Dirichlet Process Mixture Model

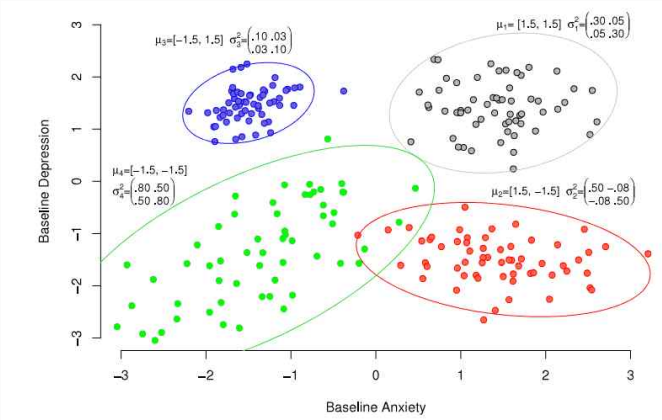
ESC 2024 - 1 파이널 프로젝트
이상윤 전제훈 정석훈 조준태 황보유민

Contents

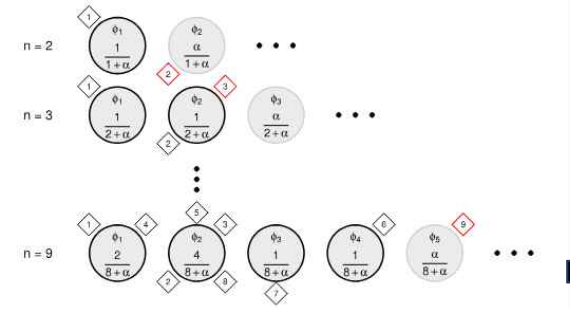
1

Infinite number of clusters

Dataset


μ3=[−1.5, 1.5] σ23=(.10 .03 .03 .10)
μ1= [1.5, 1.5] σ21=(.30 .05 .05 .30)
μ4=[−1.5, −1.5]
σ24=(.80 .50 .50 .80)
μ2=[1.5, −1.5] σ22=(.50 −.08 −.08 .50)
Baseline Depression
Baseline Anxiety
3
2
1
0
−1
−2
−3

(Review) Chinese Restaurant Process



Process

(테이블 : 클러스터, 손님 : 데이터)

- 새로 들어온 손님은 일정 확률에 따라 다른 손님이 있는 테이블에 같이 앉거나 새 테이블에 앉을 수 있다.
- 다른 손님이 있는 테이블에 앉을 확률은 그 테이블에 앉아있는 손님의 수에 비례한다.
- 테이블은 무한히 많다고 가정한다.

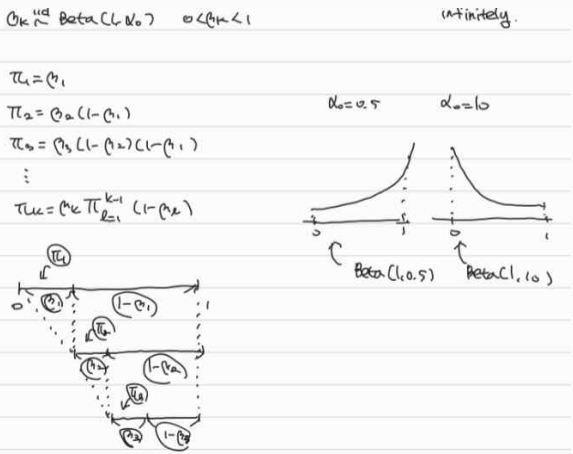
클러스터의 갯수를 정해놓지 않고도 클러스터링 가능!

기존 클러스터로 : $p(c_i|\mathbf{c}_{-i},\alpha) = \frac{n_{-i,k}}{n-1+\alpha}$

새로운 클러스터로 : $p(c_i \neq c_k \forall k \neq i|\mathbf{c}_{-i},\alpha) = \frac{\alpha}{n-1+\alpha}$

c_i : i th observation이 몇 번째 cluster에 속하는지를 나타내는 indicator variable

(Review) Stick Breaking Process



* hyperparameter α가 클수록 클러스터 생성할 확률 높음.

