EE511 PROJECT 7

MARKOV CHAIN MONTE CARLO

NAME	SONALI B SREEDHAR
EMAIL ID	sbsreedh@usc.edu
USCID	1783668369

I. PROBLEM DESCRIPTION

This project details the use of EM for one popular problem of learning a mixture. We consider the problem of learning a GMM (also called EM clustering), where both the mixture weights of the Gaussians and the parameters for each component Gaussian must be learned. We illustrate using EM to learn a GMM when there are additional constraints on the GMM parameters. In section II, we talk about generating samples from a multivariate Gaussian. In section III, we discuss generate a mixture model of multivariate Gaussian. In section III, we will get into how to perform the GMM-EM estimation to learn a GMM model with 2 dimensions and 2 subpopulations. In section IV, we use the practical data from reality and perform the EM estimation to get insight of the feature of the data.

II. MULTIVARIATE GAUSSIAN

BIVARIATE CASE:

Suppose we have two random variable, N_1, N_2, $N_1 \sim (\mu_1, \sigma_1{}^2)$ and $N_2 \sim (\mu_2, \sigma_2)$. Consider a plane with *point*(x, y) on it. If we take one sample from N_1 and take the value as the value of x and take one sample from N_2 and take the value as y. After many sampling action, for one small area on the plane, mark as (x, y) we count the number of samples, and we may get the joint probability of $f_{N_1,N_2}(N_1 = x, N_2 = y)$. In this case,

$$f(x,y) = \frac{1}{2\pi\sigma_X\sigma_Y\sqrt{1-\rho^2}} \exp\left(-\frac{1}{2(1-\rho^2)}\left[\frac{(x-\mu_X)^2}{\sigma_X^2} + \frac{(y-\mu_Y)^2}{\sigma_Y^2} - \frac{2\rho(x-\mu_X)(y-\mu_Y)}{\sigma_X\sigma_Y}\right]\right)$$

where ρ is the correlation between X and Y and where $\sigma_X > 0$ and $\sigma_Y > 0$. In this case, $\boldsymbol{\mu} = \begin{pmatrix}\mu_X \\ \mu_Y\end{pmatrix}$, $\boldsymbol{\Sigma} = \begin{pmatrix}\sigma_X^2 & \rho\sigma_X\sigma_Y \\ \rho\sigma_X\sigma_Y & \sigma_Y^2\end{pmatrix}$. And we can expand this model to higher dimention.

CONSTRUCTION AND SAMPLING:

let $Z_1, Z_2, \ldots, Z_n$ be i.i.d standard normal distribution, that is $Z_i \sim N(0,1)$. For constant $a_{i,j}$ and μ_i, define:

$X_1 = a_{11}Z_1 + a_{12}Z_2 + \cdots a_{1m}Z_m + \mu_1$

$X_2 = a_{21}Z_1 + a_{22}Z_2 + \cdots a_{2m}Z_m + \mu_2$

$\vdots$

$X_n = a_{n1}Z_1 + a_{n2}Z_2 + \cdots a_{nm}Z_m + \mu_n$

The vector $X = [X_1, X_2, \cdots, X_n] iscalledmultivariatenormal$

In matrix mode, this can be wirtten as:

$$\begin{pmatrix}X_1 \\ X_2 \\ \vdots \\ X_n\end{pmatrix} = \begin{pmatrix}a_{1,1} & a_{1,2} & \cdots & a_{1,m} \\ a_{2,1} & a_{2,2} & \cdots & a_{2,m} \\ \vdots & \vdots & \ddots & \vdots \\ a_{n,1} & a_{n,2} & \cdots & a_{n,m}\end{pmatrix} * \begin{pmatrix}Z_1 \\ Z_2 \\ \vdots \\ Z_m\end{pmatrix} + \begin{pmatrix}\mu_1 \\ \mu_2 \\ \vdots \\ \mu_n\end{pmatrix}$$

And it is easy to get that the covariance $\Sigma = A * A^T$ where A is the matrix formed by $a_{i,j}$. Therefore, if we were given μ which provides the information about the mean value of each Gaussian components and Σ the covariance between each two of the Gaussian components. We can decompose the Σ into matrix A and generate samples from standard Gaussian random variables, then we can get several samples from Gaussian multivariate.

