әл-Фараби атындағы қазақ ұлттық университеті



Зертханалық жұмыс № 1

**Пән: Теория распознавания образов**

**Тақырыбы: Порядок выполнения работы 01**

Тексерген: Гусманова Ф. Р.

Орындаған: Якуфуцзян Азати  
...

Тобы: ВТиПО

Исходные данные: число классов объектов – 2, закон распре-

деления признаков объектов – нормальный. Параметры распределе-

ния (математическое ожидание m и среднеквадратическое отклоне-

ние σ):

% given parameters:

m1=-3;

sig1=1;

m2=-1;

sig2=0.5;

p1=0.9;

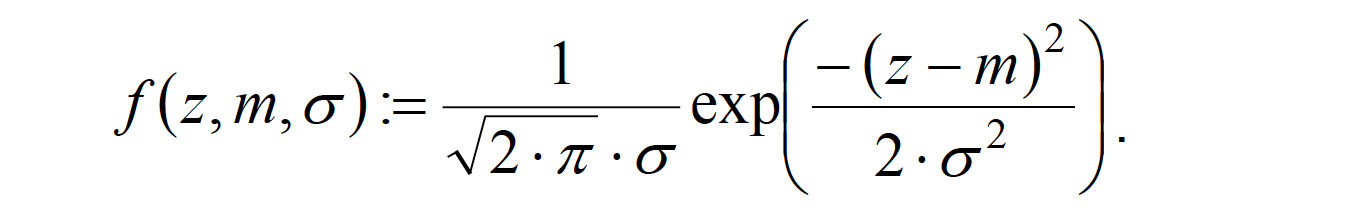
p2=0.1;

N=200;









Сформируем массив N точек (N = 200) по оси 0x, располагающихся с

равным шагом в диапазоне [xmin, xmax]. Верхнюю xmax и нижнюю xmin

границы диапазона определим по правилу «трех сигм», согласно ко-

торому случайная величина x, распределенная по нормальному за-

кону, находится в интервале значений m ± 3σ с вероятностью более

99.7%

.

Норма́льное распределе́ние, также называемое распределением Гаусса или Гаусса — Лапласа — распределение вероятностей, которое в одномерном случае задаётся функцией плотности вероятности, совпадающей с функцией Гаусса:



Реализация функции нормального распределения в MATLAB：

PDF.m

%one matlab functional file for pattern recognition class

%Student: Yaakov Azat Email:yaakovazat@gmail.com ,Teacher: Гусманова Ф. Р.% PDF ÔºöProbability Density FunctionÔºâfor ND (normal distribution)

function[pdf]=PDF(x,miu,sigma)

pdf=(1/(sqrt(2\*pi)\*sigma))\*exp((-1\*(x-miu).^2)/2\*(sigma^2));

end

Считаем, что случайные значения параметра x будут лежать в

диапазоне [x1min, x1max], если наблюдается класс 1 (x ∈ a1), и в

диапазоне [x2min, x2max], если наблюдается класс 2 (x ∈ a2), где



x1min = m1 − 3·σ1, x1max = m1 + 3·σ1,

x2min = m2 − 3·σ2, x2max = m2 + 3·σ2.

Определим нижнюю и верхнюю границы значений параметра x:

xmin := min (x1min, x2min);

xmax := max (x1max, x2max).

Реализация в Matlab: threesigma.m

%one matlab functional file for pattern recognition class

%Student: Yaakov Azat Email:yaakovazat@gmail.com ,Teacher: Гусманова Ф. Р.% function[xmin,xmax]=threesigma(miu1,sigma1,miu2,sigma2)

xmin=min(miu1-3\*sigma1,miu2-3\*sigma2);

xmax=max(miu1+3\*sigma1,miu2+3\*sigma2);

end

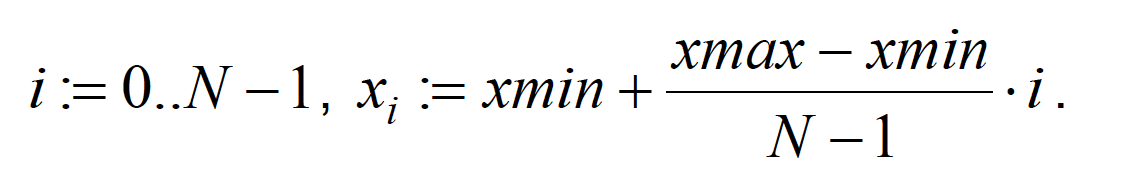
% calculate the coordinates of x :

[xmin,xmax]=threesigma(m1,sig1,m2,sig2);

Для заданных данных xmin = -6, xmax = 1.1633

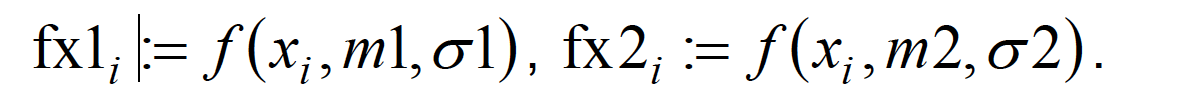
Разделим интервал [xmin, xmax] на (N − 1) часть и определим

координаты точек разделения:



Сформируем массивы значений условных по классу плотностей

вероятности (fx1i) и ( fx2i), соответствующие точкам xi:



Реализация в Matlab:

% the arrange qujian for x:

for i=1:N

x(i)=xmin+(xmax-xmin)\*(i-1)/(N-1);

end

for i=1:N

x(i);

end

% calculate the Probablity dentisy function of class 1 and class 2 ,as fx1i

% and fx2i

fx1i=PDF(x,m1,sig1);

fx2i=PDF(x,m2,sig2);

ymin=min(min(fx1i),min(fx2i));

ymax=max(max(fx1i),max(fx2i));

% draw pictures of fx1i and fx2i :

figure

subplot(2,2,1)

plot(x,fx1i,x,fx2i);

