אוניברסיטת בן-גוריון בנגב
Ben-Gurion University of the Negev

אוניברסיטת בן – גוריון

הפקולטה להנדסה

המחלקה להנדסת מחשבים

GMM, EM, Kmeans

מגיש: דן בן עמי – 316333079

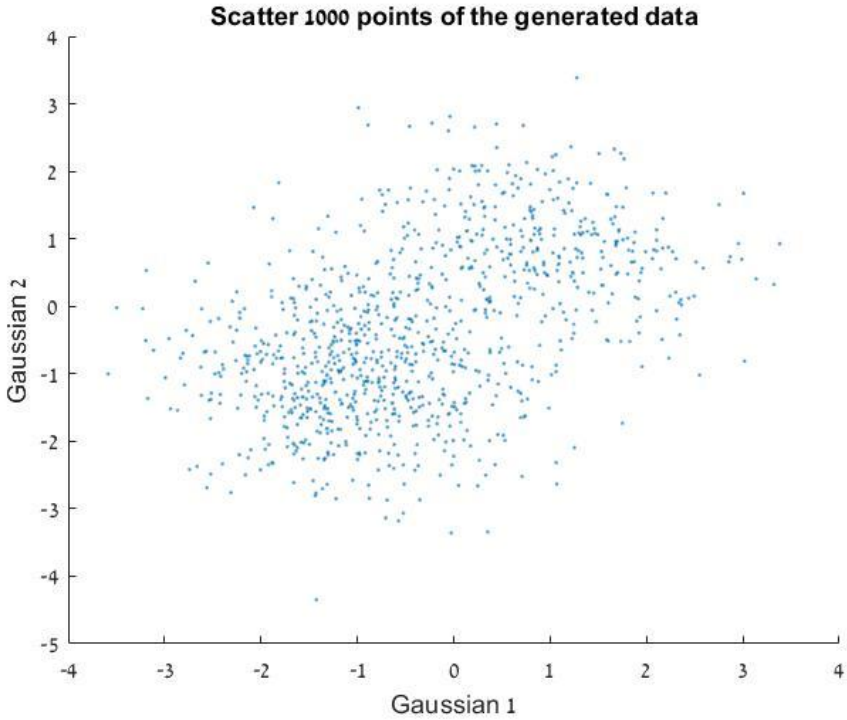
תאריך הגשה: 05.05.21

1. Data generation:

בשלב זה אצור GMM בעל שני גאוסיינים בעלי הפרמטרים הבאים:

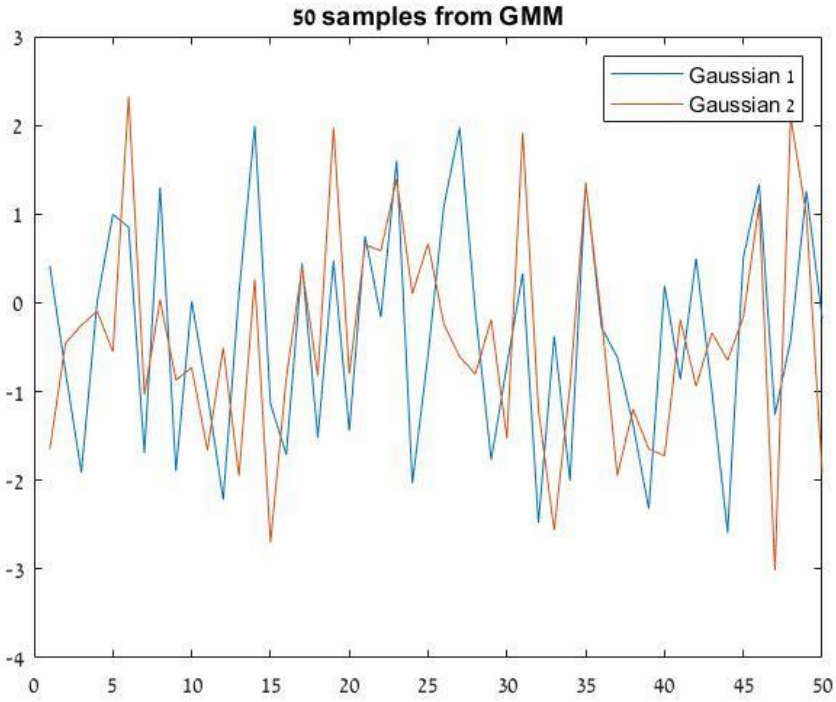
$$\mu_1 = [-1, -1]^T,$$
$$\mu_2 = [1, 1]^T,$$
$$\Sigma_1 = \begin{pmatrix} 0.8 & 0 \\ 0 & 0.8 \end{pmatrix},$$
$$\Sigma_2 = \begin{pmatrix} 0.75 & -0.2 \\ -0.2 & 0.6 \end{pmatrix},$$
$$P_Z(z = 1) = 0.7.$$

