

光谱角分类器的实现与精度评价

地信一班 罗皓文 15303096

光谱角 (SA) 为光谱空间两向量的夹角，同种地物之间的光谱角应该较小，理想情况下，同种地物在不同影调（一般是由于环境光照引起的差异）下，光谱角变化不明显。

利用此原理，本次实验分别使用光谱角分类法 (Spectral Angle Mapper, SAM) 和最小距离法对同一影像使用同一样本进行分类，并对两者进行精度比较。

给定光谱角定义如下：

$$SA = \arccos \left(\frac{\mathbf{v}_1 \times \mathbf{v}_2^T}{\|\mathbf{v}_1\| \cdot \|\mathbf{v}_2\|} \right)$$

我们认为 SA 越小的两个向量越具有相似性，某像元与某样本中心 SA 最小分为一类，依此构建分类器。

本次实验环境与使用数据如下：

CPU: Intel Xeon CPU E3-1241 v3 @ 3.50GHz 3.50GHz

RAM: 8.00 GB

OS: Windows 7 Pro SP1

IDE: MATLAB R2016a

影像: studyare.tiff

样本: sample.csv (使用 60%作为训练样本，40%用于交叉验证)

<https://github.com/codeRimoe/homework/tree/master/%E5%A4%A7%E4%BA%8C/%E9%81%A5%E6%84%9F%E5%9B%BE%E5%83%8F%E5%A4%84%E7%90%86/data/data5>

核心代码参见文末附录。

8x8 double

	1	2	3	4	5	6	7	8
1	45	0	0	10	0	55	0.1964	0.8182
2	2	26	2	0	1	31	0.1107	0.8387
3	1	14	53	0	0	68	0.2429	0.7794
4	8	1	1	46	0	56	0.2000	0.8214
5	0	15	0	0	55	70	0.2500	0.7857
6	56	56	56	56	56	280	1	1
7	0.2000	0.2000	0.2000	0.2000	0.2000	1	0.8036	0.2000
8	0.8036	0.4643	0.9464	0.8214	0.9821	1	0.2000	0.7545

SAM 分类精度 (Overall 0.8036 , Kappa 0.7545)

8x8 double

	1	2	3	4	5	6	7	8
1	38	0	0	5	0	43	0.1536	0.8837
2	1	33	1	0	0	35	0.1250	0.9429
3	2	15	55	0	0	72	0.2571	0.7639
4	15	1	0	51	0	67	0.2393	0.7612
5	0	7	0	0	56	63	0.2250	0.8889
6	56	56	56	56	56	280	1	1
7	0.2000	0.2000	0.2000	0.2000	0.2000	1	0.8321	0.2000
8	0.6786	0.5893	0.9821	0.9107	1	1	0.2000	0.7902

最小距离法分类精度 (Overall 0.8321 , Kappa 0.7902)

对比两种分类可以发现，最小距离法分类优于 SAM，我们可以观察到，在阴影区域，SAM 并没有像预想中一样取到较好的效果，反而会出现明显的噪声，这一点在阴影和纹理丰富的区域尤其明显。这是由于在向量范数较小的情况下，某分量同等的偏差导致的角度偏差远大于同一向量在范数较大的情况，而在遥感影像中，分量的偏差是离散的，只能取整数，因此分量偏差为 1 是，SA 的变化会非常大从而导致出现噪声。