2

DPMM

Dirichlet Process Mixture Model

Prior & Posterior Predictive Distribution

1. Prior Predictive Distribution

$$\tilde{y}_i \sim N(\mu_0, \sigma_0^2 + \sigma_y^2)$$

cf) $y \sim N(\mu, \sigma_y^2)$, $\mu \sim N(\mu_0, \sigma_0^2)$

평균이 μ인 normal을 따르는 데이터의 평균에 prior를 normal로 걸어주었을 때 데이터 y의 predictive distribution (posterior)

2. Posterior Predictive Distribution

$$\tilde{y} \sim \mathcal{N}(\mu_p, \sigma_p^2 + \sigma_y^2),$$

$$= \mathcal{N}\left(\frac{\bar{y}_i n_k \tau_k + \mu_0 \tau_0}{n_k \tau_k + \tau_0}, \frac{1}{(n_k \tau_k + \tau_0)} + \sigma_y^2\right)$$

cf) prior predictive distribution

$$p(\mu_k|\boldsymbol{y}) \propto p(\boldsymbol{y}|\mu_k)p(\mu_k)$$

$$\propto \exp\left(-\frac{\tau_k \sum_{i=1}^{n_k}(y_i - \mu_k)^2}{2}\right) \times \exp\left(-\frac{\tau_0(\mu_k - \mu_0)^2}{2}\right),$$

$$\sim N\left(\frac{\bar{y_k} n_k \tau_k + \mu_0 \tau_0}{n_k \tau_k + \tau_0}, \frac{1}{n_k \tau_k + \tau_0}\right)$$

k th cluster의 현재 정보를 반영하여 얻은 posterior.

DPMM

DPMM을 사용하는 이유

결국 하고자하는 것은 알고있는 정보를 최대한 사용하여 새로운 data가 어느 cluster에 들어갈지를 구하는 것으로, ci의 posterior를 구하는 것과 같다. 그러나 CRP의 경우 데이터의 평균과 분산이 cluster에 영향을 미침에도 불구하고 이를 고려하지 않고 단순히 각 클러스터에 몇명이 있는지만 고려한다. 따라서 각각의 경우의 predictive distribution을 같이 고려해주어야 하며, 이것이 DPMM이다.

DPMM=CRP+PPD(Prior/Posterior predictive distribution)

DPMM

Posterior Predictive Distribution

: 해당 cluster의 정보(size, mean, precision등)를 반영

하여 얻는 predictive distribution

$$\tilde{y} \sim \mathcal{N}(\mu_p, \sigma_p^2 + \sigma_y^2),$$
$$= \mathcal{N}\Big(\frac{\bar{y}_i n_k \tau_k + \mu_0 \tau_0}{n_k \tau_k + \tau_0}, \frac{1}{(n_k \tau_k + \tau_0)} + \sigma_y^2\Big)$$

→ 기존의 클러스터에 배정할 때 사용 !!

Prior Predictive Distribution

: 단순히 mu의 prior만 가정한 predictive distribution

$$\tilde{y_i} \sim N(\mu_0, \sigma_0^2 + \sigma_y^2)$$

→ 새 클러스터에 배정할 때 사용 !!

DPMM

- (1) 기존의 cluster에 배정되는 경우

$$p(c_i|\mathbf{c}_{-\mathbf{i}}, \mu_k, \tau_k, \alpha)$$

$$\propto p(c_i|\mathbf{c}_{-\mathbf{i}}, \alpha)p(\tilde{y}_i|\mu_k, \tau_k, \mathbf{c}_{-\mathbf{i}})$$

$$\propto \frac{n_{-i,k}}{n-1+\alpha} N(\tilde{y}_i; \frac{\bar{y}_k n_k \tau_k + \mu_0 \tau_0}{n_k \tau_k + \tau_0}, \frac{1}{n_k \tau_k + \tau_0} + \sigma_y^2)$$

CRP-기존 테이블

Posterior Predictive Dist.