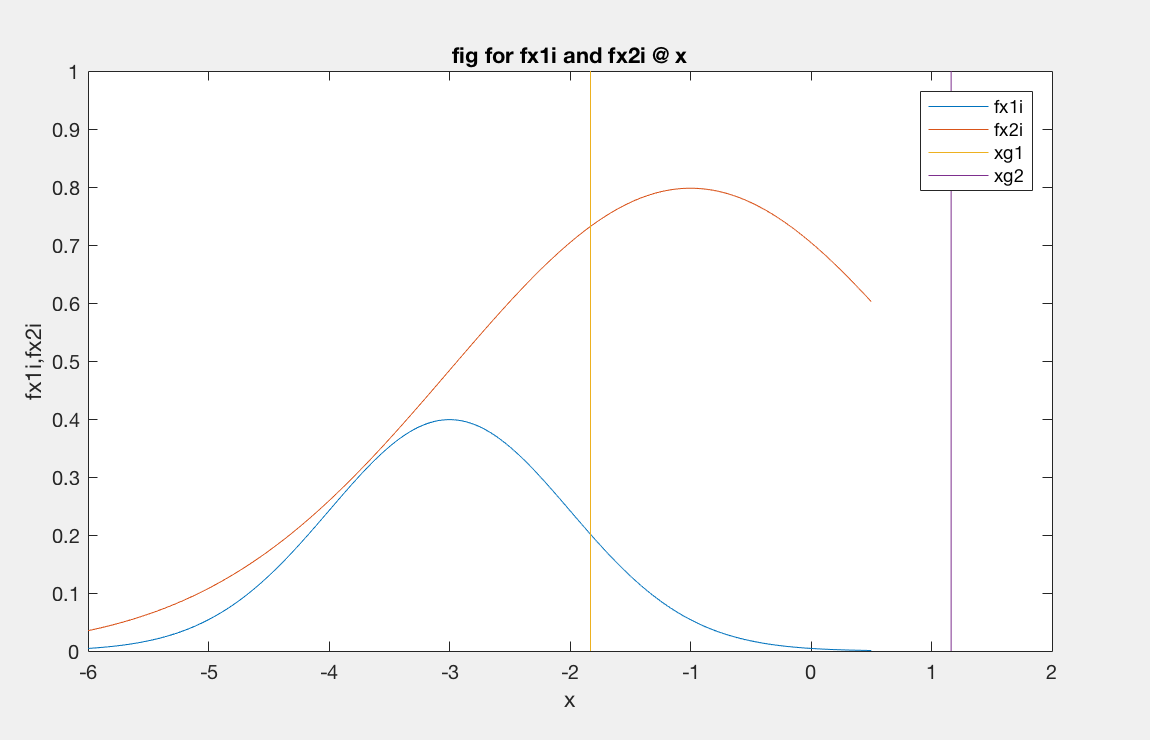
title('Subplot 1')

hold on

title('fig for fx1i and fx2i @ x')

xlabel('x')

ylabel('fx1i,fx2i')



Для определения порогов принятия решения по критерию мак-

симального правдоподобия (1.5) нужно решить уравнение

% –ø–æ—Ä–æ–≥–∏ –ø—Ä–∏–Ω—è—Ç–∏—è —Ä–µ—à–µ–Ω–∏—è:

d1=(sig1)^2;

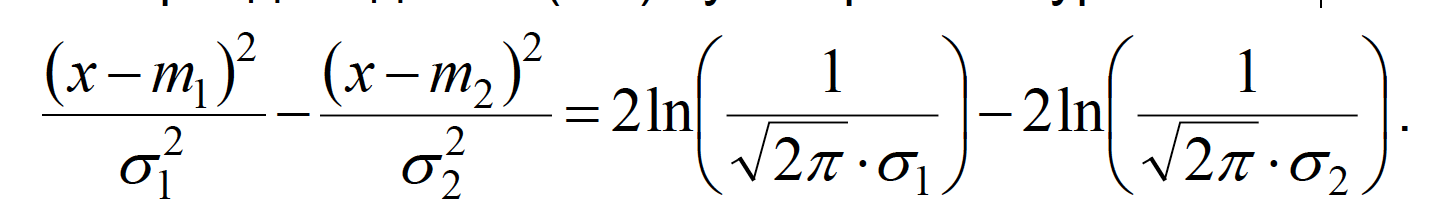
d2=(sig2)^2;

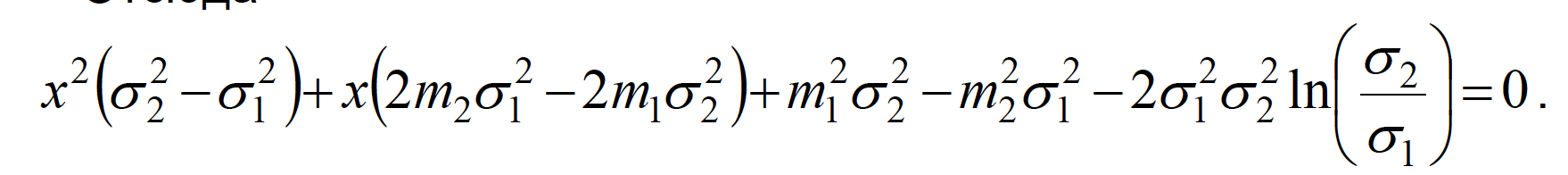
a=d2-d1;

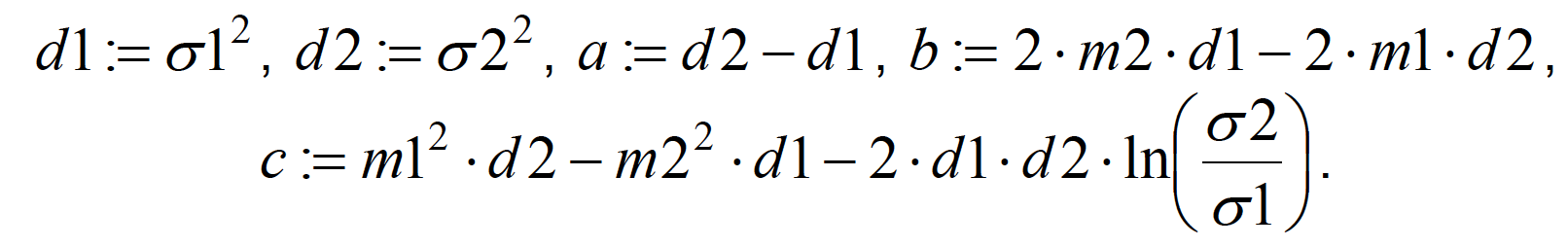
b=2\*m2\*d1-2\*m1\*d2;

c=(m1)^2\*d2-(m2)^2\*d1-2\*d1\*d2\*log(sig2/sig1);