PROBLEM 1:
CODE:

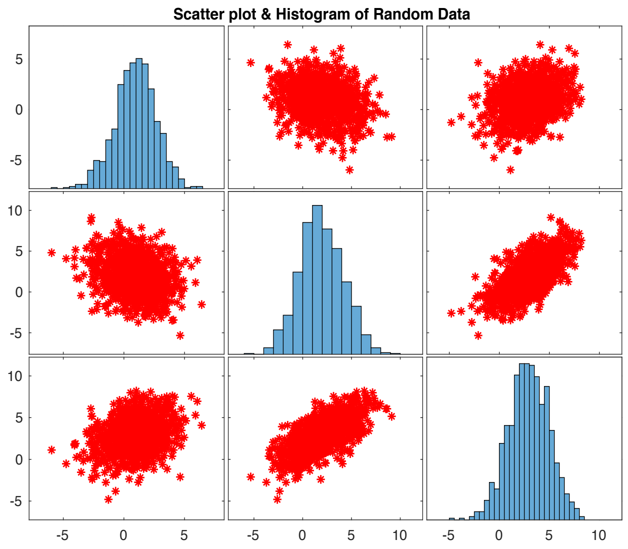
$$\text{give}: \mu = \begin{pmatrix} 1 \\ 2 \\ 3 \end{pmatrix}, \Sigma = \begin{pmatrix} 3 & -1 & 1 \\ -1 & 5 & 3 \\ 1 & 3 & 4 \end{pmatrix}$$

we generate samples from this gaussian multivariate:

```
clc;clear all;
N=1000; %Length of array of each RV
%given mu and sigma matrices
mu = [1,2,3]; sigma = [3,-1,1;-1,5,3;1,3,4];
A = chol(sigma); %Choleski lower dia matrix
X = randn(N,3)*A + repmat(mu,N,1);
M = mean(X)
C = cov(X)
plotmatrix(X,'*r')
title('Scatter plot & Histogram of Random Data')
```

RESULT:

```
M =0.9403    1.9471    2.9327

C =2.8867   -0.9406    1.0035
   -0.9406    4.9100    3.0185
    1.0035    3.0185    4.0291
```



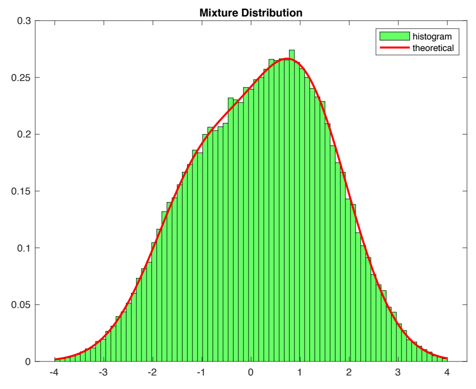
III MIXTURE MODEL

If we already have several multIvariate Gaussian , we mixture them by given each of the multivariate a weight.
When sampling we can first generate a sample from standard uniform distribution, and decide which Gaussian multivariate generator should we choose and then generate samples from that generator.

PROBLEM 2;
In this problem, we implement the following mixture distribution: $f(x) = 0.4N(-1, 1) + 0.6N(1, 1)$

CODE:
```
clc;clear all;%%---Mixture Distribution(Gaussian)---%%
N=100000;

for i=1:N
   if(rand <=0.4)
      Z(i)= normrnd(-1,1);
   else
      Z(i)= normrnd(1,1);
   end
end
histogram(Z,'BinLimits',[-4,4],'Normalization','pdf','FaceColor','g');
hold on; % overlaid the plots
%--Theoretical Distribution--%
y = -4:0.01:4;
mu1 = -1; mu2 = 1; var1 = 1; var2 = 1;
f1 = exp(-(y-mu1).^2./(2*var1))./(sqrt(2*pi*var1));
f2 = exp(-(y-mu2).^2./(2*var2))./(sqrt(2*pi*var2));
f = 0.4*f1 + 0.6*f2;
plot(y,f,'LineWidth',2,'Color','r');
legend('histogram','theoretical');title('Mixture Distribution');
```

RESULT:



IV GMM-EM ESTIMATION

Algorithm Overview:

1. **Initialization:**

Choose initial estimates $\omega_j^{(0)}, \mu_j^{(0)}, \Sigma_j^{(0)}, j = 1 \cdots k$, where j is the index of the cluster. k is the number of subpopulation.
Compute the initial log-likelihood:

$$l^{(0)} = \frac{1}{n}\sum_{i=1}^{n} log(\sum_{j=1}^{k} \omega_j^{(0)} \phi(y_i|\mu_j^{(0)}, \Sigma_j^{(0)}))$$

