PRML Assignment 3

Instruction for Source Code

The gmm model and data generator is stored in `handout/ gmm.py`. To run code, you can:

```
python source.py --d=2 --n=3000 --k=3 --mean='1,1,3,3,0,4' --
cov='1,0,0,1,0.5,0,0,0.5,0.8,0,0,0.8' --iter=60 --pr='0.5,0.2,0.3'
```

- **d**: dimension of data
- **n**: size of datasize
- **k**: number of clusters
- **mean**: if you want to assign `clutser A` with 3-d mean (v_A^0, v_A^1, v_A^2), `cluster B` with 3-d mean (v_B^0, v_B^1, v_B^2), `cluster C` with 3-d mean (v_C^0, v_C^1, v_C^2), then the input shoud be ' $v_A^0, v_A^1, v_A^2, v_B^0, v_B^1, v_B^2, v_C^0, v_C^1, v_C^2$ '
- **cov**:f you want to assign `clutser A` with 3x3 cov $[[a_{00}, a_{01}, a_{02}], [a_{10}, a_{11}, a_{12}], [a_{20}, a_{21}, a_{22}]]$, `cluster B` with 3x3 cov $[[b_{00}, b_{01}, b_{02}], [b_{10}, b_{11}, b_{12}], [b_{20}, b_{21}, b_{22}]]$, `cluster C` with 3x3 cov $[c_{00}, c_{01}, c_{02}], [c_{10}, c_{11}, c_{12}], [c_{20}, c_{21}, c_{22}]]$, then the input should be 'a_{00}, a_{01}, a_{02},a_{10},a_{11},a_{12}, a_{20}, a_{21}, a_{22}, b_{00}, b_{01}, b_{02}, b_{10}, b_{11},b_{12}, b_{20}, b_{21}, b_{22}, c_{00}, c_{01}, c_{02}, c_{10}, c_{11}, c_{12}, c_{20}, c_{21}, c_{22}'
- **iter**: number of iterations EM takes
- **pr**: the propotion of data in each cluster

Model Description

Data Generator

GMM model is generated by k multi-variant normal distributions (gaussian distribution). The data generator is designed with $k * d$ input of mean and $k * d * d$ input of covariance matrix.

The input also includes the total amount of data, N. The devision proportion of each cluster of data, p. According to p($N_1 : N_2 :\ldots: N_k = p_1 : p_2 :\ldots: p_k$), each cluster group will be assigned with N_i number of data($i = 1, 2, \ldots, k$), which sampled from gaussian distribution G_i, center of which is μ_i and covariance of which is Σ_i.

GMM with EM Solution

The EM solution to GMM in class fits with one-variance distribution. The EM solution is expanded to multi-variant model as belows.

Maximize Expectation

GMM model is defined as :

$$p(x) = \sum_{i=1}^{K} \pi_i N(x|\mu_i . \Sigma_i)$$

To maximize Expection:

$$\begin{aligned}\log p(X;\theta) &= \log \sum_z q(z) \frac{p(x,z;\theta)}{q(z)} \\ &\geq \sum_z q(z) \log \frac{p(x,z;\theta)}{q(z)} = ELBO(q,x;\theta)\end{aligned}$$

Using EM to solve GMM:

1. **E**

   Using fixed parameters μ, Σ, q, counting posterior distribution $p(z^{(n)}|x^{(n)})$

$$\gamma_{n,k} = \frac{\pi_k N(X_n|\mu_k, \Sigma_k)}{\sum_{j=1}^{K} \pi_j N(x_n|\mu_j, \Sigma_j)}$$

2. **M**

   set $q(z=i) = \gamma_{n,k}$, then evidence lower bound of Data is:

$$ELBO(\gamma, X|\mu, \Sigma, \pi) = \sum_{n=1}^{N} \sum_{k=1}^{K} \gamma_{n,k} (\ln \pi_k + \ln N(x_n|\mu_k, \Sigma_k)) + C$$

$$N(x_n|\mu_k, \Sigma_k)) = \frac{1}{\sqrt{(2\pi)^d |\Sigma|}} \cdot \exp\{-\frac{1}{2}(x-\mu)^T \Sigma^{-1} (x-\mu)\}$$

