Лабораторная работа № 9

Вариант № 4

Исследование алгоритмов кластеризации

Цель работы

Исследовать методов кластеризации на примере алгоритмов иерархической группировки и k-средних (k-means).

Задание

Реализовать классификацию объектов 3х классов на основе алгоритма k-средних. Выбрать метрику (функцию расстояния), минимизирующую ошибку классификации.

Код программы (внесённые изменения в шаблон кода выделены)

```
clear all; close all;
%1.Исходные данные для генерации образов M порождающих классов
n=2;
M=3;%размерность признакового пространства и число классов
%L - количество компонентов смеси в каждом классе
%dm - параметр, определяющий среднюю степень пересечения компонентов смесей
%romin, romax - границы значений коэффициента корреляции для задания матриц
ковариации
L=ones(1,M);%каждый класс порождается одним гауссовским распределением
dm=4;
romin=-0.9;
romax=0.9;
%Веса, математические ожидания, дисперсии и коэффициенты корреляции компонентов
смесей
ps=cell(1,M);
mM=cell(1,M);
D=cell(1,M);
ro=cell(1,M);
for i=1:M
    ps{i}=ones(1,L(i))/L(i);
    D{i}=ones(1,L(i));
    ro{i}=romin+(romax-romin)*rand(1,L(i));
end
mM{1}=[0;0]; mM{2}=[0;dm]; mM{3}=[dm;0];
% Генерация данных
Ni = 50;
NN = [Ni, Ni, Ni, Ni, Ni];
N = sum(NN); % объемы тестирующих данных
% 1. Исходные данные для генерации образов M порождающих классов (добавлено)
X = gen(n, M, NN, L, ps, mM, D, ro, 0);
Ni = 50;
NN = [Ni, Ni, Ni, Ni, Ni];
N = sum(NN); % объемы тестирующих данных
```

```
% Изменения для создания данных XN
Nmi = 0;
Ns = zeros(1, M);
XN = zeros(N, n);
for i = 1:M
    Nma = Nmi + NN(i);
    Ns(i) = Nma;
    XN(Nmi + 1:Nma, :) = X{i}';
    Nmi = Nma;
end
%2. Тестирование алгоритма с метрикой 'sqEuclidean'
options = statset('Display', 'final', 'MaxIter', 100, 'TolFun', 1e-6);
[idx, ctrs, sumd] = kmeans(XN, M, 'Distance', 'sqEuclidean', 'replicates', 5,
'Options', options);
figure(1); silhouette(XN, idx); % отображение силуэта
%3. Оценка ошибок, визуализация тестовых данных и ошибочных решений
[ercl, idxn, prM] = erclust(M, NN, idx); % оценка ошибок
disp('Индекс качества кластеризации и частость ошибок'); disp([prM, ercl]);
figure; grid on; hold on;
for i = 1:M
    plot(XN(idxn == i, 1), XN(idxn == i, 2), 'o', 'MarkerSize', 10, 'LineWidth', 1);
end
plot(ctrs(:,1), ctrs(:,2), 'k*', 'MarkerSize', 14, 'LineWidth', 2);
```

Результаты выполнения задания

Результаты классификации:

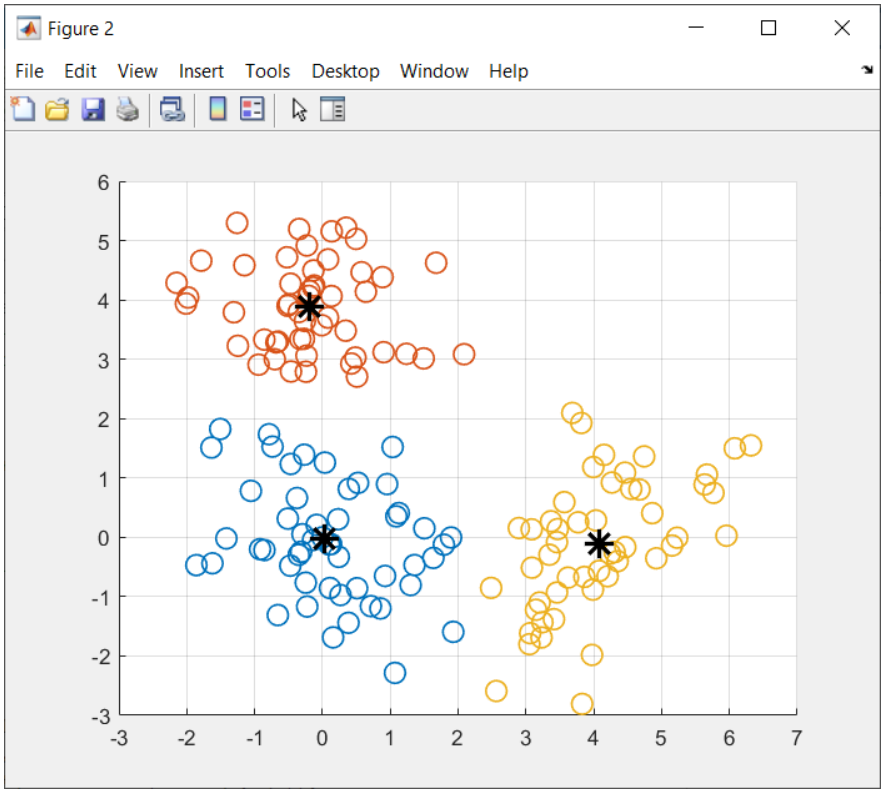

Рисунок 1 – Графическое представление кластеров методом k-средних при M=3;

