机器学习报告 # 3

从 K-means 到 FCM*

董晟渤, 统计 91, 2193510853

西安交通大学数学与统计学院

日期：2022 年 4 月

目录

*2021-2022 学年第二学期, 课程: 机器学习, 指导老师: 孟德宇.

1 概述

聚类的问题是这样: 设共有 n 个事物, $U=\{u_1,u_2,\cdots,u_n\}$ 为待分类的事物的全体, 每个事物都有 m 个特征, 对于第 j 个事物, 其特征记为 $u_j=(x_{j1},x_{j2},\cdots,x_{jm})$. 我们的目的是, 将 U 分为 c 个不同的类, 也即找到 U 的 c 个子集 $A_1,A_2,\cdots,A_c$, 使得

$$\bigcup_{i=1}^{c} A_i=U, \quad \text{且} \quad A_i\cap A_j=\varnothing, \quad i\leq j.$$

这样的分类结果可以用一个 $c\times n$ 的矩阵

$$\boldsymbol{A}=[a_{ij}]_{c\times n}=\begin{bmatrix} a_{11} & a_{12} & \cdots & a_{1n} \\ a_{21} & a_{22} & \cdots & a_{2n} \\ \vdots & \vdots & \ddots & \cdots \\ a_{c1} & a_{c2} & \cdots & a_{cn} \end{bmatrix}$$

来表示, 称为 U 的 c**-划分**. 这时候, 通常有两种划分的方式, 分别成为硬聚类和软聚类.

(1) **硬聚类**, 此时 $a_{ij}\in\{0,1\}(1\leq i\leq c, 1\leq j\leq n)$, 若 $a_{ij}=1$, 则第 j 个事物属于第 i 个类, 也即 $u_j\in A_i$. 若要求每个元素属于且仅属于一个类, 每个类至少有一个元素, 则矩阵 $\boldsymbol{A}$ 的列和与行和分别满足

$$\sum_{i=1}^{c} a_{ij}=1, \quad 0<\sum_{j=1}^{n} a_{ij}<n;$$

(2) **软聚类**, 此时 $a_{ij}\in[0,1](1\leq i\leq c, 1\leq j\leq n)$, 表示第 j 个事物属于第 i 个类的程度, 此时对矩阵 $\boldsymbol{A}$ 仍然要求

$$\sum_{i=1}^{c} a_{ij}=1, \quad 0<\sum_{j=1}^{n} a_{ij}<n.$$

本报告中研究的 K-means 聚类属于硬聚类, 而 FCM 聚类属于软聚类.

为了验证聚类的效果, 本篇报告选用的数据集为经典的鸢尾花数据集, 对其进行三分类, 并画出分类的结果图, 计算分类的准确度.

2 硬聚类: K-means 聚类

2.1 K-means 聚类的原理

K-means 是最常用的聚类算法. 在进行 K-means 聚类时, 首先需要找到每一个类的中心, 也即所谓的“聚类中心”, 再根据每个事物到中心的距离, 来判断该事物属于哪个类. 在这里为了方便, 设第 i 个类可以表示为 $A_i=\left\{u_1^{(i)},u_2^{(i)},\cdots,u_{n_i}^{(i)}\right\}$, 对于第 $k(1\le k\le m)$ 个特征, 记

$$\overline{x_k^{(i)}}=\frac{1}{n_i}\sum_{j=1}^{n_i}x_{jk}^{(i)},$$

为第 i 个类的第 k 个特征的平均值, 称

$$v_i=\left(\overline{x_1^{(i)}},\overline{x_2^{(i)}},\cdots,\overline{x_m^{(i)}}\right)$$

为第 i 个类的**聚类中心**, 并记聚类中心的全体 $V=\{v_1,v_2,\cdots,v_c\}$. K-means 算法沿用了 EM 算法 (见上一篇报告) 的思想, 重复进行以下两个步骤

- 首先, 对已知的聚类中心, 确定新的分类;
- 接下来, 对已知的分类, 计算新的聚类中心.

基于该思想, 设计算法如算法1.

