Project-2

MAYUR GUPTA[1]

[1]Indian Institute of Technology Kanpur (IITK)

April 18, 2018

INTRODUCTION AND PROBLEM DESCRIPTION

Gaussian Mixture Model (GMM) is a probability density function, which can be expressed as a weighted sum of multivariate Gaussian components.GMMs, which can effectively model complex probability distributions, are widely used in such applications as data clustering, data classification, speaker recognition, or image segmentation and classification.The standard method for estimation of GMM parameters is the expectation-maximization (EM) algorithm .A finite mixture model p(x,Θ) can be expressed by a weighted sum of K>1 components:

$$p(\mathbf{x}|\Theta) = \sum_{m=1}^{K} \alpha_m p_m(\mathbf{x}|\theta_m), (1)$$

where α_m is mth mixing proportion and p_m is the probability density function of the mth component. In (1) θm is the set of parameters defining the mth component and Θ={θ1,θ2,...,θK,α1,α2,...,αK} is the complete set of the parameters needed to define the mixture. The functional form of pm is assumed to be known; Θ is unknown and has to be estimated. The mixing proportions must satisfy the following conditions: α_m>0, m=1,...,K and $\Sigma\alpha_m$=1. The number of components K is either known a priori or has to be determined during the mixture learning process. In our problem we know value of K.

In GMMs the probability density function of the mth component is given by:

$$p_m(\mathbf{x}|\theta_m) = \frac{1}{(2\pi)^{d/2}|\Sigma_m|^{1/2}} \exp(-\frac{1}{2}(\mathbf{x} - \mu_m)^T \Sigma_m^{-1}(\mathbf{x} - \mu_m)), (2)$$

where μ_m and Σ_m denote the mean vector and covariance matrix, respectively, |·| denotes a determinant of a matrix, T denotes transposition of a matrix, and d is the dimension of the feature space. The set parameters of the mth component is θ_m={μ_m,Σ_m}. Thus, for the GMM Θ is defined by: Θ={μ1,Σ1,...,μK,Σ_k,α1,...,αK}.

Estimation of the parameters of a GMM can be performed using the maximum likelihood approach. Given a training set of independent and identically distributed feature vectors X={x1,x2,...,xN}, where x^i=[$x_1{}^i$,...,$x_d{}^i$] in R^d, the loglikelihood corresponding to the K-component GMM is given by:

$$\log p(X|\Theta) = \log \prod_{i=1}^{N} p(\mathbf{x}^i|\Theta) = \sum_{i=1}^{N} \log \sum_{m=1}^{K} \alpha_m p_m(\mathbf{x}^i|\theta_m).$$

The maximum likelihood estimate of the parameters is given by: Θ_{ml}=argmax{log$p$(X|Θ)}.

In our problem, we have to model the data using mixture of two multivariate (6-variate) normal distributions.Therefore the value of K=2 and we have a vector X in R^6.We have been given Swiss-Bank data set on original(0) and fake(1) currency and after modeling the data using Gaussian mixture model, we have to answer whether we can use this model as a classification model in this case.I did the following method to conclude my results on this:

First using the EM Algorithm of multivariate gaussian mixture model, I have calculated the MLE's of the parameter space.(code has been attached in the appendix).Then using those values of $\Theta=\{\mu 1,\Sigma 1,\mu 2,\Sigma 2,\pi\}$, calculate the pdf value for each data set given and match it with their initial value(0 or 1).In this way by calculating the difference between these two values we will get the value of percentage error which will constitute our data analysis part.

EM ALGORITHM OF MULTIVARIATE MIXTURE MODEL

The most popular approach for maximizing log likelihood function is the EM algorithm. EM algorithm is an iterative optimization technique which is operated locally.It is an iterative algorithm, which, starting from initial guess of a parameters $\Theta^{(0)}$, generates a sequence of estimations $\Theta^{(1)}$,$\Theta^{(2)}$,...,$\Theta^{(j)}$,..., with increasing loglikelihood (i.e., $\log p(X|\Theta^{(j)}) > \log p(X|\Theta^{(j-1)})$. Each iteration j of the algorithm consists of two steps called expectation step (E-step) and maximization step (M-step) followed by a convergence check. For the GMMs these steps are defined as follows:

E-step: Given the set of mixture parameters $\Theta^{(j-1)}$ from the previous iteration, for each m=1,...,K and i=1,...,N, the expected value that a feature vector x^i was generated from mth component is computed as:

$h_m^{(j)}(\mathbf{x}^i) = \frac{\alpha_m^{(j)} p_m(\mathbf{x}^i|\theta_m^{(j-1)})}{\sum_{k=1}^K \alpha_k^{(j)} p_k(\mathbf{x}^i|\theta_k^{(j-1)})}$,

where $\theta_m{}^{(j-1)}$ and $\theta_k{}^{(j-1)}$ denote parameters of components m and k, in the iteration j-1, respectively.

