HW6 Solutions

Gabe Hope

April 2017

1 Problem 1

1.1 2-Component Dataset

Parameters after fittting:

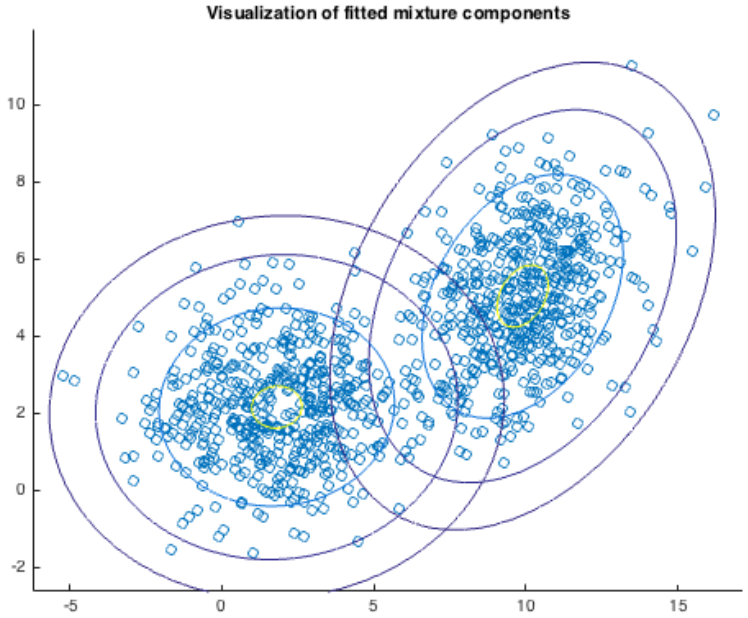
$$\pi = [0.5056, 0.4944]$$

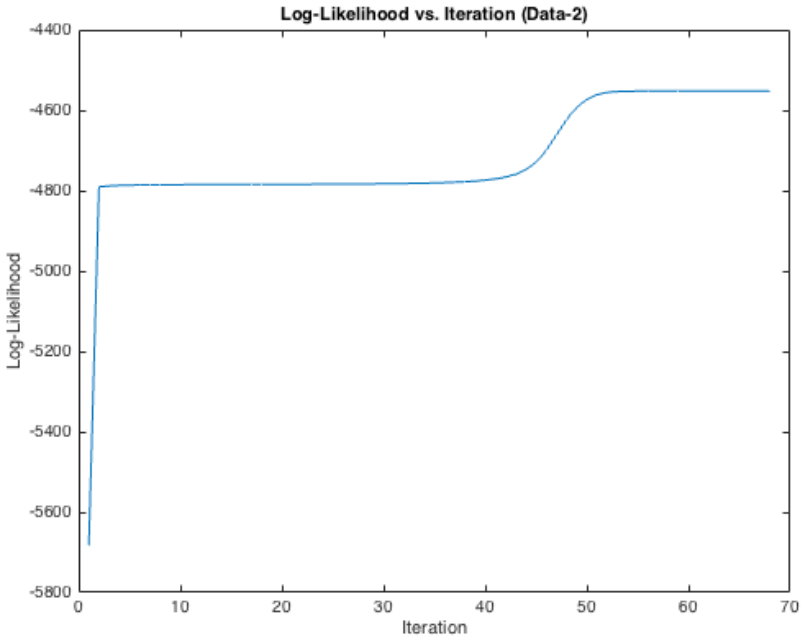
$$\mu_1 = [9.8872, 5.0366]$$

$$\mu_2 = [1.8013, 2.1581]$$

$$\sigma_1 = \begin{bmatrix} 3.1687 & 1.0606 \\ 1.0606 & 2.9010 \end{bmatrix}$$

$$\sigma_2 = \begin{bmatrix} 4.4387 & 0.1105 \\ 0.1105 & 1.9539 \end{bmatrix}$$


Visualization of fitted mixture components
10
8
6
4
2
0
-2
-5
0
5
10
15


Log-Likelihood vs. Iteration (Data-2)
-4400
-4600
-4800
-5000
-5200
-5400
-5600
-5800
Log-Likelihood
0
10
20
30
40
50
60
70
Iteration