- (2) 새로운 cluster를 만들어 배정되는 경우

$$p(c_i \neq c_k \quad \forall k \neq i|\mathbf{c}_{-i}, \mu_0, \tau_0, \alpha)$$

$$\propto p(c_i \neq c_k \quad \forall k \neq i|\mathbf{c}_{-i}, \alpha) \times \int p(\tilde{y}_i|\mu_k, \tau_k)p(\mu_k, \tau_k|\mu_0, \tau_0)d\mu_k d\tau_k$$

$$\propto \frac{\alpha}{n-1+\alpha} N(\tilde{y}_i; \mu_0, \sigma_0^2 + \sigma_y^2)$$

CRP-새로운 테이블(클러스터)

Prior Predictive Dist.

3
R code Implementation

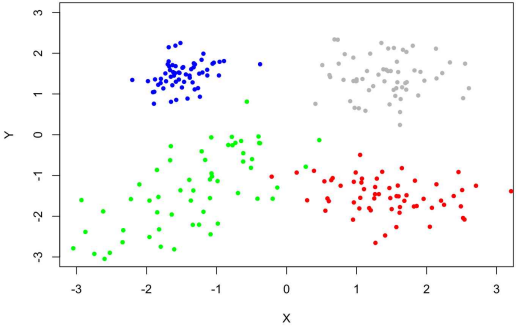
Step 1: Data Generating

$$\begin{pmatrix} Y_1 \\ Y_2 \end{pmatrix} \sim N_2\left(\begin{bmatrix} 1.5 \\ 1.5 \end{bmatrix}, \begin{bmatrix} .30 & .05 \\ .05 & .30 \end{bmatrix}\right)$$

$$\begin{pmatrix} Y_1 \\ Y_2 \end{pmatrix} \sim N_2\left(\begin{bmatrix} 1.5 \\ -1.5 \end{bmatrix}, \begin{bmatrix} .50 & -.08 \\ -.08 & .50 \end{bmatrix}\right)$$

$$\begin{pmatrix} Y_1 \\ Y_2 \end{pmatrix} \sim N_2\left(\begin{bmatrix} -1.5 \\ 1.5 \end{bmatrix}, \begin{bmatrix} .10 & .03 \\ .03 & .10 \end{bmatrix}\right)$$

$$\begin{pmatrix} Y_1 \\ Y_2 \end{pmatrix} \sim N_2\left(\begin{bmatrix} -1.5 \\ -1.5 \end{bmatrix}, \begin{bmatrix} .80 & .50 \\ .50 & .80 \end{bmatrix}\right)$$



Step 2: Defining `crp_gibbs()` Function

```
library(mvtnorm)
crp_gibbs<-function(data, alpha, mu0, sigma0, sigma_y, c_init, maxIters=1000){
  data_dim<-ncol(data) #dimension of data points
  N<-nrow(data) #number of data points
  #data= NXD matrix of data points
  #sigma_y: measurement error of data y, assumed known, same for all clusters
  #mu0, sigma0: prior mean and variance around unknown mean mu
(…)
if (newz==Nclust + 1){
        n_k<-c(n_k, 0)
        Nclust<-Nclust + 1
      }
      z[n]<-newz
      n_k[newz]<-n_k[newz] + 1
    }
    setTxtProgressBar(pb, iter)
    res[, iter]<-z
  }
  close(pb)
  invisible(res)
}
```