כעת אציג 1000 דגימות:



2. K-Means implementation:

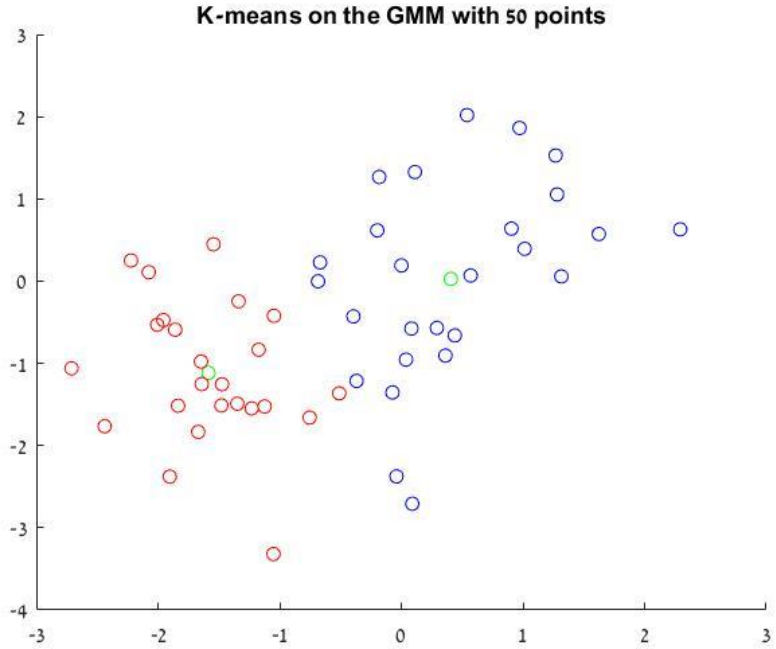
כעת נציג גרף של 50 דגימות מ-GMM שיצרנו בסעיף קודם:



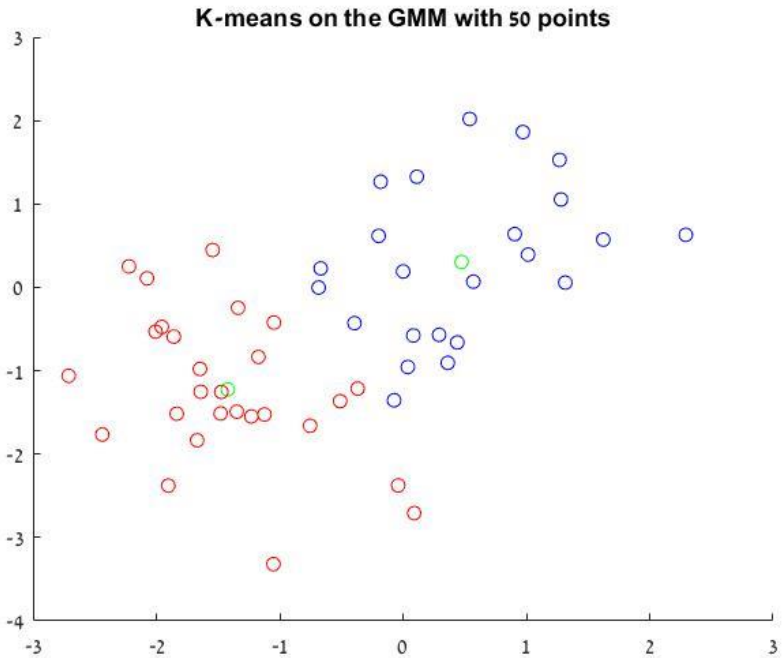
כעת נממש את אלגוריתם K-means עם שני מרכזים. נציג גרף של המרכזים ושיוך הנקודות לכל גאוסייאן לאחר כל איטרציה.

- הנקודות באדום שייכות לגאוסיין 1 ואילו הנקודות בכחול שייכות לגאוסיין 2.
- השתי נקודות הירוקות אלו המרכזים של הגאוסייאנים.

