实验七 朴素贝叶斯

实验代码：

#include <iostream>

#include <set>

#include <vector>

using namespace std;

//定义训练数据

#define M 3

#define N 15

//为了计算简单，对A2={S, M, L},令S=1, M=2, L=3；

//Y={1， -1}，令为Y={1， 2}

int A[M][N]= {

{1 , 1 , 1 , 1 , 1 , 2 , 2 , 2 , 2 , 2 , 3 , 3 , 3 , 3 , 3},

{1 , 2 , 2 , 1 , 1 , 1 , 2 , 2 , 3 , 3 , 3 , 2 , 2 , 3 , 3},

{2 , 2 , 1 , 1 , 2 , 2 , 2 , 1 , 1 , 1 , 1 , 1 , 1 , 1 , 2}

};

struct Px1

{

int x1;

int y;

double p\_x1y;

};

struct Px2

{

int x2;

int y;

double p\_x2y;

};

double p[2];

Px1 px1[6];

Px2 px2[6];

//计算先验概率和条件概率

void calP()

{

//计算先验

//double p[2];

int i, j, k;

multiset<int> m\_x1, m\_x2, m\_y;//多重集容器

multiset<int>::iterator pos1;

set<int> x1, x2, y;//集合容器

set<int>::iterator pos2, pos3;

//运用多重集容器和集合容器

for(i = 0; i < N; i++)

{

m\_x1.insert(A[0][i]);

m\_x2.insert(A[1][i]);

m\_y.insert(A[2][i]);

x1.insert(A[0][i]);

x2.insert(A[1][i]);

y.insert(A[2][i]);

}

p[0] = m\_y.count(1) / (double)N; //p(Y = 1)

p[1] = m\_y.count(2) / (double)N; //p(Y = 2)

cout << endl << "\*\*\*\*\*\*\*\*\*\*\*\*先验\*\*\*\*\*\*\*\*\*\*\*" << endl;

cout << "p(Y = 1) = " << p[0] << endl;

cout << "p(Y = 2) = " << p[1] << endl;

//计算条件概率

cout << endl;

cout << "\*\*\*\*\*\*\*\*\*条件概率\*\*\*\*\*\*\*\*" << endl;

// int px1\_num = 3 \* 2;

// int px2\_num = 3 \* 2;

j=0;

for(pos2 = y.begin(); pos2 != y.end(); pos2++)

{

for(pos3 = x1.begin(); pos3 != x1.end(); pos3++)

{

px1[j].y = \*pos2;

px1[j].x1 = \*pos3;

int count\_x1y = 0;

for(k = 0; k < N; k++)

{

if(A[0][k] == px1[j].x1 && A[2][k] == px1[j].y)

count\_x1y++;

}

px1[j].p\_x1y = count\_x1y / (double)m\_y.count(px1[j].y);//计算p(x1 | y)的概率

j++;

}

}

cout << "p(x1 | y):" << endl;

for(j = 0; j < 6; j++)

{

cout << px1[j].x1 << " " << px1[j].y << " " << px1[j].p\_x1y << endl;

}

j=0;

for(pos2 = y.begin(); pos2 != y.end(); pos2++)

{

for(pos3 = x2.begin(); pos3 != x2.end(); pos3++)

{

px2[j].y = \*pos2;

px2[j].x2 = \*pos3;

int count\_x2y = 0;

for(k = 0; k < N; k++)

{

if(A[1][k] == px2[j].x2 && A[2][k] == px2[j].y)

count\_x2y++;

}

px2[j].p\_x2y = count\_x2y / (double)m\_y.count(px2[j].y);//计算p(x2 | y)的概率

j++;

}

}

cout << "p(x2 | y):" << endl;

for(j = 0; j < 6; j++)

{

cout << px2[j].x2 << " " << px2[j].y << " " << px2[j].p\_x2y << endl;

}

}

int main()

{

int i = 0, j = 0;

//输出训练数据

cout << "\*\*\*\*\*\*\*\*\*\*\*训练数据\*\*\*\*\*\*\*\*\*\*\*\*" << endl;

for(i = 0; i < M; i++)

{

if(i == 0) cout << "X1: ";

else if(i == 1) cout << "X2: ";

else if(i == 2) cout << " Y: ";

for(int j = 0; j < N; j++)

{

cout << " "<< A[i][j];

}

cout << endl;

}

calP();//计算先验和条件概率

int s\_x1, s\_x2;

double result[2];

int class\_y = 1;

cout << "\*\*\*\*\*\*\*\*\*\*\*\*\*预测\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*" << endl;

cout << endl << endl << "Input:";

cin >> s\_x1 >> s\_x2;

for(i = 0; i < 2; i++)

{

double s\_px\_1, s\_px\_2;

for(j = 0; j < 6; j++)

{

if(s\_x1 == px1[j].x1 && px1[j].y == class\_y)

s\_px\_1 = px1[j].p\_x1y;

if(s\_x2 == px2[j].x2 && px2[j].y == class\_y)

s\_px\_2 = px2[j].p\_x2y;

}

result[i] = p[i] \* s\_px\_1 \* s\_px\_2;

class\_y++;

}

cout << endl << "all results:";

cout << result[0] << " " << result[1] << endl;

for(i = 0; i <2; i++)

{

class\_y = i;

if(result[i] < result[i+1])

{

class\_y = i+1;

}

}

cout << "("<< s\_x1 << "," << s\_x2 << ")所属的类是：" << class\_y + 1 << endl;

return 0;

}

运行结果：