[xg1,xg2]=XG(a,b,c);







Реализация в Matlab:

%–ø–æ—Ä–æ–≥–∏ –ø—Ä–∏–Ω—è—Ç–∏—è —Ä–µ—à–µ–Ω–∏—è

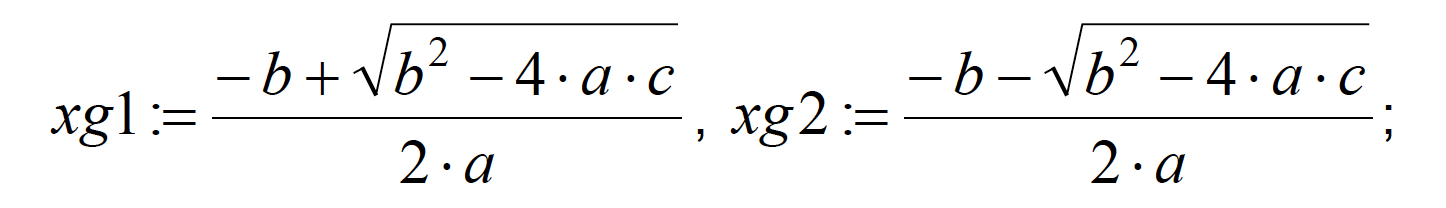
function[xg1,xg2]=XG(a,b,c)

xg2=(-b-sqrt(b^2-4\*a\*c))/(2\*a);

xg1=(-b+sqrt(b^2-4\*a\*c))/(2\*a);

end

Вычислим пороги принятия решения xg1 и xg2, xg1 < xg2:



получим xg1 = -1.8300 и xg2 = 1.1633

Изобразим на графике полученные границы раздела между

классами xg1 и xg2. Если какой-либо из порогов лежит в областях

маловероятных значений параметра x для всего множества классов

Α = {a1, a2} (в данном случае xg1∉[xg1, xg2]), то следует переопреде-

лить нижнюю и (или) верхнюю границы x:

%deside that xg1 and xg2 are or not inside the [xmin,xmax]

ok=deside(xg1,xg2,xmin,xmax);

% deside that the descision with or without in the range

function[ok]=deside(xg1,xg2,xmin,xmax)

if xg1>=xmin

ok1=1;

else ok1=o;

end

if xg2<=xmax

ok2=1;

else ok2=0;

end

ok=ok1+ok2;

end

if ok<2

disp('Have to adjust ~!')

else

disp('Normal Case ~!')

end

plot([xg1 xg1],[0 1])

plot([xg2 xg2],[0 1])

hold on

legend('fx1i','fx2i','xg1','xg2')

>>>Have to adjust ~!

xmin := if ( xmin > xg1 , xg1 , xmin);

xmax := if (xmax < xg2 , xg2 , xmax).

% adjust xmin and xmax:

[xmin,xmax]=adjust(xmin,xmax,xg1,xg2);

%cause have to adjust case then we have to calculate again :

% 2nd time calculation :

% the arrange qujian for x:

for i=1:N

x\_a(i)=xmin+(xmax-xmin)\*(i-1)/(N-1);

end

for i=1:N

x\_a(i);

end

% 2nd time calculation :

% calculate the Probablity dentisy function of class 1 and class 2 ,as fx1i

% and fx2i

fx1i\_a=PDF(x\_a,m1,sig1);

fx2i\_a=PDF(x\_a,m2,sig2);

%q1i=(PDF(x\_a,m1,sig1))/(PDF(x\_a,m1,sig1)+PDF(x\_a,m2,sig2));

%q2i=(PDF(x\_a,m2,sig2))/(PDF(x\_a,m1,sig1)+PDF(x\_a,m2,sig2));

Соответственно пересчитываются значения массивов xi, fx1i, fx2i:

% 2nd time calculation :

% draw pictures of fx1i and fx2i :

subplot(2,2,2)

plot(x\_a,fx1i\_a,x\_a,fx2i\_a);

hold on

title('fig for fx1i\_a and fx2i\_a @ x')

xlabel('x')

ylabel('fx1i\_a,fx2i\_a')

% 2nd time calculation :

% –ø–æ—Ä–æ–≥–∏ –ø—Ä–∏–Ω—è—Ç–∏—è —Ä–µ—à–µ–Ω–∏—è:

d1=(sig1)^2;

