171491218刘晓萌

人工智能实验八

一、实验内容：

1.利用贝叶斯算法进行数据分类操作

2.数据集：汽车评估数据集(见附录)

二、实验步骤

1. 仔细阅读并了解实验数据集；

2. 使用Java实现朴素贝叶斯算法

3. 利用朴素贝叶斯算法在训练数据上学习分类器,训练数据的大小分别设置为：前100个数据，前200个数据，前500个数据，前700个数据，前1000个数据，前1350个数据；

4. 利用测试数据对学习的分类器进行性能评估；

三．实验内容

1.算法基本原理

对于给出的待分类项，求解在此项出现的条件下各个类别出现的概率，哪个最大，就认为此待分类项属于哪个类别。

朴素贝叶斯分类的正式定义如下：

1、设http://latex.codecogs.com/gif.latex?x=\%7ba_1,a_2,...,a_m\%7d 为一个待分类项，而每个a为x的一个特征属性。

2、有类别集合http://latex.codecogs.com/gif.latex?C=\%7by_1,y_2,...,y_n\%7d

3、计算http://latex.codecogs.com/gif.latex?P(y_1|x),P(y_2|x),...,P(y_n|x) 。

4、如果http://latex.codecogs.com/gif.latex?P(y_k|x)=max\%7bP(y_1|x),P(y_2|x),...,P(y_n|x)\%7d ，则http://latex.codecogs.com/gif.latex?x%20\in%20y_k 。

那么现在的关键就是如何计算第3步中的各个条件概率。我们可以这么做：

1、找到一个已知分类的待分类项集合，这个集合叫做训练样本集。

2、统计得到在各类别下各个特征属性的条件概率估计。即

http://latex.codecogs.com/gif.latex?P(a_1|y_1),P(a_2|y_1),...,P(a_m|y_1);P(a_1|y_2),P(a_2|y_2),...,P(a_m|y_2);...;P(a_1|y_n),P(a_2|y_n),...,P(a_m|y_n)

3、如果各个特征属性是条件独立的，则根据贝叶斯定理有如下推导：

http://latex.codecogs.com/gif.latex?P(y_i|x)=\frac%7bP(x|y_i)P(y_i)%7d%7bP(x)%7d

因为分母对于所有类别为常数，因为我们只要将分子最大化皆可。又因为各特征属性是条件独立的，所以有：

**整个朴素贝叶斯分类分为三个阶段：**

**第一阶段:** 准备工作阶段，这个阶段的任务是为朴素贝叶斯分类做必要的准备，主要工作是根据具体情况确定特征属性，并对每个特征属性进行适当划分，然后由人工对一部分待分类项进行分类，形成训练样本集合。这一阶段的输入是所有待分类数据，输出是特征属性和训练样本。这一阶段是整个朴素贝叶斯分类中唯一需要人工完成的阶段，其质量对整个过程将有重要影响，分类器的质量很大程度上由特征属性、特征属性划分及训练样本质量决定。

**第二阶段:** 分类器训练阶段，这个阶段的任务就是生成分类器，主要工作是计算每个类别在训练样本中的出现频率及每个特征属性划分对每个类别的条件概率估计，并将结果记录。其输入是特征属性和训练样本，输出是分类器。这一阶段是机械性阶段，根据前面讨论的公式可以由程序自动计算完成。

**第三阶段:** 应用阶段。这个阶段的任务是使用分类器对待分类项进行分类，其输入是分类器和待分类项，输出是待分类项与类别的映射关系。这一阶段也是机械性阶段，由程序完成。

2.实验框架

（1）汽车评估数据集

汽车评估数据集包含1728个数据，其中训练数据1350，测试数据 个。每个数据包含6个属性，所有的数据分为4类：

Class Values: unacc, acc, good, vgood   
Attributes:   
buying: vhigh, high, med, low.   
maint: vhigh, high, med, low.   
doors: 2, 3, 4, 5more.   
persons: 2, 4, more.   
lug\_boot: small, med, big.   
safety: low, med, high.

