Contents


```
% ECE414 - Bayesian Machine Learning
% Authors   : Junbum Kim, Andy Jeong
% Project 5 : Expectation-Maximization on Gaussian Mixture Model
% Date      : November 20, 2019
% Reference : Pattern Recognition and Machine Learning by C. M. Bishop (2006)
close all; clear all; clc;    % clear workspace variables
rng(42);                      % for reproducibility
```

Equations

1 EM for Gaussian Mixtures

Goal: maximize the likelihood function with respect to the parameters
1. Initialize the means μ_k, covariances Σ_k and mixing coefficients π_k, and evaluate the initial value of the log likelihood
2. E-step:

$$\gamma(z_{nk}) = \frac{\pi_k N(x_n \mid \mu_k, \Sigma_k)}{\sum_{j=1}^{K} \pi_j N(_n | \ \mu_j, \Sigma_j)}$$

3. M-step:

$$\mu_k^{new} = \frac{1}{N_k} \sum_{n=1}^{N} \gamma(z_{nk}) x_n$$

$$\Sigma_k^{new} = \frac{1}{N_k} \sum_{n=1}^{N} \gamma(z_{nk})(x_n - \mu_k^{new})(x_n - \mu_k^{new})^T$$

$$\pi_k^{new} = \frac{N_k}{N}$$

5. Evaluate the log-likelihood:

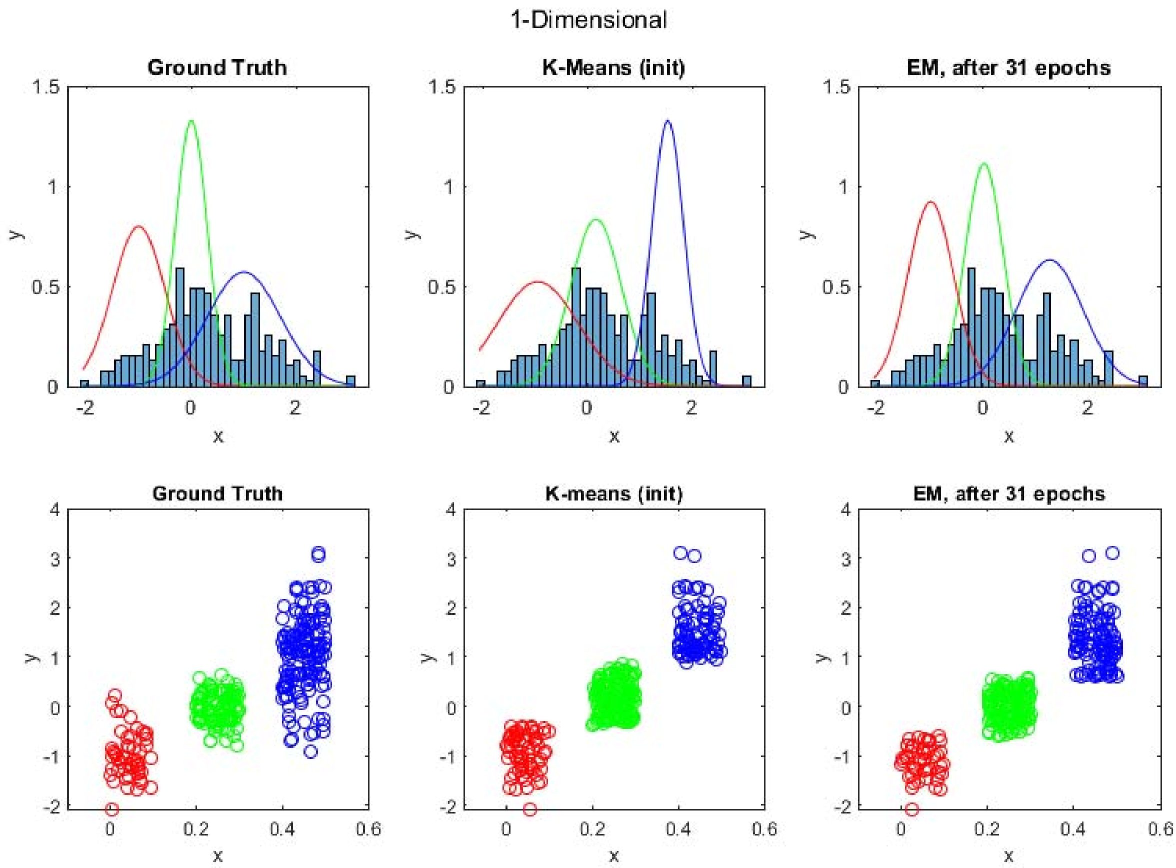
$$ln(p(X \mid \mu, \Sigma, \pi)) = \sum_{n=1}^{N} ln \sum_{k=1}^{K} \pi_k N(x_n \mid \mu_k, \Sigma_k)$$

