模式识别与机器学习实验报告

150325 班
15141009
郑琦

May 3, 2018

实验二、贝叶斯分类器

一、实验背景

Linear perceptrons allow us to learn a decision boundary that would separate two classes. They are very effective when there are only two classes, and they are well separated. Such classifiers are referred to as discriminative classifiers. In contrast, generative classifiers consider each sample as a random feature vector, and explicitly model each class by their distribution or density functions. To carry out the classification, the likelihood function should be computed for a given sample which belongs to one of candidate classes so as to assign the sample to the class that is most likely. In other words, we need to compute p(|X) for each class i. However, the density functions provide only the likelihood of seeing a particular sample, given that the sample belongs to a specific class. i.e., the density functions can be provided as p(X|) . The Bayesian rule provides us with an approach to compute the likelihood of the class for a given sample, from the density functions and related information.

二、实验原理

The essence of the Bayesian approach is to provide a mathematical rule explaining how you should change your existing beliefs in the light of new evidence. In other words, it allows us to combine new data with their existing knowledge or expertise. The canonical example is to imagine that a precocious newborn observes his first sunset, and wonders whether the sun will rise again or not. He assigns equal prior probabilities to both possible outcomes, and represents this by placing one white and one black marble into a bag. The following day, when the sun rises, the child places another white marble in the bag. The probability that a marble plucked randomly from the bag will be white (i.e., the child' s degree of belief in future sunrises) has thus gone from a half to two-thirds. After sunrise the next day, the child adds another white marble, and the probability (and thus the degree of belief) goes from two-thirds to three-quarters. And so on. Gradually, the initial belief that the sun is just as likely as not to rise each morning is modified to become a near-certainty that the sun will always rise.

In terms of classification, the Bayesian theorem allows us to combine prior

probabilities, along with observed evidence to arrive at the posterior probability. More or less, conditional probabilities represent the probability of an event occurring given evidence. According to the Bayesian Theorem, if P(ω_i) , P(X|ω_i) , i 1,2, , c, and X are known or given, the posterior probability can be derived as follows

$$P(\omega_i|X) = \frac{P(X|\omega_i)P(\omega_i)}{\sum_{j=1}^{c} P(X|\omega_i)P(\omega_i)} i = 1, \ldots, c$$

Let the series of decision actions as a , a , , a , the conditional risk of decision action a can be computed by

$$R(a_i|X) = \sum_{j=1.i=1}^{c} \lambda(a_i, \omega_i)P(\omega_j|X), i = 1, \ldots, c$$

Thus the minimum risk Bayesian decision can be found as

$$a_k^* = Arg\, min_i\, R(a_i|X) i = 1, \ldots, c$$

三、实验目标

The goals of the experiment are as follows:

(1) To understand the computation of likelihood of a class, given a sample.

(2) To understand the use of density/distribution functions to model a class.

(3) To understand the effect of prior probabilities in Bayesian classification.

(4) To understand how two (or more) density functions interact in the feature space to decide a decision boundary between classes.

(5) To understand how the decision boundary varies based on the nature of density functions.

四、实验过程

Stage 1:

(1) According to the above principle and theory in section 2.2, design a Bayesian classifier for the classification of two classes of patterns which are subjected to Gaussian normal distribution and compile the corresponding programme codes.

(2) In view of the normal cell class ω_1 , the corresponding data of sample features are extracted as

Ω_1 = -3.9847, -3.5549, -1.2401, -0.9780, -0.7932, -2.8531, -2.7605, -3.7287, -3.5414, -2.2692, -3.4549, -3.0752, -3.9934, -0.9780, -1.5799, -1.4885, -0.7431, -0.4221, -1.1186, -2.3462, -1.0826, -3.4196, -1.3193, -0.8367, -0.6579, -2.9683, and the sample features of abnormal cell class ω_2 are listed as

Ω_2 = 2.8792, 0.7932, 1.1882, 3.0682, 4.2532, 0.3271,0.9846,2.7648,2.6588

The prior probabilities of both ω_1 and ω_2 are known as $p(\omega_1) = 0.9$, $p(\omega_2) = 0.1$

Suppose the conditional probability distributions are Gaussian, find the conditional probability density functions $p(X|\omega_1)$and$p(X|\omega_2)$ and complete the design of Bayessian classifier with minimum risk, and then give a comparative analysis with the situation without considering decision loss.