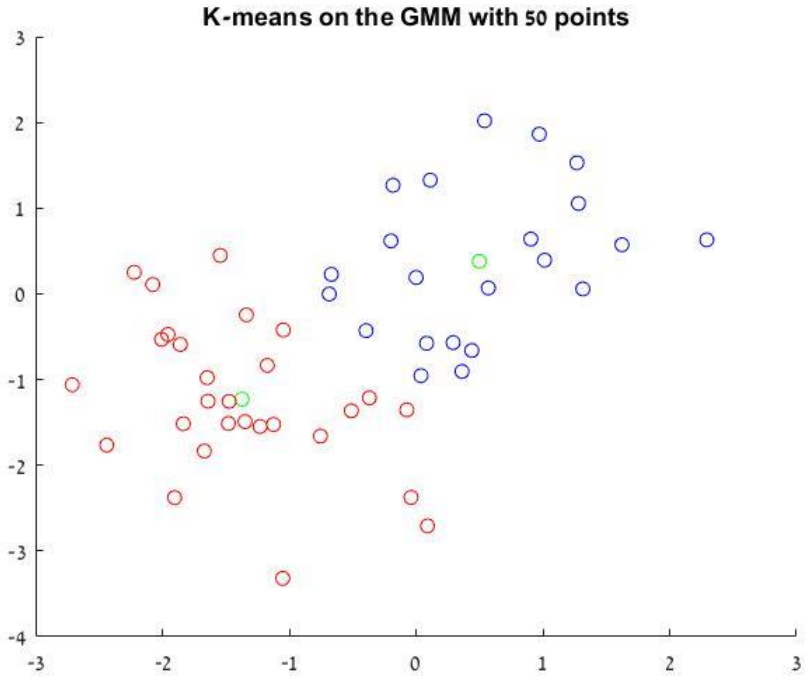
איטרציה 1:



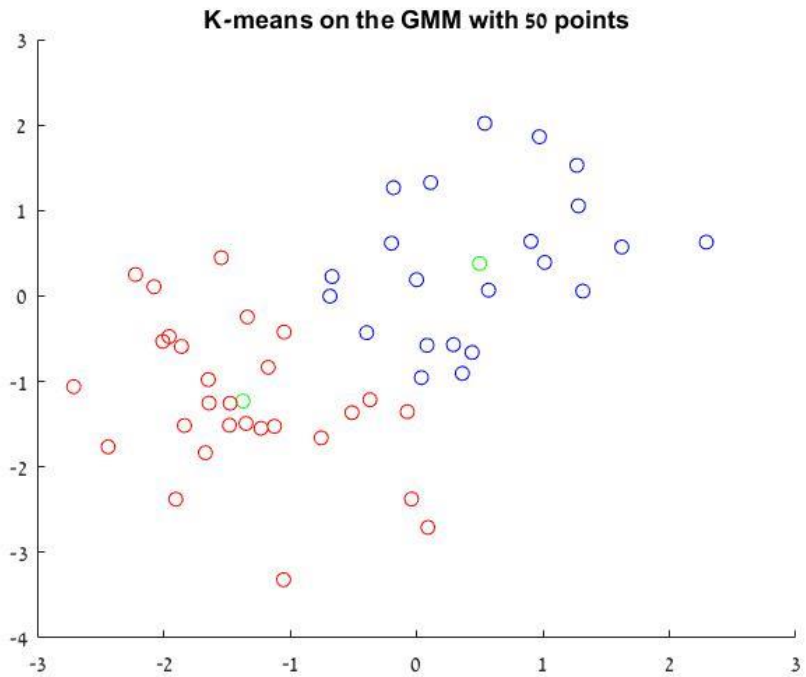
איטרציה 2:



איטרציה 3:



איטרציה 4:



המרכזים בתחילת האלגוריתם מוגרלים אקראית בין כל הנקודות, כלומר מוגרלות שתי נקודות מתוך 50 הנקודות (שגם הן נלקחו באקראי מהמודל) עבור שני המרכזים.

ניתן לראות שהאלגוריתם מבצע classification בצורה לא רעה בכלל, הוא מצליח לסווג מי מהנקודות שייכות לאיזה גאוסייאן.