1.2 3-Component Dataset

Parameters after fittting:

$$\pi = [0.3474, 0.3326, 0.3199]$$

$$\mu_1 = [2.9997, 4.6774]$$

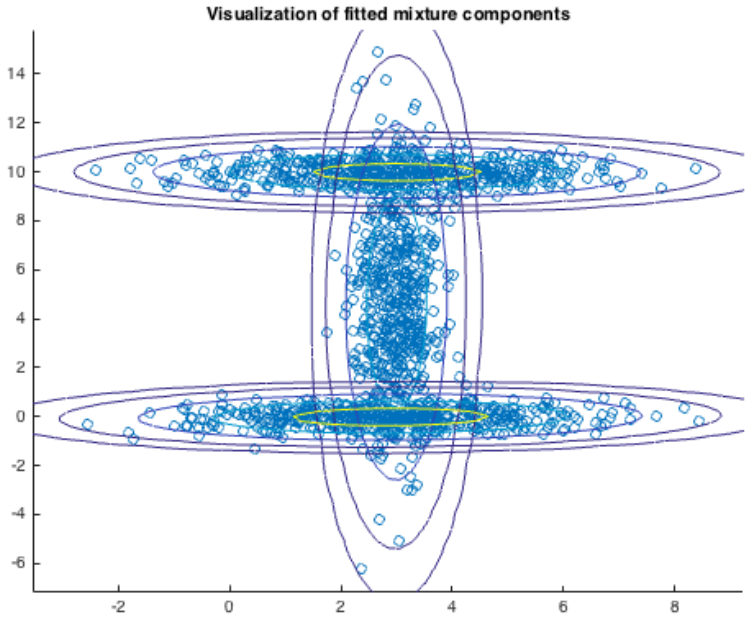
$$\mu_2 = [3.0055, 9.9904]$$

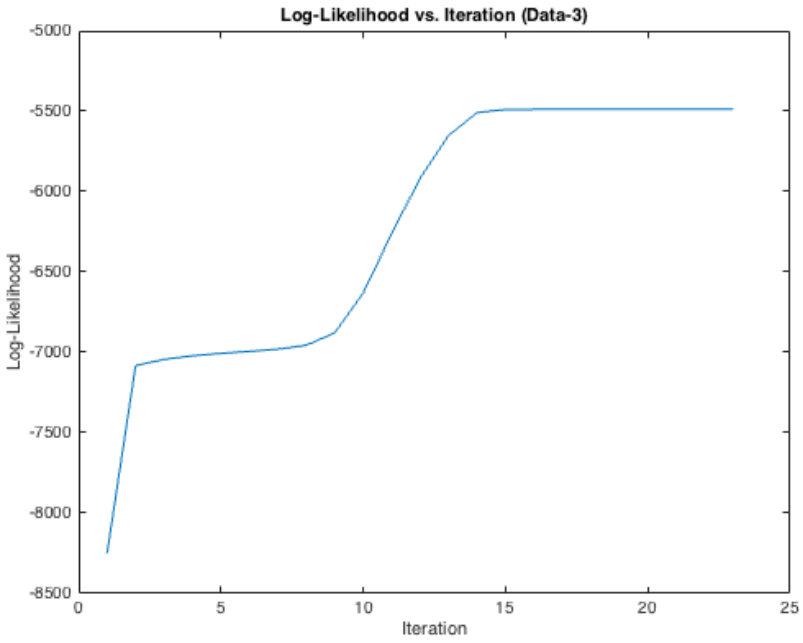
$$\mu_3 = [2.8880, -0.0166]$$

$$\sigma_1 = \begin{bmatrix} 0.1596 & 0.0271 \\ 0.0271 & 10.6048 \end{bmatrix}$$

$$\sigma_2 = \begin{bmatrix} 3.1492 & -0.0076 \\ -0.0076 & 0.1742 \end{bmatrix}$$

$$\sigma_3 = \begin{bmatrix} 3.2315 & 0.0392 \\ 0.0392 & 0.1327 \end{bmatrix}$$




Log-Likelihood vs. Iteration (Data-3)
-5000
-5500
-6000
-6500
-7000
-7500
-8000
-8500
Log-Likelihood
0
5
10
15
20
25
Iteration