Draw the curves of prior and posterior probability density functions, $p(X|\omega_1)$, $p(X|\omega_2), p(\omega_1|X), and p(X\ \omega_2)$, give the classifying decision boundary function and illustration of classification result. Stage 2 (towards an in depth study):
(1) Create a pattern dataset of multiple classes and high dimension with more than 50 samples for each class. Then design a Bayesian classifier and complete the corresponding experiments and comparative analysis by using self-supposed prior probabilities and loss parameters for the same terms as stage 1.
(2) Think and analyse the intrinsic relationship between the classifier of two classes and the one of multiple classes, give your comments.
Stage 3:
Explore the questions in the previous section and design experiments to answer these questions. Complete and submit an experiment report about all experiment results with comparative analysis and a summary of experiences about this experiment study.

五、实验结果

1. 使用 c++ 编写贝叶斯分类器分类正态分布情况下的数据集
假设以男女身体数据为例，分别读入数据，并且可以假设高斯分布的参数 $\mu_1\ \mu_2\ \theta_1\ \theta_2\ \rho$

```
//
//  main.cpp
//  线性分类器:线性分类器
//
//  Created by Qi Zheng on 2018/4/28.
//  Copyright © 2019年 Matrix. All rights reserved.
//  簡單的生活
//  何嘗不是一場華麗的冒險

#include<iostream>
#include<fstream>
#include<cmath>
#include<cstdio>
using namespace std;
const int MAXN=1000;
const double pi=3.1415926;
ifstream cin1("FEMALE.TXT");
ifstream cin2("MALE.TXT");
ifstream cin3("test2.txt");
ofstream cout1("result.txt");
struct HUMAN
{
    double height;
    double weight;
};
HUMAN female[MAXN];
HUMAN male[MAXN];
```

```
int female_num;
int male_num;

double P_female;
double P_male;
struct NORMAL
{
  double mu1;
  double mu2;
  double delta1;
  double delta2;
  double rho;
};
NORMAL female_normal;
NORMAL male_normal;
/*
读入文件数据
*/
void In()
{
     male_num=0;
     female_num=0;
     while(cin1>>female[female_num+1].height>>female[
         female_num+1].weight)
     {
          female_num++;
     }
     while(cin2>>male[male_num+1].height>>male[male_num
         +1].weight)
     {
          male_num++;
     }
}
void Init()
{
       P_female=0.5;
       P_male=0.5;
}
/*
读入样本数量个样本，并求出该样本的二维正态分布
*/
void Normalization(struct HUMAN *human,int human_num,
    struct NORMAL &human_normal)
{
     double mu1,mu2,delta1,delta2,rho;
     mu1=0;mu2=0;delta1=0;delta2=0,rho=0;
```

```
    for(int i=1;i<=human_num;i++)
    {
        mu1+=human[i].height;
        mu2+=human[i].weight;
    }
    mu1/=human_num;
    mu2/=human_num;

    for(int i=1;i<=human_num;i++)
    {
        delta1+=(human[i].height-mu1)*(human[i].height-
            mu1);
        delta2+=(human[i].weight-mu2)*(human[i].weight-
