Imperial College
London

Coursework 1: Expectation Maximization and Gaussian Mixture Models

Imperial College London

Department of Computing

495 Advanced Statistical Machine Learning and Pattern Recognition

Author:
Thomas Teh (CID: 0124 3008)

Date: February 2, 2017

1 Exercise I: Implementation of GMMs

1.1 Part a

A Gaussian distribution has an isotropic covariance means its covariance is proportional to the unit covariance, $\lambda \boldsymbol{I}$. This implies the covariance will be spherical in the variable space, which in our case (which is a 2-dimensional data), the data points should cluster and form a circle. Given that we can infer that there are 4 Gaussians, we can group the data points into 4 different clusters in yellow and red as in Figure 2 via visual inspection:

- **Yellow areas**: For theses clusters, we can see that the data points are roughly uniformly spread around the center of the yellow regions, forming a circle. Hence, it is reasonable to infer that these data points have Gaussian distributions with isotropic covariance.
- **Red areas**: For these cluster, the data points are do not spread uniformly around a center. Instead, the data points have more variance in one of the dimensions. Hence, for these two clusters, we can infer that they have Gaussian distributions with anisotropic covariance.

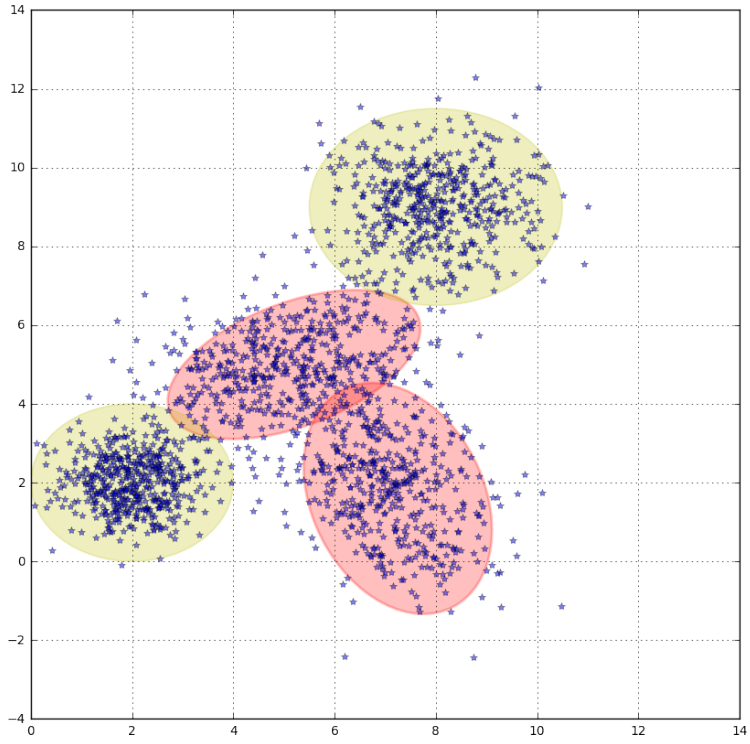

Figure 1: Covariance structure of the dataset: The yellow areas are circular while the red areas are ellipses. The colored areas are determined merely by eyeballing the data without actually imposing any model on it.

1.2 Part b

1. **Random Initialization**: The original function `random_params_generator(K)` in the Python notebook cannot be applied for several reasons.

    - The function cannot generate random covariance matrices. Covariance matrices needs to be symmetric and positive semi-definite. By generating a random array of shape (D,D) from the uniform random number generator in Numpy will not work when we input it into the multivariate normal pdf function.

    - Even if we manage to impose the structure for the covariance matrices, we have an issue of scale. We see that the variables ranges from -2 to 14, which the random number generator generates from a uniform distribution $(0, 1)$. In order to use this function, we would need to whiten our data set.

    - The mixture coefficients are not normalized. Ideally, the sum of the mixture coefficients must be 1. However, this issue does not severely affect the optimization process.