   By maximize ELBO, we can get the parameter for next iteration:

$$\begin{aligned}N_k &= \sum_{n=1}^{N} \gamma(n,k) \\ \pi_k &= \frac{N_k}{N} \\ \mu_k &= \frac{1}{N_k} \sum_{n=1}^{N} \gamma(n,k) x_n \\ \Sigma_k &= \frac{1}{N_k} \sum_{n=1} N(x_n - \mu_k)(x_n - \mu_k)^T\end{aligned}$$

In python code, π_k is stored in array `pz`, and $\gamma_{n,k}$ is stored in array px

Initialize EM

To initialize EM, two aspects need to be considered:

a. How to decide K

- A Naive method is to try different K, run EM algorithm and find the K which produce lowest `bayesian information criterion` to be the final K.(ref)
- Since deciding K is trivial in this task, I didn't implement it in my code.

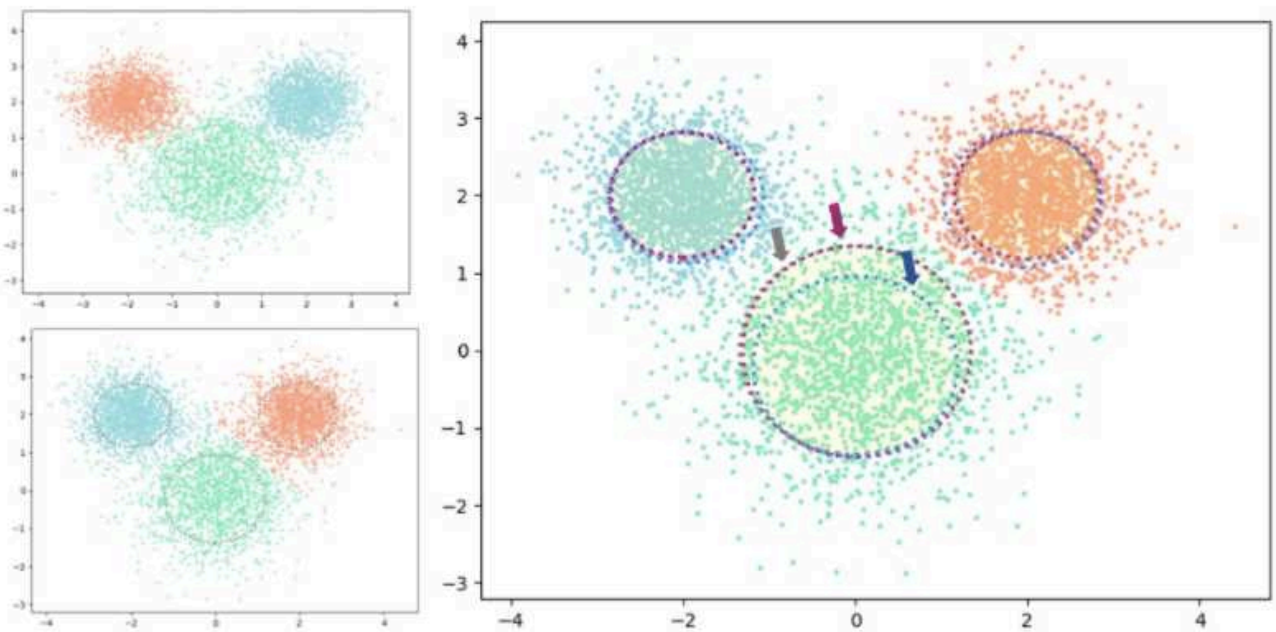
b. How to decide initial π_k , $\mu_k$and Σ_k

- π_k is defined as $\frac{1}{K}$ at start
- μ_k , $Sigma_k$has different initialization methods, from which I chose 2:
  - chose K arbitrary nodes from whole data set, setting their values as μ_k, initialize each Σ_k as I
  - run `K-means`, use the result centers as μ_k , use the covariance of clustered data groups as Σ_k

Model Performance

Data A

5000 data, unifomly sampled as A, B, C, with variance as $[0.8, 0.8]$, $[0.3, 0.3]$, $[0.3, 0.3]$



The upper figure in the left is the original data, the lower figure in the left is the result from K-means. On the right, purple arrow shows the result of EM, grey arrow shows the original data, blue arrow shows the result of K-means

EM shows better result than K-means, matching with the original data to a great extent.

