



東南大學
SOUTHEAST UNIVERSITY

模式识别

感知器算法实验报告

姓名：黄勉

学号：57118304

东南大学网络空间安全学院
School of Cyber Science & Engineering
Southeast University

2020 年 12 月

一. 问题描述

编写感知器算法程序，求下列模式分类的解向量：

$\omega_1: \{(0\ 0\ 0)^T, (1\ 0\ 0)^T, (1\ 0\ 1)^T, (1\ 1\ 0)^T\}$

$\omega_2: \{(0\ 0\ 1)^T, (0\ 1\ 1)^T, (0\ 1\ 0)^T, (1\ 1\ 1)^T\}$

设 $w(1)=(-1\ -2\ -2\ 0)^T$

二. 算法描述

1) 选择 n 个分属于 ω_1 、 ω_2 两类的模式样本构成训练样本集，对 ω_2 类样本全部乘以 (-1) ，任取权重向量初始值 $w(1)$ ，开始迭代 $t=1$

2) 用全部训练样本进行一轮迭代，计算 $w^T x_i$ 的值，并修正权重向量；

$$w_t = \begin{cases} w_{t-1}, & w_{t-1}^T x_i \geq 0, \\ w_{t-1} + \eta x_i, & w_{t-1}^T x_i < 0. \end{cases}$$

3) 只要有一个错误分类，回到步骤 2)，直至对所有样本正确分类

三. 计算过程

1. 取增量校正参数为 1。
2. 通过 `Augmented_vector` 函数将训练样本写成增广向量形式
3. 通过 `Iteration` 函数迭代，设置 `flag` 为标志位，判断是否还需要继续迭代

四. 代码实现

```
#include<iostream>
using namespace std;

//初始化训练样本
int w1[4][3]={{0,0,0},{1,0,0},{1,0,1},{1,1,0}};
int w2[4][3]={{0,0,1},{0,1,1},{0,1,0},{1,1,1}};

int l=0;//记录 W 迭代次数
int t=1;//记录迭代轮数
int W[100][4]={{-1,-2,-2,0}};
```

```

int c=1;//校正参量系数
int X[8][4];//增广向量

//将训练样本写成增广向量形式
void Augmented_vector(int a[4][3],int b[4][3])
{
    for(int i=0;i<4;i++){
        for(int j=0;j<3;j++){
            X[i][j]=a[i][j];
            X[i][3]=1;
        }
    }
    for(int i=4;i<8;i++){
        for(int j=0;j<3;j++){
            X[i][j]=-b[i-4][j];
            X[i][3]=-1;
        }
    }
    /*
    for(int i=0;i<8;i++){
        for(int j=0;j<4;j++){
            cout<<X[i][j];
        }
        cout<<endl;
    } */
}

//判别迭代
void Iteration()
{
    cout<<"第"<<t<<"轮迭代: "<<endl;
    int flag=0;
    for(int k=0;k<8;k++){
        int s=0;
        for(int j=0;j<4;j++){
            s+=W[l][j]*X[k][j];
        }
        cout<<"W["<<l+1<<"]TX"<<k+1<<"]="<<s;

        if(s>0){
            cout<<">0,故"<<endl;
            cout<<"W["<<l+2<<"]=W["<<l+1<<"]=";
            for(int j=0;j<4;j++){
                W[l+1][j]=W[l][j];
                cout<<W[l+1][j]<<" ";
            }
        }
    }
}

```

```

        }
        cout<<endl;
    }
    else{
        cout<<"<=0,故"<<endl;
        cout<<"W["<<l+2<<"]="W["<<l+1<<"]+c*X["<<k+1<<"]="";
        for(int j=0;j<4;j++){
            W[l+1][j]=W[l][j]+c*X[k][j];
            cout<<W[l+1][j]<<" ";
        }
        cout<<endl;
        flag++;
    }
    l++;
}
t++;
if(flag>0)
    Iteration();
else{
    cout<<"迭代次数: "<<t-1<<endl;
    cout<<"解向量为"<<endl;
    for(int j=0;j<4;j++){
        cout<<" "<<W[l][j];
    }
}
}

int main()
{
    Augmented_vector(w1,w2);
    Iteration();
    return 0;
}

```

五. 结果分析

第1轮迭代: W[1] TX1=0<0, 故 W[2]=W[1]+c*X[1]=-1 -2 -2 1 W[2] TX2=0<0, 故 W[3]=W[2]+c*X[2]=0 -2 -2 2 W[3] TX3=0<0, 故 W[4]=W[3]+c*X[3]=1 -2 -1 3 W[4] TX4=2>0, 故 W[5]=W[4]=1 -2 -1 3 W[5] TX5=-2<0, 故 W[6]=W[5]+c*X[5]=1 -2 -2 2 W[6] TX6=2>0, 故 W[7]=W[6]=1 -2 -2 2 W[7] TX7=0<0, 故 W[8]=W[7]+c*X[7]=1 -3 -2 1 W[8] TX8=3>0, 故 W[9]=W[8]=1 -3 -2 1	第2轮迭代: W[9] TX1=1>0, 故 W[10]=W[9]=1 -3 -2 1 W[10] TX2=2>0, 故 W[11]=W[10]=1 -3 -2 1 W[11] TX3=0<0, 故 W[12]=W[11]+c*X[3]=2 -3 -1 2 W[12] TX4=1>0, 故 W[13]=W[12]=2 -3 -1 2 W[13] TX5=-1<0, 故 W[14]=W[13]+c*X[5]=2 -3 -2 1 W[14] TX6=4>0, 故 W[15]=W[14]=2 -3 -2 1 W[15] TX7=2>0, 故 W[16]=W[15]=2 -3 -2 1 W[16] TX8=2>0, 故 W[17]=W[16]=2 -3 -2 1
--	--

第3轮迭代: W[17] TX1=1>0, 故 W[18]=W[17]=2 -3 -2 1 W[18] TX2=3>0, 故 W[19]=W[18]=2 -3 -2 1 W[19] TX3=1>0, 故 W[20]=W[19]=2 -3 -2 1 W[20] TX4=0<0, 故 W[21]=W[20]+c*X[4]=3 -2 -2 2 W[21] TX5=0<0, 故 W[22]=W[21]+c*X[5]=3 -2 -3 1 W[22] TX6=4>0, 故 W[23]=W[22]=3 -2 -3 1 W[23] TX7=1>0, 故 W[24]=W[23]=3 -2 -3 1 W[24] TX8=1>0, 故 W[25]=W[24]=3 -2 -3 1	第4轮迭代: W[25] TX1=1>0, 故 W[26]=W[25]=3 -2 -3 1 W[26] TX2=4>0, 故 W[27]=W[26]=3 -2 -3 1 W[27] TX3=1>0, 故 W[28]=W[27]=3 -2 -3 1 W[28] TX4=2>0, 故 W[29]=W[28]=3 -2 -3 1 W[29] TX5=2>0, 故 W[30]=W[29]=3 -2 -3 1 W[30] TX6=4>0, 故 W[31]=W[30]=3 -2 -3 1 W[31] TX7=1>0, 故 W[32]=W[31]=3 -2 -3 1 W[32] TX8=1>0, 故 W[33]=W[32]=3 -2 -3 1
---	---

迭代次数: 4
解向量为
3 -2 -3 1

感知器算法是一种赏罚过程: 当分类器发生分类错误时, 对分类器进行“罚”——修改权向量; 分类正确时, 对分类器进行“奖”——即权向量不变。

感知器算法是收敛的。只要模式类别是线性可分的, 就可以在有限的迭代步数中求出权向量的解。