（2）程序框架

1.获取训练样本

2.对每个类别进行统计（Class Values: unacc, acc, good, vgood）

3.对每个属性进行统计（buying: vhigh, high, med, low. maint: vhigh, high, med, low. doors: 2, 3, 4, 5more. persons: 2, 4, more. lug\_boot: small, med, big. safety: low, med, high.）

4.计算并存储每个属性在类别中的条件概率，例如P(buing:vhigh | class value:unacc) =（buing中vhigh的数量）除以 （classValue中unacc的数量）

5.获取测试集样本，将测试数据放入训练集

6.根据每一条测试数据的前六项，每一项在训练集的每个类别中都分别有一个概率值，将每一类别中六个概率值相乘，可得四个评估概率。例如，对于测试数据（low,vhigh,4,2,small,low,unacc）

评价结果V1=p(low| unacc)\* p(vhigh | unacc) \* p(4 | unacc) \* p(2 | unacc)\* p(small | unacc)\* p(low,| unacc)

V2=p(low| acc)\* p(vhigh | acc) \* p(4 | acc) \* p(2 | acc)\* p(small | acc)\* p(low,| acc)

V3=p(low| good)\* p(vhigh | good) \* p(4 | good) \* p(2 | good)\* p(small | good)\* p(low,| good)

V4=p(low| vgood)\* p(vhigh | vgood) \* p(4 | vgood) \* p(2 | vgood)\* p(small | vgood)\* p(low,| vgood)

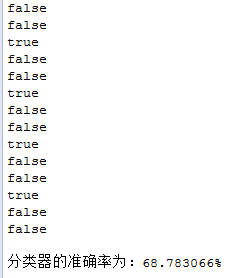
比较VI、V2、V3、V4中概率最大的评价，可得评价结果

7，对评估概率进行比价，可得最大概率，最大概率指向的类别即为测试结果

8.测试结果与测试数据最后一项进行比对，若相同表示测试正确，返回true，反之，若不相同，则返回false。

9.统计true的数量，除以测试样本数量，可得分类器的准确率。

（3）实验结果



种训练集下得到的样本测试准确率为68.78%左右。朴素贝叶斯的测试性能与训练集有关，训练集好则准确率高。

四．实验代码

package rgzn;

import java.io.BufferedReader;

import java.io.File;

import java.io.FileNotFoundException;

import java.io.FileReader;

import java.io.InputStreamReader;

import java.math.BigDecimal;

import java.net.URL;

import java.util.ArrayList; import java.util.Scanner;

import java.util.Vector;

//汽车属性

class Car {

public String buying;// vhigh,high,med,low

public String maint;// vhigh,high,med,low

public String doors;// 2,3,4,5more

public String persons;// 2,4,more

public String lug\_boot;// small ,med,big

public String safety;// low,med,high

public String ClassValues;// unacc,acc, good,vgood

public String[] predictResult = new String[5];// 记录预测结果

}

public class NavieBayes {

Vector<Car> vector;

int testTotal = 0;// 训练样本数量

int predictTotal = 0;// 测试样本的数据

int predictSucess = 0;// 预测成功的数量

String[] ClassValueName = { "unacc", "acc", "good", "vgood" };

// 存储数量

int[] ClassValueTotal = new int[4];// unacc-0 acc-1 good-2 vgood-3

int[][] buying\_Vlaue = new int[4][4]; // 前面是自己的属性，后面是value的属性

int[][] maint\_Value = new int[4][4];

int[][] doors\_Value = new int[4][4];

int[][] persons\_Value = new int[3][4];

int[][] lugboot\_Value = new int[3][4];

int[][] safety\_Value = new int[3][4];

// 存储概率

float[] ClassValueTotal\_gl = new float[4];// unacc-0 acc-1 good-2 vgood-3

float[][] buying\_Vlaue\_gl = new float[4][4]; // 前面是自己的属性，后面是value的属性

float[][] maint\_Value\_gl = new float[4][4];

float[][] doors\_Value\_gl = new float[4][4];

float[][] persons\_Value\_gl = new float[3][4];

float[][] lugboot\_Value\_gl = new float[3][4];

float[][] safety\_Value\_gl = new float[3][4];

public NavieBayes() {

vector = new Vector<Car>();//存储数据

}

/\*\*

\* 主函数

\*/

public static void main(String[] args) throws Exception {

NavieBayes pSbys = new NavieBayes();

pSbys.getData();// 获取训练样本

pSbys.dataTest();//传入测试样本数据，检测其分类器的性能

pSbys.showGL();// 展示结果

}

/\*\*

\* 统计数量，为计算概率做准备

\*/

private void dataTongJi(Car car) {

for(int i=0;i<4;i++){

if(car.ClassValues.equals(ClassValueName[i])){

ClassValueTotal[i]++;