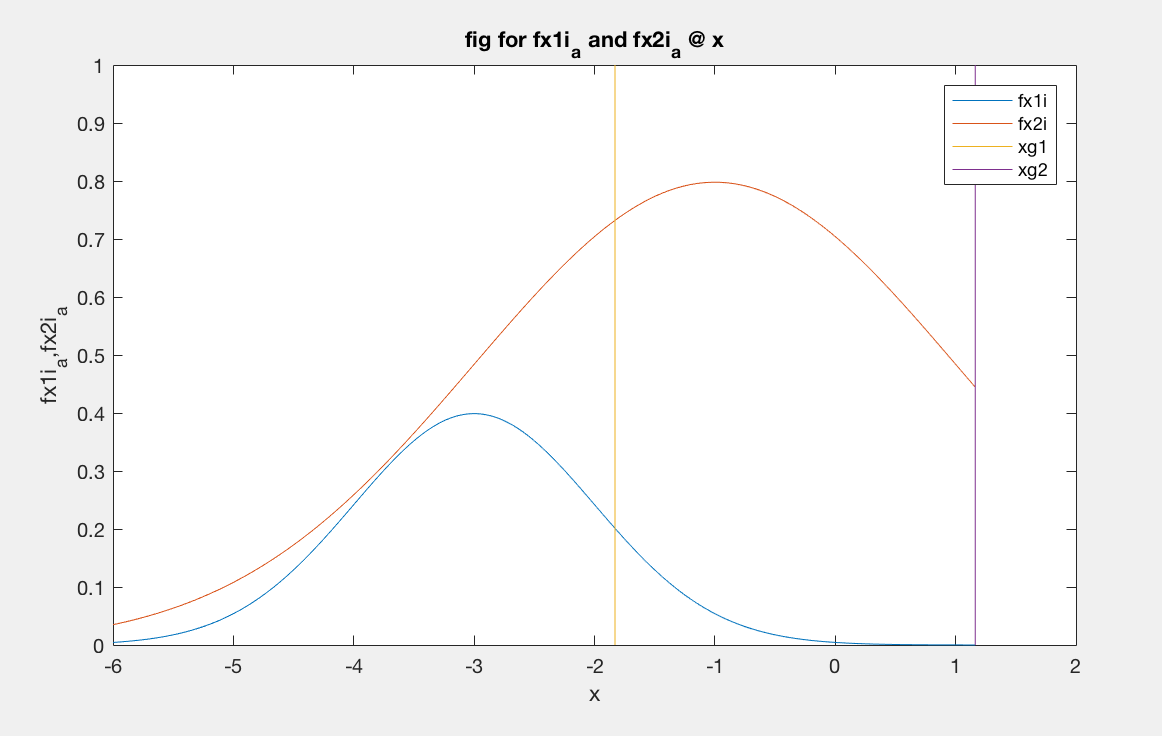
d2=(sig2)^2;

a=d2-d1;

b=2\*m2\*d1-2\*m1\*d2;

c=(m1)^2\*d2-(m2)^2\*d1-2\*d1\*d2\*log(sig2/sig1);

[xg1\_a,xg2\_a]=XG(a,b,c);



fgi := if ( xg1 < xi < xg2 , 0.5 , 0 )

%adjust xmin and xmax

function[xmin,xmax]=adjust(xmin,xmax,gx1,gx2)

xmin=min(min(xmin,xmax),min(gx1,gx2));

xmax=max(max(xmin,xmax),max(gx1,gx2));

end

% juesehanshu

function[fg]=juese(x,xg1,xg2)

% ÂÖàÁúãÂ∑ÆÂÄº 1

[m,n]=size(x);

fg=zeros(1,200);

XG1=fg+xg1;

XG2=fg+xg2;

M1=x-XG1;

M2=XG2-x;

a1=find(M1<0);

M1(a1)=0;

b1=find(M1>0);

M1(b1)=0.5;

a2=find(M2<0);

M1(a2)=0;

b2=find(M2>0);

M2(b2)=1;

MM=M2-M1;

c=find(MM==0.5);

MM(c)=0.5;

c1=find(MM<0.5);

MM(c1)=0;

c2=find(MM>0.5);

MM(c2)=0;

fg=MM;

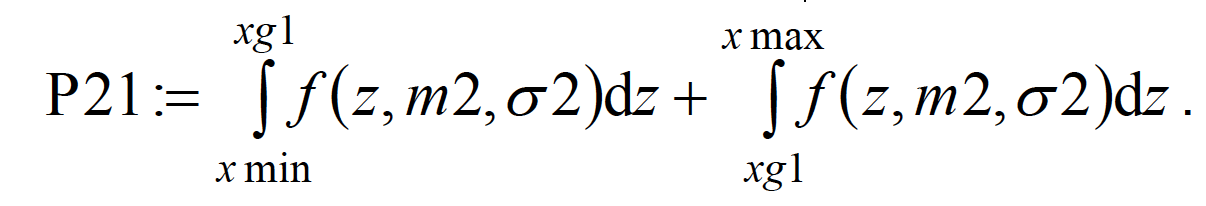
end

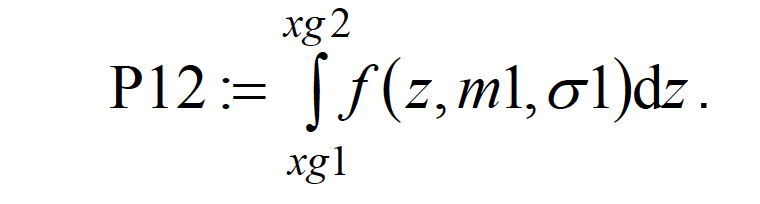
Для оценки эффективности решающего правила (1.5) рассчи-

таем теоретические величины вероятностей ошибок распознавания.

Вероятность отнести наблюдаемый признак к классу a 1 , когда он

в действительности принадлежит классу a2:





ok\_a=deside(xg1,xg2,xmin,xmax);

if ok\_a<2

disp('Have to adjust ~!')

else

disp('Normal Case ~!')

end

plot([xg1 xg1],[0 1])

plot([xg2 xg2],[0 1])

hold on

legend('fx1i','fx2i','xg1','xg2')

% finished normal

% juece hanshu

fg=juese(x\_a,xg1\_a,xg2\_a);

a\_a=find(fg==0);

b\_b=find(fg==0.5);

fg(a\_a)=0.5;

fg(b\_b)=0;

% desided already

%division

syms z pdf1 pdf2;

pdf21\_1\_1=int((1/(sqrt(2\*pi)\*sig2))\*exp((-1\*(z-m2).^2)/2\*(sig2^2)),z,xmin,xg1\_a);

pdf21\_1=vpa(pdf21\_1\_1);

pdf21\_2\_1=int((1/(sqrt(2\*pi)\*sig2))\*exp((-1\*(z-m2).^2)/2\*(sig2^2)),z,xg1\_a,xmax);

pdf21\_2=vpa(pdf21\_2\_1);

P21=pdf21\_1+pdf21\_2;

pdf1=int((1/(sqrt(2\*pi)\*sig1))\*exp((-1\*(z-m1).^2)/2\*(sig1^2)),z,xg1\_a,xg2\_a);

P12=vpa(pdf1);