            mu2);
    }
    delta1/=human_num;
    delta2/=human_num;
    delta1=sqrt(delta1);
    delta2=sqrt(delta2);
    for(int i=1;i<=human_num;i++)
    {
        rho+=human[i].height*human[i].weight;
    }
    rho/=human_num;
    rho-=mu1*mu2;
    rho/=(delta1*delta2);
    human_normal.mu1=mu1;
    human_normal.mu2=mu2;
    human_normal.delta1=delta1;
    human_normal.delta2=delta2;
    human_normal.rho=rho;
    cout<<mu1<<" "<<delta1<<" "<<mu2<<" "<<delta2<<" "<<
        rho<<endl;
}
/*
在分布为normal的条件下特征为(x1,x2)的条件概率
*/

double P(struct NORMAL &normal,double x1,double x2)
{
    double ans;
    double mu1=normal.mu1;
    double mu2=normal.mu2;
    double delta1=normal.delta1;
    double delta2=normal.delta2;
    double rho=normal.rho;
```

```
    rho=0;
    ans=(1/(2*pi*delta1*delta2*          sqrt(1-rho*rho)
                ))*exp(-1/(2*sqrt(1-rho*rho)    )*(
                 ((x1-mu1)*(x1-mu1))/(delta1*delta1)
       +    ((x2-mu2)*(x2-mu2))/(delta2*delta2)    -     (2*
       rho*(x1-mu1)*(x2-mu2))/(delta1*delta2)
                                                )    );
    return ans;
}
/*
归为normal的后验概率
t为0表示female，1表示male
*/
double Posterior_probability1(double x1,double x2,bool t)
{
    double Pw;
    struct NORMAL normal;
    if(t==0)return (P(female_normal,x1,x2)*P_female)/(P(
        female_normal,x1,x2)*P_female+P(male_normal,x1,x2)
        *P_male);
    else return (P(male_normal,x1,x2)*P_male)/(P(
        female_normal,x1,x2)*P_female+P(male_normal,x1,x2)
        *P_male);
}
/*
判断是哪个类别，返回0表示female,1表示male
*/
bool Classify(double x1,double x2)
{
    //cout<<Posterior_probability1(x1,x2,0)<<" "<<
        Posterior_probability1(x1,x2,1)<<endl;
    if(Posterior_probability1(x1,x2,0)>=
        Posterior_probability1(x1,x2,1))return 0;
    else return 1;
}
/*
得到错误率并将错误率输出到result.txt中
*/
void Find_error_rate()
{
    double height,weight;
    char c;
    int right_num=0;
    int wrong_num=0;
    while(cin3>>height>>weight>>c)
    {
```

```
        if(    (c=='f' || c=='F')   && Classify(height,
            weight)==0)//分类为女性并且正确
              right_num++;
        else if(    (c=='m' || c=='M')   && Classify(
            height,weight)==1)//分类为男性并且正确
              right_num++;
        else
             wrong_num++;

        cout1<<height<<" "<<weight<<" "<<Classify(height,
            weight)<<endl;
    }
    cout<<"error rate is "<<(double)wrong_num/(double)(
        wrong_num+right_num)<<endl;
}
int main()
{
    In();
    Init();
    Normalization(female,female_num,female_normal);
    Normalization(male,male_num,male_normal);
    //female_normal.rho=0;
    //male_normal.rho=0;
    Find_error_rate();
    return 0;
}
```