בנוסף, ניתן לראות כי המרכזים של הגאוסיינים מתכנסים לתוחלת האמיתית

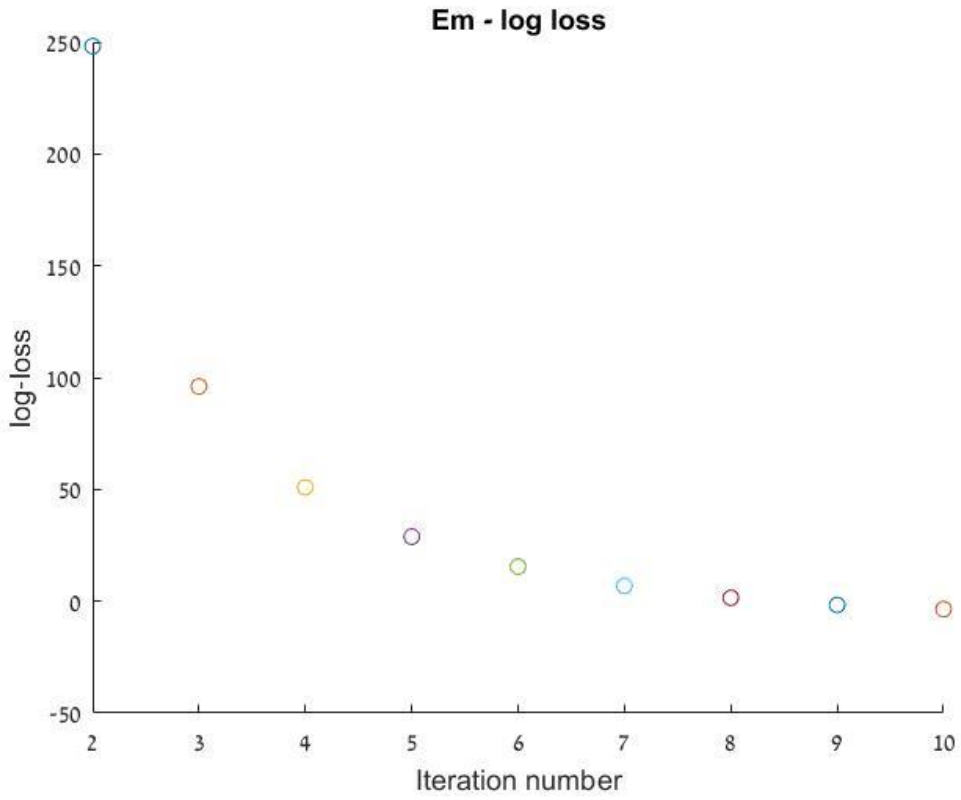
([-1,-1;1,1])

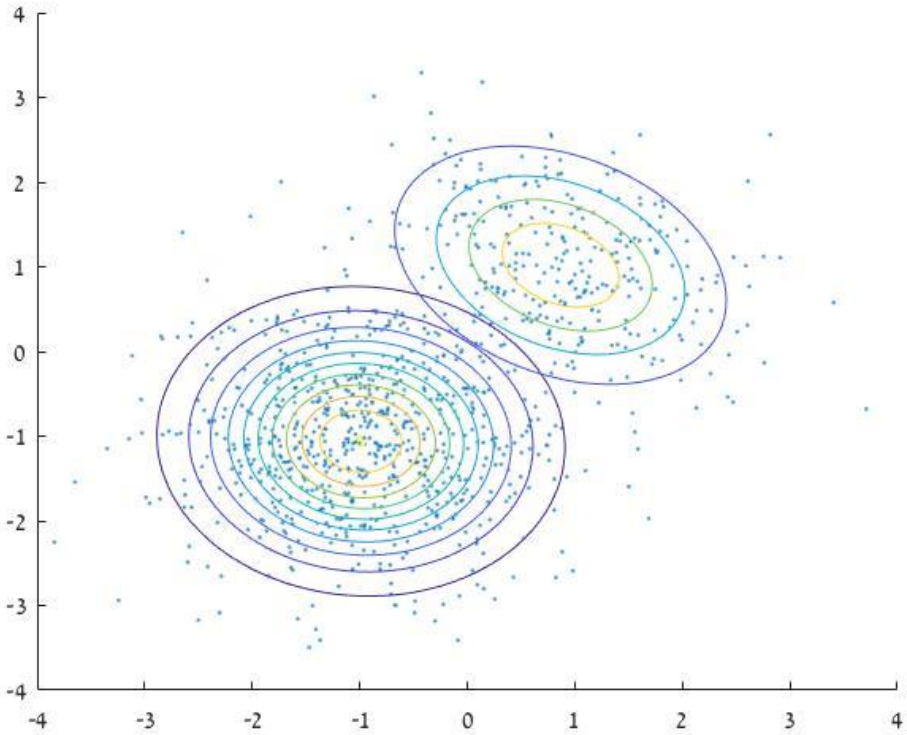
Centroids [-1.1070,-1.1600;0.9468,0.9056]

3. EM implementation:

תחילה נגריל 1000 דגימות מההתפלגות של ה- GMM. לאחר מכן נממש את אלגוריתם EM עבור שני גאוסייאנים.