2. **E-Step:**

For *j* = 1,⋯ ,*k* compute

$$\gamma_{ij}^{(m)} = \frac{\omega_j^{(m)} \phi(y_i|\mu_j^{(m)}, \Sigma_j^{(m)})}{\sum_{l=1}^{k} \omega_l^{(m)} \phi(y_i|\mu_l^{(m)}, \Sigma_l^{(m)})}$$

and

$$n_j^{(m)} = \sum_{i=1}^{n} \gamma_{ij}^{(m)}$$

3. **M-Step:**

For $j = 1, \cdots, k$ compute the new estimates

$$\omega_j^{(m+1)} = \frac{n_j^{(m)}}{n}$$

$$\mu_j^{(m+1)} = \frac{1}{n_j^{(m)}} \sum_{i=1}^{n} \gamma_{ij}^{(m)} y_i$$

$$\Sigma_j^{(m+1)} = \frac{1}{n_j^{(m)}} \sum_{i=1}^{n} \gamma_{ij}^{(m)} (y_i - \mu_j^{(m+1)})(y_i - \mu_j^{(m+1)})^T$$

4. **Convergence check:**
5.

$$l^{(m+1)} = \frac{1}{n}\sum_{i=1}^{n} log(\sum_{j=1}^{k} \omega_j^{(m+1)} \phi(y_i|\mu_j^{(m+1)}, \Sigma_j^{(m+1)}))$$

Return to 2 step if $|l^{(m+1)} - l^{(m)}| > \delta$ for some threshold δ.

PROBLEM 3:
CODE:

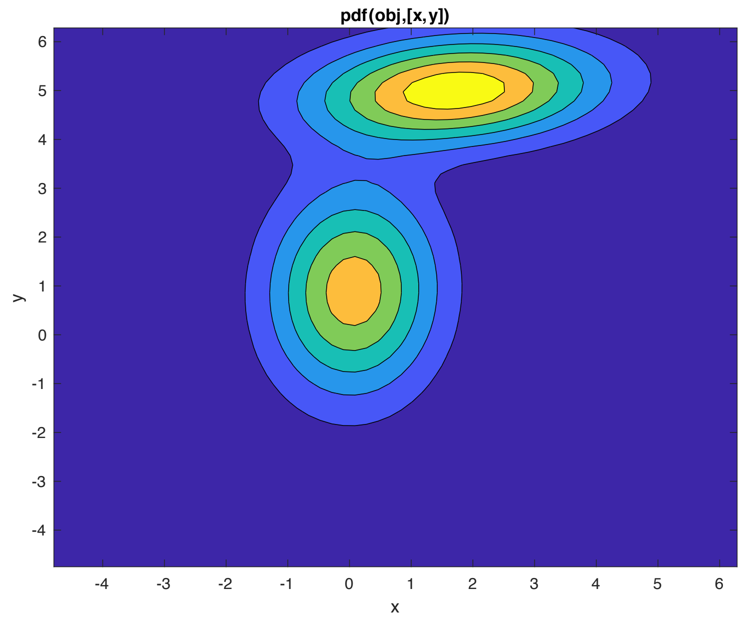
```
clear; close all;

% Generate data X
n = 300;
mu1 = [ 2 5];
```

```matlab
sigma1 = [3 1; 1 0.5];
mu2 = [0 1];
sigma2 = [1 0.5; 0.5 2];

for i = 1:n/2
    X1(i,:) = (chol(sigma1)*[normrnd(0,1) normrnd(0,1)]' + mu1')';
    X2(i,:) = (chol(sigma2)*[normrnd(0,1) normrnd(0,1)]' + mu2')';
end

X = [X1; X2];

% Initialization
k = randperm(n);
mu = X(k(1:2), :);

sigma1 = cov(X);
sigma2 = cov(X);

p = [0.2 0.8];

ud1 = bsxfun(@minus, X, mu(1,:));
ud2 = bsxfun(@minus, X, mu(2,:));

phi(:,1)=exp(-1/2*sum((ud1*inv(sigma1).*ud1),2))/sqrt((2*pi)^2*det(sigma1));
phi(:,2)=exp(-1/2*sum((ud2*inv(sigma2).*ud2),2))/sqrt((2*pi)^2*det(sigma2));

L_before = sum(log(p(1).*phi(:,1)+p(2).*phi(:,2)))/n;

for i = 1:2000
    %E-step
    phi_w = bsxfun(@times, phi, p);

    gamma = bsxfun(@rdivide, phi_w, sum(phi_w, 2));

    mu_temp = mu;

    % M-step
    mu(1,:) = gamma(:, 1)' * X ./ sum(gamma(:, 1), 1);
    mu(2,:) = gamma(:, 2)' * X ./ sum(gamma(:, 2), 1);

    XS1 = bsxfun(@minus, X, mu(1, :));
    XS2 = bsxfun(@minus, X, mu(2, :));

    for j=1:n
        sigma1 = sigma1 + (gamma(j, 1) .* (XS1(j, :)' * XS1(j, :)));
        sigma2 = sigma2 + (gamma(j, 2) .* (XS2(j, :)' * XS2(j, :)));
    end

    sigma1 = sigma1 ./ sum(gamma(:, 1));
    sigma2 = sigma2 ./ sum(gamma(:, 2));

    p = [mean(gamma(:,1)) mean(gamma(:,2))];

    ud1 = bsxfun(@minus, X, mu(1,:));
    ud2 = bsxfun(@minus, X, mu(2,:));
    phi(:,1)=exp(-
1/2*sum((ud1*inv(sigma1).*ud1),2))/sqrt((2*pi)^2*det(sigma1));
    phi(:,2)=exp(-
1/2*sum((ud2*inv(sigma2).*ud2),2))/sqrt((2*pi)^2*det(sigma2));

    L_after = sum(log(p(1).*phi(:,1)+p(2).*phi(:,2)))/n;

    % Check convergence
    if L_after == L_before
        break;
    else
        L_before = L_after;
    end

end

sigma = cat(3,sigma1,sigma2);
% build GMM
obj = gmdistribution(mu,sigma,p);

% 2D projection
figure;
ezcontourf(@(x,y) pdf(obj,[x y]));

mu
sigma1
sigma2
p
```