1-Dimensional

```
% define parameters
mu = [-1, 0, 1];              % mean
sigma = [0.5, 0.3, 0.7];      % standard deviation
N = [50, 100, 150];           % number of data points per class

% generate random data points of varying means, standard deviations
data1 = normrnd(mu(1), sigma(1), [N(1), 1]);
data2 = normrnd(mu(2), sigma(2), [N(2), 1]);
data3 = normrnd(mu(3), sigma(3), [N(3), 1]);
data = [data1;data2;data3];

% obtain initial approximation with K-means (K = 3)
[IDX, C] = kmeans(data,3);

% initialize parameters according to K-means clustering
mu_k = C;
sigma_k = sqrt([var(data1) var(data2) var(data3)]);
pi_k = mean([IDX==1 IDX==2 IDX==3]);

num_bins = 40;
figure('Renderer', 'painters', 'Position', [100 100 900 600]);
figure(1);
% plot histogram and pdf of the ground truth
subplot(2,3,1);
plot_hist(data, mu, sigma, num_bins, ['r', 'g', 'b']);
title('Ground Truth');
```

```
% plot histogram and pdf of the initial K-means approx.
subplot(2,3,2);
plot_hist(data, mu_k, sigma_k, num_bins, ['g', 'b', 'r']);
title('K-Means (init)');

% responsibility variable
gamma = zeros(sum(N),3);

% set likelihood to an extreme value to start
llh_prev = -1e100;
llh = -1e99;
epoch = 0;
convergence_criterion = 1e-1/2;
while llh - llh_prev > convergence_criterion
    % Expectation
    gamma(:,1) = pi_k(1) * normpdf(data, mu_k(1), sigma_k(1));
    gamma(:,2) = pi_k(2) * normpdf(data, mu_k(2), sigma_k(2));
    gamma(:,3) = pi_k(3) * normpdf(data, mu_k(3), sigma_k(3));

    tmp = gamma;
    gamma = gamma ./ sum(gamma,2);

    % Maximization
    pi_k = mean(gamma);
    Nk = sum(gamma);

    mu_k = sum(gamma .* data) ./ Nk;
    sigma_k = sqrt(sum(gamma .* (data - mu_k).^2)./Nk);

    llh_prev = llh;
    llh = sum(log(sum(tmp,2)));
    epoch = epoch + 1;
end

% plot histogram and pdf of the final after EM
subplot(2,3,3);
plot_hist(data, mu_k, sigma_k, num_bins, ['g', 'b', 'r']);
title(sprintf('EM, after %d epochs',epoch));

sprintf('1-D: EM took %d epochs', epoch)

% plot data by classes
subplot(2,3,4);
plot1D(data1,data2,data3);
title("Ground Truth");

subplot(2,3,5);
plot1D(data(IDX==3), data(IDX==1), data(IDX==2));
title("K-means (init)");

[~, label] = max(gamma,[],2);
subplot(2,3,6);
plot1D(data(label==3),data(label==1),data(label==2));
title(sprintf('EM, after %d epochs',epoch));

% adjust the main title
t = suptitle('1-Dimensional');
set(t, 'FontSize', 12, 'FontWeight', 'normal');
```

```
ans =

    '1-D: EM took 31 epochs'
```

