<과제물 작성시 주의사항>

[공통]

과제물 제출시 완성된 **소스파일 및 보고서**를 반드시 **'HW_02_학번.zip'** 형식으로 압축하여 첨부합니다.

(이름 약어.py, HW_02_학번.pdf)

[소스파일]

1. 소스파일은 **.py파일만 작성**하며 반드시 문제에서 지시 또는 요구한 조건에 맞추어서 작성합니다.

(jupyter로 작성하였어도 코드를 제출 시 py파일로 작성하여 제출하여야 합니다.)

2. 각 코드마다 **반드시 주석을 달아 주셔야 합니다.** 주석을 달지 않을 경우, 부분적으로 감점이 있을 수 있습니다. **이번 과제 특히**

3. 결과가 올바르더라도 과정이 옳지 않을 경우, 부분적으로 감점이 있을 수 있습니다.

4. 제출한 파일이 실행되지 않을 경우, 제출한 과제물은 0점 처리됩니다.

[보고서]

1. **PDF**로 제출하며, 표지를 포함해야 합니다.

2. 보고서에는 **(과제 제목 및 목적), (소스 코드에 대한 설명), (실행 결과 + Plot), (참고문헌)**이 포함되어야 합니다.

3. 자신의 코드에 대한 설명이 명확하지 않거나 copy한 글이라면 0점 처리됩니다.

4. **실행 결과는 실행 결과를 캡처하여 첨부하도록 합니다.**

5. 참고문헌은 반드시 적어도 한 개 이상을 명시하여야 합니다.

EM ALGORITHM USING KMEANS FOR GMM

Machine learning homework-2

Assistant : Choongseop Lee, cndtjq97@gmail.com

EM ALGORITHM USING KMEANS FOR GMM

- You should build 6 python functions at Name.py
  - initialization (10 points)
  - multivariate_gaussian_distribution (10 points)
  - expectation (20 points)
  - maximization (20 points)
  - fit (20 points)
  - plotting (20 points) including report scores

- An unclear comment is a deduction factor
- Implement using Python built-in functions and Numpy functions only + plotting methods
- Submission : change below files to zip file and submit it by KLAS
  - Name.py (please change CSL.py to your name.py)
    - E.g. if your name is 홍길동 --> (CSL.py --> GDH.py)
  - HW_02_student ID.pdf (E.g. HW_02_202110605.pdf) <= report

EM ALGORITHM USING KMEANS FOR GMM

Algorithm 1: EM algorithm for GMM

Input : a given data $X = \{x_1, x_2, \ldots, x_n\}$

Output : $\pi = \{\pi_1, \pi_2, \ldots, \pi_K\}$,

$\mu = \{\mu_1, \mu_2, \ldots, \mu_K\}$,

$\Sigma = \{\Sigma_1, \Sigma_2, \ldots, \Sigma_K\}$

1 Randomly initialize π, μ, Σ

2 **for** $t = 1 : T$ **do**

3   // E-step

4   **for** $n = 1 : N$ **do**

5   **for** $k = 1 : K$ **do**

6       $\gamma(z_{nk}) = \frac{\pi_k N(x_n | \mu_k, \Sigma_k)}{\sum_{j=1}^{K} \pi_j N(x_n | \mu_j, \Sigma_j)}$

7   **end**

8   **end**

9   // M-step

10   **for** $k = 1 : K$ **do**

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

12     $\Sigma_k = \frac{\sum_{n=1}^{N} \gamma(z_{nk})(x_n - \mu_k)(x_n - \mu_k)^T}{\sum_{n=1}^{N} \gamma(z_{nk})}$

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

14   **end**

15 **end**

- See comments for additional information
- In initialization step – line 1
- Initialize mean, sigma, and pi values

EM ALGORITHM USING KMEANS FOR GMM

Algorithm 1: EM algorithm for GMM

Input : a given data $X = \{x_1, x_2, \ldots, x_n\}$

Output : $\pi = \{\pi_1, \pi_2, \ldots, \pi_K\}$,
$\mu = \{\mu_1, \mu_2, \ldots, \mu_K\}$,
$\Sigma = \{\Sigma_1, \Sigma_2, \ldots, \Sigma_K\}$

Randomly initialize π, μ, Σ
for $t = 1 : T$ **do**
  // E-step
  for $n = 1 : N$ **do**
    for $k = 1 : K$ **do**
      $\gamma(z_{nk}) = \frac{\pi_k \boxed{N(x_n|\mu_k, \Sigma_k)}}{\sum_{j=1}^{K} \pi_j N(x_n|\mu_j, \Sigma_j)}$
    end
  end
  // M-step
  for $k = 1 : K$ **do**
    $\mu_k = \frac{\sum_{n=1}^{N} \gamma(z_{nk}) x_n}{\sum_{n=1}^{N} \gamma(z_{nk})}$
    $\Sigma_k = \frac{\sum_{n=1}^{N} \gamma(z_{nk})(x_n - \mu_k)(x_n - \mu_k)^T}{\sum_{n=1}^{N} \gamma(z_{nk})}$
    $\pi_k = \frac{1}{N} \sum_{n=1}^{N} \gamma(z_{nk})$
  end
end

$$g_{(\mu,\Sigma)}(\mathbf{x}) = \frac{1}{\sqrt{2\pi}^{d} \sqrt{\det(\Sigma)}} e^{-\frac{1}{2}(\mathbf{x}-\mu)^T \Sigma^{-1} (\mathbf{x}-\mu)}$$

