

机器学习与知识发现报告

卫星影像的AI分类与识别

2018年4月14日

目录

1 参赛情况	3
1.1 小组成员及分工	3
1.2 实验条件	3
2 赛题内容	3
2.1 题目背景	3
2.2 任务描述	3
3 模型设计	4
3.1 图像处理	4
3.2 模型训练	4
3.3 模型测试	6
4 项目实现及结果	6
4.1 项目训练部分	6
4.2 项目测试部分	9
5 讨论	11
5.1 矩阵循环	11
6 总结	11
A training.m	12
B testing.m	14

1 参赛情况

1.1 小组成员及分工

本小组成员共一人，即本人，学号SC17023010，用户ID为7604701.个人完成算法设计、项目实现和报告书写的工作。由于项目开始较晚，所以在完成本项目的时候，初赛已经结束，所以没有比赛成绩。

1.2 实验条件

本实验的运行环境为Ubuntu 16.04，基于无监督的K-means聚类算法使用MATLAB完成设计。

2 赛题内容

2.1 题目背景

遥感影像解译，作为数字图像分析的一个重要组成部分，长期以来被广泛应用于国土、测绘、国防、城市、农业、防灾减灾等各个领域。随着机器学习技术的发展，如地表覆盖分类等基于遥感影像的数字图像分析技术也得到了一定程度的发展。但是长期以来，基于遥感影像的应用仍停留在目视解译的阶段，自动化的程度较低。一个重要的原因即遥感影像的机器学习分析方法效率不高，还不足以支撑现有的应用。

本赛题目标为在基于一定量的目视解译样本基础上，通过各类图像处理、机器学习算法，提取影像中各类地物的光谱或形状等特别的特征。计算其统计信息，同时用这些种子类别对模型进行训练，随后用训练好的模型去对其他待分数据进行分类。使每个像元按不同的规则将其划分到和其最相似的样本类，以此完成对整个图像的分类。本次大赛官方将提供在2015年某地区的高分辨率遥感影像，包括基于该遥感影像目视解译出来的地表覆盖样本数据（图片），参赛队伍需要对其进行数据挖掘和必要的机器学习训练。模型建立完毕后，官方会提供采用该高分辨率遥感影像数据来做评测，检测算法是否能准确的识别出地表覆盖物。

2.2 任务描述

参赛者需要对官方提供的2015年的高分辨率遥感数据以及相应的样本数据进行预处理、特征提取、训练等各个维度进行数据挖掘和特征创建，并自行创建训练数据中的负样本，进行合适的机器学习训练。在我们提供2015年的数据中，通过您的算法或模型准确的对该地区的地表覆盖物进行分类。

分类结果以准确率（overall accuracy）作为主要评估标准。

3 模型设计

3.1 图像处理

由于官方提供的遥感图像为16bit图片，在屏幕上显示出的图像为全黑，如图1，所以需要将图像转为8bit图像，如图2。而在MATLAB中，处理数据的基本类型应为double型，所以在转为8bit图像后，还要利用im2double函数将图像转为double型，并将范围在0至255之间的rgb值进行归一化。将rgb三维图像分为red、green、blue三个维度，再将这三个维度合在一起做k-means聚类。