Step 3-(1): Find Maximum-A-Posteriori of the Class Membership

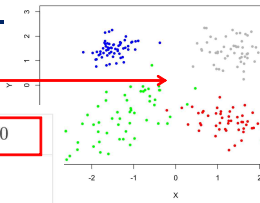
```
alpha<-0.01
mu0<-matrix(rep(0,2), ncol=2, byrow=TRUE)
sigma0<-diag(2)*3^2
sigma_y<-diag(2)*1
c_init<-rep(1, nrow(datc))
results<-crp_gibbs(data=datc, alpha=alpha, mu0=mu0, sigma0=sigma0, sigma_y=sigma_y,
c_init=rep(1, nrow(datc)))
tab<-apply( results, 1, FUN=function(x){
        tab<-table(x)
        ans<-names(tab[which.max(tab)])
        return(ans)})
table(tab)

tab
1 2 3 4
68 40 64 68
```

Step 3-(2): Hyperparameter Tuning; *Effect of Input Value*

```
set.seed(11)
alpha<-c(0.01, 1.00, 3.00, 5.00)
sigma_y<-c(0.50, 1.00, 1.50, 3.00)
for (i in 1:length(alpha)){
        for (j in 1:length(sigma_y)){
                results<-crp_gibbs(data=datc, alpha=alpha[i], mu0=matrix(rep(0, 2),
                        ncol=2, byrow=TRUE), sigma0 = diag(2)*3^2,
                        sigma_y=diag(2)*sigma_y[j], c_init=rep(1, nrow(datc)))
                tab<-apply(results, 1, FUN=function(x){
                        tab<-table(x)
                        ans<-names(tab[which.max(tab)])
                        return(ans)})
                cat("alpha= ", alpha[i], "sigma_y= ", sigma_y[j], "\n")
                print(table(tab))
        }
}
```

Step 3-(2): Hyperparameter Tuning; *interpreting effect*

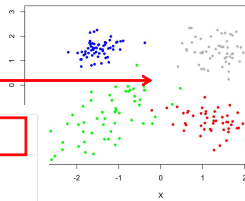


$\alpha \backslash \sigma_y^2$	$\sigma_y^2 = 0.50$	$\sigma_y^2 = 1.00$	$\sigma_y^2 = 1.50$	$\sigma_y^2 = 3.00$
$\alpha = 0.01$	c_i 1 2 3 4 5 n_k 61 38 60 60 21	1 2 3 4 79 65 27 59	1 2 141 99	1 240
$\alpha = 1.00$	c_i 1 2 3 4 5 n_k 61 60 44 58 17	1 2 3 4 60 75 66 39	1 2 3 4 83 32 68 57	1 240
$\alpha = 3.00$	c_i 1 2 3 4 5 n_k 61 35 23 60 61	1 2 3 4 76 38 60 66	1 2 3 4 94 51 93 2	1 240
$\alpha = 5.00$	c_i 1 2 3 4 5 n_k 1 60 54 61 64	1 2 3 4 5 51 9 70 33 77	1 2 3 4 42 70 34 94	1 240

기존 클러스터로 : $p(c_i|\mathbf{c}_{-i}, \alpha) = \dfrac{n_{-i,k}}{n-1+\alpha}$

새로운 클러스터로 : $p(c_i \neq c_k \forall k \neq i|\mathbf{c}_{-i}, \alpha) = \dfrac{\alpha}{n-1+\alpha}$

Step 3-(2): Hyperparameter Tuning; *interpreting effect*



$\alpha \backslash \sigma_y^2$	$\sigma_y^2 = 0.50$	$\sigma_y^2 = 1.00$	$\sigma_y^2 = 1.50$	$\sigma_y^2 = 3.00$
$\alpha = 0.01$	c_i 1 2 3 4 5 n_k 61 38 60 60 21	1 2 3 4 79 65 27 59	1 2 141 99	1 240
$\alpha = 1.00$	c_i 1 2 3 4 5 n_k 61 60 44 58 17	1 2 3 4 60 75 66 39	1 2 3 4 83 32 68 57	1 240
$\alpha = 3.00$	c_i 1 2 3 4 5 n_k 61 35 23 60 61	1 2 3 4 76 38 60 66	1 2 3 4 94 51 93 2	1 240
$\alpha = 5.00$	c_i 1 2 3 4 5 n_k 1 60 54 61 64	1 2 3 4 5 51 9 70 33 77	1 2 3 4 42 70 34 94	1 240

