Путин Павел Александрович, группа 7-1

Лабораторная работа № 9

**Вариант № 6**

Исследование алгоритмов кластеризации

**Цель работы**

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

**Задание**

Получить у преподавателя вариант задания и написать код, реализующий соответствующий алгоритм обработки информации. Для ответа на поставленные в задании вопросы провести численный эксперимент или статистическое имитационное моделирование и представить соответствующие графики. Провести анализ полученных результатов и представить его в виде выводов по проделанной работе.

Реализовать классификацию объектов 5ти классов на основе алгоритма k-средних. Выбрать метрику (функцию расстояния), минимизирующую ошибку классификации.

**Код программы (внесённые изменения в шаблон кода выделены)**

% Файл pr72\_kmeans.Программа для тестирования алгоритма кластеризации

% на основе метода K-means

close all;

%% 1. Исходные данные для генерации образов M порождающих классов

n=2; M=5;%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]; mM{4}=[dm;dm]; mM{5}=[-dm;-dm];

Ni=50; NN=[Ni,Ni,Ni,Ni,Ni]; N=sum(NN); % объемы тестирующих данных

%% 2. Тестирование алгоритма

options=statset('Display','final','MaxIter',100,'TolFun',1e-6);

X=gen(n,M,NN,L,ps,mM,D,ro,0);

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;

[idx,ctrs,sumd] = kmeans(XN,M,'Distance','sqeuclidean','replicates',5,'Options',options);

% idx - индекс принадлежности данных каждому кластеру

% ctrx - центры каждомго кластера

% sumd - сумма квадратов эвклидова расстояния точек внутри каждого кластера до центра

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 (sqeuclidean)'); disp([prM,ercl]);

figure; grid on; hold on;

title('sqeuclidean')

plot(XN(1:Ns(1),1),XN(1:Ns(1),2),'ko','MarkerSize',10,'LineWidth',1);

plot(XN(Ns(1)+1:Ns(2),1),XN(Ns(1)+1:Ns(2),2),'r^','MarkerSize',10,'LineWidth',1);

plot(XN(Ns(2)+1:Ns(3),1),XN(Ns(2)+1:Ns(3),2),'b+','MarkerSize',10,'LineWidth',1);

plot(XN(Ns(3)+1:Ns(4),1),XN(Ns(3)+1:Ns(4),2),'m<','MarkerSize',10,'LineWidth',1);

plot(XN(Ns(4)+1:Ns(5),1),XN(Ns(4)+1:Ns(5),2),'g\*','MarkerSize',10,'LineWidth',1);

plot(XN(idxn==1,1),XN(idxn==1,2),'ko','MarkerSize',10,'LineWidth',1);

plot(XN(idxn==2,1),XN(idxn==2,2),'r^','MarkerSize',10,'LineWidth',1);

plot(XN(idxn==3,1),XN(idxn==3,2),'b+','MarkerSize',10,'LineWidth',1);

plot(XN(idxn==4,1),XN(idxn==4,2),'m<','MarkerSize',10,'LineWidth',1);

plot(XN(idxn==5,1),XN(idxn==5,2),'g\*','MarkerSize',10,'LineWidth',1);

plot(ctrs(:,1),ctrs(:,2),'k\*','MarkerSize',14,'LineWidth',2);

legend('Cluster 1','Cluster 2','Cluster 3','Cluster 4','Cluster 5'); hold off;

% + Добавить кластеризацию с другими метриками

[idx,ctrs,sumd] = kmeans(XN,M,'Distance','cityblock','replicates',5,'Options',options);

figure(1); silhouette(XN,idx); %отображение силуэта

[ercl,idxn,prM] = erclust(M,NN,idx);%оценка ошибок

disp('Индекс качества кластеризации и частость ошибок (cityblock)'); disp([prM,ercl]);

figure; grid on; hold on;

title('cityblock')

plot(XN(1:Ns(1),1),XN(1:Ns(1),2),'ko','MarkerSize',10,'LineWidth',1);

plot(XN(Ns(1)+1:Ns(2),1),XN(Ns(1)+1:Ns(2),2),'r^','MarkerSize',10,'LineWidth',1);

plot(XN(Ns(2)+1:Ns(3),1),XN(Ns(2)+1:Ns(3),2),'b+','MarkerSize',10,'LineWidth',1);

plot(XN(Ns(3)+1:Ns(4),1),XN(Ns(3)+1:Ns(4),2),'m<','MarkerSize',10,'LineWidth',1);

plot(XN(Ns(4)+1:Ns(5),1),XN(Ns(4)+1:Ns(5),2),'g\*','MarkerSize',10,'LineWidth',1);

plot(XN(idxn==1,1),XN(idxn==1,2),'ko','MarkerSize',10,'LineWidth',1);

plot(XN(idxn==2,1),XN(idxn==2,2),'r^','MarkerSize',10,'LineWidth',1);

plot(XN(idxn==3,1),XN(idxn==3,2),'b+','MarkerSize',10,'LineWidth',1);

plot(XN(idxn==4,1),XN(idxn==4,2),'m<','MarkerSize',10,'LineWidth',1);

plot(XN(idxn==5,1),XN(idxn==5,2),'g\*','MarkerSize',10,'LineWidth',1);

plot(ctrs(:,1),ctrs(:,2),'k\*','MarkerSize',14,'LineWidth',2);

legend('Cluster 1','Cluster 2','Cluster 3','Cluster 4','Cluster 5'); hold off;