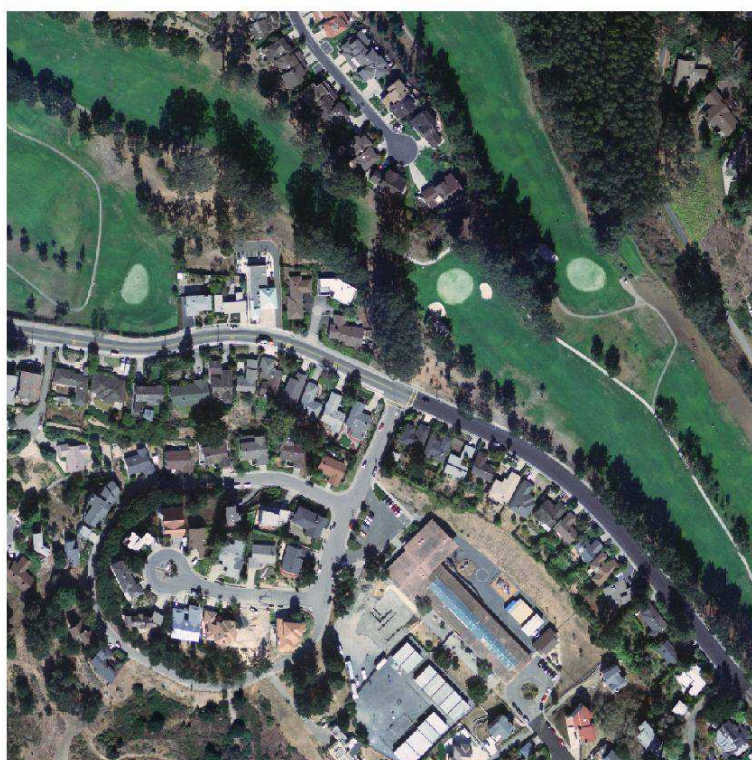
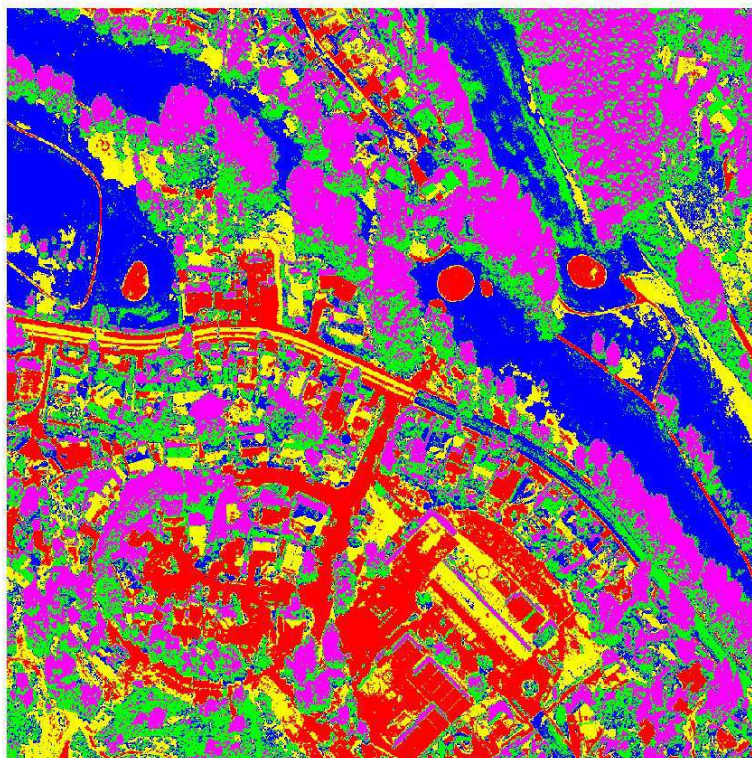
SAM 分类器代码摘录如下，具体代码与数据已上传 Github:

<https://github.com/codeRimoe/homework/tree/master/%E5%A4%A7%E4%BA%8C/%E9%81%A5%E6%84%9F%E5%9B%BE%E5%83%8F%E5%A4%84%E7%90%86/classify>

