

## ISODATA聚类算法的matlab程序

作者: 凯鲁嘎吉 - 博客园 <http://www.cnblogs.com/kailugaji/>

参考: [Kmeans及ISODATA算法的matlab实现](#)

算法简介: 聚类算法: ISODATA算法

数据见：[MATLAB实例：PCA降维](#)中的iris数据集，保存为：iris.data，最后一列是类标签。

## demo\_isodata.m

```
clear
clc
data_load=dlmread('iris.data');
[~,dim]=size(data_load);
x=data_load(:,1:dim-1);
K=3;
theta_N=1;
theta_S=1;
theta_c=4;
L=1;
I=5;
ISODATA(x,K,theta_N,theta_S,theta_c,L,I)
```

# ISODATA.m

[illegible]

```

%% step1
n = size(x,1);
N_c = K;
mean = cell(K,1);
for i=1:K
    mean{i} = x(i,:);
end
ite = 1;
while ite<I
    flag = 1;
    while flag
        %% step2
        class = cell(size(mean));
        for i=1:n
            num = Belong2(x(i,:),mean);
            class{num} = [class{num};x(i,:)];
        end
        %% step3
        for i=1:N_c
            size_i = size(class{i},1);
            if size_i<theta_N
                class_i = class{i};
                mean = DeleteRow(mean,i);
                class = DeleteRow(class,i);
                N_c = N_c-1;
                for j=1:size_i
                    class_ij = class_i(j,:);%the j'th row of class{i}
                    num = Belong2(class_ij,mean);
                    class{num} = [class{num};class_ij];
                end
            end
        end
        end

        %% step4
        for i=1:N_c
            if ~isempty(mean{i})
                mean{i} = sum(class{i})./size(class{i},1);
            end
        end
        end
        %% step5
        Dis = zeros(N_c,1);
        for i=1:N_c
            if ~isempty(class{i})
                N_i =size(class{i},1);
                tmp = bsxfun(@minus,class{i},mean{i});
                Dis(i) = sum(arrayfun(@(x)norm(tmp(x,:)),1:N_i))/N_i;
            end
        end
    end
end

```

```

end
%% step6
D = 0;
for i=1:N_c
    if ~isempty(class{i})
        N_i =size(class{i},1);
        D = D + N_i*Dis(i);
    end
end
D = D/n;
%% step7
flag = 0;
if ite == I
    theta_c = 0;
    flag = 0;
elseif ~(N_c > K/2)
    flag = 1;
elseif mod(ite,2)==0 || ~(N_c<2*K)
    flag = 0;
end
%% 分裂处理
%% step8
if flag
    flag = 0;
    delta = cell(N_c,1);
    for i=1:N_c
        if ~isempty(class{i})
            N_i =size(class{i},1);
            tmp = bsxfun(@minus,class{i},mean{i});
            delta{i} = arrayfun(@(x) norm(tmp(:,x)),1:size(tmp,2))/N_i;
        end
    end
end

%% step9
delta_max = cell(N_c,1);
for i=1:N_c
    if ~isempty(class{i})
        max_i = max(delta{i});
        sub = find(delta{i}==max_i,1);
        delta_max{i} = [max_i,sub];
    end
end
%% step10
for i=1:N_c
    if delta_max{i}(1) > theta_S
        N_i =size(class{i},1);
        con1 = (Dis(i)>D && N_i>2*(theta_N + 1));
    end
end

```

```

con2 = ~(N_c>K/2);
if con1 || con2
    %%%%这里分裂%%%%%%%%
    flag = 1;%一旦发生分裂，那么分裂一次后就返回第二步；若没发生分裂，则直接进入合并处理步
    lamda = 0.5;
    max_sub = delta_max{i}(2);
    mean{i}(max_sub) = mean{i}(max_sub) + lamda * delta_max{i}(1);
    addOneMean = mean{i};
    addOneMean(max_sub) = addOneMean(max_sub) - lamda * delta_max{i}(1);
    mean = [mean;addOneMean];
    N_c = N_c+1;
    break;
end
end
end

end

end
%% 合并处理
if L
    %% step11
    Distance = zeros(N_c,N_c);
    for i=1:N_c-1
        for j=i:N_c
            Distance(i,j) = norm(mean{i}-mean{j});
        end
    end
    %% step12
    index = find(-Distance>theta_c);
    keepIndex = [Distance(index),index];
    [~, index] = sort(keepIndex(:,1));
    if size(index,1) > L
        index = index(1:L,:);
    end
    %% step13
    if size(index,1) ~= 0
        for id=1:size(index,1)
            [m_i m_j]= seq2idx(index(id),N_c);
            %%%%这里合并%%%%%%%%
            N_mi = size(class{m_i},1);
            N_mj = size(class{m_j},1);
            mean{m_i} = (N_mi*mean{m_i} + N_mj*mean{m_j})/(N_mi+N_mj);
            mean = DeleteRow(mean,m_j);
            class{m_i} = [class{m_i};class{m_j}];
            class = DeleteRow(class,m_j);
        end
    end
end