% + Добавить кластеризацию с другими метриками

[idx,ctrs,sumd] = kmeans(XN,M,'Distance','correlation','replicates',5,'Options',options);

figure(1); silhouette(XN,idx); % отображение силуэта

[ercl,idxn,prM] = erclust(M,NN,idx); % оценка ошибок

disp('Индекс качества кластеризации и частость ошибок (correlation)'); disp([prM,ercl]);

figure; grid on; hold on;

title('correlation')

plot(XN(1:Ns(1),1),XN(1:Ns(1),2),'ko','MarkerSize',10,'LineWidth',1);

plot(XN(Ns(1)+1:Ns(2),1),XN(Ns(1)+1:Ns(2),2),'r^','MarkerSize',10,'LineWidth',1);

plot(XN(Ns(2)+1:Ns(3),1),XN(Ns(2)+1:Ns(3),2),'b+','MarkerSize',10,'LineWidth',1);

plot(XN(Ns(3)+1:Ns(4),1),XN(Ns(3)+1:Ns(4),2),'m<','MarkerSize',10,'LineWidth',1);

plot(XN(Ns(4)+1:Ns(5),1),XN(Ns(4)+1:Ns(5),2),'g\*','MarkerSize',10,'LineWidth',1);

plot(XN(idxn==1,1),XN(idxn==1,2),'ko','MarkerSize',10,'LineWidth',1);

plot(XN(idxn==2,1),XN(idxn==2,2),'r^','MarkerSize',10,'LineWidth',1);

plot(XN(idxn==3,1),XN(idxn==3,2),'b+','MarkerSize',10,'LineWidth',1);

plot(XN(idxn==4,1),XN(idxn==4,2),'m<','MarkerSize',10,'LineWidth',1);

plot(XN(idxn==5,1),XN(idxn==5,2),'g\*','MarkerSize',10,'LineWidth',1);

legend('Cluster 1','Cluster 2','Cluster 3','Cluster 4','Cluster 5'); hold off;

% + Добавить кластеризацию с другими метриками

[idx,ctrs,sumd] = kmeans(XN,M,'Distance','cosine','replicates',5,'Options',options);

figure(1); silhouette(XN,idx); % отображение силуэта

[ercl,idxn,prM] = erclust(M,NN,idx); % оценка ошибок

disp('Индекс качества кластеризации и частость ошибок (cosine)'); disp([prM,ercl]);

% Сделить чтоб Индекс качества кластеризации равнялся 1 и частость ошибок

% была минимальной

figure; grid on; hold on;

title('cosine')

plot(XN(1:Ns(1),1),XN(1:Ns(1),2),'ko','MarkerSize',10,'LineWidth',1);

plot(XN(Ns(1)+1:Ns(2),1),XN(Ns(1)+1:Ns(2),2),'r^','MarkerSize',10,'LineWidth',1);

plot(XN(Ns(2)+1:Ns(3),1),XN(Ns(2)+1:Ns(3),2),'b+','MarkerSize',10,'LineWidth',1);

plot(XN(Ns(3)+1:Ns(4),1),XN(Ns(3)+1:Ns(4),2),'m<','MarkerSize',10,'LineWidth',1);

plot(XN(Ns(4)+1:Ns(5),1),XN(Ns(4)+1:Ns(5),2),'g\*','MarkerSize',10,'LineWidth',1);

plot(XN(idxn==1,1),XN(idxn==1,2),'ko','MarkerSize',10,'LineWidth',1);

plot(XN(idxn==2,1),XN(idxn==2,2),'r^','MarkerSize',10,'LineWidth',1);

plot(XN(idxn==3,1),XN(idxn==3,2),'b+','MarkerSize',10,'LineWidth',1);

plot(XN(idxn==4,1),XN(idxn==4,2),'m<','MarkerSize',10,'LineWidth',1);

plot(XN(idxn==5,1),XN(idxn==5,2),'g\*','MarkerSize',10,'LineWidth',1);

plot(ctrs(:,1),ctrs(:,2),'k\*','MarkerSize',14,'LineWidth',2);

legend('Cluster 1','Cluster 2','Cluster 3','Cluster 4','Cluster 5'); hold off;