//buying:vhigh,high,med,low

if(car.buying.equals("vhigh")) buying\_Vlaue[0][i]++;

else if(car.buying.equals("high"))buying\_Vlaue[1][i]++;

else if(car.buying.equals("med")) buying\_Vlaue[2][i]++;

else buying\_Vlaue[3][i]++;

//maint\_Value:vhigh,high,med,low

if(car.maint.equals("vhigh")) maint\_Value[0][i]++;

else if(car.maint.equals("high")) maint\_Value[1][i]++;

else if(car.maint.equals("med")) maint\_Value[2][i]++;

else maint\_Value[3][i]++;

//doors\_Value:2,3,4,5more

if(car.doors.equals("2")) doors\_Value[0][i]++;

else if(car.doors.equals("3")) doors\_Value[1][i]++;

else if(car.doors.equals("4")) doors\_Value[2][i]++;

else doors\_Value[3][i]++;

//persons\_Value:2,4,more

if(car.persons.equals("2")) persons\_Value[0][i]++;

else if(car.persons.equals("4")) persons\_Value[1][i]++;

else persons\_Value[2][i]++;

//lugboot\_Value:small ,med,big

if(car.lug\_boot.equals("small")) lugboot\_Value[0][i]++;

else if(car.lug\_boot.equals("med")) lugboot\_Value[1][i]++;

else lugboot\_Value[2][i]++;

//safety\_Value:low,med,high

if(car.safety.equals("low")) safety\_Value[0][i]++;

else if(car.safety.equals("med")) safety\_Value[1][i]++;

else safety\_Value[2][i]++;

}

}

}

/\*\*

\* 学习过程获取训练样本,统计个数，计算概率

\*

\* @param url

\*/

private void getData() throws Exception {

InputStreamReader in = new InputStreamReader(getClass()

.getResourceAsStream("learn.txt"));

Scanner scanner = new Scanner(in);

while (scanner.hasNext()) {

testTotal++;

String[] temp = scanner.nextLine().split(",");

Car car = new Car();

car.buying = temp[0];

car.maint = temp[1];

car.doors = temp[2];

car.persons = temp[3];

car.lug\_boot = temp[4];

car.safety = temp[5];

car.ClassValues = temp[6];

dataTongJi(car);// 统计数据的个数

}

gailvTongJi();// 统计概率

}

/\*\*

\* 概率统计

\*/

private void gailvTongJi() {

for (int i = 0; i < ClassValueTotal.length; i++) {

ClassValueTotal\_gl[i] = (float) ClassValueTotal[i] / testTotal;

}

for (int i = 0; i < buying\_Vlaue\_gl.length; i++) {

for (int j = 0; j < buying\_Vlaue\_gl[0].length; j++) {

buying\_Vlaue\_gl[i][j] = (float) buying\_Vlaue[i][j]

/ ClassValueTotal[j];

maint\_Value\_gl[i][j] = (float) maint\_Value[i][j]

/ ClassValueTotal[j];

doors\_Value\_gl[i][j] = (float) doors\_Value[i][j]

/ ClassValueTotal[j];

}

}

for (int i = 0; i < persons\_Value\_gl.length; i++) {

for (int k = 0; k < persons\_Value\_gl[0].length; k++) {

persons\_Value\_gl[i][k] = (float) persons\_Value[i][k]

/ ClassValueTotal[k];

lugboot\_Value\_gl[i][k] = (float) lugboot\_Value[i][k]

/ ClassValueTotal[k];

safety\_Value\_gl[i][k] = (float) safety\_Value[i][k]

/ ClassValueTotal[k];

}

}

}

//结果显示

//for (int j = 0; j < c.predictResult.length; j++) {

//System.out.print(c.predictResult[j] + "\t\t");

//}

private void showGL() {

for (int i = 0; i < predictTotal; i++) {

Car c = vector.get(i);

