图像分类

（最小距离分类、最大似然分类、K 均值分类）

一、实验目的与原理

图像分类的目的是将图像中每个像元根据其不同波段的光谱亮度、空间结构 特征或者其他信息，按照某种规则或算法划分为不同的类别。遥感图像分类就是对地球表面及其环境在遥感图像上的信息进行识别和分类，从而达到识别图像信 息所对应的实际地物，提取所需地物信息的目的。

遥感图像分类的原理：由于地物的成分、性质、分布情况的复杂程度和成像 条件不同，以及一个像元或瞬时试场里往往有两种或多种地物的情况，即混合像元，使得同类地物的特征向量也不尽相同，而且使得不同地物类型的特征向量之

间的差别也不都是显著的。遥感图像的光谱特征通常是以地物在多光谱图像上的灰度体现出来的，即不同地物在同一波段图像上表现的灰度一般互不相同；同时，

不同地物在多个波段上的灰度呈现规律也不相同，这就构成了我们在图像上区分

不同地物的物理依据。

遥感图像传统的计算机分类方式有监督分类和非监督分类两种。监督分类包

括最小距离分类法、平行多面体分类法、最大似然分类法。非监督分类包括 K-

均值聚类法、ISODATA 分类法。本实验将用 Matlab 实现监督分类的最小距离分

类和最大似然分类，以及非监督分类中的 K-均值分类。

三、 算法描述

（1） 最小距离分类算法描述

1.读取各个波段文件，并加载各类地物在各个波段的均值文件。

2.计算每个像元向量到各类地物的距离。

3.比较像元到各类地物的距离，取最小的距离，并将像元归为该类。

4.为各类赋予一定的颜色。

（2）最大似然分类算法描述：

1.读取 BIP 文件，将每个波段读为一列。

2.加载各类地物在各个波段的均值和方差

3.对每个像元向量计算属于各个类的概率，比较概率的大小，概率最大的将像元

归为该类。

4.对每个类赋予一定的颜色。

5.显示分类后的图像。

（3）K-means 分类算法描述：

1．为每个聚类确定一个初始的聚类中心（一定要在最小值和最大值之间，这一 步非常重要）

2．将样本集中的每一个样本按照最小距离原则,分配到 k 个聚类中的某一个

3.使用每个聚类中所有样本的均值作为新的聚类中心 4．如果聚类中心有变化则重复 2、3 步直到聚类中心不再变化为止5．最后得到的 k 个聚类中心就是聚类的结果

四、 Matlab 源代码

（1）最小距离分类代码：

clear allclc

x = 400;

y = 640;

N = 6;

image = zeros(x\*y,N);for i = 1:N

img=fopen('sample\_BIP','rb');

b0 = fread(img,i-1,'uint8');

image(:,i) = fread(img,x\*y,'uint8',N-1);

fclose(img);

end

load crop.txt;

load forest.txt;

load water.txt;

load soil1.txt;

load soil2.txt;

load soil3.txt;

dis = zeros(x\*y,N);

dis(:,1) = sqrt(sum((image-repmat(crop(1,:),x\*y,1)).^2,2));

dis(:,2) = sqrt(sum((image-repmat(forest(1,:),x\*y,1)).^2,2));

dis(:,3) = sqrt(sum((image-repmat(water(1,:),x\*y,1)).^2,2));

dis(:,4) = sqrt(sum((image-repmat(soil1(1,:),x\*y,1)).^2,2));

dis(:,5) = sqrt(sum((image-repmat(soil2(1,:),x\*y,1)).^2,2));

dis(:,6) = sqrt(sum((image-repmat(soil3(1,:),x\*y,1)).^2,2));

min\_dis = min(dis,[],2);

classes = zeros(x\*y,3);

for i = 1:x\*y

ind = find(dis(i,:)==min\_dis(i));

switch ind

case 1

classes(i,:) = [255,0,0];

case 2

classes(i,:) = [0,255,0];

|  |  |  |  |
| --- | --- | --- | --- |
|  | case 3  classes(i,:) = [0,0,255];  case 4  classes(i,:) = [255,255,0];  case 5  classes(i,:) = [255,0,255]; |  |  |

case 6

classes(i,:) = [0,255,255];

end

end

img2 = reshape(classes,y,x,3);