2. **Random Sampling For Initialization**: I wrote the function `EMInitRandom(Data, nbCluster, subsampling)` in order to do random initialization.

    - Instead of generating random numbers for the parameter values, we do random sampling and estimate the parameters using the random sample from the data set. For each mixture component, we will do the random sampling and estimation.

    - Effectively, we can solve the shortcomings of the previous functions: (1) we can work with data that are not whiten as scale will not be an issue, (2) By default, the estimation will have all the properties of a covariance matrix.

3. **K-means vs Random Initialization**: When we used the 2nd method above to initialize the parameters, we find that the algorithm converges much slower relative to k-means initialization. The reason is that the K-means algorithm has already determined the centroids of the clusters. What the GMM does differently from the K-means is merely assigning the data point the probability to which Gaussian distribution it may belong to. In short, K-means algorithm is a hard classification whereas the GMM is a soft classification.

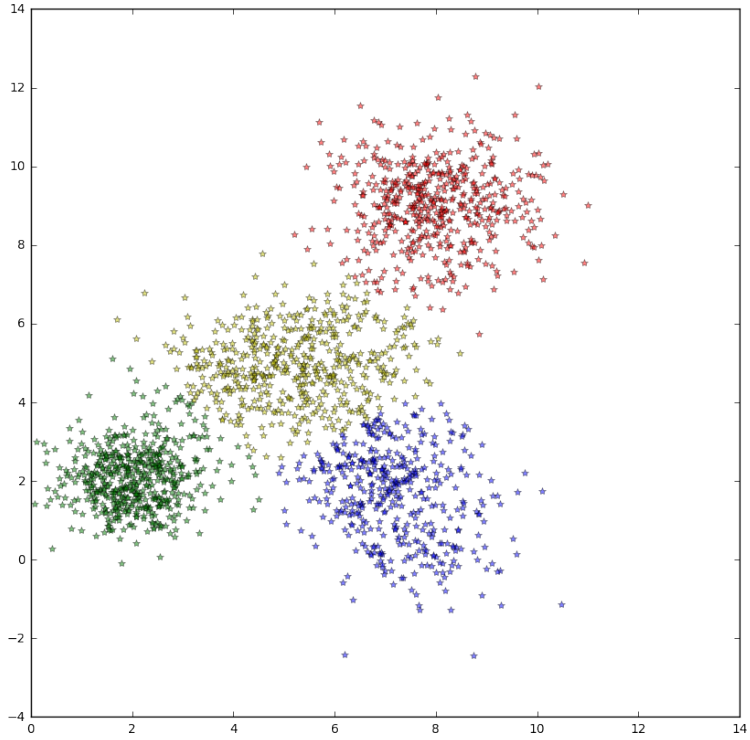

Figure 2: Clustering using K-Means algorithm.

4. **Comment**: From the GMM, we find that indeed that two of the Gaussians have covariances which are close to isotropic while the other two Gaussians' covariances are anisotropic. Please refer to the results from a run of step 10 using the K-means initialization on the next page.

$$\mu_1 = \begin{bmatrix} 7.9763470 & 9.0751523 \end{bmatrix}$$
$$\mu_2 = \begin{bmatrix} 7.0258555 & 1.9268015 \end{bmatrix}$$
$$\mu_3 = \begin{bmatrix} 5.0051005 & 5.0372016 \end{bmatrix}$$
$$\mu_4 = \begin{bmatrix} 1.98810413 & 2.0064038 \end{bmatrix}$$

$$\Sigma_1 = \begin{bmatrix} 1.0339488 & -0.0077491 \\ -0.0077491 & 0.9729311 \end{bmatrix}$$
$$\Sigma_2 = \begin{bmatrix} 1.0787892 & -0.5235914 \\ -0.5235914 & 1.9189258 \end{bmatrix}$$
$$\Sigma_3 = \begin{bmatrix} 1.8969002 & 0.5102359 \\ 0.5102359 & 0.9838895 \end{bmatrix}$$
$$\Sigma_4 = \begin{bmatrix} 0.4927645 & 0.0065982 \\ 0.0065982 & 0.4585648 \end{bmatrix}$$

$$\pi_1 = 0.249784$$
$$\pi_2 = 0.248498$$
$$\pi_3 = 0.250999$$
$$\pi_4 = 0.250720$$