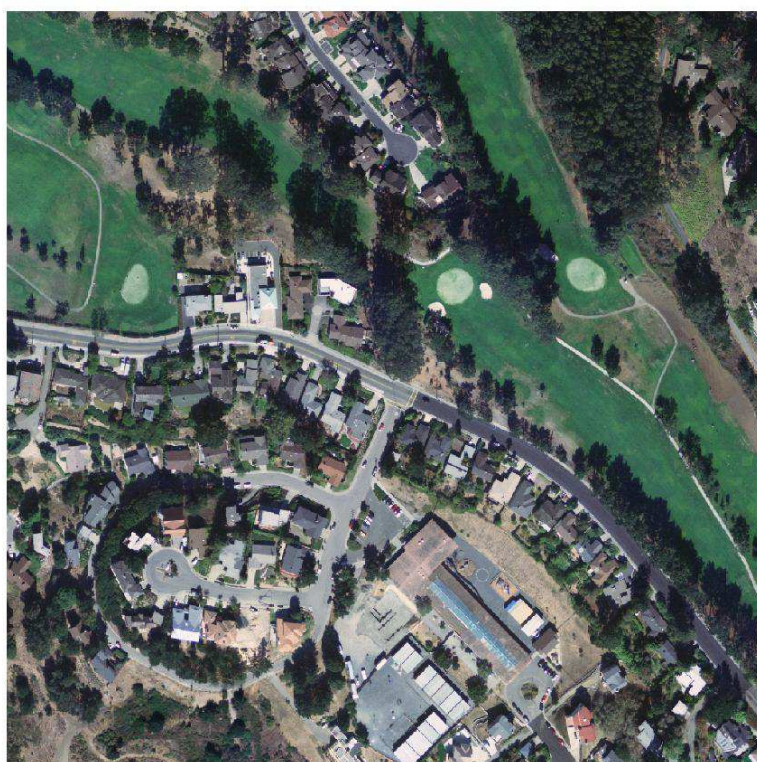
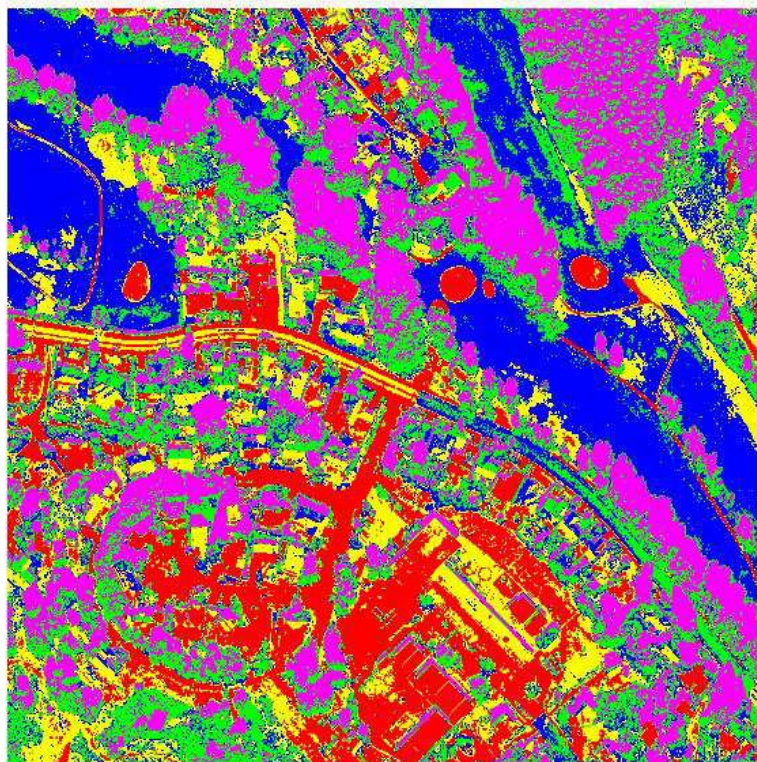
```
function [sign,Overall_Accuracy,Kappa,Acc_Table,class]
    =saMap(data,trainSam,testSam)

    nrows=size(data,1);
    ncols=size(data,2);
    nclass=size(trainSam,2);
    nsign=size(trainSam{1},2);
    sign=[];
    for i=1:nclass
        for j=9:nsign
            sign(i,j-8)=mean(trainSam{i}(:,j));
        end
    end
    % 此处省略交叉验证部分代码 %
    class=zeros(nrows,ncols);
    for i=1:nrows
        for j=1:ncols
            for k=1:(nsign-8)
                feature(k)=double(data(i,j,k));
            end
            class(i,j)=classify(sign,feature);
        end
    end
end

function [type]=classify(sign,feature)
    nClass=size(sign,1);
    typePos=[];
    for i=1:nClass
        typePosArr1=sign(i,:)*feature';
        typePosArr2=sqrt(gLength2(sign(i,:))*gLength2(feature));
        typePos(i)=typePosArr1/typePosArr2;
    end
    type=find(typePos==max(typePos));
end
% 函数 gLength2(A)用于计算某向量范数之平方%
```



SAM 分类结果



最小距离法分类结果