RGB\_classes = cat(3,img2(:,:,1)',img2(:,:,2)',img2(:,:,3)');

imshow(uint8(RGB\_classes));

title('最小距离分类');

（3）最大似然分类代码：

clear allclc

x = 400;

y = 640;

N = 6;

image = zeros(x\*y,N);

for i = 1:N

img=fopen('sample\_BIP','rb');

b0 = fread(img,i-1,'uint8');

image(:,i) = fread(img,x\*y,'uint8',N-1);

fclose(img);

end

load crop.txt;

load forest.txt;

load water.txt;

load soil1.txt;

load soil2.txt;

load soil3.txt;

all=cat(3,crop,forest,water,soil1,soil2,soil3);

cov = zeros(6,6,6);

mean = zeros(1,6,6);

for i = 1:6

mean(1,:,i) = all(1,:,i);

cov(:,:,i)=diag(all(2,:,i));

end

dk = zeros(x\*y,6);

for i = 1:x\*y

for j = 1:6

dk(i,j) = -log(det(cov(:,:,j)))\*0.5 - 0.5\*(image(i,:) - mean(1,:,j))/cov(:,:,j) \* (image(i,:) -

|  |  |  |  |
| --- | --- | --- | --- |
|  | mean(1,:,j))';  end  end  dk\_max = max(dk,[],2);  classes = zeros(x\*y,3);  for i = 1:x\*y |  |  |

ind = find(dk(i,:)==dk\_max(i));

switch ind

case 1

classes(i,:) = [255,0,0];case 2

classes(i,:) = [0,255,0];

case 3

classes(i,:) = [0,0,255];

case 4

classes(i,:) = [255,255,0];case 5

classes(i,:) = [255,0,255];

case 6

classes(i,:) = [0,255,255];

end

end

img2 = reshape(classes,y,x,3);

RGB\_classes = cat(3,img2(:,:,1)',img2(:,:,2)',img2(:,:,3)');

imshow(uint8(RGB\_classes));

title('最大似然分类');

（3）K-means分类代码：

clear all

clc

x = 400;

y = 640;

N = 6;

image = zeros(x\*y,N);

for i = 1:N

img=fopen('sample\_BIP','rb');

b0 = fread(img,i-1,'uint8');

image(:,i) = fread(img,x\*y,'uint8',N-1);

fclose(img);

end

load ini\_center.txt;

center = zeros(size(ini\_center));

threshold = 1e-3;

max\_d = 1;

while max\_d > threshold

dis = zeros(x\*y,N);

dis(:,1) = sqrt(sum((image-repmat(ini\_center(1,:),x\*y,1)).^2,2));

dis(:,2) = sqrt(sum((image-repmat(ini\_center(2,:),x\*y,1)).^2,2)); dis(:,3) =sqrt(sum((image-repmat(ini\_center(3,:),x\*y,1)).^2,2));

dis(:,4) = sqrt(sum((image-repmat(ini\_center(4,:),x\*y,1)).^2,2));dis(:,5) = sqrt(sum((image-repmat(ini\_center(5,:),x\*y,1)).^2,2)); dis(:,6) = sqrt(sum((image-repmat(ini\_center(6,:),x\*y,1)).^2,2));

min\_dis = min(dis,[],2);

classes = zeros(x\*y,3);crop = zeros(1,6); forest = zeros(1,6);water = zeros(1,6); soil1 = zeros(1,6);

soil2 = zeros(1,6);

soil3 = zeros(1,6);

n\_crop = 0; n\_forest = 0; n\_water = 0;n\_soil1 = 0;n\_soil2 = 0;n\_soil3 = 0;

for i = 1:x\*y

ind = find(dis(i,:) == min\_dis(i));

switch ind(1,1)

case 1

crop = crop + image(i,:);