נציג בגרף את log-loss עבור כל איטרציה של האלגוריתם:





ניתן לראות כי ה – log-loss דועך לאפס, כלומר השיערוך הולך ומתקרב לגאוסייאנים המקוריים וכן הסיווג של הנקודות מתבצע כהלכה. בנוסף ניתן לראות כי השערוך של התוחלת והשונות שהאלגוריתם מצא קרובים מאוד לערכים המקוריים.

תוחלת

[-0.9889,-1.0611;0.8674,1.0181]

שונות

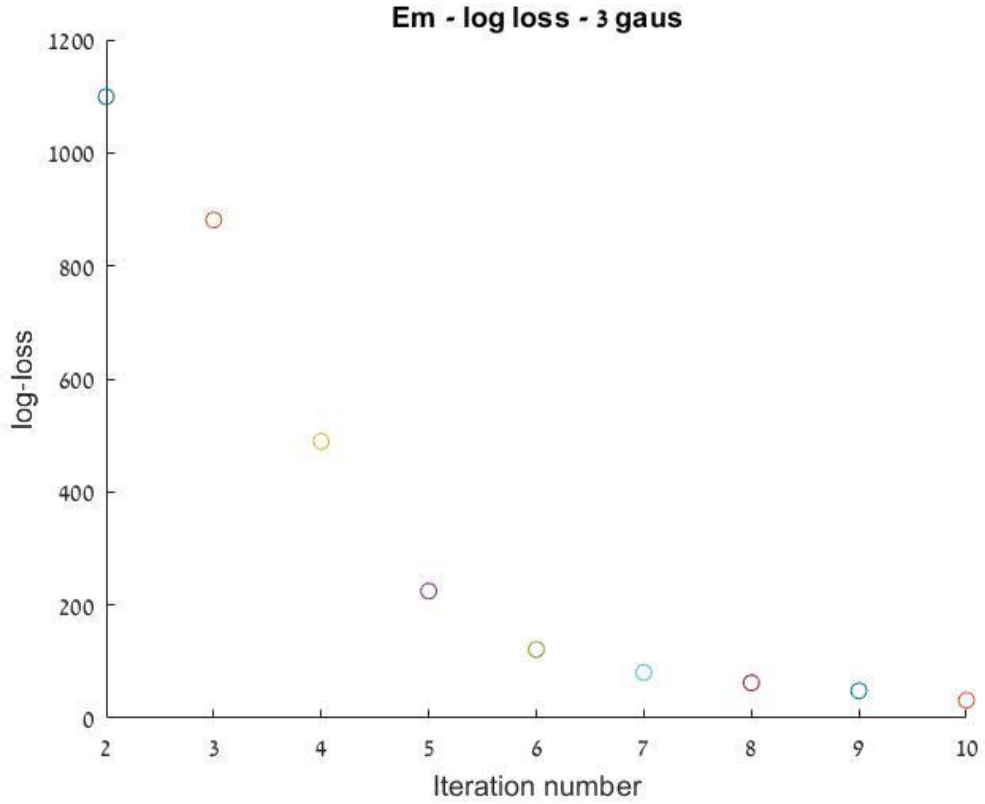
```
val(:,:,1) =

    0.7489   -0.0274
   -0.0274    0.6950


val(:,:,2) =

    0.7497   -0.2036
   -0.2036    0.6236
```