2 Exercise II

2.1 Formulation

Let $\boldsymbol{z}$ be a K-dimensional binary random variable with 1-of-K representation.

$$p(z_k = 1) = \pi_k \Rightarrow p(\boldsymbol{z}) = \prod_{i=1}^{K} \pi_k^{z_k}$$

Conditional probability of $\boldsymbol{x}$ given a particular value for latent variable $\boldsymbol{z}$:

$$p(\boldsymbol{x}|\boldsymbol{z}) = \prod_{k=1}^{K} \mathcal{N}(\boldsymbol{x}|\boldsymbol{\mu}_k, \Sigma)^{z_k}$$

Using Bayes theorem and marginalize the latent variable $\boldsymbol{z}$

$$p(\boldsymbol{x}) = \sum_{\boldsymbol{z}} p(\boldsymbol{z}) p(\boldsymbol{x}|\boldsymbol{z}) = \sum_{k=1}^{K} \pi_k \mathcal{N}(\boldsymbol{x}|\boldsymbol{\mu}_k, \Sigma)$$

Similarly, the posterior probability of $\boldsymbol{z}$ is given by

$$\gamma(z_k) = \frac{p(\boldsymbol{x}|\boldsymbol{z}) p(\boldsymbol{z})}{p(\boldsymbol{x})} = \frac{\pi_k \mathcal{N}(\boldsymbol{x}|\boldsymbol{\mu}_k, \Sigma)}{\sum_{j=1}^{K} \pi_j \mathcal{N}(\boldsymbol{x}|\boldsymbol{\mu}_j, \Sigma)}$$

2.2 Expectation Maximization

1. Writing down the log-likelihood for the joint distribution:

$$p(\boldsymbol{X}, \boldsymbol{Z}|\boldsymbol{\mu}, \Sigma, \boldsymbol{\pi}) = \prod_{n=1}^{N} \prod_{k=1}^{K} \pi_k^{z_{nk}} \mathcal{N}(\boldsymbol{x}_n|\boldsymbol{\mu}_k, \Sigma)^{z_{nk}}$$

$$\ln p(\boldsymbol{X}, \boldsymbol{Z}|\boldsymbol{\mu}, \Sigma, \boldsymbol{\pi}) = \sum_{n=1}^{N} \sum_{k=1}^{K} z_{nk} \Big(\ln \pi_k + \ln \mathcal{N}(\boldsymbol{x}_n|\boldsymbol{\mu}_k, \Sigma) \Big)$$

2. Taking the expectation on the log likelihood (the E-step):

$$\begin{aligned} G(\boldsymbol{\theta}) &= \mathbb{E}_{p(\boldsymbol{Z}|\boldsymbol{X},\boldsymbol{\theta})} [\ln p(\boldsymbol{X}, \boldsymbol{Z}|\boldsymbol{\mu}, \Sigma, \boldsymbol{\pi})] \\ &= \sum_{n=1}^{N} \sum_{k=1}^{K} \mathbb{E}_{p(\boldsymbol{Z}|\boldsymbol{X},\boldsymbol{\theta})}[z_{nk}] \Big(\ln \pi_k + \ln \mathcal{N}(\boldsymbol{x}_n|\boldsymbol{\mu}_k, \Sigma) \Big) \\ &= \sum_{n=1}^{N} \sum_{k=1}^{K} \gamma(z_{nk}) \Big(\ln \pi_k - \frac{F}{2} \ln 2\pi - \frac{1}{2} \ln |\Sigma| - \frac{1}{2} (\boldsymbol{x_n} - \boldsymbol{\mu_k})^\top \Sigma^{-1} (\boldsymbol{x_n} - \boldsymbol{\mu_k}) \Big) \end{aligned}$$