Data B

Skewed 'Ellipse-shaped' Data

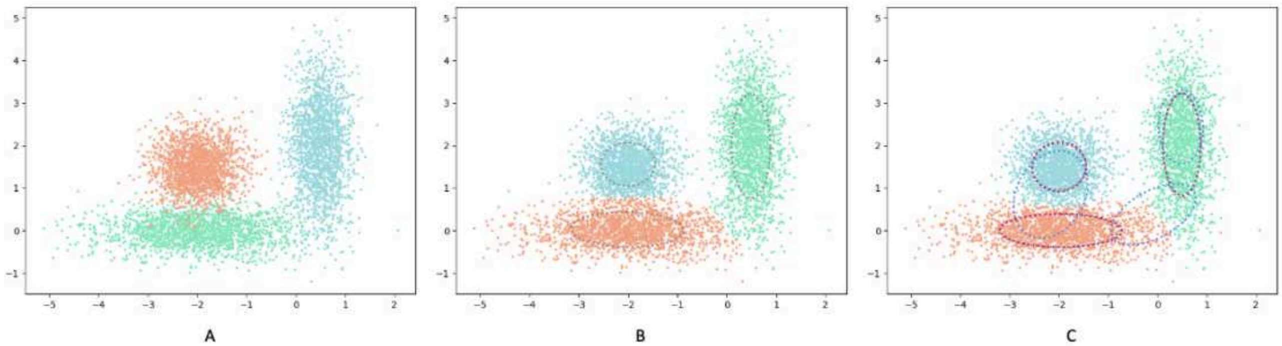


Figure A is the original data , while B is the result of K-means, C is the result of EM

Juding from the blue circule(k-means result in initialization) in `figure C`, we know k-means can be unstable when data is not in `circle` shape. However, EM can reduces this trouble.

Data C

Data with non-zero covariance

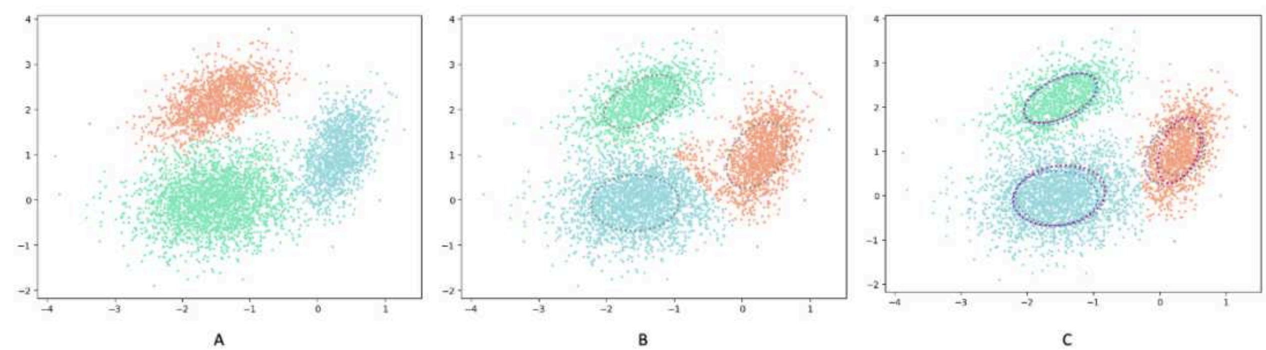


Figure A is the original data , while B is the result of K-means, C is the result of EM

K-means can't use the information of probability distribution. However, EM overcomes this shortcoming. It gave better results under this circumstance.

