CSC412 – Assignment #3 Solution

March 19, 2002

- Everybody did very well on this assignment.
- Some people did not have enough time to run all the 1000 runs to the required precision. The reason the code runs slow is because there are too many for loops. To make your code faster just vectorize as much as you can.

0.1 What to hand in

- The equation you are using to update the covariance matrices in the M-step. (5 marks)

$$\Sigma_k^{new} = \frac{\sum_n r_{nk}(\mathbf{x}^n - \mu_k^{new})(\mathbf{x}^n - \mu_k^{new})^T}{\sum_n r_{nk}}$$

- A plot of the best fit (highest likelihood) that you found for each number of components, showing the training data as points and the one-standard deviation ellipses of the Gaussians. (5 marks)
- A histogram, for each number of components, of the final log likelihoods of the training data and of the number of iterations of EM it took for training to converge. (7 marks)
  - You should plot the true log likelihoods, not the log likelihoods up to a constant.
- Your MATLAB code, properly commented. (35 marks)
  - The marks break down is as follows:
    * 5 marks – initialization.
    * 3 marks – loop until convergence.
    * 10 marks – E step.
    * 10 marks – M step.
    * 5 marks – handling of small variances.

– Here is a sample code for fitting mixtures of Gaussians.

```
function [lik,chi,mu,sigma] = mog(xx,K,tol);
% Mixture of Gaussians with maximum likelihood.
%
% Columns of xx are data vectors.  K is the number of clusters, and tol is
% the tolerance in the likelihood to determine convergence.
%
% Returns the likelihood, mixing proportions (chi), the means (columns of mu),
% and covariances (matrices in sigma).

if nargin <3, tol=1e-6; end;
if nargin <2, K=1; end;

iter=0; lik=-Inf;

% D is the number of dimensions, N number of data vectors.
[D,N] = size(xx);

% initialize chi, mu and sigma.
chi=ones(1,K)/K;
cc=cov(xx');
mu = sqrtm(cc)*randn(D,K);
sigma = repmat(diag(diag(cc)),[1,1,K]);

% Likelihoods for each mixture component.
ll=zeros(K,N);

% Some useful constants
debug = 0;
oneN = ones(1,N);
oneK = ones(1,K);
oneD = ones(1,D);
const = -D/2*log(2*pi);

% Fit parameters here

while ((iter<2) | (lik-oldlik)/abs(lik) >tol)

  % E-step

  for kk=1:K
    cc = chol(sigma(:,:,kk));
    delta = cc'\(xx-mu(:,kk*oneN));
    ll(kk,:) = log(chi(kk)) + const - sum(log(diag(cc))) - .5*sum(delta.^2,1);
  end

  ltot = logsum(ll,1);

  oldlik=lik;
  lik=sum(ltot);

  if debug>0,
    fprintf(1,'iter: %5d\t lik: %g \t diff:%g\n',iter,lik,lik-oldlik);
  end;
```

```
  rr = exp(ll-ltot(oneK,:));      % responsibilities
  sumrr = sum(rr,2);
  rr=rr./sumrr(:,oneN);


  % M-step

  chi = sumrr/N;
  mu = xx*rr';
  for kk=1:K
    delta = xx-mu(:,kk*oneN);
    sigma(:,:,kk) = delta*(delta.*rr(oneD*kk,:))'+1e-2*diag(diag(cc));
  end


  if debug>1,
    % Draw stuff
    clf;
    plot(xx(1,:),xx(2,:),'.');
    for kk=1:K
      plotGauss(mu(:,kk),sigma(:,:,kk));
    end
    drawnow;
  end;

  iter=iter+1;

end
```

– Here is a sample code for what you have to do.

```
load a3geyser

numtrials = 1000;
tol = 1e-6;
KK = [1 2 3 6];

liks      = zeros(length(KK),numtrials);
bestmu    = cell(length(KK),1);
bestsigma = cell(length(KK),1);
bestchi   = cell(length(KK),1);

% Fit mixtures
for ii=1:length(KK),
  K = KK(ii);
  bestlik = -Inf;
  for tt=1:numtrials,
    [lik chi mu sigma] = mog(xx,K,tol);
    liks(ii,tt) = lik;
    if lik>bestlik,
      bestchi{ii}   = chi;
      bestmu{ii}    = mu;
      bestsigma{ii} = sigma;
      bestlik       = lik;
    end;
    fprintf(1,'K%d tt%d  \r',ii,tt);
  end;
end;

% Plots histograms and mixtures
for ii=1:length(KK),
  subplot(4,2,ii*2-1);
  hist(liks(ii,:),50);
  title(['Distribution of likelihoods, K=' num2str(KK(ii))]);
  subplot(4,2,ii*2);
  plot(xx(1,:),xx(2,:),'.');
  for kk=1:KK(ii),
    plotGauss(bestmu{ii}(:,kk),bestsigma{ii}(:,:,kk));
  end;
  title(['Best Mixture, K=' num2str(KK(ii))]);
end;
```

– Here is a sample output.