MATLAB CODE FILES

1. **main.m**

```
clear
load GMM_data.mat
loglik = 1:10;
O = 2;          %Number of coefficients in a vector
T = 1000;       %Number of vectors in a sequence
nex = 1;        %Number of sequences
data = reshape(X',O,T,nex);

for k = 1:10
M = k;          %Number of mixtures
Q = 1;          %Number of states
cov_type = 'spherical';

%data = randn(O,T,nex);
% initial guess of parameters
prior0 = normalise(rand(Q,1));
transmat0 = mk_stochastic(rand(Q,Q));

[mu0, Sigma0] = mixgauss_init(Q*M, data, cov_type);
mu0 = reshape(mu0, [O Q M]);
Sigma0 = reshape(Sigma0, [O O Q M]);
mixmat0 = mk_stochastic(rand(Q,M));

[model.LL, model.prior1, model.transmat1, model.mu1, model.Sigma1,
model.mixmat1] =     mhmm_em(data, prior0, transmat0, mu0, Sigma0,
mixmat0, 'max_iter', 10);


save(sprintf('model1a%d.mat',k) ,'model');

end
save ans1a1
```

2. **main1a2.m**

```
clear
load GMM_data.mat
loglik = 1:10;
for k=1:10
load(sprintf('model1a%d',k));
loglik(k) = GMM_llhood(model,Xho);
end
plot(1:10,loglik);
```

3. **main1a3.m**

```
clear
load GMM_data.mat
load model1a3.mat
O = 2;          %Number of coefficients in a vector
T = 1000;       %Number of vectors in a sequence
nex = 1;        %Number of sequences
data = reshape(X',O,T,nex);

M = 3;          %Number of mixtures
Q = 1;          %Number of states
cov_type = 'spherical';

prior0 = normalise(rand(Q,1));
transmat0 = mk_stochastic(rand(Q,Q));

[mu0, Sigma0] = mixgauss_init(Q*M, data, cov_type);
mu0 = reshape(mu0, [O Q M]);
Sigma0 = reshape(Sigma0, [O O Q M]);
mixmat0 = mk_stochastic(rand(Q,M));

[LL, prior, transmat, mu, Sigma, mixmat, loglik3, mu1, sigma1, mu2, sigma2,
mu3, sigma3] =  mhmm_em_dump(data, prior0, transmat0, mu0, Sigma0,
mixmat0, 'max_iter',10);

h1 = figure();
hold on
   contour_plot(mu1(:,1)',sigma1(:,:,1,1),'r');
   contour_plot(mu1(:,2)',sigma1(:,:,1,2),'g');
   contour_plot(mu1(:,3)',sigma1(:,:,1,3),'b');

h2 = figure();
hold on
   contour_plot(mu2(:,1)',sigma2(:,:,1,1),'r');
   contour_plot(mu2(:,2)',sigma2(:,:,1,2),'g');
```

```
    contour_plot(mu2(:,3)',sigma2(:,:,1,3),'b');

h3 = figure();
hold on
    contour_plot(mu3(:,1)',sigma3(:,:,1,1),'r');
    contour_plot(mu3(:,2)',sigma3(:,:,1,2),'g');
    contour_plot(mu3(:,3)',sigma3(:,:,1,3),'b');

h4 = figure();
plot(3:3:9,loglik3);
```

4. GMM_llhood.m

```
function llhood = GMM_llhood(model,X)

% Input  :  model - structure containing parameters of GMM
%           X - matrix containing data (one instance per row)
% Output :  llhood - loglikelihood of the data X given the model,
%                    i.e. log P(X|model)

% Please change the code below appropriately
O = 2;          %Number of coefficients in a vector
T = 1000;         %Number of vectors in a sequence
nex = 1;        %Number of sequences
data = reshape(X',O,T,nex);

llhood = mhmm_logprob(data, model.prior1, model.transmat1, model.mu1,
model.Sigma1, model.mixmat1);

end
```

5. contour_plot.m

```
function contour_plot(mu, sigma, linespec)
% Input  : mu - is the mean of a 2-D Gaussian.
%          sigma - is the 2x2 covariance matrix of the Gaussian.
%          linespec - use this to specify the color of the plot,
%          e.g. 'r', 'g', 'b', etc.

% Output : Contour plot for the Gaussian
% Tip    : Use "hold on;" and call this three times (one for each Gaussian).
x1range=0:0.1:4;
x2range=0:0.1:4;
[X1 X2] = meshgrid(x1range, x2range);

Z = zeros(length(x2range), length(x1range));
for n1 = 1:length(x2range)
   for n2 = 1:length(x1range)
      Z(n1,n2) = mvnpdf_cd([X1(n1,n2) X2(n1,n2)], mu, sigma);
   end
end
contour(X1, X2, Z, 2, linespec);
axis square;
```