2 Code

2.1 Code

```
function [ lik ] = GaussPDF_AllPairs( X, mu, sigma )
%Computes the multivariate Guassian pdf for every
% observation using each of the K components.
%
%   X:      The input observations (N x D)
%   mu:     Component means (K x D)
%   sigma:  Component covariance matrices (K, D, D)
%
%   lik:    Output pdf values (N x K)

    N = size(X, 1);
    K  = size(mu, 1);

    lik = zeros(N, K);
    for k = 1:K
        mu_k = mu(k,:);
        sigma_k = squeeze(sigma(k,:,:));

        lik(:, k) = mvnpdf(X, mu_k, sigma_k);
    end
end

function [ llik ] = llik_GMM( X, pi, mu, sigma )
%Compute the log-likelihood of the data given
% the current set of parameters
%
%   X:      Input observations (N x D)
%   pi:     Component probabilities (1 x K)
%   mu:     Component means (K x D)
%   sigma:  Component covariances (K x D x D)

    llik = sum(log(GaussPDF_AllPairs( X, mu, sigma ) *
        pi'));
end

function [ mu_best, sigma_best, pi_best,
   llik_trace_best ] = GMM_EM( X, K, trials, tol,
   minvar, covmul )
%Run the EM algorithm to fit a Gaussian mixture model.
%
%   X:      Input observations (N x D)
```

```
%   K:      Number of components to fit
%   trials: Number of random initializations to try
%   tol:    Convergence tolerance
%   minvar: Minimum allowed variance in any component
%   covmul: Constant to use for covariance init.

    %Default values for hyperparameters
    if nargin < 3
       trials = 20;
    end
    if nargin < 4
       tol = 1e-6;
    end
    if nargin < 5
        minvar = 1e-6;
    end
    if nargin < 6
        covmul = 1;
    end

    N = size(X, 1);
    D = size(X, 2);

    %Best seen values
    llik_best = -inf;
    mu_best = 0;
    sigma_best = 0;
    pi_best = 0;
    llik_trace_best = 0;

    %Run for a given number of random inits.
    for trial = 1:trials

        %Initialize parameters with ranomly selected
           obs.
        smplcov = covmul * cov(X);
        sigma = repmat(reshape(smplcov, 1, D, D), K,
           1, 1);
        mu = X(randsample(size(X,1), K), :);
        pi = ones(1,K) / K;

        llik_prev = -inf;
        llik = llik_GMM( X, pi, mu, sigma );

        llik_trace = llik;
```

```
        %Run until update are smaller than given
           tolerance
        while llik - llik_prev > tol
            llik_prev = llik;

            %E-Step
            likmat = GaussPDF_AllPairs( X, mu, sigma )
               ;
            likmat = bsxfun(@times, likmat, pi);
            resp = bsxfun(@rdivide, likmat, sum(likmat
               , 2));

            %M-Step
            Nk = sum(resp, 1);
            mu = bsxfun(@rdivide, (resp' * X), Nk');
            for k = 1:K
                X_mu = bsxfun(@minus, X, mu(k,:));
                sigma_k = bsxfun(@times, X_mu, resp(:,
                    k))' * X_mu;
                sigma_k = sigma_k / Nk(k);
                sigma_k(1:D+1:end) = max(diag(sigma_k)
                   , minvar);
                sigma(k, :, :) = sigma_k;
            end
            pi = Nk ./ N;

            llik = llik_GMM( X, pi, mu, sigma );
            llik_trace = [llik_trace, llik];
        end

        %Update best found results
        if llik > llik_best
            llik_best = llik;
            mu_best = mu;
            sigma_best = sigma;
            pi_best = pi;
            llik_trace_best = llik_trace;
        end
    end
end

function [ ] = plot_GMM( X, mu, sigma )
%Function to visualize mixture model after
% fitting with the EM algorithm
%
%   X:      Input data (N x D)
```

```
%   mu:     Component means (K x D)
%   sigma:  Component covariances (K x D x D)

    figure
    hold on

    %Plot the dataset
    scatter(X(:,1), X(:,2))

    %Setup plot boundaries
    maxes = max(X, [], 1) + 1;
    mins = min(X, [], 1) - 1;

    %Grid for contour plots
    x1 = mins(1):.2:maxes(1);
    x2 = mins(2):.2:maxes(2);
    [X1,X2] = meshgrid(x1,x2);

    %For each component distribution make a contour
    % plot of the Guassian pdf
    K = size(mu, 1);
    for k = 1:K
        F = mvnpdf([X1(:) X2(:)],mu(k,:), squeeze(
           sigma(k,:,:)));
        F = reshape(F,length(x2),length(x1));
        contour(x1,x2,F,[.0001 .001 .01 .05:.1:.95 .99
            .999 .9999]);
    end

    title('Visualization of fitted mixture components'
       )
    hold off
end

%Run the experiment for the 2-component dataset
load data2
[ mu, sigma, pi, llik_trace ] = GMM_EM( data, 2, 10);

figure
plot(llik_trace)
title('Log-Likelihood vs. Iteration (Data-2)')
xlabel('Iteration')
ylabel('Log-Likelihood')

plot_GMM(data, mu, sigma)
```

```
%Run the experiment for the 3-component dataset
load data3
[ mu, sigma, pi, llik_trace ] = GMM_EM( data, 3, 10);

figure
plot(llik_trace)
title('Log-Likelihood vs. Iteration (Data-3)')
xlabel('Iteration')
ylabel('Log-Likelihood')

plot_GMM(data, mu, sigma)
```