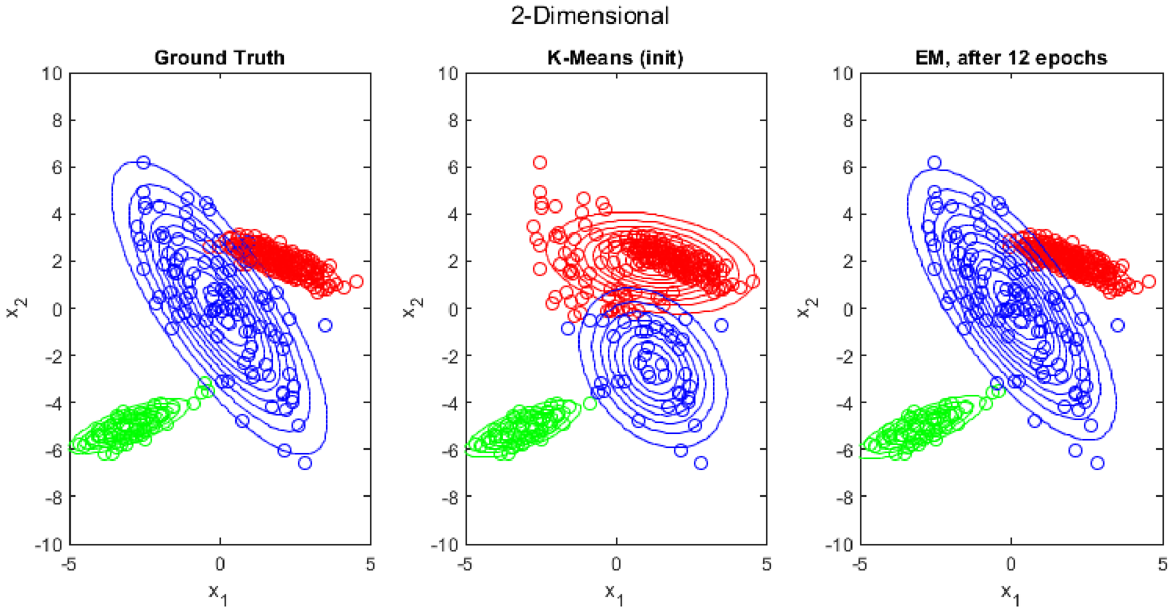


2-Dimensional

```
% define parameters for data
mu1 = [-3,-5];                    % means
mu2 = [0, 0];
mu3 = [2, 2];
sigma1 = [0.9, 0.4; 0.4, 0.3];    % covariances
sigma2 = inv(sigma1);
sigma3 = [0.9, -0.4; -0.4, 0.3];
N = [50, 100, 150];               % number of data points per class

% generate random data points of varying means, covariances
data1 = mvnrnd(mu1,sigma1,N(1));
data2 = mvnrnd(mu2,sigma2,N(2));
data3 = mvnrnd(mu3,sigma3,N(3));
data = [data1;data2;data3];

% obtain initial approximation with K-means (K = 3)
C = kmeans(data,3);

% initialize parameters according to K-means clustering
mu_1 = mean(data(C==1,:));      % means
mu_2 = mean(data(C==2,:));
mu_3 = mean(data(C==3,:));
sigma_1 = cov(data(C==1,:));    % covariances
sigma_2 = cov(data(C==2,:));
sigma_3 = cov(data(C==3,:));
pi_k = mean([C==1 C==2 C==3]);  % mixing probability

figure('Renderer', 'painters', 'Position', [100 100 900 400]);
figure(2);
% plot data by classes
subplot(1,3,1);
plot2D(data3,data1,data2); hold on;
plot_circle(mu1, sigma1,'g');
plot_circle(mu2, sigma2,'b');
plot_circle(mu3, sigma3,'r');
xlim([-5 5]); title("Ground Truth");

% plot circles for the initial guess
subplot(1,3,2);
plot2D(data(C==1,:), data(C==2,:), data(C==3,:)); hold on;
plot_circle(mu_1, sigma_1,'r');
plot_circle(mu_2, sigma_2,'g');
```

```
plot_circle(mu_3, sigma_3,'b');
xlim([-5 5]); title('K-Means (init)');

% responsibility variable
gamma1 = zeros(sum(N),2);
gamma2 = zeros(sum(N),2);
gamma3 = zeros(sum(N),2);

% set likelihood to an extreme value to start
llh_prev = -1e100;
llh = -1e99;
epoch = 0;
convergence_criterion = 1e-2;
while llh - llh_prev > convergence_criterion
    % Expectation
    gamma1 = pi_k(1) * mvnpdf(data, mu_1, sigma_1);
    gamma2 = pi_k(2) * mvnpdf(data, mu_2, sigma_2);
    gamma3 = pi_k(3) * mvnpdf(data, mu_3, sigma_3);

    gamma = [gamma1 gamma2 gamma3];
    tmp = gamma;
    gamma = gamma ./ sum(gamma,2);

    % Maximazation
    pi_k = mean(gamma);
    Nk = sum(gamma);

    mu_1 = sum(gamma(:,1) .* data) ./ Nk(1);
    mu_2 = sum(gamma(:,2) .* data) ./ Nk(2);
    mu_3 = sum(gamma(:,3) .* data) ./ Nk(3);