RESULT:

Experiment 1: no covariance

$\omega = \begin{pmatrix} 0.5 \\ 0.5 \end{pmatrix}$ $\mu_1 = \begin{pmatrix} 2 \\ 5 \end{pmatrix}$ $\mu_2 = \begin{pmatrix} 0 \\ 1 \end{pmatrix}$ $\Sigma_1 = \begin{pmatrix} 3 & 0 \\ 0 & 0.5 \end{pmatrix}$ $\Sigma_2 = \begin{pmatrix} 1 & 0 \\ 0 & 2 \end{pmatrix}$



```
mu =
       1.7241    5.0062
        0.0663    0.8925
```

```
sigma2 =0.8913    0.0449
           0.0449    2.1972
```

```
sigma1 =
       2.6075    0.1779
      0.1779    0.5517
```

```
p =0.5071    0.4929
```

Experiment 2: with covariance

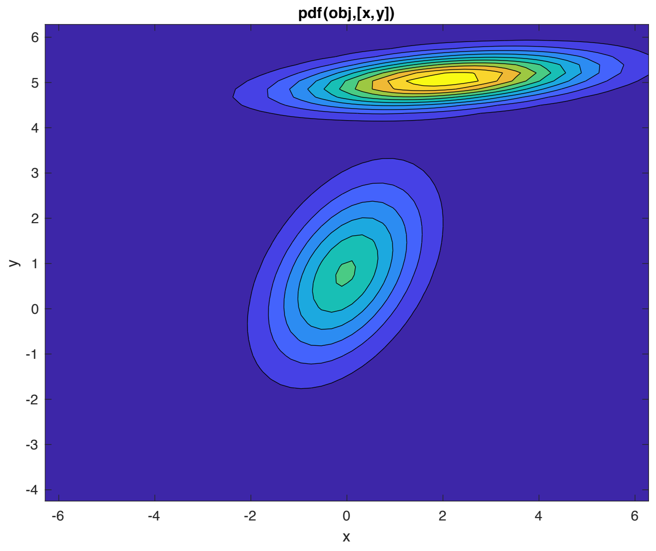
$$\omega = \begin{pmatrix} 0.5 \\ 0.5 \end{pmatrix} \mu_1 = \begin{pmatrix} 2 \\ 5 \end{pmatrix} \mu_2 = \begin{pmatrix} 0 \\ 1 \end{pmatrix} \Sigma_1 = \begin{pmatrix} 3 & 1 \\ 1 & 0.5 \end{pmatrix} \Sigma_2 = \begin{pmatrix} 1 & 0.5 \\ 0.5 & 2 \end{pmatrix}$$

```
mu =

  -0.0333    0.7785
   2.0457    5.0608
```

```
sigma2 =

   4.1006    0.3189
   0.3189    0.1592
```

