入地理解了模糊模式识别的概念和方法。在实验过程中，我首先学会了如何对数据进行正规化处理，通过最大最小值归一化，消除了不同量纲对模糊聚类的影响，为后续的计算提供了基础。

使用欧式距离计算模糊关系矩阵的过程中，我注意到欧式距离法是一种常用的距离度量方式，通过矩阵的计算，能够有效地反映出数据点之间的相似性。生成的模糊关系矩阵为后续的模糊等价关系提供了基础。

在计算模糊等价矩阵时，我深入理解了模糊等价关系的性质，即对称性、自反性、传递性等，这些性质使得模糊等价矩阵在模糊聚类中能够更好地刻画数据点之间的关系。

选择不同的λ值进行截断矩阵并进行聚类的过程中，我深刻体会到λ值对聚类结果的影响。合适的λ值能够更准确地反映数据点之间的相似性，从而获得更合理的聚类效果。通过观察不同λ值下的聚类结果，我对模糊聚类算法的参数调整有了更深入的认识。

最后，通过生成聚类树谱图，我能够直观地看到不同度量方式下的聚类效果。这种可视化方式使得对聚类结果的比较更加直观，也有助于找到合适的度量方式以及参数。