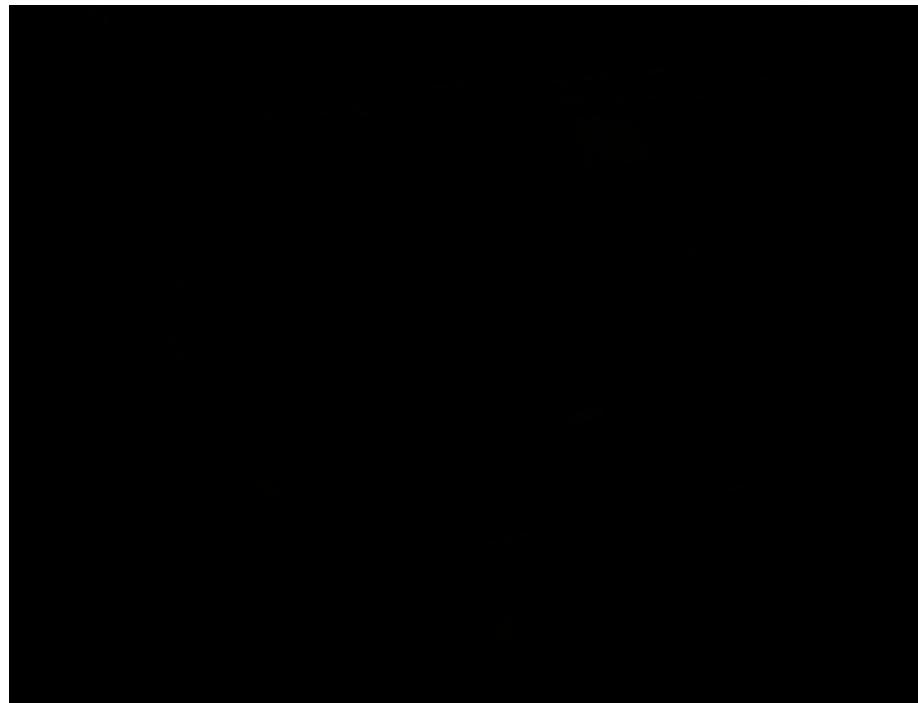


图 1: 16bit图像在屏幕中显示全黑

3.2 模型训练

利用无监督的k-means算法对训练图像做聚类，得到并保存聚类中心于codebook中，作为测试集的依据。通过不断地试验，最终发现选择6个聚类中心能够得到比较好的聚类结果。



图 2: 8bit训练图像

3.3 模型测试

在测试中，首先加载codebook，载入训练得到的聚类结果。通过for循环计算rgb距离和为像素赋值。for循环中的中间变量每次循环写入一个聚类中心。对图像的每个像素，计算其rgb值到每个中心的rgb距离，由于只比较相对大小，所以不用求平方根，距离如下，其中imr为图像的r值，cr为聚类中心的r值，g和b类同：

$$(imr - cr)^2 + (img - cg)^2 + (imb - cb)^2.$$

计算rgb距离后找出rgb距离的最小值及其对应的聚类中心，将聚类中心的rgb值赋给该像素，同时将每个像素的分类结果输出csv文件。最后输出测试图像。

4 项目实现及结果

本项目使用MATLAB进行项目的实现。

4.1 项目训练部分

训练中的直接使用MATLAB中的kmeans函数进行聚类。如图3为对训练图像进行聚类后的可视化结果。

在选择训练算法时，我还尝试了模糊C均值算法，即Fuzzy C Means (FCM)。但是，FCM训练的结果相对k-means聚类的结果并没有更好，如图4为FCM的训练结果。并且FCM的训练时间是k-means算法的三倍多，如图5。所以最终我选择了k-means算法进行项目实现。



图 3: 训练图像的可视化聚类结果



图 4: FCM的训练结果

in training (time 10)
时间已过 106.486032 秒。
时间已过 25.434649 秒。

图 5: 两种算法训练时间对比

4.2 项目测试部分

定义中间变量rgbdist矩阵，用于保存rgb距离。利用MATLAB自带的min函数，可得到最小值和最小值的位置。最小值的位置便是该像素的分类结果，即该像素属于哪一中心。然后将分类结果输出到csv文件。如图6为测试图像，图7为分类后的测试图像，其中蓝色为水体，红色为道路，黄色为房屋，绿色为植被，黑白为土地。



图 6: 测试图像



图 7: 分类后的测试图像

5 讨论

5.1 矩阵循环

在测试阶段，我使用的是for循环，但是在MATLAB中for循环的效率较低，可以尝试使用矩阵方式进行循环。大体思想为创建与聚类中心有相同rgb值且与测试图像相同大小的图像，亦即矩阵。分别将测试图像的每个像素与聚类中心的像素的相应像素进行比对，若找到距离最小的聚类中心，则将聚类中心的rgb值赋给该像素。如下为初步的实现。

```
for i = 1:sizeC(1)
    eval(['center' num2str(i) '=' zeros(size(testim));']);
    for j = 1:3

        eval(['center' num2str(i) '((:,:,1)' num2str(j) ...
            ') = C(' num2str(i) ',' num2str(j) ');']);
    end
end
center1 = imresize(center1, 0.1);
imshow(center1);
```

6 总结

通过本次实验，我了解了一种无监督的聚类算法，一个人实现了一个小的项目。虽然项目很小，但是已经知道了机器学习应用在图像分析上的大概流程。最后，感谢陈老师一个学期的教授，感谢各位助教的辛苦付出。