>> P12

P12 =

0.12097571058004356418129488085748

>> P21

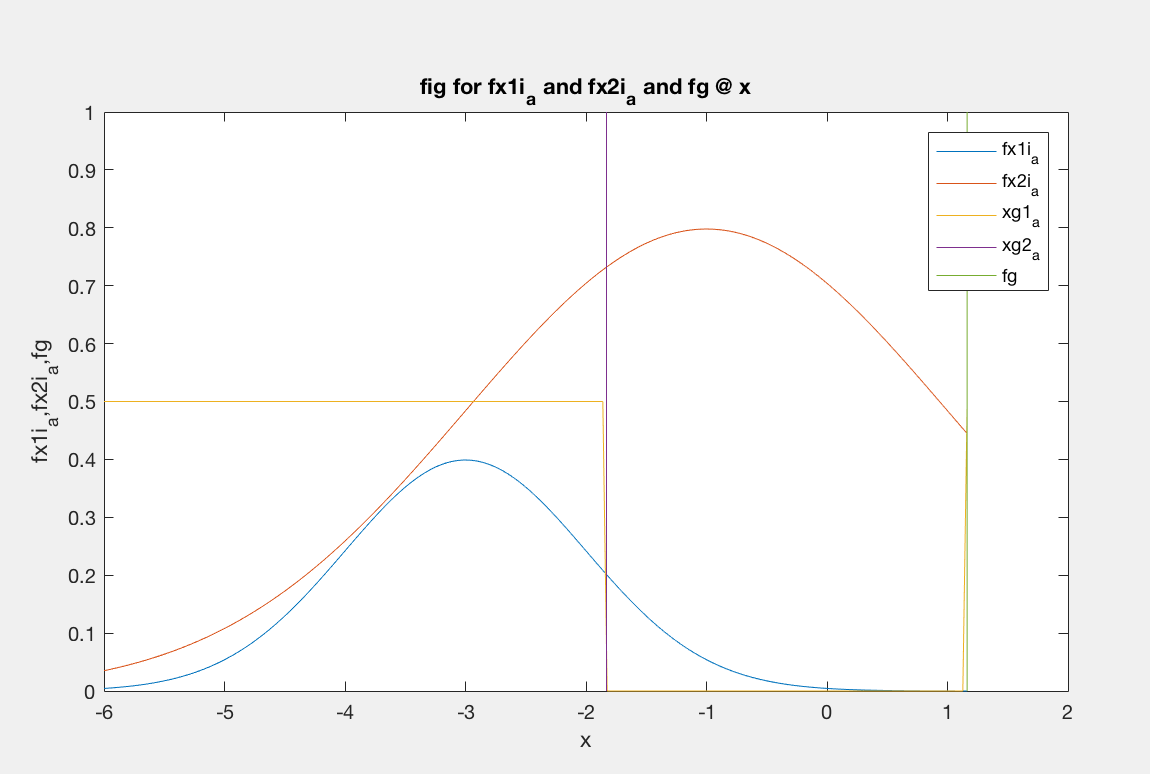
P21 =

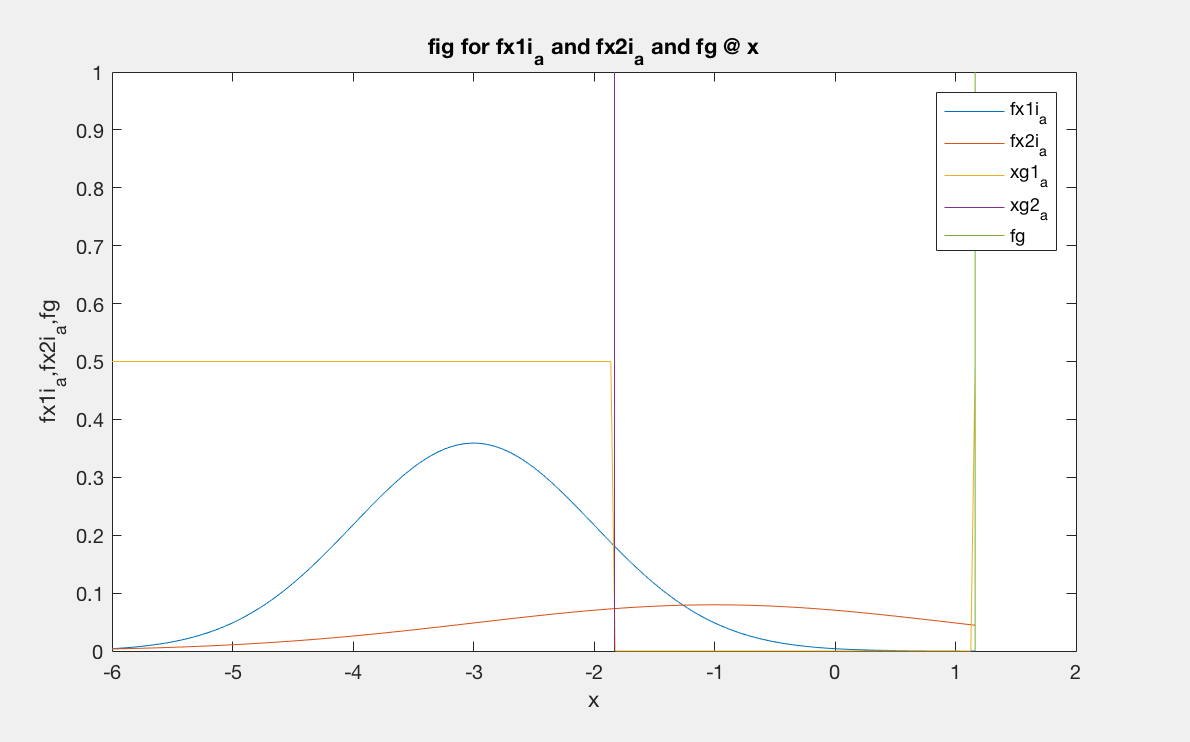
3.4163399197223171614205347754465

>> P

P =

-0.76865781515118036280091482815197



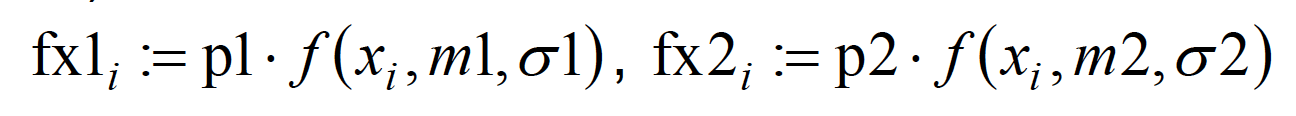


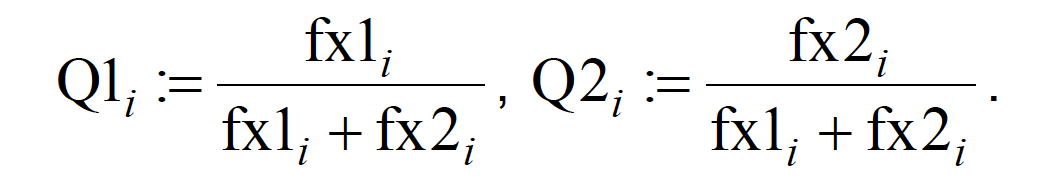
Для построения решающего правила по критерию максималь-