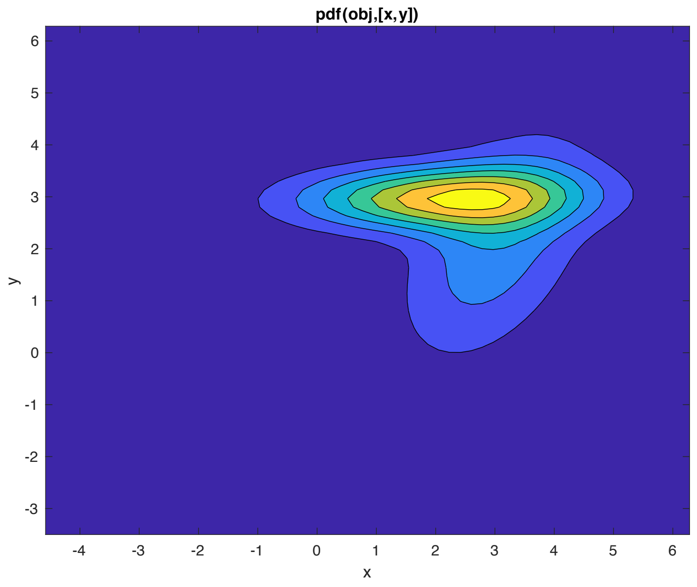
```
sigma1 =

1.1391    0.6304
 0.6304    1.7785
```

```
p =0.4935    0.5065
```



Experiment 3: Close

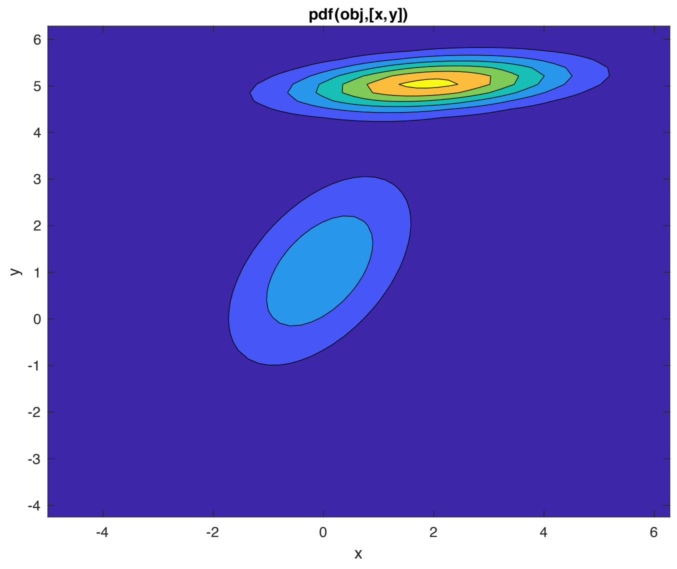
$$\omega = \begin{pmatrix} 0.5 \\ 0.5 \end{pmatrix} \mu_1 = \begin{pmatrix} 2 \\ 3 \end{pmatrix} \mu_2 = \begin{pmatrix} 3 \\ 2 \end{pmatrix} \Sigma_1 = \begin{pmatrix} 3 & 1 \\ 1 & 0.5 \end{pmatrix} \Sigma_2 = \begin{pmatrix} 1 & 0.5 \\ 0.5 & 2 \end{pmatrix}$$



```
mu =2.0612    2.9760
    3.0193    2.0611
                                                          sigma1 =2.5252    0.0456
                                                                  0.0456    0.1742

  Sigma2=1.1311    0.7126                                     p =0.5308    0.4692
         0.7126    2.1307
```