算法 1: K-means 算法

input 数据集 $U=\{u_1,u_2,\cdots,u_n\}$、分类数 c 与迭代次数 k;

set $l=0$, 初始化 $v_1^{(0)},v_2^{(0)},\cdots,v_c^{(0)}$;

for $l\in\{0,1,2,\cdots,k\}$ **do**

  $d_{ij}^{(l)}=\|u_j-v_i\|,1\le i\le c,1\le j\le n$;

  $c_j=\arg\min_i d_{ij},1\le j\le n$;

  计算第 i 个类的聚类中心 $v_i,1\le i\le c$;

end

output $c_1^{(k)},c_2^{(k)},\cdots,c_n^{(k)}$.

2.2 K-means 聚类的应用

将 K-means 算法应用到鸢尾花数据集上, 并选取鸢尾花的后两个属性进行聚类, 所得的结果如图 **1**所示. 分类的准确度达到了 95.33%.

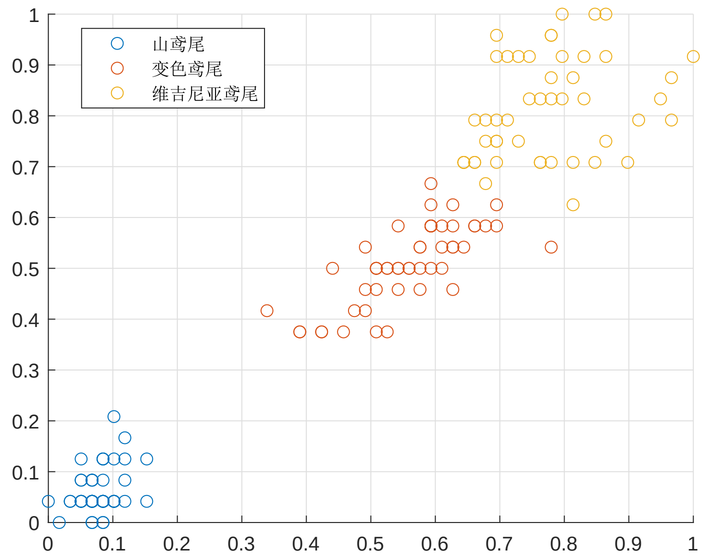

图 1: K-means 聚类的结果

3 软聚类: FCM 聚类

3.1 FCM 聚类的原理

FCM 聚类是一种基于目标函数的模糊聚类算法, 主要用于数据的聚类分析.

在 FCM 聚类中, 对每一个类, 同样都需要先找到它的中心, 再判断事物接近每个类中心的程度. 同样记第 i 个类为 $A_i = \left\{u_1^{(i)}, u_2^{(i)}, \cdots, u_{n_i}^{(i)}\right\}$, 对于第 $k(1 \leq k \leq m)$ 个特征, 记第 i 个类的第 k 个特征的平均值

$$\overline{x_k^{(i)}} = \frac{1}{n_i} \sum_{j=1}^{n_i} x_{jk}^{(i)},$$

记第 i 个类的聚类中心

$$v_i = \left(\overline{x_1^{(i)}}, \overline{x_2^{(i)}}, \cdots, \overline{x_m^{(i)}}\right)$$

并记聚类中心的全体 $V = \{v_1, v_2, \cdots, v_c\}$.

接下来, 设加权指数 $r > 1$, 定义目标函数

$$J_m(\boldsymbol{A}, V) = \sum_{i=1}^{c} \sum_{j=1}^{n} (a_{ij})^r \|u_j - v_i\|^2,$$

其是 U 中的每个事物到聚类中心 V 的距离按照一定权重的加权平方和的和. FCM 聚类的目标在于, 找到 $\boldsymbol{A} = [a_{ij}]_{c \times n}$ 和 $V = \{v_1, v_2, \cdots, v_c\}$, 使得 $J_m(\boldsymbol{A}, \boldsymbol{V})$ 最小.

