**实验2 朴素贝叶斯分类器**

**一、实验目的**

（1）理解朴素贝叶斯分类器的工作原理。

（2）编程实现朴素贝叶斯分类器。

**二、实验内容**

1、（难度1）编程实现朴素贝叶斯分类器的训练。假设数据只涉及连续属性。

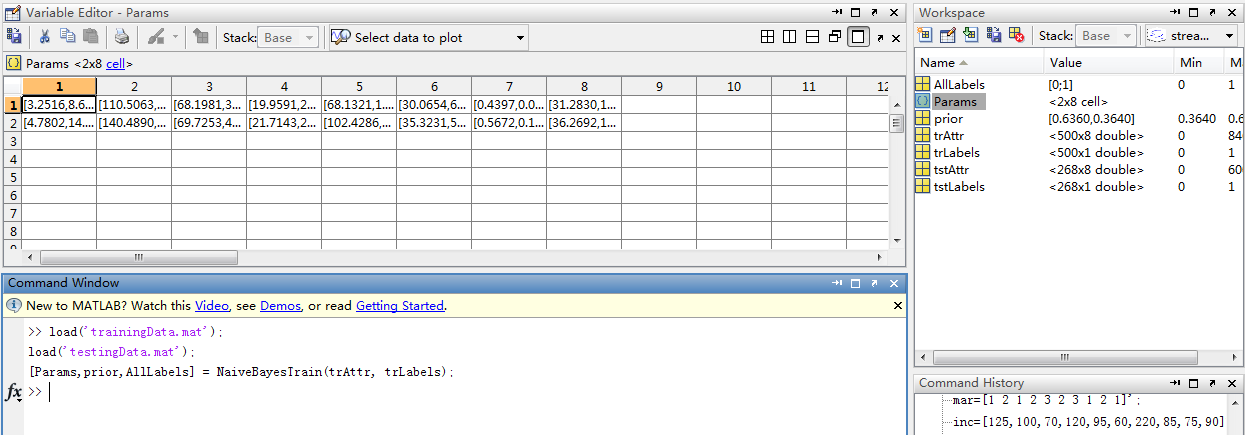
测试代码如下：

load('trainingData.mat');

load('testingData.mat');

[Params,prior,AllLabels] = NaiveBayesTrain(trAttr, trLabels);

**（将执行正确的代码粘贴在此处，核心代码要求有注释）**



function [Params,prior,AllLabels] = NaiveBayesTrain(trAttr,trLabels)

AllLabels = unique(trLabels);

numClass = length(AllLabels);%取得所有类标号的种类个数

Ind4Class{numClass,1} = [];

for i =1 : numClass

ind = find(trLabels == AllLabels(i));

Ind4Class{i,1} = ind;

prior(i) = length(ind)/length(trLabels);

end

numAttr = size(trAttr,2);

Params{numClass,numAttr} = [];

for AttrInd = 1 : numAttr

AttrVals = trAttr(:,AttrInd);

for i = 1 : numClass

Params{i,AttrInd} = GetParams4ContiAttr(AttrVals,Ind4Class{i,1});

end

end

end

function Paramsl = GetParams4ContiAttr(AttrVals,ind)

Attrvals4ThisClass = AttrVals(ind);

Paramsl(1) = mean(Attrvals4ThisClass);

Paramsl(2) = var(Attrvals4ThisClass);

end

2、（难度2）编程实现朴素贝叶斯分类器的训练。假设数据既涉及连续属性，也涉及分类属性。

测试代码如下：

refund=[1 2 2 1 2 2 1 2 2 2]';

mar=[1 2 1 2 3 2 3 1 2 1]';

inc=[125,100,70,120,95,60,220,85,75,90]';