- In multivariate_gaussian_distribution
- Since iris has four features, a multivariate normal distribution(MVN) should be used
- Use the linear algebraic functions of Numpy

EM ALGORITHM USING KMEANS FOR GMM

Algorithm 1: EM algorithm for GMM

Input : a given data $X = \{x_1, x_2, ..., x_n\}$
Output: $\pi = \{\pi_1, \pi_2, ..., \pi_K\}$,
$\mu = \{\mu_1, \mu_2, ..., \mu_K\}$,
$\Sigma = \{\Sigma_1, \Sigma_2, ..., \Sigma_K\}$

1 Randomly initialize π, μ, Σ
2 **for** $t = 1 : T$ **do**
3   // E-step
4   **for** $n = 1 : N$ **do**
5   **for** $k = 1 : K$ **do**
6             $\gamma(z_{nk}) = \frac{\pi_k N(x_n|\mu_k, \Sigma_k)}{\sum_{j=1}^{K} \pi_j N(x_n|\mu_j, \Sigma_j)}$
7   **end**
8   **end**
9   // M-step
10   **for** $k = 1 : K$ **do**
11         $\mu_k = \frac{\sum_{n=1}^{N} \gamma(z_{nk}) x_n}{\sum_{n=1}^{N} \gamma(z_{nk})}$
12         $\Sigma_k = \frac{\sum_{n=1}^{N} \gamma(z_{nk})(x_n - \mu_k)(x_n - \mu_k)^T}{\sum_{n=1}^{N} \gamma(z_{nk})}$
13         $\pi_k = \frac{1}{N} \sum_{n=1}^{N} \gamma(z_{nk})$
14   **end**
15 **end**

- In expectation step – line 4~8
- Use MVN to calculate a posterior probability

EM ALGORITHM USING KMEANS FOR GMM

Algorithm 1: EM algorithm for GMM

Input : a given data $X = \{x_1, x_2, \ldots, x_n\}$
Output: $\pi = \{\pi_1, \pi_2, \ldots, \pi_K\}$,
$\mu = \{\mu_1, \mu_2, \ldots, \mu_K\}$,
$\Sigma = \{\Sigma_1, \Sigma_2, \ldots, \Sigma_K\}$

1 Randomly initialize π, μ, Σ
2 **for** $t = 1 : T$ **do**
3 // E-step
4 **for** $n = 1 : N$ **do**
5 **for** $k = 1 : K$ **do**
6 $\gamma(z_{nk}) = \frac{\pi_k N(x_n|\mu_k, \Sigma_k)}{\sum_{j=1}^{K} \pi_j N(x_n|\mu_j, \Sigma_j)}$
7 **end**
8 **end**
9 // M-step
10 **for** $k = 1 : K$ **do**
11 $\mu_k = \frac{\sum_{n=1}^{N} \gamma(z_{nk}) x_n}{\sum_{n=1}^{N} \gamma(z_{nk})}$
12 $\Sigma_k = \frac{\sum_{n=1}^{N} \gamma(z_{nk})(x_n - \mu_k)(x_n - \mu_k)^T}{\sum_{n=1}^{N} \gamma(z_{nk})}$
13 $\pi_k = \frac{1}{N} \sum_{n=1}^{N} \gamma(z_{nk})$
14 **end**
15 **end**

- In maximization step – line 10~14
- Use the posterior probability to calculate the parameters for MVN

EM ALGORITHM USING KMEANS FOR GMM

Algorithm 1: EM algorithm for GMM

Input : a given data $X = \{x_1, x_2, ..., x_n\}$

Output: $\pi = \{\pi_1, \pi_2, ..., \pi_K\}$,

$\mu = \{\mu_1, \mu_2, ..., \mu_K\}$,

$\Sigma = \{\Sigma_1, \Sigma_2, ..., \Sigma_K\}$

1 Randomly initialize π, μ, Σ

2 **for** $t = 1 : T$ **do**

3 // E-step

4 **for** $n = 1 : N$ **do**

5 **for** $k = 1 : K$ **do**

6 $\gamma(z_{nk}) = \frac{\pi_k N(x_n|\mu_k, \Sigma_k)}{\sum_{j=1}^{K} \pi_j N(x_n|\mu_j, \Sigma_j)}$

7 **end**

8 **end**

9 // M-step

10 **for** $k = 1 : K$ **do**

11 $\mu_k = \frac{\sum_{n=1}^{N} \gamma(z_{nk}) x_n}{\sum_{n=1}^{N} \gamma(z_{nk})}$

12 $\Sigma_k = \frac{\sum_{n=1}^{N} \gamma(z_{nk})(x_n - \mu_k)(x_n - \mu_k)^T}{\sum_{n=1}^{N} \gamma(z_{nk})}$