성능은 α보다 sigma에 더 민감

기존 클러스터로 : $p(c_i|\mathbf{c}_{-i}, \alpha) = \frac{n_{-i,k}}{n-1+\alpha}$

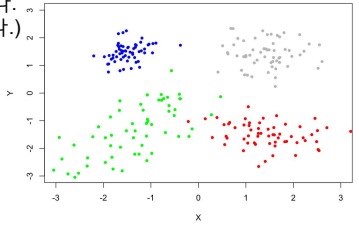
새로운 클러스터로 : $p(c_i \neq c_k \forall k \neq i|\mathbf{c}_{-i}, \alpha) \propto \frac{\alpha}{n-1+\alpha}$

α가 클수록 새 클러스터를 생성할 확률이 높음!

Step 4: Building Confusion Matrix; *Information on Misclassification*

True \ Modal	1	2	3	4
1	60			
2		60		
3				60
4		7	36	17

Clustering은 Unsupervised Learning이지만,
데이터를 직접 생성했으므로 일종의 Confusion Matrix를 생각해 볼 수 있다.
(Unsupervised Learning이므로 학습 과정에서 Label이 사용되지는 않는다.)



Step 5: Estimating Cluster Mean & Covariances *with best hyperparameters*

```
 cluster  1
      [,1]
[1,] 1.415
[2,] 1.412
      [,1]  [,2]
[1,] 0.314 0.011
[2,] 0.011 0.285
```

```
 cluster  4
      [,1]
[1,] -1.337
[2,]  1.131
      [,1]   [,2]
[1,]  0.222 -0.146
[2,] -0.146  0.598
```

```
 cluster  2
      [,1]
[1,] -1.805
[2,] -1.931
      [,1]  [,2]
[1,] 0.626 0.399
[2,] 0.399 0.713
```

```
 cluster  3
      [,1]
[1,]  1.289
[2,] -1.446
      [,1]   [,2]
[1,]  0.689 -0.098
[2,] -0.098  0.253
```

$$\begin{pmatrix} Y_1 \\ Y_2 \end{pmatrix} \sim N_2\left(\begin{bmatrix} 1.5 \\ 1.5 \end{bmatrix}, \begin{bmatrix} .30 & .05 \\ .05 & .30 \end{bmatrix}\right)$$

$$\begin{pmatrix} Y_1 \\ Y_2 \end{pmatrix} \sim N_2\left(\begin{bmatrix} 1.5 \\ -1.5 \end{bmatrix}, \begin{bmatrix} .50 & -.08 \\ -.08 & .50 \end{bmatrix}\right)$$

$$\begin{pmatrix} Y_1 \\ Y_2 \end{pmatrix} \sim N_2\left(\begin{bmatrix} -1.5 \\ 1.5 \end{bmatrix}, \begin{bmatrix} .10 & .03 \\ .03 & .10 \end{bmatrix}\right)$$

$$\begin{pmatrix} Y_1 \\ Y_2 \end{pmatrix} \sim N_2\left(\begin{bmatrix} -1.5 \\ -1.5 \end{bmatrix}, \begin{bmatrix} .80 & .50 \\ .50 & .80 \end{bmatrix}\right)$$

4

Further Researches

① Testing DPMM empirically with Larger Dataset

② Testing DPMM empirically with More Complex Clusters

③ Comparing DPMM with Gaussian Mixture Model and K-Means Clustering

4-①

Larger Dataset

① Testing DPMM empirically with Larger Dataset

Dataset의 크기가 커진다면?

Prior를 가정했지만 data의 size가 커질수록 bayesian estimator에서의 prior의 영향력은 줄어든다.