ной апостериорной вероятности (1.3) зададим априорные вероятно-

сти p1 и p2 появления классов a1 и a2, p1 + p2 = 1:

P1=0.9;p2=0.1;





%p1+p2=1

% Q

for i=1:N

q1(i)=fx1i\_a(i)/(fx1i\_a(i)+fx2i\_a(i));

q2(i)=fx2i\_a(i)/(fx1i\_a(i)+fx2i\_a(i));

end

% plot Q

subplot(2,2,4)

plot(x\_a,q1,x\_a,q2);

hold on

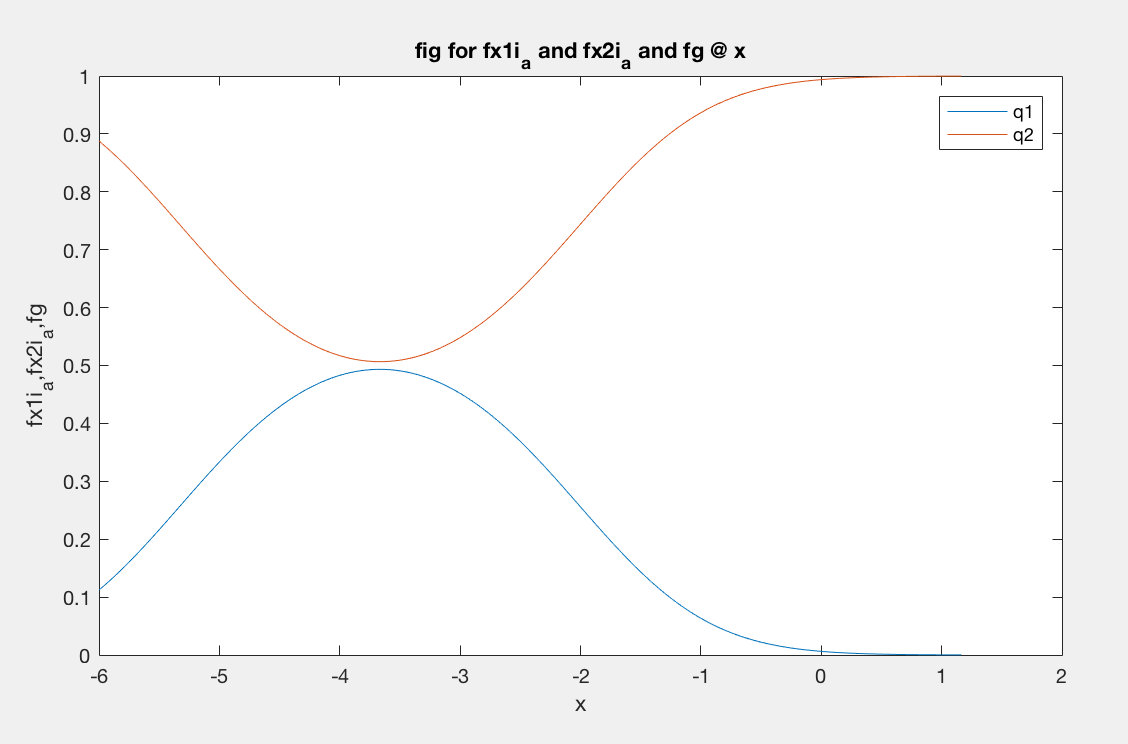
title('fig for fx1i\_a and fx2i\_a and fg @ x')