2. 对病人的病情作出判断

```
//
//  main.cpp
//  线性分类器:线性分类器
//
//  Created by Qi Zheng on 2018/4/28.
//  Copyright © 2019年 Matrix. All rights reserved.
//  簡單的生活
//  何嘗不是一場華麗的冒險

#include <cmath>
#include <iostream>
#include <iomanip>
#include <list>
#include <vector>
#include <random>
using namespace std;
```

```
double normalCFD(double value)
{
    return 0.5 * erfc(-value * M_SQRT1_2);
}

int main(){
    vector<double> omega1
        ={-3.9847,-3.5549,-1.2401,-0.9780,-0.7932,-2.8531,-2.7605,-3.7287,
          -3.5414,-2.2692,-3.4549,-3.0752,-3.9934,
        -0.9780,-1.5799,-1.4885,
        -0.7431,-0.4221,-1.1186,-2.3462,-1.0826,-3.4196,-1.3193,-0.8367,
         -0.6579,-2.9683};
    vector<double> omega2
        ={2.8792,0.7932,1.1882,3.0682,4.2532,0.3271,0.9846,2.7648,2.6588};

    int count1=(int)omega1.size();
    int count2=(int)omega2.size();
    vector<double>::iterator iterator1,iterator2;
    //计算两类样本数据的均值与方差
    double sum=0,squrt=0;
    double average1,average2;
    double sigma1,sigma2;
    for (iterator1=omega1.begin(); iterator1!=omega1.end
        (); iterator1++) {
        sum+=*iterator1;
        squrt+=(*iterator1)*(*iterator1);
    }
    average1=sum/count1;
    sigma1=squrt/count1-average1*average1;
    sum=squrt=0;
    for (iterator2=omega2.begin(); iterator2!=omega2.end
        (); iterator2++) {
        sum+=*iterator2;
        squrt+=(*iterator2)*(*iterator2);
    }
    average2=sum/count2;
    sigma2=squrt/count2-average2*average2;
    cout<<"正常类的平均值与方差为："<<average1<<sigma1<<
        endl;
    cout<<"非正常类的平均值与方差为："<<average2<<sigma2
       <<endl;
    double viewdata;
    cout<<"请输入观测值"<<endl;cin>>viewdata;
    //带入标准正态分布计算条件概率
    double value1=(viewdata-average1)/sigma1;
    double value2=(viewdata-average2)/sigma2;
```

```
    double x_omega1=normalCFD(value1),x_omega2=normalCFD(
        value2);
    //根据贝叶斯概率分布定理计算后验概率
    double omega1_x=(x_omega1*0.9)/(x_omega1*0.9+x_omega2
        *0.1);
    double omega2_x=(x_omega2*0.1)/(x_omega1*0.9+x_omega2
        *0.1);
    double alpha1_x=omega2_x*1;
    double alpha2_x=omega1_x*6;
    if (alpha1_x>alpha2_x) {
        cout<<"应该选择决策2，即将样本归入非正常类"<<endl
            ;
    }
    else{
        cout<<"应该选择决策1，即将样本归入正常类"<<endl;
    }
    return 0;
}
```

六、实验分析

本次试验的主要内容是贝叶斯分类器。贝叶斯概率公式是概率统计中重要的结论之一，它揭示了两个事件之间的概率存在一定的关系，也提供了计算条件概率的方法。如果运用到实际中的话，那么最关键的概念就是先验概率的概念。有了先验概率，我们对样本可能的区间有了大致的估计，也有了一定的参考，这就体现了一个学习的过程：通过学习，可以得到先验概率，再带入到贝叶斯定理之中，得到后验概率，计算风险，风险最小的，就是我们的最佳决策。

另一个重要的概念是自相似先验概率。根据中心极限定理，大多数的时间都满足高斯分布（也就是正态分布）。我们可以通过正太分布的参数估计来模拟出一个类中的数据所遵循的正态分布。具体的参数估计公式为：

$$\mu = \frac{1}{n}\sum_{i=1}^{n} x_i$$

$$\sigma = \frac{1}{n}\sum_{i=1}^{n} x_i^2 - (\frac{1}{n}\sum_{i=1}^{n} x_i)^2$$

通过简单的变换就可以把一般的正太分布变换为标准正态分布 N(0,1):

