

Господарикова Варвара Сергеевна, группа 2-1

Лабораторная работа №6

Распознавание образов на основе непараметрических алгоритмов оценивания плотности распределения случайной величины

Вариант №1

Цель работы

Исследовать алгоритмы распознавания образов на основе оценивания плотности распределения случайных величин и случайных векторов при использовании методов Парзена и k ближайших соседей.

Задание

Реализуйте алгоритм распознавания образов, применив оценивание по методу Парзена. Вычислите вероятности ошибок распознавания на основе метода скользящего контроля.

Код программы

```
clear all; close all;
```

```
n=2; M=3;
```

```
K = 1000;
```

```
m = [8 -4; -6 5; 3 -1]';
```

```
pw = [0.33, 0.33, 0.33];
```

```
np=sum(pw); pw=pw/np;
```

```
C = [3 1; 1 3];
```

```
C_ = C^-1;
```

```
D = C(1,1);
```

```
N = K * M;
```

```
NN = zeros(M, 1);
```

```
for k = 1 : M - 1
```

```
    NN(k) = uint16(N * pw(k));
```

```
end;
```

```
NN(M) = N - sum(NN);
```

```
label = {'bo', 'r+', 'k*', 'gx'};
```

```
IMS = [];
```

```
for i=1:M
```

```
    ims = repmat(m(:,i), [1, NN(i)]) + randncor(n,NN(i),C);
```

```

    IMS = [IMS, ims];
end;
Ks = fix(K * pw);
Ks(end) = K - sum(Ks(1 : end - 1));
for i=1:M, %???? ?? ???????
    XN{i} = repmat(m(:,i), [1, Ks(i)]) + randncor(n,Ks(i),C(i));
end;

G=zeros(M,n+1); PIJ=zeros(M); l0_=zeros(M);
for i = 1 : M,
    G(i,1:n)=(C_*m(:,i))';    G(i,n+1)=-0.5*m(:,i)'*C_*m(:,i);
    for j=i+1:M,
        l0_(i,j)=log(pw(j)/pw(i));
        h=0.5*(m(:,i)-m(:,j))*C_*(m(:,i)-m(:,j)); sD=sqrt(2*h);
        PIJ(i,j)=normcdf(l0_(i,j),h,sD); PIJ(j,i)=1-normcdf(l0_(i,j),-h,sD);
    end;
    PIJ(i,i)=1-sum(PIJ(i,:));
end;

r=0.5; kl_kernel=11;
Pc1=zeros(M);
p1_=zeros(M,1);
for i=1:M,
    N=Ks(i);
    XNi=XN{i}; XNi_=zeros(n,N-1);
    indi=[1:i-1,i+1:M];
    for j=1:N,
        x=XNi(:,j); indj=[1:j-1,j+1:N];
        XNi_(:,1:j-1)=XNi(:,1:j-1); XNi_(:,j:end)=XNi(:,j+1:end);
        h_N=N^(-r/n);
        p1_(i)=vkernel(x,XNi_,h_N,11);
        for t=1:M-1,
            ij=indi(t);
            h_N=Ks(ij)^(-r/n);
            p1_(ij)=vkernel(x,XN{ij},h_N,11);
        end;
        [ui1,iai1]=max(p1_);
        Pc1(i,iai1)=Pc1(i,iai1)+1;
    end;
    Pc1(i,:)=Pc1(i,:)/N;
end;

x=ones(n+1,1); Pc_=zeros(M);
for k=1:K,
    for i=1:M,
        [x_px]=randncor(n,1,C);
        x(1:n,1)=m(:,i)+x_;
    end;
end;

```

```

u=G*x+log(pw');
[ui,iai]=max(u);
Pc_(i,iai)=Pc_(i,iai)+1;
end;
end;
Pc_=Pc_/K;
disp('Теоретическая матрица ошибок');disp(PIJ);
disp('Экспериментальная матрица ошибок');disp(Pc_);
disp(' Матрица ошибок скользящего контроля');disp(Pc1);

```

Результат выполнения задания

В ходе выполнения работы было создано решения для оценивания по методу Парзена. Ошибка вычислялась по методу скользящего контроля.

Для выполнения задания были использованы следующие центры классов:

$$M1 = \{8, -4\};$$

$$M2 = \{-6, 5\};$$

$$M3 = \{3, -1\};$$

Вероятность попадания случайной величины в каждый из классов была принята одинаковой и равной 0.33. А матрица ковариаций представляла собой:

$$C = \{ (3,1) , (1,3) \};$$

Далее были произведены вычисления и получены следующие матрицы:

1. Теоретическая матрица ошибок

0.9789	0.0000	0.0211
0.0000	0.9999	0.0001
0.0211	0.0001	0.9788

2. Экспериментальная матрица ошибок

0.9800	0	0.0200
0	1.0000	0
0.0210	0	0.9790

3. Матрица ошибок скользящего контроля

0.9790	0	0.0210
0	1.0000	0
0.0120	0	0.9880

Вывод

В результате проделанных исследований мы можем сделать вывод о том, что метод скользящих соседей довольно быстр для исполнения кода, а также достаточно точен, ведь, как видно из матриц приведенных выше, значения его оценки ошибок отличаются от экспериментальных в среднем на тысячные доли.