学号 专业 姓名

实验日期 教师签字 成绩

实验报告

【实验名称】 机器学习基础算法

【实验目的】

用Matlab学习并实现简单的分类和聚类算法

【实验原理】

分类：给定训练集，其中每个训练样本包含一个特征向量 𝐱 和一个类别标签𝑦；训练一个模型，使之能够根据一个未知的特征向量 𝐱^′来判断其类别标签𝑦′

聚类：给定数据集，其中每个样本仅包含一个特征向量 𝐱，根据高内聚低耦合的原则来计算每个样本的类别标签 𝑦

【源码】

实现k-近邻算法用于分类任务

clc;close all;clear;

X = rand(100,2)\*2-1; %每行一个训练样本，每列一个特征

Y = sum(X,2)>randn(100,1)/10; %每行一个标签

k = 9; %k临近参数

[p1,p2] = meshgrid(-1:0.01:1); %生成测试样本

for i = 1:size(p1)

for j = 1:size(p2)

dis = pdist2([p1(i,j),p2(i,j)],X);

[~,list] = sort(dis,'ascend'); %list获得从近到远的索引

if sum(Y(list(1:k))) > (k+1)/2 %如果1的数量大于0的数量，则求和会大于(k+1)/2

Y1(i,j) = 1;

else

Y1(i,j) = 0; %确定每个测试样本的标签

end

end

end

hold on;box on; %绘制训练样本的分类面

plot(X(Y,1),X(Y,2),'ok','MarkerFaceColor','b');

plot(X(~Y,1),X(~Y,2),'ok','MarkerFaceColor','r');

contour(p1,p2,Y1,1,'LineWidth',1);

实现k-means算法用于聚类任务

clc;close all;clear;

X = rand(300,2); %每行一个样本，每列一个特征

X(sum(X.^2,2)>0.5&sum((1-X).^2,2)>0.5,:) = [];

k = 3; %k-means的参数

C = X(randperm(end,k),:);%生成初始中心点

for j = 1:100 %重复迭代100次

for i = 1:size(X,1) %循环求每个样本的类

[~,list] = sort(pdist2(X(i,:),C),'ascend');%list获得从近到远的C的索引

Y(i) = list(1); %X的Y是最近的C对应的值

end

C(Y(i),:) = mean([C(Y(i),:);X(i,:)],1);%更新中心点

end

hold on;box on;

for i = 1:k

B = (Y == i); %绘制不同聚类的结果

plot(X(B,1),X(B,2),'ok','MarkerFaceColor',rand(1,3));

end

【运行结果截图】

实现k-近邻算法用于分类任务，分别取k=3，5，7，9，用plot()绘制出训练集，并用contour()绘制分类面，分类算法图如下：

|  |  |
| --- | --- |
|  |  |
|  |  |

实现k-means算法用于聚类任务，用plot()绘制出聚类后的数据集，绘制k分别等于2、3、4、5时的聚类结果：

|  |  |
| --- | --- |
|  |  |
|  |  |

【实验总结】

根据本题PPT上的相关提示部分给出的代码，实现本题难度不大；

一定要先理解测试集和训练集的关系，我第一次做的时候由于对两者的关系不太清楚，所以始终不理解绘制分类面为什么要生成grid，导致在contour函数上面花了很多时间；

由于水平有限，未能完成本任务的进阶目标：用十行代码实现分类聚类，对于PPT中使用的hist函数做了相关学习了解，无果，最后还是选择了用简单的for循环加if判断来实现分类；

--

X = rand(100,2)\*2-1; %每行一个训练样本，每列一个特征

Y = sum(X,2)>randn(100,1)/10; %每行一个标签

--

plot(X(Y,1),X(Y,2),'ok','MarkerFaceColor','b');

plot(X(~Y,1),X(~Y,2),'ok','MarkerFaceColor','r');

上述两部分代码展示了matlab矩阵编程的思想，值得学习借鉴，否则绘制时需要对Y的值进行if判断，使得代码不够简洁。