$$\widehat{x} = \frac{x-\mu}{\sigma}$$

这样，我们就可以根据样本自身的数据求得条件概率。

经测试，一个测试结果为：

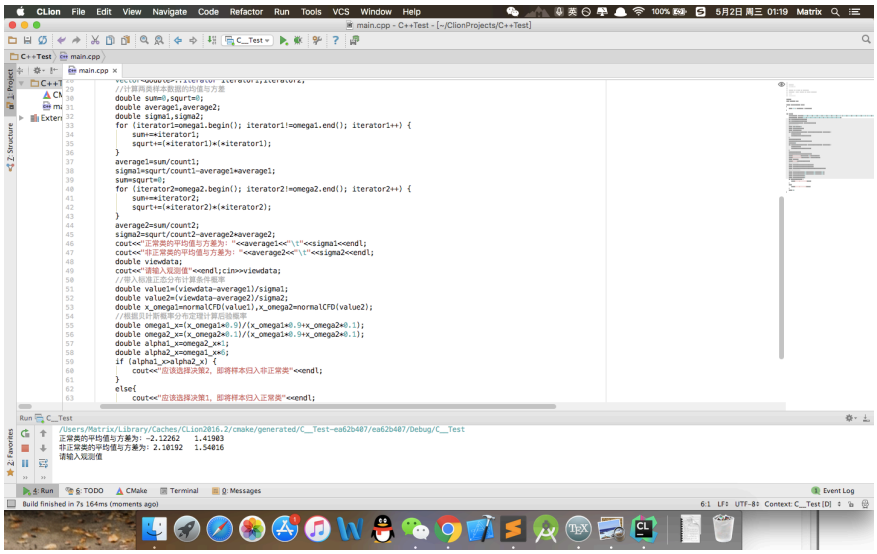

Figure 1: example caption

如果想要增加样本数量，只需要在开头处修改 vector 容器中的样本数据和数量即可：

```
vector<double>vector1,vector2;
for(int i=0;i<100;i++){
        vector1.push_back(-random()%2);//样本一中为正常样本数据
        vector2.push_back(random()%2);//样本二中为非正常样本数据
}
```

使用了 MATLAB 来绘制先验概率函数的图像：

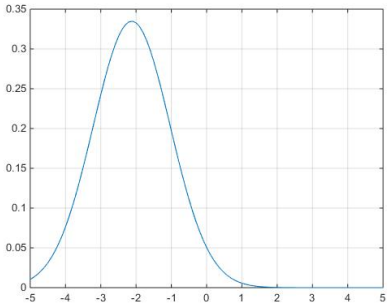

Figure 2: prior probability ω_1

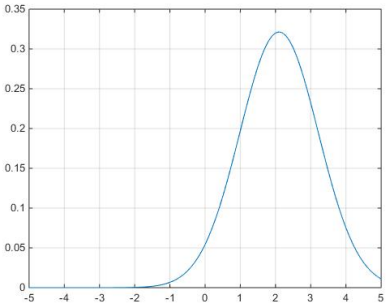

Figure 3: prior probability ω_2

再绘制后验概率函数的图像：

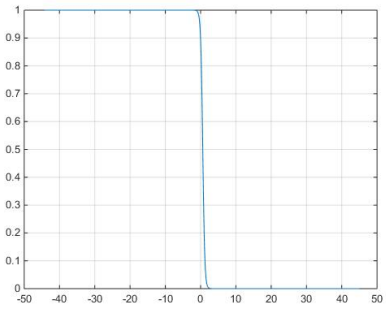

Figure 4: posterior probability ω_1

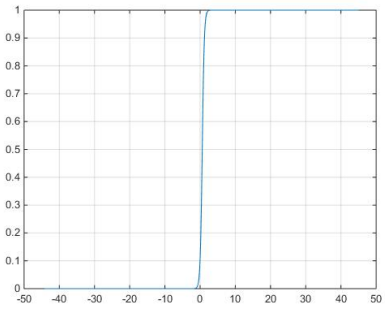

Figure 5: posterior probability ω_2

可以很明显地看出，先验概率密度函数和后验概率密度函数有明显的不同。先验概率密度函数很好的遵从了高斯分布，在 μ 附近的函数分布达到极大值。这个对于 ω_1 和 ω_2 均适用。计算后验概率密度函数之后绘图，显然可以看出两个概率密度函数分为两极，大概以 x=0 处为分界线。这说明贝叶斯分类器起到了很好的分类效果。