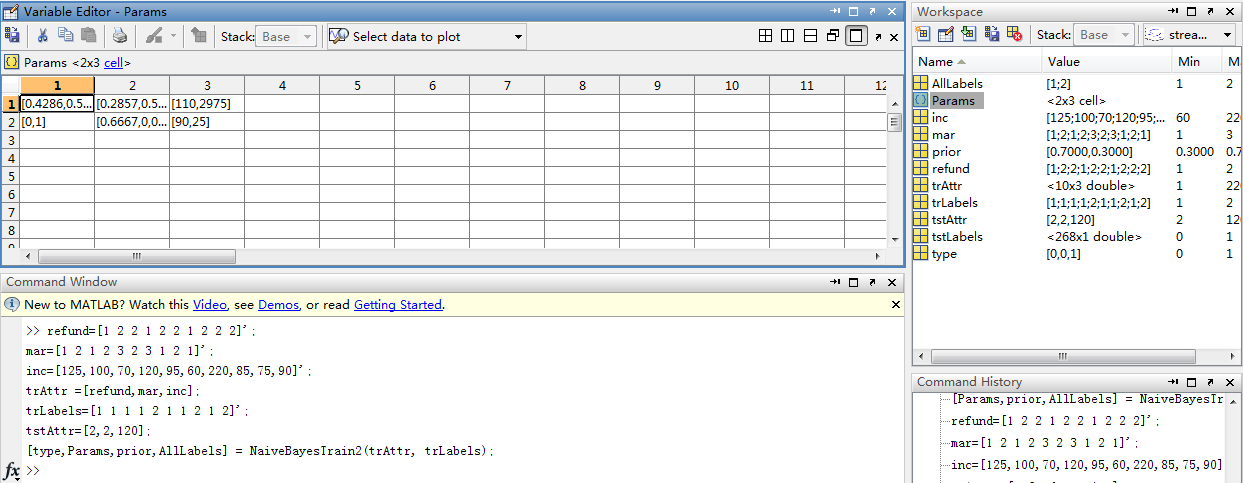
trAttr =[refund,mar,inc];

trLabels=[1 1 1 1 2 1 1 2 1 2]';

tstAttr=[2,2,120];

[type,Params,prior,AllLabels] = NaiveBayesTrain2(trAttr, trLabels);

**（将执行正确的代码粘贴在此处，核心代码要求有注释）**

****

function [type,Params,prior,AllLabels] = NaiveBayesTrain2(trAttr,trLabels)

AllLabels = unique(trLabels);

numClass = length(AllLabels);%取得所有类标号的种类个数

Ind4Class{numClass,1} = [];

for i =1 : numClass

ind = find(trLabels == AllLabels(i));

Ind4Class{i,1} = ind;

prior(i) = length(ind)/length(trLabels);

end

numAttr = size(trAttr,2);

Params{numClass,numAttr} = [];

for AttrInd = 1 : numAttr

AttrVals = trAttr(:,AttrInd);

numAttrValsClass = length(unique(AttrVals)); % 取得所有属性标号的种类个数

maxAttrVals = max(unique(AttrVals)); % 取得当前的最大属性标号

if ( isempty(find(rem(AttrVals,1)~=0) )&& numAttrValsClass< 10 && maxAttrVals ==numAttrValsClass )

%%%% 若属性为分类属性k种取值情况，要求取值为1，2，。。。k

%%%%属性为分类属性，type=0

type(AttrInd)=0;

for i=1:numClass

Params{i,AttrInd}=GetParams4CategoricalAttr(AttrVals,Ind4Class{i,1});

end

else

type(AttrInd)=1;

for i=1:numClass

Params{i,AttrInd}=GetParams4ContiAttr(AttrVals,Ind4Class{i,1});

end

end

end

end

function Params1=GetParams4CategoricalAttr(AttrVals,ind)

Params1=[];

AllAttrVals =unique(AttrVals);

numAttrValsClass = length(AllAttrVals);

AttrVals4ThisClass=AttrVals(ind);

nj=length(ind);

for i=1:numAttrValsClass

nij=length(find(AttrVals4ThisClass==AllAttrVals(i)));

Params1(i)=nij/nj;

end

end

3、（难度3）编程实现朴素贝叶斯分类器的训练和分类。

load('trainingData.mat');

load('testingData.mat');

refund=[1 2 2 1 2 2 1 2 2 2]';

mar=[1 2 1 2 3 2 3 1 2 1]';

inc=[125,100,70,120,95,60,220,85,75,90]';