A training.m

```
clc;
close all;
clear;

%% kmeans 聚类训练
tic
train = imread('train1_8bit.png');
train = im2double(train);
train = imresize(train, 0.3);
[m,n,k] = size(train);
red = train(:, :, 1);
green = train(:, :, 2);
blue = train(:, :, 3);
red = red(:);
green = green(:);
blue = blue(:);
im = [red(:) green(:) blue(:)];
[IDX, C] = kmeans(im, 6); % 图类分个中心较好 %16
C255 = C * 255; % 聚类中心% 保存codebook
save('codebook', 'C');

im = double(im); %依次对同一类别的像元将其亮度值修改为该类亮度的平均值
%
for i = 1:5
    pp = find(IDX==i);
    avg = mean(im(pp,:));
    for j = 1:length(pp)
        im(pp(j), :) = avg;
    end
end %输出分类结果图
```

```

red = im(:,1);
green = im(:,2);
blue = im(:,3);
red = reshape(red,m,n);
green = reshape(green,m,n);
blue = reshape(blue,m,n);
train(:,:,1) = red;
train(:,:,2) = green;
train(:,:,3) = blue;
% img=uint8(img);
figure,
imshow(train);

toc
%% 模糊均值 (Cfuzzy C,) 算法实现meansFCM

% tic
% train=imread('train1_8bit.png');
% train=im2double(train);
% train=imresize(train, 0.3);
% [m,n,k]=size(train);
% red=train(:,:,1);green=train(:,:,2);blue=train(:,:,3);
% red=red(:);green=green(:);blue=blue(:);
% im=[red(:) green(:) blue(:)];
% [C, U, obj_fcn] =fcm(im,6);
% save('codebook', 'C');
% maxU = max(U);
% index1 = find(U(1,:)==maxU);
% index2 = find(U(2,:)==maxU);
% index3 = find(U(3,:)==maxU);
% index4 = find(U(4,:)==maxU);
% index5 = find(U(5,:)==maxU);
% 依次对同一类别的像元将其亮度值修改为该类亮度的平均值%

```

```
% red(index1)=mean(red(index1));green(index1)=mean(green(index1))
    ;blue(index1)=mean(blue(index1));
% red(index2)=mean(red(index2));green(index2)=mean(green(index2))
    ;blue(index2)=mean(blue(index2));
% red(index3)=mean(red(index3));green(index3)=mean(green(index3))
    ;blue(index3)=mean(blue(index3));
% red(index4)=mean(red(index4));green(index4)=mean(green(index4))
    ;blue(index4)=mean(blue(index4));
% red(index5)=mean(red(index5));green(index5)=mean(green(index5))
    ;blue(index5)=mean(blue(index5));
% 输出分类结果图%
% red=reshape(red,m,n);green=reshape(green,m,n);blue=reshape(
    blue,m,n);
% train(:,:,1)=red;train(:,:,2)=green;train(:,:,3)=blue;
% % img=uint8(img);
% figure,
% imshow(train);
%
% toc
% imwrite(train,'');
```

B testing.m

```
clc;
close all;
clear;

%% 加载codebook
load codebook.mat;
testim = imread('testing1_8bit.png');
testim = im2double(testim); %若修改%范围，这里也要改rgb
sizeC = size(C); %若修改%范围，则要改rgbC
sizeim = size(testim)
% figure, imshow(testim);
```

```
%% 尝试使用矩阵方式进行循环

% % 创建与聚类中心有相同值且与测试图像相同大小的图像rgb
% for i = 1:sizeC(1)
%     eval(['center' num2str(i) '=' zeros(size(testim));']);
%     for j = 1:3
%         若修改          %范围, 则要改rgbC
%             eval(['center' num2str(i) '(:, :, ' num2str(j) ...
%                     ') = C(' num2str(i) ',' num2str(j) ');']);
%     end
% end

% center1 = imresize(center1, 0.1);
% imshow(center1);

%% 图像对比
tic
% center = zeros(sizeim)保存最小;%值rgb
rgbdist = zeros(sizeim(1),sizeim(2),sizeC(1)); 保存%距离rgb

% 计算每个像素的距离rgb
for j = 1:sizeim(1)
    for k = 1:sizeim(2)
        for i = 1:sizeC(1)若修改
            %范围, 则要改rgbC
            rgbdist(j,k,i) = (testim(j,k,1)-C(i,1))^2 + ...
                (testim(j,k,2)-C(i,2))^2 + (testim(j,k,3)-C(i,3))^2;
        end
    end
end
```

```
% 最小值和最小值的位置
[valmin posmin] = min(rgbdist,[],3);
csvwrite('csv_testing1', posmin);% 输出结果csv
C = [0 0 1; 1 0 0; 1 1 1; 0 1 0; 0.8 0.8 0.1; 0 0 0];

% 为每个像素赋值
for i = 1:sizeim(1)
    for j = 1:sizeim(2)
        centnum = posmin(i,j); %中聚类中心的行号codebook
        cbrgb = C(centnum,:);
        for k = 1:sizeim(3)
            testim(i,j,k) = cbrgb(k);
        end
    end
end

toc

figure, imshow(testim);
```