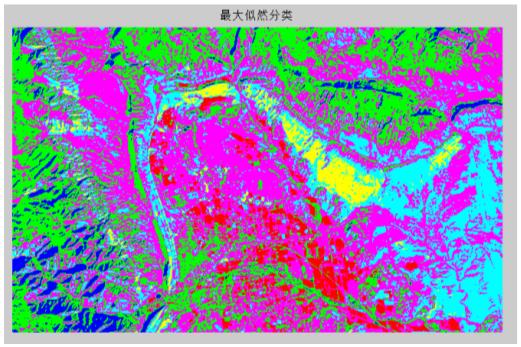
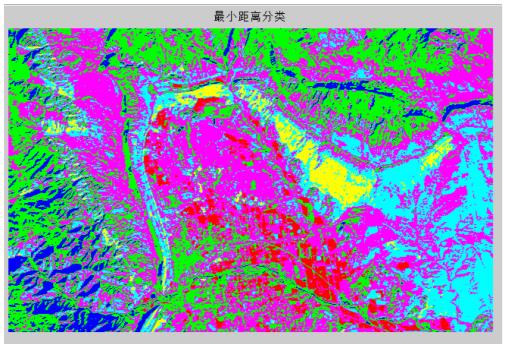
n\_crop = n\_crop + 1;

classes(i,:) = [255,0,0];

case 2

forest = forest + image(i,:);

n\_forest = n\_forest + 1;

classes(i,:) = [0,255,0];

case 3

water = water + image(i,:);

n\_water = n\_water + 1;

classes(i,:) = [0,0,255];

case 4

soil1 = soil1 + image(i,:);

n\_soil1 = n\_soil1 + 1;

classes(i,:) = [255,255,0];

case 5

soil2 = soil2 + image(i,:);

n\_soil2 = n\_soil2 + 1;

classes(i,:) = [255,0,255];

case 6

soil3 = soil3 + image(i,:);

n\_soil3 = n\_soil3 + 1;

classes(i,:) = [0,255,255];

end

|  |  |  |  |
| --- | --- | --- | --- |
|  | end  center(1,:) = crop./n\_crop;  center(2,:) = forest./n\_forest;  center(3,:) = water./n\_water;  center(4,:) = soil1./n\_soil1;  center(5,:) = soil2./n\_soil2; |  |  |

center(6,:) = soil3./n\_soil3;

max\_d = max(max(abs(center-ini\_center)));

ini\_center = center;

end

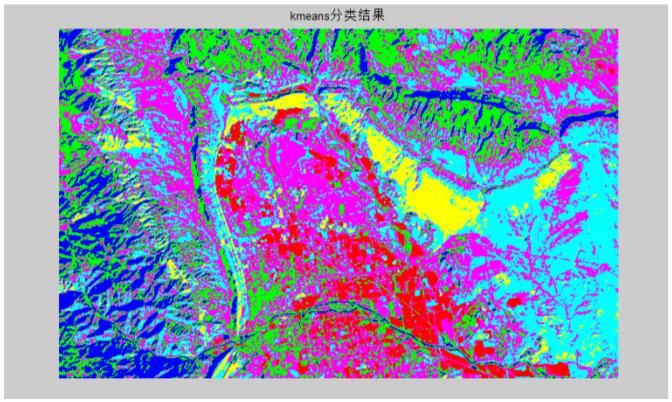
img2 = reshape(classes,y,x,3);

RGB\_classes = cat(3,img2(:,:,1)',img2(:,:,2)',img2(:,:,3)');

imshow(uint8(RGB\_classes)),title('kmeans·分类');

四、运行结果

图1：最小距离分类 图2：最大似然分类

图3：K-means分类