In [3]:

```
def loadDataSet():  #加载表4.1数据
    x1=[1,1,1,1,1,2,2,2,2,2,3,3,3,3,3]
    x2=[’S’,’M’,’M’,’S’,’S’,’S’,’M’,’M’,’L’,’L’,’L’,’M’,’M’,’L’,’L’]
    Y=[-1,-1,1,1,-1,-1,-1,1,1,1,1,1,1,1,-1]
    return x1,x2,Y

def nb(x1,x2,Y):  #朴素贝叶斯算法部分
    x1_s=set(x1)  #特征1去重：建立特征一的词典
    x2_s=set(x2)  #特征二去重
    p(x1,x1_s,Y)  #计算特征一分别属于1跟-1的概率，就是就是书上1，3两行
    p(x2,x2_s,Y) #计算特征二属于1，-1概率，书上2，4两行

def p(xj,x_s,Y):
    for x in x_s:  #对每一个特征
        xcount = 0
        _xcount=0
        for i in range(15):#对每一个样本
            if Y[i]==1 and xj[i]==x: #再y=1时，每个特征出现次数
                xcount+=1
            elif Y[i]==-1 and xj[i]==x:#y=-1时，每个特征出现次数
                _xcount+=1

        print(’x=%s,Y=%d,p=%f’%(x,1,xcount/Y.count(1)))  #打印概率
        print(’x=%s,Y=%d,p=%f’%(x,-1,_xcount/Y.count(-1)))
x1,x2,Y=loadDataSet()
nb(x1,x2,Y)
```

```
x=1,Y=1,p=0.222222
x=1,Y=-1,p=0.500000
x=2,Y=1,p=0.333333
x=2,Y=-1,p=0.333333
x=3,Y=1,p=0.444444
x=3,Y=-1,p=0.166667
x=S,Y=1,p=0.111111
x=S,Y=-1,p=0.500000
x=M,Y=1,p=0.444444
x=M,Y=-1,p=0.333333
x=L,Y=1,p=0.444444
x=L,Y=-1,p=0.166667
```

In [2]:

```
def loadDataSet():
    x1=[1,1,1,1,1,2,2,2,2,2,3,3,3,3,3]
    x2=['S','M','M','S','S','S','M','M','L','L','L','M','M','L','L']
    Y=[-1,-1,1,1,-1,-1,-1,1,1,1,1,1,1,1,-1]
    return x1,x2,Y
```

输入样本

```
def nb(x1,x2,Y):
    py1=Y.count(1)/len(Y)
    # print('py1=',py1)
    py2=1-py1
    # print('py2=',py2)
    x1_s=set(x1)
    x2_s=set(x2)
    for xi in x1_s:
        p(x1,xi,Y)
    for xj in x2_s:
        p(x2,xj,Y)
    return py1,py2
```

朴素贝叶斯

去重 词典.

去重x2.

对. X1列

Xj x2 -

```
def p(xj,value,Y):
    xcount = 0
    _xcount=0
    for i in range(15):
        if Y[i]==1 and xj[i]==value:
            xcount+=1
        elif Y[i]==-1 and xj[i]==value:
            _xcount+=1
    #print('x=%s,Y=%d,p=%f' % (value, 1, xcount / Y.count(1)))
    #print('x=%s,Y=%d,p=%f' % (value, -1, _xcount / Y.count(-1)))
    pxy1=xcount / Y.count(1)
    pxy2=_xcount / Y.count(-1)
    return pxy1,pxy2
```

任意特征.

任意样本点.

打印概率.

```
def classify(x1,x2,inX,Y):
    px1y1,px1y2=p(x1,int(inX[0]),Y)  #传入特征一
    px2y1,px2y2=p(x2,inX[1],Y)    #传入特征2
    py1,py2=nb(x1,x2,Y)
    p1Vec=py1*px1y1*px2y1
    p2Vec=py2*px1y2*px2y2
    print('Y类别为1,-1时概率分别为',p1Vec, p2Vec)
    if p1Vec>p2Vec:                     #看y=1概率大还是y=-1概率大
        print('x=(%s,%s)被分为Y=1'%(inX[0],inX[1]))
    else:print('x=(%s,%s)被分为Y=-1'%(inX[0],inX[1]))
x1,x2,Y=loadDataSet()
nb(x1,x2,Y)
inX=input('').split()
classify(x1,x2,inX,Y)
```

```
2 S
Y类别为1,-1时概率分别为 0.022222222222222222 0.06666666666666667
x=(2,S)被分为Y=-1
```

In [4]:

```
def loadDataSet():  #加载表4.1数据
    x1=[1,1,1,1,1,2,2,2,2,2,3,3,3,3,3]
    x2=['S','M','M','S','S','S','M','M','L','L','L','M','M','L','L']
    Y=[-1,-1,1,1,-1,-1,-1,1,1,1,1,1,1,1,-1]
    return x1,x2,Y

def nb(x1,x2,Y):  #朴素贝叶斯算法部分
    x1_s=set(x1)  #特征1去重：建立特征一的词典
    x2_s=set(x2)  #特征二去重
    s1=len(x1_s)
    s2=len(x2_s)
    p(x1,x1_s,s1,Y)  #计算特征一分别属于1跟-1的概率，就是就是书上1，3两行
    p(x2,x2_s,s1,Y) #计算特征二属于1，-1概率，书上2，4两行

def p(xj,x_s,sj,Y):
    for x in x_s:  #对每一个特征
        xcount = 0
        _xcount=0
        for i in range(15):#对每一个样本
            if Y[i]==1 and xj[i]==x: #再y=1时，每个特征出现次数
                xcount+=1
            elif Y[i]==-1 and xj[i]==x:#y=-1时，每个特征出现次数
                _xcount+=1

        print('x=%s,Y=%d,p=%f'%(x,1,(xcount+1)/(Y.count(1)+sj)))  #打印概率
        print('x=%s,Y=%d,p=%f'%(x,-1,(xcount+1)/(Y.count(-1)+sj)))

x1,x2,Y=loadDataSet()
nb(x1,x2,Y)
```

```
x=1,Y=1,p=0.250000
x=1,Y=-1,p=0.333333
x=2,Y=1,p=0.333333
x=2,Y=-1,p=0.444444
x=3,Y=1,p=0.416667
x=3,Y=-1,p=0.555556
x=S,Y=1,p=0.166667
x=S,Y=-1,p=0.222222
x=M,Y=1,p=0.416667
x=M,Y=-1,p=0.555556
x=L,Y=1,p=0.416667
x=L,Y=-1,p=0.555556
```