**Результаты выполнения задания**

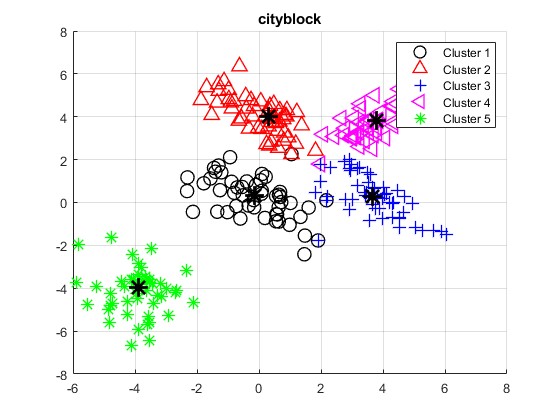


Рисунок 1 - Кластеризация алгоритмом k-means с использованием метрики cityblock

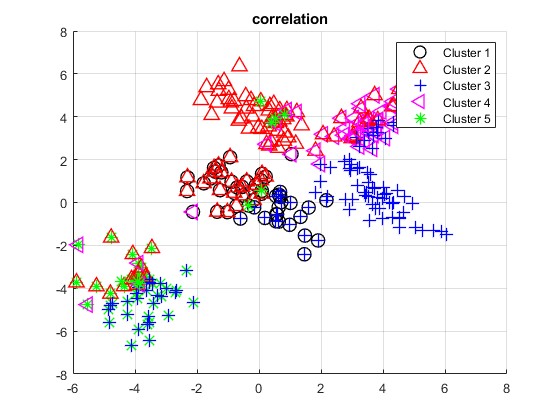


Рисунок 2 - Кластеризация алгоритмом k-means с использованием метрики correlation

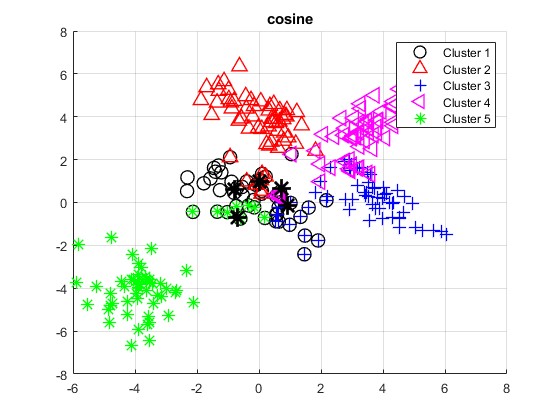


Рисунок 3 - Кластеризация алгоритмом k-means с использованием метрики cosine

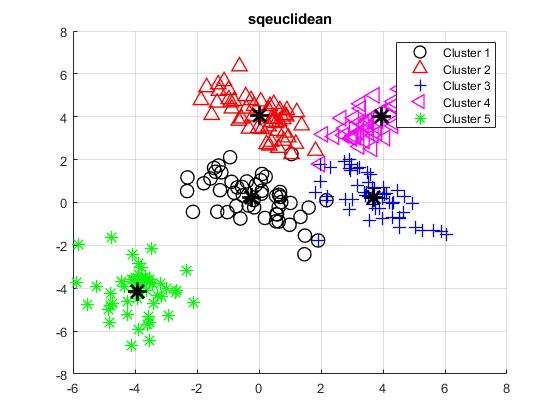


Рисунок 4 - Кластеризация алгоритмом k-means с использованием метрики sqeucledean

Таблица 1 - Анализ кластеризации

|  |  |  |
| --- | --- | --- |
| Метрика | Индекс качества | Частость ошибок |
| sqeuclidean | 1.0000 | 0.0520 |
| cityblock | 1.0000 | 0.0680 |
| correlation | 1.0000 | 0.6480 |
| cosine | 1.0000 | 0.1760 |

Таблица 2 - Определения метрик

|  |  |  |
| --- | --- | --- |
| Метрика | Описание | Формула |
| sqeuclidean | Квадрат евклидова расстояния. Каждый центр тяжести – это среднее значение точек в этом кластере. |  |
| cityblock | Сумма абсолютных разностей, т.е. расстояние L1. Каждый центроид является покомпонентной медианой точек в этом кластере. |  |
| cosine | Единица минус косинус включенного угла между точками (рассматриваемыми как векторы). Каждый центр тяжести представляет собой среднее значение точек в этом кластере после приведения этих точек к единице евклидовой длины. |  |
| correlation | Единица минус выборочная корреляция между точками (рассматриваемыми как последовательности значений). Каждый центроид представляет собой среднее значение по компонентам точек в этом кластере после центрирования и нормализации этих точек до нулевого среднего значения и единицы стандартного отклонения. |  |

# Выводы

1. Лучший результат кластеризации показывает использование евклидовой метрики.
2. Дендрограмма – это полное дерево вложенных кластеров.
3. В алгоритме k-средних минимизируется частость ошибок.