    sigma_1 = (data - mu_1)' * ((data - mu_1) .* gamma(:,1))./Nk(1);
    sigma_2 = (data - mu_2)' * ((data - mu_2) .* gamma(:,2))./Nk(2);
    sigma_3 = (data - mu_3)' * ((data - mu_3) .* gamma(:,3))./Nk(3);

    llh_prev = llh;
    llh = sum(log(sum(tmp,2)));
    epoch = epoch + 1;
end

% plot histogram and pdf of the final after EM
subplot(1,3,3);
[~, label] = max(gamma,[],2);
plot2D(data(label==1,:),data(label==2,:),data(label==3,:)); hold on;
plot_circle(mu_1, sigma_1,'r');
plot_circle(mu_2, sigma_2,'g');
plot_circle(mu_3, sigma_3,'b');
xlim([-5 5]); title(sprintf('EM, after %d epochs',epoch));

sprintf('2-D: EM took %d epochs', epoch)

% adjust the main title
t = suptitle('2-Dimensional');
set(t, 'FontSize', 12, 'FontWeight', 'normal');
```

```
ans =

    '2-D: EM took 12 epochs'
```



Functions

```
% plot data points in 1-D space
function [p] = plot1D(arg1, arg2, arg3)
    sep = 0.2;                                  % spacing in x in the plot
    axis1 = rand(length(arg1),1) * 0.1;
    axis2 = rand(length(arg2),1) * 0.1 + sep;
    axis3 = rand(length(arg3),1) * 0.1 + sep * 2;
    p = plot(axis1, arg1, 'ro', axis2, arg2, 'go', axis3, arg3, 'bo');
    xlim([-0.1,0.6]); xlabel('x'); ylabel('y');
end

% plot data points in 2-D space
function [p] = plot2D(arg1,arg2,arg3)
    p = plot(arg1(:,1), arg1(:,2),'ro', arg2(:,1),arg2(:,2),'go',arg3(:,1),arg3(:,2),'bo');
end

% plot histogram and probability density function
function plot_hist(data, mu_guess, sigma_guess, num_bins, color)
    histogram(data, num_bins, 'Normalization', 'pdf'); hold on;       % histogram
    x = linspace(min(data),max(data),100);
    for i=1:length(mu_guess)
        plot(x, normpdf(x, mu_guess(i), sigma_guess(i)), color(i));    % distribution
    end
    hold off; xlabel('x'); ylabel('y'); ylim([0 1.5]);
end

% plot circles around the cluster means
function plot_circle(mu, sigma, color)
    [x, y] = meshgrid(linspace(-10,10,100));
    z = mvnpdf([x(:), y(:)], mu, sigma);
    z = reshape(z, size(x));

    contour(x, y, z, 'LineColor', color); hold on;
    xlabel('x_1'); ylabel('x_2');
end
```

Published with MATLAB® R2018a