Data D

Data with overlaps

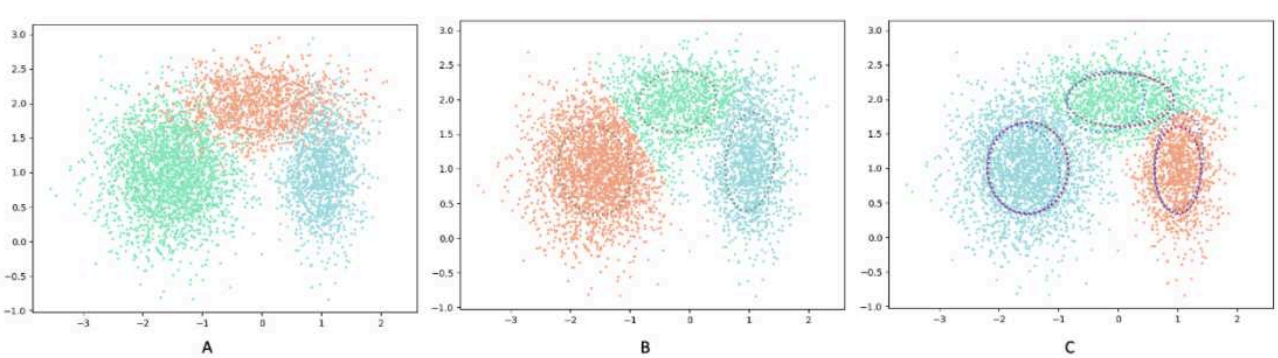


Figure A is the original data , while B is the result of K-means, C is the result of EM

EM outperforms K-means when dealing with the margin of overlapping data.

Data E

Overlapping Data with same centers

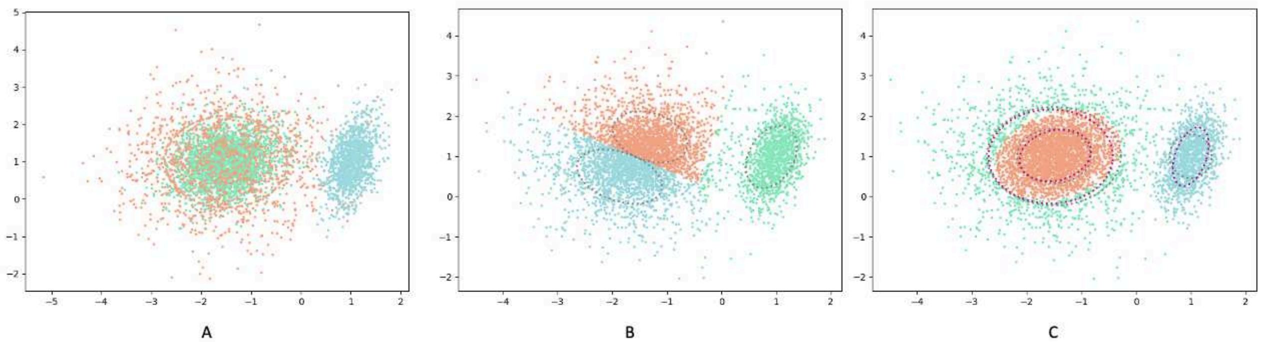


Figure A is the original data , while B is the result of K-means, C is the result of EM

While It's hard for EM to distinguish overlapping data, It can make good use of probability distribution, therefore, its prediction of data center and covariance obviously outperforms K-means.

Initialization

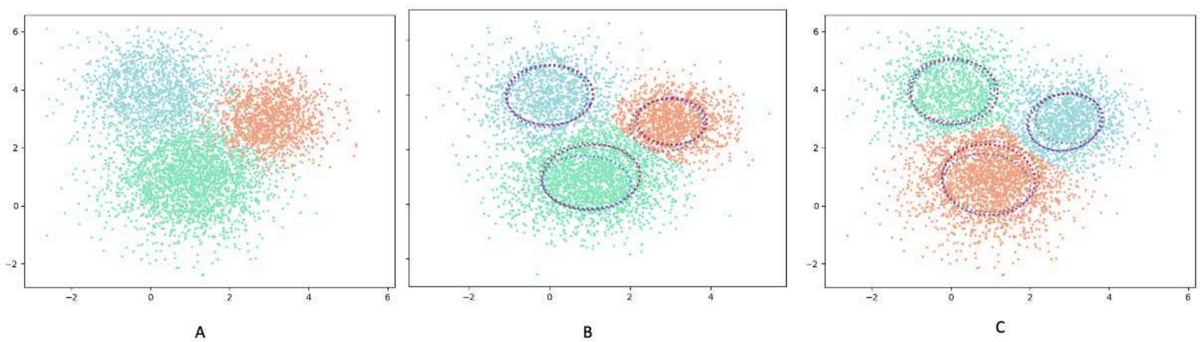


Figure A is the original data , while B is the result of EM with K-means initialization, C is the result of EM with arbitary-data initialization

The two methods don't have significant difference, however, K-means initialization can help to produce faster convergence.

Conclusion and Discussion

EM shows good result in culstering task for GMM model. Utilizing probability distribution, it outperforms K-means method in dealing with the margin of data group. However, when data has large overlaps, EM can't help to distinguish.