实验报告

1. 实验名称

k-means 聚类

1. 实验内容及数据描述

在Matlab的平台上，实现k-means聚类算法，对给定的数据集进行聚类，并统计聚类正确率。

数据描述如下

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 名称 | 维数 | 样本数 | 类别数 | 训练集大小 | 测试集大小 |
| Iris | 4 | 150 | 3 | 100 | 50 |

1. 实验流程
2. 编写k-mean算法
3. 导入Iris数据集并进行聚类分析
4. 聚类精度计算
5. Matlab代码

1、k-mean.m

function [ idx, ctr ] = k\_means( data, k, iterations )

%{

函数功能：

    对数据实现k-means聚类

参数说明：

    data：待聚类的数据，没有类别信息

    k：期望聚类的类别数目

    iterations：期望的算法的迭代次数（可不给）

算法停止的两个条件满足一个即可：达到预定迭代次数，聚类的质心不再改变或者改变很小。

函数返回：

    idx：数据及其类别标号

    ctr：存储k个聚类中的位置

%}

% m表示数据的规模，n表示数据的维度

[m, n] = size(data);

if k > m

    disp('你需要聚类的数目已经大于数据的数目，无法聚类！');

    return;

end

idx = zeros(m, 1);

ctr = zeros(k, n);

% nargin是用来判断输入变量个数的函数，这样就可以针对不同的情况执行不同的功能。

if nargin == 2

    iterations = 0;

End

% 保存上一次的聚类中心

u = zeros(k, n);

% 保存更新后的聚类中心

c = zeros(k, n);

% 选定初始质心

t = 1;

for i=1:k

    % 初始质心的选取方式为：从第一个数据开始，每隔m/k间隔选取一个数据点，直至得到k个类别中心

    u(i, :) = data(t, :);

    t = t + m/k;

End

iteration = 1;

while true

    % 计算每个数据点到类别中心的距离，把数据点归入到与之最近的类别中

    for i=1:m

        % dis保存每个数据点到k个类别中心的距离

        dis = zeros(k, 1);

        for j=1:k

            % 这里的数据可以是任意维度的，距离度量使用欧式距离

            sum\_dis = 0;

            for t=1:n

                sum\_dis = sum\_dis + (u(j, t) - data(i, t))^2;

            end

            dis(j) = sqrt(sum\_dis);

        end

        % 找出数据点与k个类别中心中，距离最小的一个，该数据点归入到这一类中

        [~, index] = sort(dis);

        idx(i, 1:n) = data(i, :);

        idx(i, n+1) = index(1);

    end

    % 每一次聚类之后应该重新计算类别中心

    for i=1:k

        total\_dis = zeros(1, n);

        num\_i = 0;

        for j=1:m

            if idx(j, 3) == i

                for t=1:n

                    total\_dis(1, t) = total\_dis(1, t) + data(j, t);

                end

                num\_i = num\_i + 1;

            end

        end

        c(i, :) = total\_dis(1, :)/num\_i;

    end

    % 算法结束

    % 给定了迭代次数并且已经迭代了iterations次，退出算法

    if iterations ~= 0 && iteration == iterations

        ctr = c;

        break;

    elseif iterations == 0 && norm(c-u) < 0.01

        ctr = c;

        break;

    end

    iteration = iteration + 1; u = c;

End

end

1. accuraction.m

function [accurate] = accuraction(predict, truth, change)

%{

函数功能：

    计算准确性

参数说明：

    predict：预测的结果

    turth：实际结果

change：重新标记的数据

函数返回：

    accurate：准确率

%}

accuracy = 0;

[n, ~] = size(predict);

for i = 1:n

    if (change(predict(i)) == truth(i))

        accuracy = accuracy + 1;

    end

End

accurate = accuracy / n \* 100;

end

1. main.m

clear;

clc;

load Iris.mat data;

t\_data = data(:, 1:4);

[ idx, ctr ] = k\_means(t\_data, 3, 1);

accurac = [0, 0, 0, 0, 0, 0];

accurac(1) = accuraction(idx(:, 5), data(:, 5), [1, 2, 3]);

accurac(2) = accuraction(idx(:, 5), data(:, 5), [1, 3, 2]);

accurac(3) = accuraction(idx(:, 5), data(:, 5), [2, 1, 3]);

accurac(4) = accuraction(idx(:, 5), data(:, 5), [2, 3, 1]);

accurac(5) = accuraction(idx(:, 5), data(:, 5), [3, 1, 3]);

accurac(6) = accuraction(idx(:, 5), data(:, 5), [3, 2, 1]);

1. 测试结果

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 排列组合 | 1、2、3 | 1、3、2 | 2、1、3 | 2、3、1 | 3、1、2 | 3、2、1 |
| 正确率 | 0.273 | 0.120 | 0.046 | 0.427 | 0.267 | 0.833 |

根据测试结果可知，取最高值，排列组合的顺序为3、2、1，准确率为83.3%，k-means聚类算法符合要求。

1. 实验心得
2. 了解了k-means聚类算法的原理和使用
3. 熟悉了matlab函数的运行和使用