定理 3.1 (FCM 聚类)**.** 设待分类的数据集 $U=\{u_1,u_2,\cdots,u_n\}$, 其中对 $1\le j\le n$, 有 $u_j=(x_{j1},x_{j2},\cdots,x_{jm})$. 并设分类数 $2\le c\le n-1$, 加权指数 $r>1$, 目标函数

$$J_m(\boldsymbol{A},V)=\sum_{i=1}^{c}\sum_{j=1}^{n}(a_{ij})^r\|u_j-v_i\|^2,$$

其中 $\boldsymbol{A}=[a_{ij}]_{c\times n}$, $\sum\limits_{i=1}^{c}a_{ij}=1$, $0<\sum\limits_{j=1}^{n}a_{ij}<n$, $V=\{v_1,v_2,\cdots,v_c\}$, 则仅当

$$a_{ij}=1\Bigg/\sum_{j=1}^{c}\left(\frac{\|u_j-v_i\|}{\|u_j-v_j\|}\right)^{\frac{2}{r-1}},\quad 1\le i\le c,\quad 1\le j\le n$$

及

$$v_i=\sum_{j=1}^{n}(a_{ij})^r u_j\Bigg/\sum_{j=1}^{n}(a_{ij})^r,\quad 1\le i\le c$$

时, $J_m(\boldsymbol{A},V)$ 取得局部最小值.

证明. 首先对 $\boldsymbol{A}$ 最小化, 此时约束条件为 $\sum\limits_{i=1}^{c}a_{ij}=1(1\le j\le n)$, 引入 Lagrange 函数

$$L(\boldsymbol{A},\boldsymbol{\lambda})=\sum_{i=1}^{c}\sum_{j=1}^{n}(a_{ij})^r\|u_j-v_i\|^2-\sum_{j=1}^{n}\lambda_j\cdot\left(\sum_{i=1}^{c}a_{ij}-1\right),$$

其中 $\boldsymbol{\lambda}=(\lambda_1,\lambda_2,\cdots\lambda_n)$, 并由

$$\frac{\partial L(\boldsymbol{A},\boldsymbol{\lambda})}{\partial a_{ij}}=r(a_{ij})^{r-1}\|u_j-v_i\|^2-\lambda_j=0,$$
$$\frac{\partial L(\boldsymbol{A},\boldsymbol{\lambda})}{\partial \lambda_j}=\sum_{i=1}^{c}a_{ij}-1=0,$$

解得

$$a_{ij}=\left(\frac{\lambda_j}{r\|u_j-v_i\|^2}\right)^{\frac{1}{r-1}}=1\Bigg/\sum_{k=1}^{c}\left(\frac{\|u_j-v_i\|}{\|u_j-v_k\|}\right)^{\frac{2}{r-1}}.$$

接下来对 V 最小化, 此时无约束条件, 令

$$\frac{\partial J_m(\boldsymbol{A},V)}{\partial v_i}=-\sum_{j=1}^{n}2(a_{ij})^r(\mu_j-v_i)=0,$$

解得

$$v_i=\sum_{j=1}^{n}(a_{ij})^r u_j\Bigg/\sum_{j=1}^{n}(a_{ij})^r.$$

这便给出了 $J_m(\boldsymbol{A},V)$ 取局部最小值的必要条件. □

一般来说, 直接应用定理3.1来求解结果 $\boldsymbol{A}$ 和 V 是相当困难的. 通过使用参考文献 [2] 给出的 FCM 算法2进行迭代, 可以在实际应用中解决该问题. 在得到了 $\boldsymbol{A}$ 之后, 对于第 j 个事物, 记 $c_j=\arg\max_{1\le i\le c}a_{ij}$ 为其所属的类.