System.out.print(c.predictResult[c.predictResult.length-1] + "\t\t");

System.out.println();

}

// 分类器的准确率

float t = (float) predictSucess / predictTotal;

t = (t \* 1000) / 10;

System.out.println("\n分类器的准确率为：" + t + "%");

}

/\*\*

\* 给定测试样本，测试其分类器性能如何

\* 读取数据

\* @throws FileNotFoundException

\*/

private void dataTest() throws FileNotFoundException {

InputStreamReader in = new InputStreamReader(getClass()

.getResourceAsStream("test.txt"));

Scanner scanner = new Scanner(in);

while (scanner.hasNext()) {

predictTotal++;

String[] temp = scanner.nextLine().split(",");

Car car = new Car();

car.buying = temp[0];

car.maint = temp[1];

car.doors = temp[2];

car.persons = temp[3];

car.lug\_boot = temp[4];

car.safety = temp[5];

car.ClassValues = temp[6];

vector.add(car);

xingnengTest(car);// 性能测试

}

}

/\*\*

\* 对分类器进行性能测试，判断其成功率为多少

\*

\* @param car

\*/

private void xingnengTest(Car car) {

// unacc,acc,good,vgood, P(yi)--ClassValueTotal\_gl 、 P(x|yi)=

// low,vhigh,4,2,small,low,unacc 第一条

float itemGl;// 每一条的概率

int b, m, d, p, l, s;

b = m = d = p = l = s = -1;

float MaxGl = 0;// 临时概率

if (car.buying.equals("vhigh")) {

b = 0;

} else if (car.buying.equals("high")) {

b = 1;

} else if (car.buying.equals("med")) {

b = 2;

} else {

b = 3;

}

if (car.maint.equals("vhigh")) {

m = 0;

} else if (car.maint.equals("high")) {

m = 1;

} else if (car.maint.equals("med")) {

m = 2;

} else {

m = 3;

}

if (car.doors.equals("2")) {

d = 0;

} else if (car.doors.equals("3")) {

d = 1;

} else if (car.doors.equals("4")) {

d = 2;

} else {

d = 3;

}

if (car.persons.equals("2")) {

p = 0;

} else if (car.persons.equals("4")) {

p = 1;

} else {

p = 2;

}

if (car.lug\_boot.equals("small")) {

l = 0;

} else if (car.lug\_boot.equals("med")) {

l = 1;

} else {

l = 2;

}

if (car.safety.equals("low")) {

s = 0;

} else if (car.safety.equals("med")) {

s = 1;

} else {

s = 2;

}

int t = 0;// 记录最大概率的下标

int i;

// 如：

// unacc

// low(0.038),vhigh(0.329),4(0.2323),2(0.4545),small(0.3737),low(0.4545),unacc()

for (i = 0; i < ClassValueTotal\_gl.length; i++) {// 计算在unacc,acc,good,vgood下的概率

itemGl = 0;

BigDecimal[] bigDecimal = {

new BigDecimal(Float.toString(ClassValueTotal\_gl[i])),

new BigDecimal(Float.toString(buying\_Vlaue\_gl[b][i])),

new BigDecimal(Float.toString(maint\_Value\_gl[m][i])),

new BigDecimal(Float.toString(doors\_Value\_gl[d][i])),

new BigDecimal(Float.toString(persons\_Value\_gl[p][i])),

new BigDecimal(Float.toString(lugboot\_Value\_gl[l][i])),

new BigDecimal(Float.toString(safety\_Value\_gl[s][i])), };

for (int j = 1; j < bigDecimal.length; j++) {//除:a.divide(b,2);//2为精度取值

bigDecimal[0] = bigDecimal[0].multiply(bigDecimal[j]);//multiply乘

}

itemGl = bigDecimal[0].floatValue();

car.predictResult[i] = itemGl + " ";

if (MaxGl < itemGl) {

MaxGl = itemGl;

t = i;

}

}

// 判断结果是否正确

if (car.ClassValues.equals(ClassValueName[t])) {// 预测结果和开始给定的结果相等

car.predictResult[i] = "true";

predictSucess++;

} else {

car.predictResult[i] = "false";

}

}

}