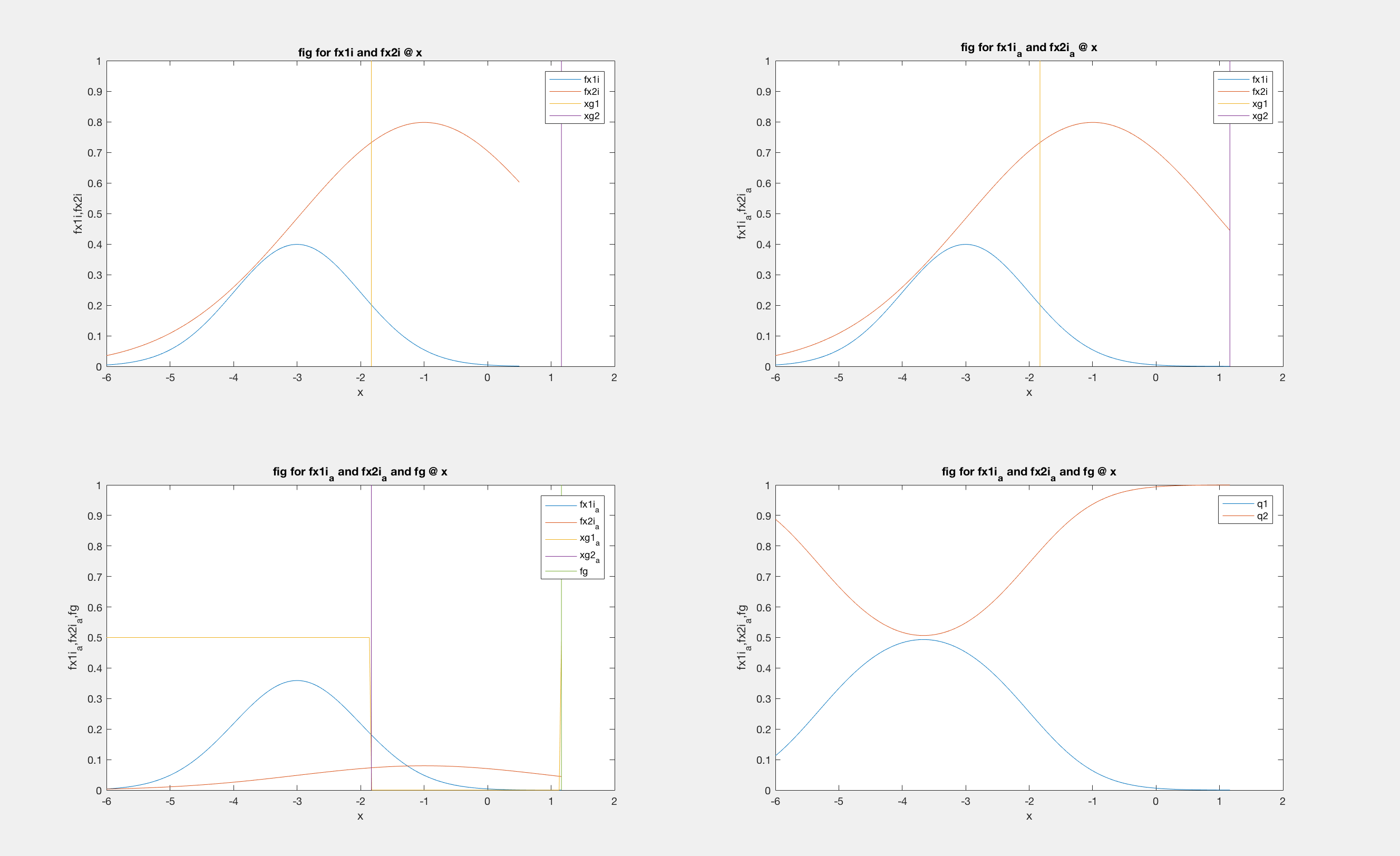
xlabel('x')

ylabel('fx1i\_a,fx2i\_a,fg')

legend('q1','q2')

disp('successful!')





Matlab Code:

#firstwork.m

%one matlab functional file for pattern recognition class

%Student: Yaakov Azat Email:yaakovazat@gmail.com ,Teacher: Гусманова Ф. Р.% clc;

clear;

% given parameters:

m1=-3;

sig1=1;

m2=-1;

sig2=0.5;

p1=0.9;

p2=0.1;

N=200;

% calculate the coordinates of x :

[xmin,xmax]=threesigma(m1,sig1,m2,sig2);

% the arrange qujian for x:

for i=1:N

x(i)=xmin+(xmax-xmin)\*(i-1)/(N-1);

end

for i=1:N

x(i);

end

% calculate the Probablity dentisy function of class 1 and class 2 ,as fx1i

% and fx2i

fx1i=PDF(x,m1,sig1);

fx2i=PDF(x,m2,sig2);

ymin=min(min(fx1i),min(fx2i));

ymax=max(max(fx1i),max(fx2i));

% draw pictures of fx1i and fx2i :

figure

subplot(2,2,1)

plot(x,fx1i,x,fx2i);

title('Subplot 1')

hold on

title('fig for fx1i and fx2i @ x')

xlabel('x')

ylabel('fx1i,fx2i')

% –ø–æ—Ä–æ–≥–∏ –ø—Ä–∏–Ω—è—Ç–∏—è —Ä–µ—à–µ–Ω–∏—è:

d1=(sig1)^2;

d2=(sig2)^2;

a=d2-d1;

b=2\*m2\*d1-2\*m1\*d2;

c=(m1)^2\*d2-(m2)^2\*d1-2\*d1\*d2\*log(sig2/sig1);

[xg1,xg2]=XG(a,b,c);

%deside that xg1 and xg2 are or not inside the [xmin,xmax]

ok=deside(xg1,xg2,xmin,xmax);

if ok<2

disp('Have to adjust ~!')

else

disp('Normal Case ~!')

end

plot([xg1 xg1],[0 1])

plot([xg2 xg2],[0 1])

hold on

legend('fx1i','fx2i','xg1','xg2')

% adjust xmin and xmax:

[xmin,xmax]=adjust(xmin,xmax,xg1,xg2);

%cause have to adjust case then we have to calculate again :

% 2nd time calculation :

% the arrange qujian for x:

for i=1:N

x\_a(i)=xmin+(xmax-xmin)\*(i-1)/(N-1);

end

for i=1:N

x\_a(i);

end

% 2nd time calculation :

% calculate the Probablity dentisy function of class 1 and class 2 ,as fx1i

% and fx2i

fx1i\_a=PDF(x\_a,m1,sig1);

fx2i\_a=PDF(x\_a,m2,sig2);

%q1i=(PDF(x\_a,m1,sig1))/(PDF(x\_a,m1,sig1)+PDF(x\_a,m2,sig2));

%q2i=(PDF(x\_a,m2,sig2))/(PDF(x\_a,m1,sig1)+PDF(x\_a,m2,sig2));

% 2nd time calculation :

% draw pictures of fx1i and fx2i :

subplot(2,2,2)

plot(x\_a,fx1i\_a,x\_a,fx2i\_a);

hold on

title('fig for fx1i\_a and fx2i\_a @ x')

xlabel('x')

ylabel('fx1i\_a,fx2i\_a')

% 2nd time calculation :

% –ø–æ—Ä–æ–≥–∏ –ø—Ä–∏–Ω—è—Ç–∏—è —Ä–µ—à–µ–Ω–∏—è:

d1=(sig1)^2;

d2=(sig2)^2;

a=d2-d1;

b=2\*m2\*d1-2\*m1\*d2;

c=(m1)^2\*d2-(m2)^2\*d1-2\*d1\*d2\*log(sig2/sig1);