Experiment 4: different weight

$$\omega = \begin{pmatrix} 0.2 \\ 0.8 \end{pmatrix} \mu_1 = \begin{pmatrix} 2 \\ 5 \end{pmatrix} \mu_2 = \begin{pmatrix} 0 \\ 1 \end{pmatrix} \Sigma_1 = \begin{pmatrix} 3 & 1 \\ 1 & 0.5 \end{pmatrix} \Sigma_2 = \begin{pmatrix} 1 & 0.5 \\ 0.5 & 2 \end{pmatrix}$$



```
mu=  2.2261    3.0026
      2.8270    2.0268


sigma1 =

   3.0284    0.0567
   0.0567    0.1234
```

```
sigma2 =

   1.3758    0.7124
   0.7124    1.9792


p =

   0.5241    0.4759
```

Performance Analysis:

The weight actually has not very much influence on the speed although if the weight vary a lot from our initial guess, it may take more iterations. However, if the two Gaussian Multivariate are close, it really took a lot more iterations. And it can be seen from the data. If the threshold is set to 0.0001, the variance between the true mean and the estimation is actually lower than 5% . The covariance may vary a bit between the true parameter and the estimation, that basically is due to the amounts of samples are really small and if large amounts of samples are given, we may get better estimation.

V . PRACTICAL DATA EM

PROBLEM 4:

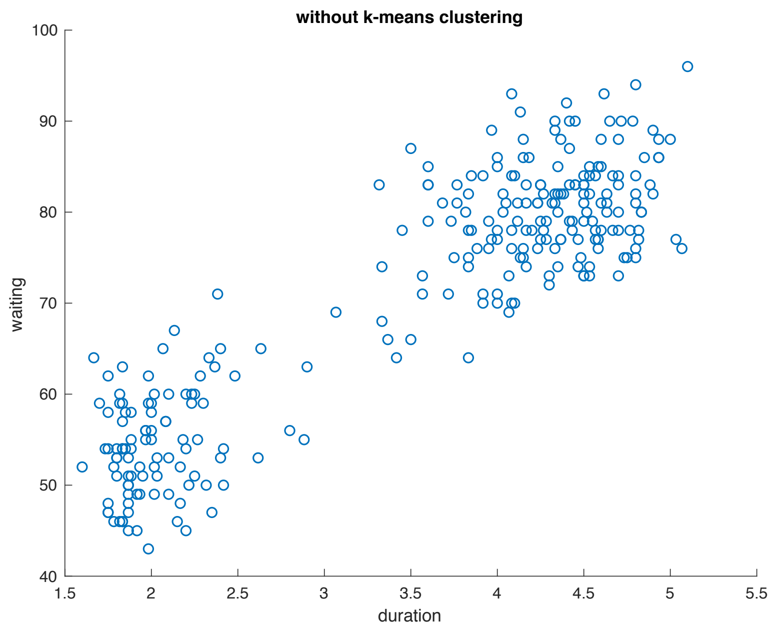
A geyser is a hot spring characterized by an intermittent discharge of water and steam. Old Faithful is a famous cone geyser in Yellowstone National Park, Wyoming. It has a predicable geothermal discharge and since 2000, it has errupted every 44 to 125 minutes. Refer to the addendum data file that contains waiting times and the durations for 272 erruptions.

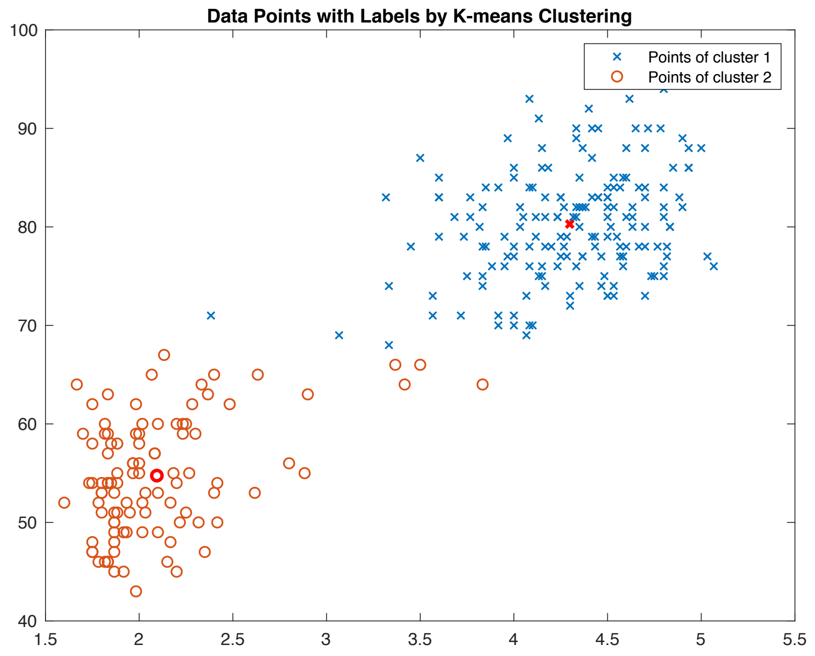
CODE:

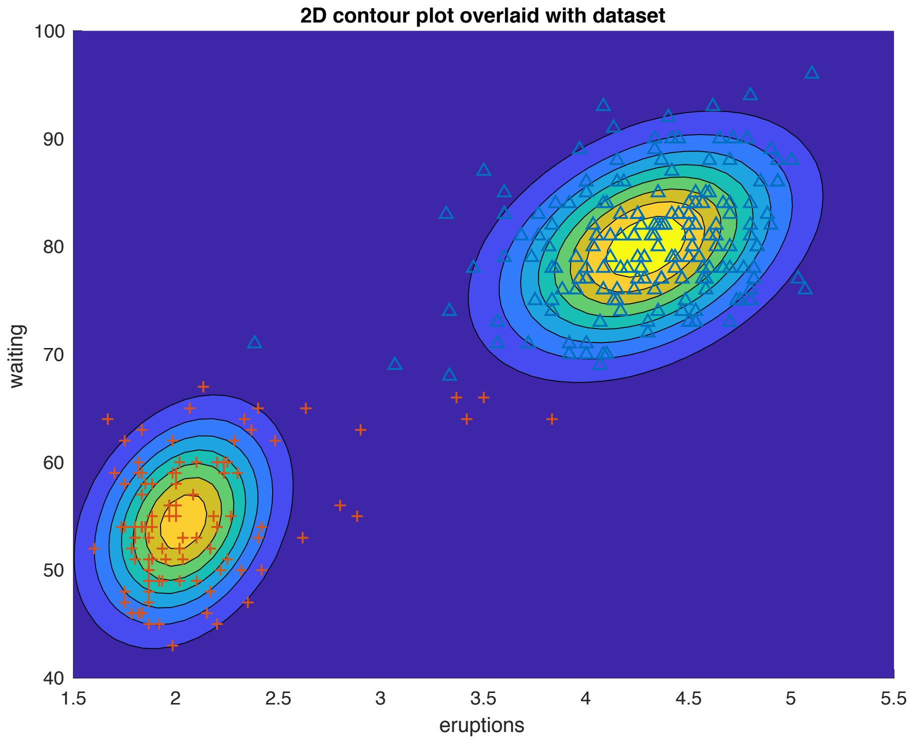
```
clc;clear all;close all;
duration = load('duration_data.mat');
duration = duration.duration;
waiting = load('waiting_data.mat');
waiting = waiting.waiting;
scatter(duration,waiting);
X =[duration,waiting];
title('without k-means clustering')
xlabel('duration'); ylabel('waiting');
%cascade data points
X = [duration,waiting];
```

```
%shuffle
X = X(randperm(size(X,1)),:);
%kmeans
[y,C] = kmeans(X,2);
figure(2)
plot(X(y==1,1),X(y==1,2), 'x');
hold on;
plot(X(y==2,1),X(y==2,2), 'o');
plot(C(1,1),C(1,2), 'rx','LineWidth',2);
plot(C(2,1),C(2,2), 'ro','LineWidth',2);
legend('Points of cluster 1','Points of cluster 2')
title('Data Points with Labels by K-means Clustering')
hold off;
% GMM Distribution
EM = gmdistribution.fit(X,2);
% 2D projection
figure;
hold on
ezcontourf(@(x,y) pdf(EM,[x y]),[1.5 5.5, 40 100]);

plot(X(y==1,1),X(y==1,2),'^');
plot(X(y==2,1),X(y==2,2),'+');
title('2D contour plot overlaid with dataset');
xlabel('eruptions');
ylabel('waiting');
```

RESULT:







Therefore, we can use the GMM model to estimate the behaviour of the geyser by sampling from the GMM model with parameters derived from above.