3. Solving the following optimization problem

$$\max_{\boldsymbol{\theta}} G(\boldsymbol{\theta}) - \lambda\left(\sum_{k=1}^{K}\pi_k - 1\right)$$
$$s.t. \sum_{k=1}^{K}\pi_k = 1$$

Let

$$L(\boldsymbol{\theta}) = G(\boldsymbol{\theta}) - \lambda\left(\sum_{k=1}^{K}\pi_k - 1\right)$$

(a) Solving for $\boldsymbol{\mu_k}$. We take the derivative of $L(\boldsymbol{\theta})$ against $\boldsymbol{\mu_k}$

$$\frac{\partial L(\boldsymbol{\theta})}{\partial \boldsymbol{\mu_k}} = -\frac{1}{2}\sum_{n=1}^{N} 2\gamma(z_{nk})\boldsymbol{\Sigma^{-1}}(\boldsymbol{x_n} - \boldsymbol{\mu_k}) = 0$$
$$\sum_{n=1}^{N}\gamma(z_{nk})\boldsymbol{\Sigma^{-1}}(\boldsymbol{x_n} - \boldsymbol{\mu_k}) = 0$$
$$\sum_{n=1}^{N}\gamma(z_{nk})\boldsymbol{x_n} = \boldsymbol{\mu_k}\sum_{n=1}^{N}\gamma(z_{nk})$$

Rearranging the terms

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

(b) Solving for Σ. We take the derivative of $L(\boldsymbol{\theta})$ against Σ

$$\frac{\partial L(\boldsymbol{\theta})}{\partial \Sigma} = \sum_{n=1}^{N}\sum_{k=1}^{K}\gamma(z_{nk})\left(-\frac{1}{2}\boldsymbol{\Sigma^{-1}} + \frac{1}{2}\boldsymbol{\Sigma^{-1}}(\boldsymbol{x_n} - \boldsymbol{\mu}_k)(\boldsymbol{x_n} - \boldsymbol{\mu}_k)^\top\boldsymbol{\Sigma^{-1}}\right) = 0$$
$$\sum_{n=1}^{N}\sum_{k=1}^{K}\gamma(z_{nk})\left(\mathbf{1} - (\boldsymbol{x_n} - \boldsymbol{\mu}_k)(\boldsymbol{x_n} - \boldsymbol{\mu}_k)^\top\boldsymbol{\Sigma^{-1}}\right) = 0$$

$$\sum_{n=1}^{N}\sum_{k=1}^{K}\gamma(z_{nk})\mathbf{1} = \sum_{n=1}^{N}\sum_{k=1}^{K}\gamma(z_{nk})(\boldsymbol{x_n} - \boldsymbol{\mu}_k)(\boldsymbol{x_n} - \boldsymbol{\mu}_k)^\top\boldsymbol{\Sigma^{-1}}$$
$$\Sigma = \frac{1}{N}\sum_{n=1}^{N}\sum_{k=1}^{K}\gamma(z_{nk})(\boldsymbol{x_n} - \boldsymbol{\mu}_k)(\boldsymbol{x_n} - \boldsymbol{\mu}_k)^\top$$

(c) Solving for π_k. We take the derivative of $L(\boldsymbol{\theta})$ against π_k and use the fact that $\sum_{k=1}^{K} \pi_k = 1$

$$\frac{\partial L(\boldsymbol{\theta})}{\partial \pi_k} = \sum_{n=1}^{N} \frac{\gamma(z_{nk})}{\pi_k} - \lambda = 0$$

$$\pi_k = \frac{1}{\lambda} \sum_{n=1}^{N} \gamma(z_{nk})$$

$$\sum_{k}^{K} \pi_k = \frac{1}{\lambda} \sum_{n=1}^{N} \sum_{k}^{K} \gamma(z_{nk})$$

$$\lambda = N$$

$$\pi_k = \frac{1}{N} \sum_{n=1}^{N} \gamma(z_{nk})$$