算法 2: FCM 算法

input 数据集 $U=\{u_1,u_2,\cdots,u_n\}$、分类数 c、加权指数 r 与迭代次数 k;

set $l=0$, 初始化 $\boldsymbol{A}^{(0)}$;

for $l\in\{0,1,2,\cdots,k\}$ **do**

$$v_i^{(l)}=\sum_{j=1}^{n}\left(a_{ij}^{(l)}\right)^r u_j\Bigg/\sum_{j=1}^{n}\left(a_{ij}^{(l)}\right)^r,1\le i\le c;$$

$$a_{ij}^{(l+1)}=1\Bigg/\sum_{j=1}^{c}\left(\frac{\left\|u_j-v_i^{(l)}\right\|}{\left\|u_j-v_j^{(l)}\right\|}\right)^{\frac{2}{r-1}},1\le i\le c,1\le j\le n;$$

end

output $\boldsymbol{A}^{(k+1)}=\left[a_{ij}^{(k+1)}\right]_{c\times n}$ 与 $V^{(k)}=\left\{v_1^{(k)},v_2^{(k)},\cdots,v_c^{(k)}\right\}$.

3.2 FCM 聚类的应用

将 FCM 算法应用到鸢尾花数据集上, 所得的结果如图 2所示.

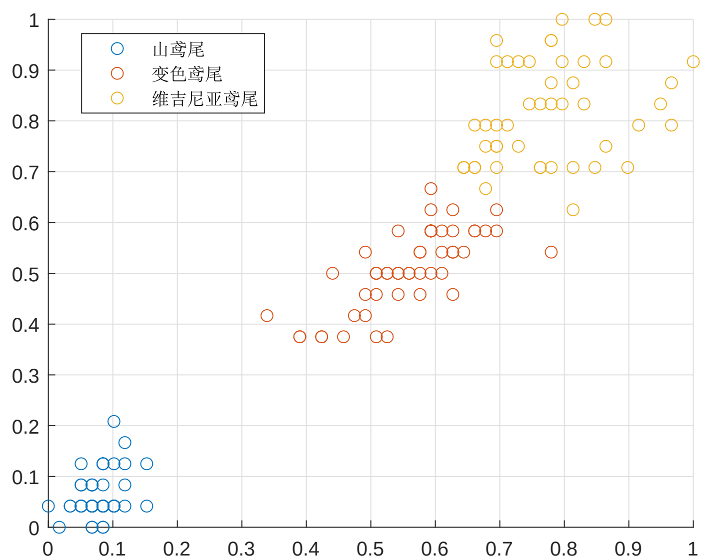

图 2: FCM 聚类的结果

此时, 聚类的准确度达到了 96%. 相较于 K-means 聚类, FCM 聚类的准确度较高, 并且给出了每个事物对每个类的“隶属度”, 能够反映更多信息.

参考文献

[1] 周志华. 机器学习 [M]. 北京: 清华大学出版社, 2016.

[2] Bezdek J. C.. *A convergence theorem for the fuzzy ISODATA clustering algorithms*. IEEE Transactions on Pattern Analysis and Machine Intelligence, 1980:2, 1-8.

附录

A 代码 1: K-means 聚类

代码使用 MATLAB 编写.

```
%% 初始化

clc;
clear;
close all;

%% 数据预处理

data = xlsread('/Iris.csv');
x = data(:, 4 : 5);
x = (x - min(x).*ones(size(x)))./(max(x).*ones(size(x)) - min(x).*ones
    (size(x))); % 用极差进行正规化
n = size(x, 1); % 数据量
c = ones(1 ,n); % 分类, 1表示setosa, 2表示versicolor, 3表示virginica

%% p-距离

p = 2;
d = @(x, y) sum(abs(x - y).^p)^(1/p);

%% K-means聚类

m = 100; % 迭代次数
center = [x(1, :); x(51, :); x(101, :)];
```

```
for k = 1 : m

    % 根据距离分类
    for i = 1 : n
        dist = [d(x(i, :), center(1, :)), d(x(i, :), center(2, :)), d(
            x(i, :), center(3, :))];
        c(i) = find(dist == min(dist));
    end

    % 计算新的质心
    sum = zeros(3, size(x, 2));
    count = zeros(1, 3);
    for i = 1 : n
        sum(c(i), :) = sum(c(i), :) + x(i, :);
        count(c(i)) = count(c(i)) + 1;
    end
    center = sum ./ count';
end

%% 画出结果图

x1 = zeros(count(1), size(x, 2));
x2 = zeros(count(2), size(x, 2));
x3 = zeros(count(3), size(x, 2));
count = zeros(1, 3);
for i = 1 : n
    count(c(i)) = count(c(i)) + 1;
    if c(i) == 1
        x1(count(1), :) = x(i, :);
    elseif c(i) == 2
        x2(count(2), :) = x(i, :);
    else
        x3(count(3), :) = x(i, :);
    end
end
scatter(x1(:, 1), x1(:, 2));
```

```
hold on;
scatter(x2(:, 1), x2(:, 2));
hold on;
scatter(x3(:, 1), x3(:, 2));
hold on;
legend('山鸢尾', '变色鸢尾', '维吉尼亚鸢尾');
grid on;

%% 计算准确度

right = 0;
for i = 1 : n
    if (c(i) == 1 && i <= 50) || (c(i) == 2 && 51 <= i && i <= 100) ||
        (c(i) == 3 && i > 101)
        right = right + 1;
    end
end
right / n
```