[xg1\_a,xg2\_a]=XG(a,b,c);

%deside that xg1 and xg2 are or not inside the [xmin,xmax]

ok\_a=deside(xg1,xg2,xmin,xmax);

if ok\_a<2

disp('Have to adjust ~!')

else

disp('Normal Case ~!')

end

plot([xg1 xg1],[0 1])

plot([xg2 xg2],[0 1])

hold on

legend('fx1i','fx2i','xg1','xg2')

% finished normal

% juece hanshu

fg=juese(x\_a,xg1\_a,xg2\_a);

a\_a=find(fg==0);

b\_b=find(fg==0.5);

fg(a\_a)=0.5;

fg(b\_b)=0;

% desided already

%division

syms z pdf1 pdf2;

pdf21\_1\_1=int((1/(sqrt(2\*pi)\*sig2))\*exp((-1\*(z-m2).^2)/2\*(sig2^2)),z,xmin,xg1\_a);

pdf21\_1=vpa(pdf21\_1\_1);

pdf21\_2\_1=int((1/(sqrt(2\*pi)\*sig2))\*exp((-1\*(z-m2).^2)/2\*(sig2^2)),z,xg1\_a,xmax);

pdf21\_2=vpa(pdf21\_2\_1);

P21=pdf21\_1+pdf21\_2;

pdf1=int((1/(sqrt(2\*pi)\*sig1))\*exp((-1\*(z-m1).^2)/2\*(sig1^2)),z,xg1\_a,xg2\_a);

P12=vpa(pdf1);

% calculating P

P=1-0.5\*(P21+P12)

%plot again

subplot(2,2,3)

plot(x\_a,p1\*fx1i\_a,x\_a,p2\*fx2i\_a,x\_a,fg);

hold on

title('fig for fx1i\_a and fx2i\_a and fg @ x')

xlabel('x')

ylabel('fx1i\_a,fx2i\_a,fg')

plot([xg1 xg1],[0 1])

plot([xg2 xg2],[0 1])

hold on

legend('fx1i\_a','fx2i\_a','xg1\_a','xg2\_a','fg')

%p1+p2=1

% Q

for i=1:N

q1(i)=fx1i\_a(i)/(fx1i\_a(i)+fx2i\_a(i));

q2(i)=fx2i\_a(i)/(fx1i\_a(i)+fx2i\_a(i));

end

% plot Q

subplot(2,2,4)

plot(x\_a,q1,x\_a,q2);

hold on

title('fig for fx1i\_a and fx2i\_a and fg @ x')

xlabel('x')

ylabel('fx1i\_a,fx2i\_a,fg')

legend('q1','q2')

disp('successful!')

#juese.m

% juesehanshu

function[fg]=juese(x,xg1,xg2)

% ÂÖàÁúãÂ∑ÆÂÄº 1

[m,n]=size(x);

fg=zeros(1,200);

XG1=fg+xg1;

XG2=fg+xg2;

M1=x-XG1;

M2=XG2-x;

a1=find(M1<0);

M1(a1)=0;

b1=find(M1>0);

M1(b1)=0.5;

a2=find(M2<0);

M1(a2)=0;

b2=find(M2>0);

M2(b2)=1;

MM=M2-M1;

c=find(MM==0.5);

MM(c)=0.5;

c1=find(MM<0.5);

MM(c1)=0;

c2=find(MM>0.5);

MM(c2)=0;

fg=MM;

end

#adjust.m

%adjust xmin and xmax

function[xmin,xmax]=adjust(xmin,xmax,gx1,gx2)

xmin=min(min(xmin,xmax),min(gx1,gx2));

xmax=max(max(xmin,xmax),max(gx1,gx2));

end

#deside.m

% deside that the descision with or without in the range

function[ok]=deside(xg1,xg2,xmin,xmax)

if xg1>=xmin

ok1=1;

else ok1=o;

end

if xg2<=xmax

ok2=1;

else ok2=0;

end

ok=ok1+ok2;

end

#XG.m

%one matlab functional file for pattern recognition class

%Student: Yaakov Azat Email:yaakovazat@gmail.com ,Teacher: Гусманова Ф. Р.% function[xmin,xmax]=threesigma(miu1,sigma1,miu2,sigma2)

xmin=min(miu1-3\*sigma1,miu2-3\*sigma2);

xmax=max(miu1+3\*sigma1,miu2+3\*sigma2);

end

#PDF.m

%one matlab functional file for pattern recognition class

%Student: Yaakov Azat Email:yaakovazat@gmail.com ,Teacher: Гусманова Ф. Р.% % PDF ÔºöProbability Density FunctionÔºâfor ND (normal distribution)

function[pdf]=PDF(x,miu,sigma)

pdf=(1/(sqrt(2\*pi)\*sigma))\*exp((-1\*(x-miu).^2)/2\*(sigma^2));

end

#fzmsig.m

function[fzms]=fzmsig (z,m,sig)

%–û–ø—Ä–µ–¥–µ–ª–∏–º –ø–æ–ª—å–∑–æ–≤–∞—Ç–µ–ª—å—Å–∫—É—é —Ñ—É–Ω–∫—Ü–∏—é ‚Äì –ø–ª–æ—Ç–Ω–æ—Å—Ç—å –Ω–æ—Ä–º–∞–ª—å-

%–Ω–æ–≥–æ —Ä–∞—Å–ø—Ä–µ–¥–µ–ª–µ–Ω–∏—è —Å –ø–∞—Ä–∞–º–µ—Ç—Ä–∞–º–∏ m, œÉ:

fzms=(1/(sqrt(2\*pi)\*sig))\*exp((-1\*(z-m).^2)/2\*(sig^2));

end

#getq.m

function [Q]=getq(f1,f2)

pdf=(1/(sqrt(2\*pi)\*sigma))\*exp((-1\*(x-miu).^2)/2\*(sigma^2));

fx1i=PDF(x,miu,sigma);

fx2i=PDF(x,miu,sigma);

end

%one matlab functional file for pattern recognition class

%Student: Yaakov Azat Email:yaakovazat@gmail.com ,Teacher: Гусманова Ф. Р.%