-> **잘못 분류된 데이터의 수가 줄어들 것이다!**

① **-1:** *n=600 X 4*

```
n<-600
m1<-c(1.5, 1.5)
(…)
c_modal<-apply(results, 1, FUN=function(x){
         tab<-table(x)
         ans<-names(tab[which.max(tab)])
         return(ans)
})
c_true<-rep(1:4, each=600)
table(c_true, c_modal)

table(tab)
tab
1 2 3 4
651 631 599 519

table(c_true, c_modal)
         c_modal
c_true     1 2 3 4
         1 3 1 0 596
         2 0 594 6 0
         3 600 0 0 0
         4 48 37 512 3
```

① -1 : *n=600 X 4*

Building Confusion Matrix

T \ M	1	2	3	4
1	3	1		596
2		594	6	
3	600			
4	48	37	512	3

각 군집 60개, 군집의 수 4개, 총 240개에서
각 군집 600개, 군집의 수 4개, 총 2400개로

Estimating Cluster Parameters

```
cluster  1
      [,1]
[1,] -1.403
[2,]  1.416
       [,1]   [,2]
[1,]  0.176 -0.058
[2,] -0.058  0.231
```

```
cluster  2
      [,1]
[1,]  1.436
[2,] -1.458
       [,1]   [,2]
[1,]  0.582 -0.123
[2,] -0.123  0.224
```

```
cluster  3
      [,1]
[1,] -1.686
[2,] -1.646
       [,1]  [,2]
[1,] 0.599 0.310
[2,] 0.310 0.566
```

```
cluster  4
      [,1]
[1,] 1.503
[2,] 1.511
       [,1]  [,2]
[1,] 0.302 0.053
[2,] 0.053 0.295
```

① **-2 :** *n=2400*

```
n<-2400
m1<-c(1.5, 1.5)
(…)
c_modal<-apply(results, 1, FUN=function(x){
        tab<-table(x)
        ans<-names(tab[which.max(tab)])
        return(ans)
})
c_true<-rep(1:4, each=600)
table(c_true, c_modal)

table(tab)
tab
1 2 3 4
2598 2548 599 519

table(c_true, c_modal)
        c_modal
c_true     1 2 3 4 5
        1 15 2375 10 0 0
        2 0 2 2389 9 0
        3 2400 0 0 0 0
        4 211 25 182 1981 1
```

① **-2 :** *n=2400*

Building Confusion Matrix

T \ M	1	2	3	4	5
1	15	2375	10		
2		2	2389	9	
3	2400				
4	211	25	182	1981	1

각 군집 60개, 군집의 수 4개, 총 240개에서
각 군집 2400개, 군집의 수 4개, 총 9600개로

Estimating Cluster Parameters

```
cluster  1
      [,1]
[1,] -1.437
[2,]  1.372
       [,1]   [,2]
[1,]  0.158 -0.054
[2,] -0.054  0.279
```

```
cluster  3
      [,1]
[1,]  1.406
[2,] -1.453
       [,1]   [,2]
[1,]  0.584 -0.138
[2,] -0.138  0.244
```

```
cluster  2
      [,1]
[1,] 1.487
[2,] 1.513
      [,1]  [,2]
[1,] 0.287 0.057
[2,] 0.057 0.294
```

```
cluster  4
      [,1]
[1,] -1.724
[2,] -1.749
      [,1]  [,2]
[1,] 0.557 0.283
[2,] 0.283 0.554
```

② 안 좋았던 hyperparameter의 경우

c_true \ c_modal	1	2	3	4	5
1	37	0	0	16	7
2	0	0	56	4	0
3	60	0	0	0	0
4	13	33	14	0	0