```

```

        end
    end
    %% step14
    ite=ite+1;
end
for i=1:N_c
    fprintf(' 第%d类聚类中心为\n', i);
    disp(mean{i});
    fprintf(' 第%d类中元素为\n', i);
    disp(class{i});
end
end

```

%%%

```

function number = Belong2(x_i, means)
    INF = 10000;
    min = INF;
    kk = size(means, 1);
    number = 1;
    for i=1:kk
        if ~isempty(means{i})
            if norm(x_i - means{i}) < min
                min = norm(x_i - means{i});
                number = i;
            end
        end
    end
end

```

```

function A_del = DeleteRow(A, r)
    n = size(A, 1);
    if r == 1
        A_del = A(2:n, :);
    elseif r == n
        A_del = A(1:n-1, :);
    else
        A_del = [A(1:r-1, :); A(r+1:n, :)];
    end
end

```

```

function [row col] = seq2idx(id, n)
    if mod(id, n) == 0
        row = n;
        col = id/n;
    else

```

```
        row = mod(id,n);
        col = ceil(id/n);
    end
end
```

# 结果

```
>> demo_isodata
```

第1类聚类中心为

6.6016	2.9857	5.3841	1.9159
--------	--------	--------	--------

第1类中元素为

7.0000	3.2000	4.7000	1.4000
6.4000	3.2000	4.5000	1.5000
6.9000	3.1000	4.9000	1.5000
6.5000	2.8000	4.6000	1.5000
6.3000	3.3000	4.7000	1.6000
6.6000	2.9000	4.6000	1.3000
6.7000	3.1000	4.4000	1.4000
5.9000	3.2000	4.8000	1.8000
6.3000	2.5000	4.9000	1.5000
6.6000	3.0000	4.4000	1.4000
6.8000	2.8000	4.8000	1.4000
6.7000	3.0000	5.0000	1.7000
6.0000	2.7000	5.1000	1.6000
6.7000	3.1000	4.7000	1.5000
6.3000	3.3000	6.0000	2.5000
5.8000	2.7000	5.1000	1.9000
7.1000	3.0000	5.9000	2.1000
6.3000	2.9000	5.6000	1.8000
6.5000	3.0000	5.8000	2.2000
7.6000	3.0000	6.6000	2.1000
7.3000	2.9000	6.3000	1.8000
6.7000	2.5000	5.8000	1.8000
7.2000	3.6000	6.1000	2.5000
6.5000	3.2000	5.1000	2.0000
6.4000	2.7000	5.3000	1.9000
6.8000	3.0000	5.5000	2.1000
5.7000	2.5000	5.0000	2.0000
5.8000	2.8000	5.1000	2.4000
6.4000	3.2000	5.3000	2.3000
6.5000	3.0000	5.5000	1.8000
7.7000	3.8000	6.7000	2.2000
7.7000	2.6000	6.9000	2.3000
6.0000	2.2000	5.0000	1.5000

6.9000	3.2000	5.7000	2.3000
5.6000	2.8000	4.9000	2.0000
7.7000	2.8000	6.7000	2.0000
6.3000	2.7000	4.9000	1.8000
6.7000	3.3000	5.7000	2.1000
7.2000	3.2000	6.0000	1.8000
6.2000	2.8000	4.8000	1.8000
6.1000	3.0000	4.9000	1.8000
6.4000	2.8000	5.6000	2.1000
7.2000	3.0000	5.8000	1.6000
7.4000	2.8000	6.1000	1.9000
7.9000	3.8000	6.4000	2.0000
6.4000	2.8000	5.6000	2.2000
6.3000	2.8000	5.1000	1.5000
6.1000	2.6000	5.6000	1.4000
7.7000	3.0000	6.1000	2.3000
6.3000	3.4000	5.6000	2.4000
6.4000	3.1000	5.5000	1.8000
6.0000	3.0000	4.8000	1.8000
6.9000	3.1000	5.4000	2.1000
6.7000	3.1000	5.6000	2.4000
6.9000	3.1000	5.1000	2.3000
5.8000	2.7000	5.1000	1.9000
6.8000	3.2000	5.9000	2.3000
6.7000	3.3000	5.7000	2.5000
6.7000	3.0000	5.2000	2.3000
6.3000	2.5000	5.0000	1.9000
6.5000	3.0000	5.2000	2.0000
6.2000	3.4000	5.4000	2.3000
5.9000	3.0000	5.1000	1.8000