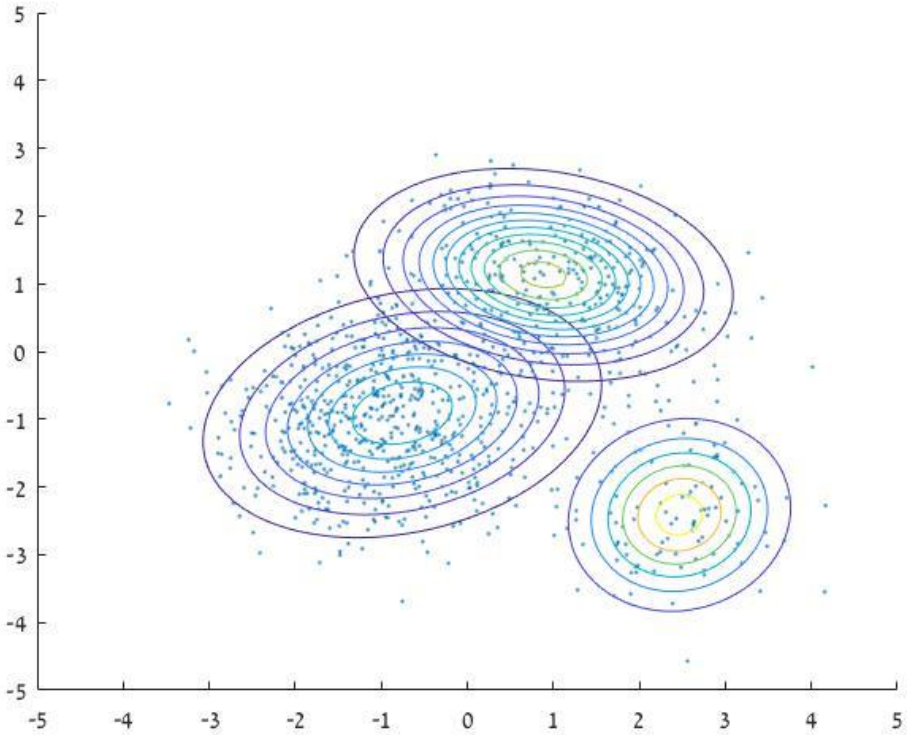
התוחלות והשונויות ההתחלתיות נלקחו באופן אקראי מהדגימות הנתונות (שגם הן נלקו באקראי). במהלך ריצת האלגוריתם שמתי לב כי אם התוחלת ההתחלתית רחוקה מאוד

מהתוחלת המקורית האלגוריתם לא מתכנס לתוצאה הנכונה. דבר זה הגיוני שכן מימד האקראיות כאן הוא גדול (גם התוחלת וגם השונות אקראיים לגמרי).

כעת נפעיל שוב את אלגוריתם EM על GMM עם שלושה גאוסייאנים כאשר הגאוסיין השלישי בעלת תוחלת [2.5,-2.5] ושונות [0.5 0,0 0.5]. בנוסף שיניתי את ההסתברות לכל גאוסייאן כך: לגאוסייאן הראשון הסתברות 0.6, לשני 0.3 ולשלישי 0.1.

גרף ה- log-loss עבור 3 גאוסייאנים:





גם כאן ניתן לראות כי ה- log-loss דועך לאפס עם כל איטרציה וכן האלגוריתם מבצע סיווג נכון של הנקודות. בדומה התוחלות והשוניות שהאלגוריתם מחשב קרובות למקור.

<u>שונות</u>:

```
val(:,:,1) =

    1.2891    0.2308
    0.2308    0.8109


val(:,:,2) =

    0.9963   -0.1319
   -0.1319    0.5055


val(:,:,3) =

    0.4430    0.0363
    0.0363    0.5382
```

<u>תוחלת</u>:

[-0.7561,-0.9097;0.8852,1.1297;2.4662,-2.4124]