M-step: Given the expected values of $h_m{}^{(j)}(x^i)$ obtained in the E-step the set of parameters Θ(j) is calculated as:

$\alpha_m^{(j)} = \frac{1}{N}\sum_{i=1}^{N} h_m^{(j)}(\mathbf{x}^i)$

$\mu_m^{(j)} = \frac{\sum_{i=1}^N h_m^{(j)}(\mathbf{x}^i) * \mathbf{x}^i}{\sum_{i=1}^N h_m^{(j)}(\mathbf{x}^i)}$

$\Sigma_m^{(j)} = \frac{\sum_{i=1}^N h_m^{(j)}(\mathbf{x}^i)(\mathbf{x}^i-\mu_m^{(j)})(\mathbf{x}^i-\mu_m^{(j)})^T}{\sum_{i=1}^N h_m^{(j)}(\mathbf{x}^i)}$

Convergence check: The loglikelihood $\log p(X|\Theta^{(j)})$ is computed according to likelihood function. The algorithm is terminated if the following convergence criterion is met.

$\frac{\log p(X|\Theta^{(j)}) - \log p(X|\Theta^{(j-1)})}{\log p(X|\Theta(j))} < \epsilon$,

where [?]1 is a user defined termination threshold. If the convergence criterion is not met algorithm proceeds to Step 1.

For our question we have value of k=2 i.e. we have two components of gaussian mixture.The above generalized method for K components of gaussian mixture can be used to derive the formula for two component model.

DATA ANALYSIS

In data analysis, i will analyze the results on the basis of the values of the parameter space i.e. Θ={µ1,Σ1,µ2,Σ2,π} which i have calculated using the EM algorithm.(The code has been attached in the appendix for the reference of algorithm).

First I initialized the means and the covariance of the components using k-means and giving input of initial values.Then i have calculated the expected value using the E-step formula and then using the M-step formula, i have calculated the value of Θ={µ1,Σ1,µ2,Σ2,π} for the next stage.

Then giving the convergence condition, i have formulated the while loop which will rum till the convergence of these values does not take place.Once the convergence take place, then we will get the final values of Θ={µ1,Σ1,µ2,Σ2,π} using EM-Algorithm.

INITIALISATION:

My initial values that are used by me for the initialisation step are as follows:

pi= [0.5 , 0.5]

mu1= [214 , 130 , 130.5 , 8.1 , 9.7 , 140.5]

mu2= [214 , 130 , 130.5 , 8.1 , 9.7 , 140.5]

sigma1= [1, 1, 1, 1, 1, 1 ; 1, 1, 1, 1, 1, 1 ; 1, 1, 1, 1, 1, 1 ; 1, 1, 1, 1, 1, 1 ; 1, 1, 1, 1, 1, 1 ; 1, 1, 1, 1, 1, 1]
(6 by 6 matrix)

sigma2= [1, 1, 1, 1, 1, 1 ; 1, 1, 1, 1, 1, 1 ; 1, 1, 1, 1, 1, 1 ; 1, 1, 1, 1, 1, 1 ; 1, 1, 1, 1, 1, 1 ; 1, 1, 1, 1, 1, 1]
(6 by 6 matrix)

CLASSIFICATION METHOD and RESULTS

In the classification step, based on the final values of parameter space Θ={µ1,Σ1,µ2,Σ2,π}, I have calculated the expected values of all the 200 data points and classified them as fake currency (if the value is close to 1) and original currency (if the value is close to 0).Also by comparing this data with the original data we can calculate the number of data points which show a difference in their original status of currency and see how precise is the EM-Algorithm.