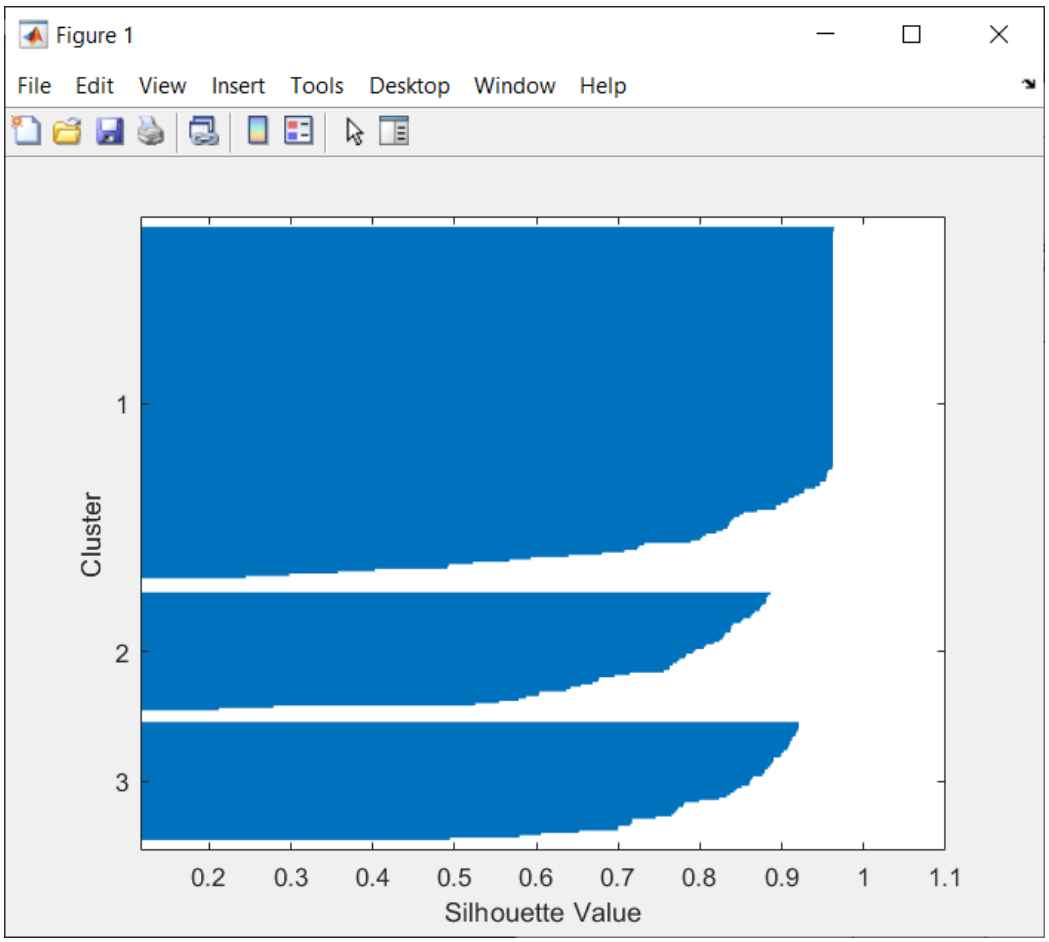

Рисунок 2 – Диаграмма кластеризации методом k-средних при M=3;

```
>> lab9_kmeans
Replicate 1, 2 iterations, total sum of distances = 264.142.
Replicate 2, 2 iterations, total sum of distances = 264.157.
Replicate 3, 3 iterations, total sum of distances = 264.142.
Replicate 4, 2 iterations, total sum of distances = 264.157.
Replicate 5, 3 iterations, total sum of distances = 264.157.
Best total sum of distances = 264.142
Индекс качества кластеризации и частость ошибок
    1.0000    0.0080
```

Рисунок 3 – Результат работы программы

Из результатов мы видим, что на первой попытке алгоритм выполнил 2 итерации, при этом сумма квадратов расстояний получилась равной 264.142.

На второй попытке алгоритм выполнил 2 итерации, при этом увеличив сумму квадратов расстояний до 264.157.

На дальнейших повторах алгоритм получил расстояния меньше или такие же. Значит, именно этот результат и будет лучшим.

"Индекс качества кластеризации" равен 1, что является отличным показателем. "Частость ошибок" равна 0.0080, что означает, что всего около 0,8% данных были классифицированы неверно.

Таким образом, результаты говорят о том, что алгоритм успешно провел кластеризацию с небольшой частотой ошибок, и это лучший результат с минимальной общей суммой расстояний.

Выводы

1. Дендрограмма — это диаграмма, представляющая собой визуализацию иерархии объединения объектов или групп в агломеративных методах кластерного анализа. На дендрограмме объекты или группы представлены в виде ветвей, а расстояния между ними отражают степень их схожести или различия.
2. В алгоритме k-средних минимизируется сумма квадратов расстояний между каждой точкой данных и центроидом ее кластера. Этот критерий известен как "инерция" или "сумма квадратов отклонений". Алгоритм стремится найти такие центроиды для кластеров, чтобы минимизировать общую сумму квадратов расстояний от каждой точки данных до центроида ее кластера.