- 클러스터의 갯수 틀림 - 원래는 4개
- 클러스터링한 결과도 안좋음.
- 클러스터링이 잘 안된 클러스터의 경우 50 프로 정도만 맞춤

c_true \ c_modal	1	2	3	4
1	589	7	4	0
2	0	0	598	2
3	0	600	0	0
4	2	87	50	461

- 클러스터의 갯수 맞춤
- 클러스터링이 잘 안된 클러스터의 경우에도 75프로가 넘는 정확도를 보임

4-②

More Complex Cluster

② Testing DPMM empirically with More Complex Clusters

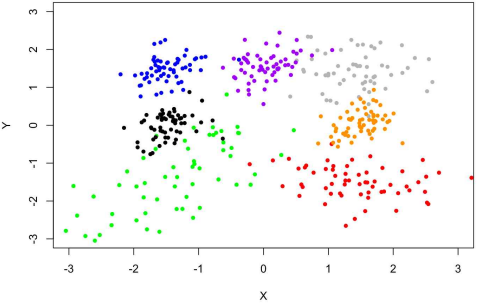
클러스터의 개수가 늘어난다면?

기존 논문에서 제공된 데이터 셋은 4개의 클러스터로부터 생성된 것으로, 그림상 각각의 클러스터가 비교적 잘 구분되었다. 그러나 클러스터의 수가 많아진다면 CRP에서 생성해야하는 클러스터의 수도 많아지고 overlap 되는 경우도 많아질 것이므로 DPMM의 성능을 제대로 확인할 수 있을 것이다.

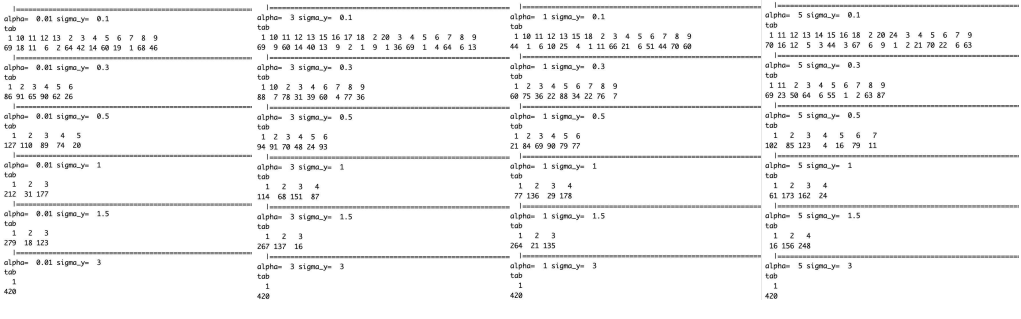
② - ⓐ Data Generating

```
m1<-c(1.5, 1.5)
S1<-matrix(c(0.3, 0.05, 0.05, 0.3), ncol=2)
clus1<-mvrnorm(n, m1, S1)
m2<-c(1.5, -1.5)
S2<-matrix(c(0.5, -0.08, -0.08, 0.2), ncol=2)
clus2<-mvrnorm(n, m2, S2)
(…)

S1.5<-matrix(c(0.1, 0.03, 0.03, 0.1), ncol=2)
clus1.5<-mvrnorm(n, m1.5, S1.5)
m2.5<-c(1.5, 0)
S2.5<-matrix(c(0.1, 0.03, 0.03, 0.1), ncol=2)
clus2.5<-mvrnorm(n, m2.5, S2.5)
m3.5<-c(-1.5, 0)
S3.5<-matrix(c(0.1, 0.03, 0.03, 0.1), ncol=2)
clus3.5<-mvrnorm(n, m3.5, S3.5)
datc<-rbind(clus1, clus2, clus3, clus4, clus1.5,
clus2.5, clus3.5)
```