RUNNING E-STEP AND M-STEP ALGORITHM:

After running the EM-Algorithm code, the final values of parameter space that i got are as follows:

```
>> sigma1
sigma1 =

   0.145028   0.030963   0.020639  -0.116367  -0.016361   0.095671
   0.030963   0.129696   0.104282   0.205531   0.099061  -0.196448
   0.020639   0.104282   0.158897   0.271320   0.126035  -0.225564
  -0.116367   0.205531   0.271320   2.153845   0.123016  -1.049045
  -0.016361   0.099061   0.126035   0.123016   0.647815  -0.531413
   0.095671  -0.196448  -0.225564  -1.049045  -0.531413   1.328164

>> sigma2
sigma2 =

   0.145063   0.030978   0.020655  -0.116365  -0.016361   0.095695
   0.030978   0.129711   0.104296   0.205548   0.099068  -0.196451
   0.020655   0.104296   0.158915   0.271342   0.126043  -0.225567
  -0.116365   0.205548   0.271342   2.153980   0.123023  -1.049104
  -0.016361   0.099068   0.126043   0.123023   0.647855  -0.531445
   0.095695  -0.196451  -0.225567  -1.049104  -0.531445   1.328258
```

Figure 1: **Values of sigma1 and sigma2**

```
>> p
p =  0.50002
>> mean1
mean1 =

   214.8981
   130.1274
   129.9581
     9.4247
    10.6496
   140.4917

>> mean2
mean2 =

   214.8929
   130.1247
   129.9552
     9.4230
    10.6493
   140.4882
```

Figure 2: **Value of** π *and mean values*

CONCLUSION

Using the EM-Algorithm, I have calculated the value of MLE's of Θ={µ1,Σ1,µ2,Σ2,π} for 200 multivariate data points whose pdf was a mixture of two gaussian distribution(one with probability density pi and other with probability density (1-pi). Also after calculating the expected value of each of these data point, I have classified them as a fake currency(1) or the original currency(0).

APPENDIX

```
load('swiss-bank.dat');

X = Data'; % 200x6 data set
D = size(X,2); % dimension
N = size(X,1); % number of samples
K = 2; % number of Gaussian Mixture components

% Initialization
p = [0.5, 0.5]; % arbitrary pi
[idx,mu] = kmeans(X,K); % initial means of the components

sigma = zeros(D,D,K);
fork = 1:K
  sigma(:,:,k) = cov(X(idx==k,:));
end

converged = 0;
prevLogl = Inf;
prevMu = mu;
prevSigma = sigma;
prevPi = p;
r = 0;
while(converged ~= 1)
  r = r +1
  gm = zeros(K,N); % gaussian component in the nominator
  sumGM = zeros(N,1); % denominator
  % E-step: Evaluate using the current parameters
  % compute the nominator and denominator
  fork = 1:K
      fori = 1:N
```

```
            Xmu = X-mu;
            % Using log
            logPdf = log(1/sqrt(det(sigma(:,:,k))*(2*pi)^D)) + (-0.5*Xmu*(sigma(:,:,k)\Xmu'));
            gm(k,i) = log(p(k)) * logPdf;
            sumGM(i) = sumGM(i) + gm(k,i);
        end
    end

    % calculation
    res = zeros(K,N);
    Nk = zeros(4,1);
    fork = 1:K
        fori = 1:N
            res(k,i) = gm(k,i)/sumGM(i);
        end
        Nk(k) = sum(res(k,:));
    end
     % M-step: Re-estimate the parameters
    fork = 1:K
        fori = 1:N
            mu(k,:) = mu(k,:) + res(k,i).*X(k,:);
            sigma(:,:,k) = sigma(:,:,k) + res(k,i).*(X(k,:)-mu(k,:))*(X(k,:)-mu(k,:))';
        end
        mu(k,:) = mu(k,:)./Nk(k);
        sigma(:,:,k) = sigma(:,:,k)./Nk(k);
        p(k) = Nk(k)/N;
    end
% Evaluate the log-likelihood and check for convergence of either
% the parameters or the log-likelihood. If not converged, go to E-step.
    logl = 0;
    fori = 1:N
        logl = logl + log(sum(gm(:,i)));
    end
```

```
    % Check for convergence of parameters
    errorLogl = abs(logl-prevLogl);
    if(errorLogl <= eps)
        converged = 1;
    end

    errorMu = abs(mu(:)-prevMu(:));
    errorSigma = abs(sigma(:)-prevSigma(:));
    errorPi = abs(p(:)-prevPi(:));

    if(all(errorMu <= eps) && all(errorSigma <= eps) && all(errorPi <= eps))
        converged = 1;
    end

    prevLogl= logl;
    prevMu = mu;
    prevSigma = sigma;
    prevPi = p;

end% while
```

.