第2类聚类中心为

5.6838	2.6784	4.0919	1.2676
--------	--------	--------	--------

第2类中元素为

5.5000	2.3000	4.0000	1.3000
5.7000	2.8000	4.5000	1.3000
4.9000	2.4000	3.3000	1.0000
5.2000	2.7000	3.9000	1.4000
5.0000	2.0000	3.5000	1.0000
5.9000	3.0000	4.2000	1.5000
6.0000	2.2000	4.0000	1.0000
6.1000	2.9000	4.7000	1.4000
5.6000	2.9000	3.6000	1.3000
5.6000	3.0000	4.5000	1.5000
5.8000	2.7000	4.1000	1.0000
6.2000	2.2000	4.5000	1.5000

5.6000	2.5000	3.9000	1.1000
6.1000	2.8000	4.0000	1.3000
6.1000	2.8000	4.7000	1.2000
6.4000	2.9000	4.3000	1.3000
6.0000	2.9000	4.5000	1.5000
5.7000	2.6000	3.5000	1.0000
5.5000	2.4000	3.8000	1.1000
5.5000	2.4000	3.7000	1.0000
5.8000	2.7000	3.9000	1.2000
5.4000	3.0000	4.5000	1.5000
6.0000	3.4000	4.5000	1.6000
6.3000	2.3000	4.4000	1.3000
5.6000	3.0000	4.1000	1.3000
5.5000	2.5000	4.0000	1.3000
5.5000	2.6000	4.4000	1.2000
6.1000	3.0000	4.6000	1.4000
5.8000	2.6000	4.0000	1.2000
5.0000	2.3000	3.3000	1.0000
5.6000	2.7000	4.2000	1.3000
5.7000	3.0000	4.2000	1.2000
5.7000	2.9000	4.2000	1.3000
6.2000	2.9000	4.3000	1.3000
5.1000	2.5000	3.0000	1.1000
5.7000	2.8000	4.1000	1.3000
4.9000	2.5000	4.5000	1.7000

第3类聚类中心为

5.0060	3.4180	1.4640	0.2440
--------	--------	--------	--------

第3类中元素为

5.1000	3.5000	1.4000	0.2000
4.9000	3.0000	1.4000	0.2000
4.7000	3.2000	1.3000	0.2000
4.6000	3.1000	1.5000	0.2000
5.0000	3.6000	1.4000	0.2000
5.4000	3.9000	1.7000	0.4000
4.6000	3.4000	1.4000	0.3000
5.0000	3.4000	1.5000	0.2000
4.4000	2.9000	1.4000	0.2000
4.9000	3.1000	1.5000	0.1000
5.4000	3.7000	1.5000	0.2000
4.8000	3.4000	1.6000	0.2000
4.8000	3.0000	1.4000	0.1000
4.3000	3.0000	1.1000	0.1000
5.8000	4.0000	1.2000	0.2000
5.7000	4.4000	1.5000	0.4000
5.4000	3.9000	1.3000	0.4000



5.1000	3.5000	1.4000	0.3000
5.7000	3.8000	1.7000	0.3000
5.1000	3.8000	1.5000	0.3000
5.4000	3.4000	1.7000	0.2000
5.1000	3.7000	1.5000	0.4000
4.6000	3.6000	1.0000	0.2000
5.1000	3.3000	1.7000	0.5000
4.8000	3.4000	1.9000	0.2000
5.0000	3.0000	1.6000	0.2000
5.0000	3.4000	1.6000	0.4000
5.2000	3.5000	1.5000	0.2000
5.2000	3.4000	1.4000	0.2000
4.7000	3.2000	1.6000	0.2000
4.8000	3.1000	1.6000	0.2000
5.4000	3.4000	1.5000	0.4000
5.2000	4.1000	1.5000	0.1000
5.5000	4.2000	1.4000	0.2000
4.9000	3.1000	1.5000	0.1000
5.0000	3.2000	1.2000	0.2000
5.5000	3.5000	1.3000	0.2000
4.9000	3.1000	1.5000	0.1000
4.4000	3.0000	1.3000	0.2000
5.1000	3.4000	1.5000	0.2000
5.0000	3.5000	1.3000	0.3000
4.5000	2.3000	1.3000	0.3000
4.4000	3.2000	1.3000	0.2000
5.0000	3.5000	1.6000	0.6000
5.1000	3.8000	1.9000	0.4000
4.8000	3.0000	1.4000	0.3000
5.1000	3.8000	1.6000	0.2000
4.6000	3.2000	1.4000	0.2000
5.3000	3.7000	1.5000	0.2000
5.0000	3.3000	1.4000	0.2000

2019-10-10 16:10:14