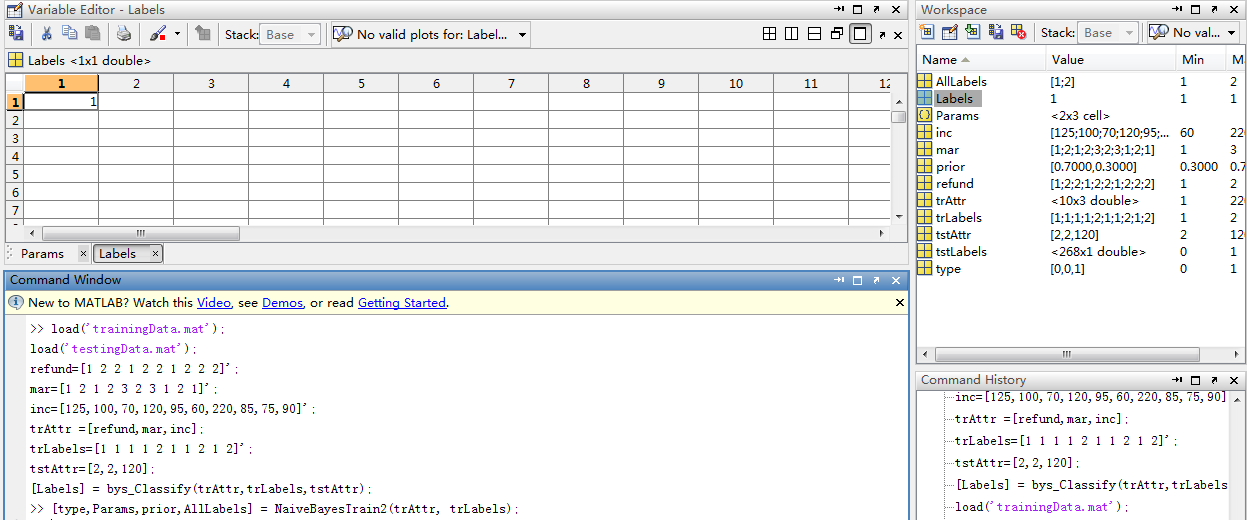
trAttr =[refund,mar,inc];

trLabels=[1 1 1 1 2 1 1 2 1 2]';

tstAttr=[2,2,120];

[Labels] = bys\_Classify(trAttr,trLabels,tstAttr);

**（将执行正确的代码粘贴在此处，核心代码要求有注释）**



function [Labels] = bys\_Classify(trAttr,trLabels,tstAttr)

[type,Params,prior,AllLabels] = NaiveBayesTrain2(trAttr, trLabels);

N = size(tstAttr,1);

Labels = zeros(N,1);

for i=1 : N

tstAttrSample = tstAttr(i,:);

post = NaiveBayesPrdeict(type,Params,prior,tstAttrSample);

[maxValue,index] = max (post);

Labels(i) = AllLabels(index);

end

end

function post = NaiveBayesPrdeict(type,Params,prior,tstAttr)

numClass = length(prior);

numAttr = length(tstAttr);

post = [];

for i = 1: numClass

p\_AttrCond = 1;

for AttrInd = 1 : numAttr

AttrVals = tstAttr(AttrInd);

if(type(AttrInd) == 0)%属性为分类属性 type = 0

p = Params{i,AttrInd}(AttrVals);% 属性编号是 1，2 属性编号必须从 1 开始

p\_AttrCond=p\_AttrCond\*p;

else %属性为连续属性 type = 1

mean = Params{i,AttrInd}(1);

sig = sqrt(Params{i,AttrInd}(2));

p = (1/(sqrt(2\*pi)\*sig))\*exp(-(AttrVals-mean)^2/(2\*sig^2));

p\_AttrCond = p\_AttrCond\*p;

end

end

post(i) = p\_AttrCond\*prior(i);

end

end