B 代码 2: FCM 聚类

代码使用 MATLAB 编写.

```
%% 初始化

clc;
clear;
close all;

%% 数据预处理

data = xlsread('/Iris.csv');
x = data(:, 4 : 5);
x = (x - min(x).*ones(size(x)))./(max(x).*ones(size(x)) - min(x).*ones
   (size(x))); % 用极差进行正规化
n = size(x, 1); % 数据量
```

```
c = ones(1 ,n); % 分类，1表示setosa，2表示versicolor，3表示virginica
m = 2; % 加权指数
center = [x(1, :); x(75, :); x(150, :)]; % 聚类中心
u = zeros(3, n); % 参数矩阵
for i = 1 : n
    for j = 1 : 3
        % u(j, i) = sum(center(j, :).*x(i, :)) / (norm(center(j, :))*
            norm(x(i, :))); % 利用夹角余弦给出参数矩阵初值
        u(j, i) = exp(-1/2*norm(center(j, :) - x(i, :))^2); % 利用
            Gauss核给出参数矩阵初值
    end
end

%% p-距离

p = 2;
d = @(x, y) sum(abs(x - y).^p).^(1/p);

%% FCM聚类

for k = 1 : 100

    % 计算新的中心
    for j = 1 : 3
        center(j, :) = (u(j, :).^m * x) / sum(u(j, :).^m);
    end

    % 计算参数矩阵
    for j = 1 : 3
        for i = 1 : n
            u(j, i) = 1/(d(x(i, :), center(j, :))^(2/(m-1)) * (d(x(i,
                :), center(1, :))^(-2/(m-1)) + d(x(i, :), center(2, :))
                ^(-2/(m-1)) + d(x(i, :), center(3, :))^(-2/(m-1))));
        end
    end
end
```

```matlab
count = zeros(1, 3);
for i = 1 : n
    for j = 1 : 3
        if u(1, i) > u (2, i) && u (1, i) > u(3, i)
            c(i) = 1;
        elseif u(2, i) > u(3, i)
            c(i) = 2;
        else
            c(i) = 3;
        end
    end
    count(c(i)) = count(c(i)) + 1;
end

%% 画出结果图

x1 = zeros(count(1), size(x, 2));
x2 = zeros(count(2), size(x, 2));
x3 = zeros(count(3), size(x, 2));
count = zeros(1, 3);
for i = 1 : n
    count(c(i)) = count(c(i)) + 1;
    if c(i) == 1
        x1(count(1), :) = x(i, :);
    elseif c(i) == 2
        x2(count(2), :) = x(i, :);
    else
        x3(count(3), :) = x(i, :);
    end
end
scatter(x1(:, 1), x1(:, 2));
hold on;
scatter(x2(:, 1), x2(:, 2));
hold on;
scatter(x3(:, 1), x3(:, 2));
```

```
hold on;
legend('山鸢尾', '变色鸢尾', '维吉尼亚鸢尾');
grid on;

%% 计算准确度

right = 0;
for i = 1 : n
    if (c(i) == 1 && i <= 50) || (c(i) == 2 && 51 <= i && i <= 100) ||
        (c(i) == 3 && i > 101)
        right = right + 1;
    else
        i
    end
end
right / n * 100
```