קוד במטלב:

```
%Section 1

mu = [-1 -1;1 1];
sigma = cat(3,[.8 0; 0 .8],[.75 -0.2; -0.2 .6]);
p = [0.7,0.3];
gm = gmdistribution(mu,sigma,p);
X = random(gm,1000);
figure(1)
scatter(X(:,1),X(:,2),10,'.') % Scatter plot with points of size 10
xlabel('Gaussian 1');
ylabel('Gaussian 2');
title('Scatter 1000 points of the generated data')

%Section 2A
X = random(gm,50);
figure(2)
plot(X);
legend('Gaussian 1','Gaussian 2');
title('50 samples from GMM')

%Section 2 B+c
Centroids =
[X(randi(50),1),X(randi(50),2);X(randi(50),1),X(randi(50),2)];
C = zeros(50,1);
Arg1 = 0;
Arg2 = 0;

for k=1:4
    mone1=[0,0];
    mechane1=0;
    mone2=[0,0];
    mechane2=0;
   for i=1:50
          Arg1 = norm(X(i,:)- Centroids(1,:),2);
          Arg2 = norm(X(i,:)- Centroids(2,:),2);
          if Arg1<Arg2
              C(i)=1;
              mone1 = mone1+X(i,:);
              mechane1= mechane1+1;
          else
              C(i)=2;
              mone2 = mone2+X(i,:);
              mechane2 = mechane2 + 1;
          end
   end
    Centroids = [mone1/mechane1; mone2/mechane2];
    figure(k+2)
    scatter(Centroids(1,1),Centroids(1,2),'g');
    title('K-means on the GMM with 50 points')
    hold on
    scatter(Centroids(2,1),Centroids(2,2),'g');
    for i=1:50
        if C(i)==1
            scatter(X(i,1),X(i,2),'r');
            hold on
        else
```

```
            scatter(X(i,1),X(i,2),'b');
            hold on
        end

    end
end

%Section 3 a+b+c
X = random(gm,1000);
p1 = zeros(1000,1); %probability that realaziation i is from gaussian
1
mu_es = [X(randi(1000),:);X(randi(1000),:)];
sigma_es =  cat(3,[X(randi(1000),:);
X(randi(1000),:)],[X(randi(1000),:); X(randi(1000),:)]);
close all
original_log_likeli = 0;
for i=1:1000
        given1 = 1/(2*pi*sqrt(abs(det(sigma(:,:,1)))))*exp((-
1/2)*(X(i,:)-mu(1,:))*(sigma(:,:,1)\(X(i,:)-mu(1,:))'));
        given2 = 1/(2*pi*sqrt(abs(det(sigma(:,:,2)))))*exp((-
1/2)*(X(i,:)-mu(2,:))*(sigma(:,:,2)\(X(i,:)-mu(2,:))'));
        original_log_likeli = original_log_likeli +
log(given1*0.7+given2*0.3);
end

for k=1:10
    iteration_log_likeli=0;
    for i=1:1000
        given1 = 1/(2*pi*sqrt(abs(det(sigma_es(:,:,1)))))*exp((-
1/2)*(X(i,:)-mu_es(1,:))*(sigma_es(:,:,1)\(X(i,:)-mu_es(1,:))'));
        given2 = 1/(2*pi*sqrt(abs(det(sigma_es(:,:,2)))))*exp((-
1/2)*(X(i,:)-mu_es(2,:))*(sigma_es(:,:,2)\(X(i,:)-mu_es(2,:))'));
        p1(i) = given1*0.7/(given1*0.7+given2*0.3);
        iteration_log_likeli = iteration_log_likeli +
log(given1*0.7+given2*0.3);
    end
    mu_es = [X(:,1)'*p1/sum(p1) X(:,2)'*p1/sum(p1);X(:,1)'*(1-
p1)/sum(1-p1) X(:,2)'*(1-p1)/sum(1-p1)];
    acc1=[0 0; 0 0];
    acc2=[0 0; 0 0];
    for i=1:1000
        acc1=acc1+p1(i)*(X(i,:)-mu_es(1,:))'*(X(i,:)-mu_es(1,:));
        acc2=acc2+(1-p1(i))*(X(i,:)-mu_es(2,:))'*(X(i,:)-mu_es(2,:));
    end
    sigma_es = cat(3,acc1/sum(p1),acc2/sum(1-p1));
    if k>1
        figure(1)
        scatter(k,original_log_likeli-iteration_log_likeli);
        title('Em - log loss')
        xlabel('Iteration number')
        ylabel('log-loss')
        hold on
    end
end
%Section 3 d
close all
x = -4:.1:4 ; %// x a x i s
y = -4:.1:4 ; %// y a x i s
scatter(X(:,1),X(:,2),10,'.') % Scatter plot with points of size 10
```

```
hold on
[X ,Y] = meshgrid (x ,y) ; %// a l l c ombin a ti on s o f x , y
Z1 = mvnpdf ([X( : ),Y( : )] ,mu_es(1,:), sigma_es(:,:,1)) ; %//