② - ⓑ Hyperparameter tuning



② - ⓒ **With Best hyperparameter;** *alpha = 5, sigma_y=0.5*

Building Confusion Matrix

T \ M	1	2	3	4	5	6	7
1			17	43			
2		59			1		
3	60						
4	1	2			4	21	32
5	21			39			
6		28	32				
7	4				22		34

② - ⓒ With Best hyperparameter; *alpha = 5, sigma_y=0.5*

Estimating clustering coefficients

```
 cluster  1
       [,1]
[1,] -1.166
[2,]  1.407
      [,1]  [,2]
[1,] 0.369 0.021
[2,] 0.021 0.189

 cluster  2
       [,1]
[1,]  1.415
[2,] -1.070
       [,1]   [,2]
[1,]  0.380 -0.053
[2,] -0.053  0.535

 cluster  3
      [,1]
[1,] 1.672
[2,] 0.558
      [,1]  [,2]
[1,] 0.191 0.114
[2,] 0.114 0.276

 cluster  4
       [,1]
[1,] 0.760
[2,] 1.594
      [,1]  [,2]
[1,] 0.421 0.029
[2,] 0.029 0.176

 cluster  5
       [,1]
[1,] -1.235
[2,] -0.027
       [,1]   [,2]
[1,]  0.323 -0.148
[2,] -0.148  0.359

 cluster  6
       [,1]
[1,] -2.155
[2,] -2.501
      [,1]  [,2]
[1,] 0.701 0.236
[2,] 0.236 0.457

 cluster  7
       [,1]
[1,] -1.317
[2,] -0.521
      [,1]  [,2]
[1,] 0.277 0.018
[2,] 0.018 0.372
```

4-③

Comparing with other Clustering methods

③ Comparing DPMM with GMM & K-Means Clustering

DPMM

- 사전 분포가 필요함.
  - i.e. Bayesian Method
- 각 Datapoint가 어떤 군집에 속할 수 있는지 확률로 표현 가능함
  - i.e. Soft Labeling
- 군집의 수 k 가 확률적으로 정해짐.

GMM

- k 개의 Gaussian의 linear combination으로 표현
- 각 Datapoint가 어떤 군집에 속할 수 있는지 확률로 표현 가능함
  - i.e. Soft Labeling
- 군집의 수 *k* 는 고정된 hyperparameter임.

K-Means Clustering

- k 개의 중심점을 기준으로 거리가 가까운 것끼리 그룹화
- 각 Datapoint는 오직 하나의 군집에만 속함
  - i.e. Hard Labeling
- 군집의 수 *k* 는 고정된 hyperparameter임

③ - ⓐ Comparing DPMM with Gaussian Mixture Model

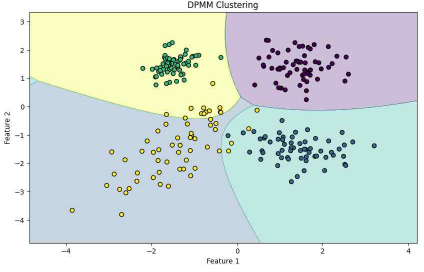
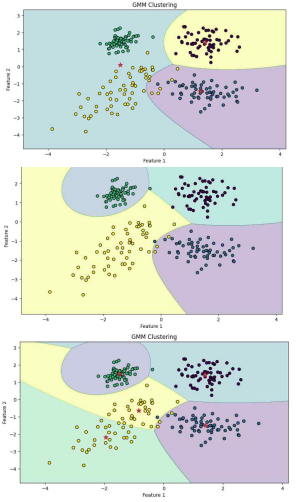


Figure 1 DPMM vs. Gaussian Mixture Model



③ - ⓑ Comparing DPMM with K-Means Clustering

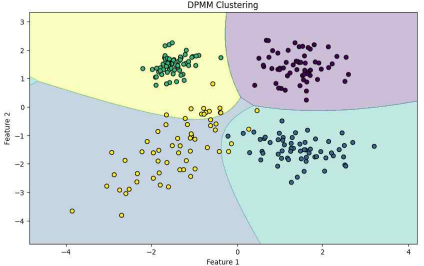
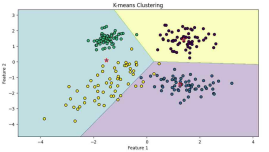
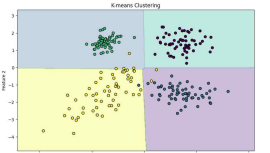
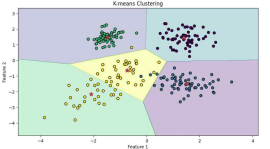


Figure 2 DPMM vs. K-Means Clustering







감사합니다