compute Gaussian pd f
Z1 = reshape(Z1,size(X));
Z2 = mvnpdf ([X( : ),Y( : )] ,mu_es(2,:), sigma_es(:,:,2)) ; %//
compute Gaussian pd f
Z2 = reshape(Z2,size(X));
contour(X,Y,Z1);
contour(X,Y,Z2);
close all
%=====================================================================
================
%Section 3 e
mu = [-1 -1;1 1;2.5 -2.5];
sigma = cat(3,[.8 0; 0 .8],[.75 -0.2; -0.2 .6],[.5 0; 0 .5]);
p = [0.6,0.3,0.1];
gm = gmdistribution(mu,sigma,p);
X = random(gm,1000);
scatter(X(:,1),X(:,2),10,'.') % Scatter plot with points of size 10
p1 = zeros(1000,1); %probability that realaziation i is from gaussian
1
p2 = zeros(1000,1); %probability that realaziation i is from gaussian
2
mu_es = [X(randi(1000),:);X(randi(1000),:);X(randi(1000),:)];
sigma_es =  cat(3,[X(randi(1000),:);
X(randi(1000),:)],[X(randi(1000),:);
X(randi(1000),:)],[X(randi(1000),:); X(randi(1000),:)]);
close all
original_log_likeli = 0;
for i=1:1000
        given1 = 1/(2*pi*sqrt(abs(det(sigma(:,:,1)))))*exp((-
1/2)*(X(i,:)-mu(1,:))*(sigma(:,:,1)\(X(i,:)-mu(1,:))'));
        given2 = 1/(2*pi*sqrt(abs(det(sigma(:,:,2)))))*exp((-
1/2)*(X(i,:)-mu(2,:))*(sigma(:,:,2)\(X(i,:)-mu(2,:))'));
        given3 = 1/(2*pi*sqrt(abs(det(sigma(:,:,3)))))*exp((-
1/2)*(X(i,:)-mu(3,:))*(sigma(:,:,3)\(X(i,:)-mu(3,:))'));
        original_log_likeli = original_log_likeli +
log(given1*0.6+given2*0.2+given3*0.1);
end

for k=1:10
    iteration_log_likeli=0;
    for i=1:1000
        given1 = 1/(2*pi*sqrt(abs(det(sigma_es(:,:,1)))))*exp((-
1/2)*(X(i,:)-mu_es(1,:))*(sigma_es(:,:,1)\(X(i,:)-mu_es(1,:))'));
        given2 = 1/(2*pi*sqrt(abs(det(sigma_es(:,:,2)))))*exp((-
1/2)*(X(i,:)-mu_es(2,:))*(sigma_es(:,:,2)\(X(i,:)-mu_es(2,:))'));
        given3 = 1/(2*pi*sqrt(abs(det(sigma(:,:,3)))))*exp((-
1/2)*(X(i,:)-mu(3,:))*(sigma(:,:,3)\(X(i,:)-mu(3,:))'));
        p1(i) = given1*0.6/(given1*0.6+given2*0.2+given3*0.1);
        p2(i) = given2*0.2/(given1*0.6+given2*0.2+given3*0.1);
        iteration_log_likeli = iteration_log_likeli +
log(given1*0.6+given2*0.2+given3*0.1);
    end
    mu_es = [X(:,1)'*p1/sum(p1) X(:,2)'*p1/sum(p1);X(:,1)'*p2/sum(p2)
X(:,2)'*p2/sum(p2);X(:,1)'*(1-p1-p2)/sum(1-p1-p2) X(:,2)'*(1-p1-
p2)/sum(1-p1-p2)];
    acc1=[0 0; 0 0];
```

```
    acc2=[0 0; 0 0];
    acc3=[0 0; 0 0];
    for i=1:1000
        acc1=acc1+p1(i)*(X(i,:)-mu_es(1,:))'*(X(i,:)-mu_es(1,:));
        acc2=acc2+p2(i)*(X(i,:)-mu_es(2,:))'*(X(i,:)-mu_es(2,:));
        acc3=acc3+(1-p1(i)-p2(i))*(X(i,:)-mu_es(3,:))'*(X(i,:)-
mu_es(3,:));
    end
    sigma_es = cat(3,acc1/sum(p1),acc2/sum(p2),acc3/sum(1-p1-p2));
    if k>1
        figure(1)
        scatter(k,original_log_likeli-iteration_log_likeli);
        title('Em - log loss - 3 gaus')
        xlabel('Iteration number')
        ylabel('log-loss')
        hold on
    end
end
%Section 3 d
close all
x = -5:.1:5 ; %// x a x i s
y = -5:.1:5 ; %// y a x i s
scatter(X(:,1),X(:,2),10,'.') % Scatter plot with points of size 10
hold on
[X ,Y] = meshgrid (x ,y) ; %// a l l c ombin a ti on s o f x , y
Z1 = mvnpdf ([X( : ),Y( : )] ,mu_es(1,:), sigma_es(:,:,1)) ; %//
compute Gaussian pd f
Z1 = reshape(Z1,size(X));
Z2 = mvnpdf ([X( : ),Y( : )] ,mu_es(2,:), sigma_es(:,:,2)) ; %//
compute Gaussian pd f
Z2 = reshape(Z2,size(X));
Z3 = mvnpdf ([X( : ),Y( : )] ,mu_es(3,:), sigma_es(:,:,3)) ; %//
compute Gaussian pd f
Z3 = reshape(Z3,size(X));
contour(X,Y,Z1);
contour(X,Y,Z2);
contour(X,Y,Z3);
```