13 $\pi_k = \frac{1}{N} \sum_{n=1}^{N} \gamma(z_{nk})$

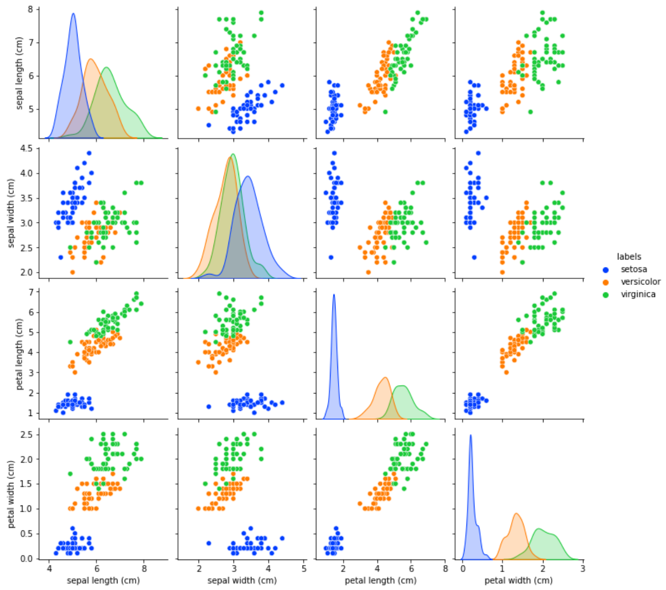
14 **end**

15 **end**

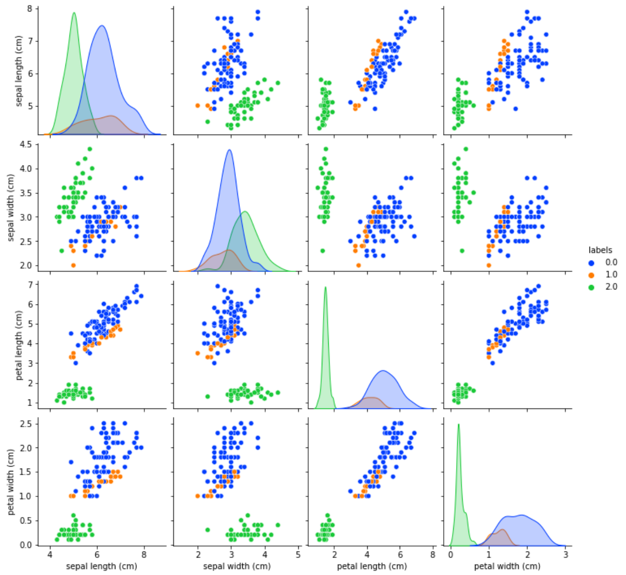
- In fit clustering – line 1 ~ 15
- Repeat the preceding functions to find the final posterior
- Implement termination conditions (Shallow copy caution)

EM ALGORITHM USING KMEANS FOR GMM

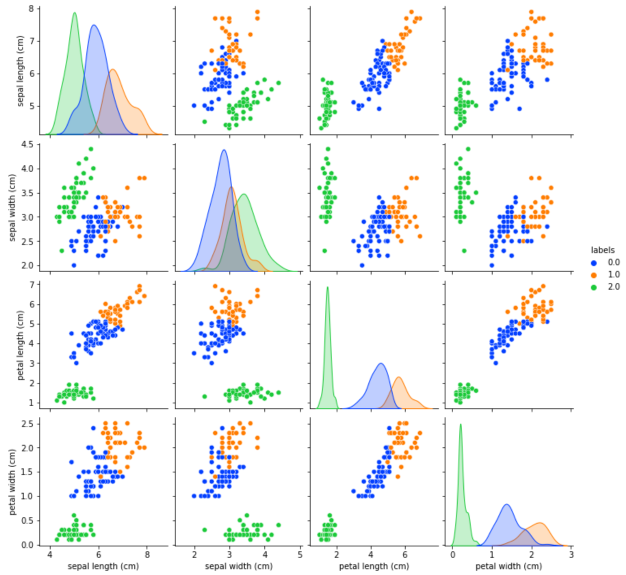
Origin



EM



KMeans



- In plotting function
- Implement a plot function for comparison
- The default is pairplot
- The plot should be attached to your report
- It's 20 points, including the report score

RESULT

- Print out the results at least 10 times and write them in the report with plot figures

```
In [1]: runcell(0, 'C:/Users/이충섭/Desktop/solution.py')
pi :             [0.36747348 0.29919319 0.33333333]
count / total : [0.36666667 0.3        0.33333333]
EM Accuracy: 0.97    Hit: 145 / 150
KM Accuracy: 0.89    Hit: 134 / 150
```

```
In [3]: runcell(0, 'C:/Users/이충섭/Desktop/solution.py')
pi :             [0.05915848 0.32031078 0.62053075]
count / total : [0.06 0.32 0.62]
EM Accuracy: 0.66    Hit: 99 / 150
KM Accuracy: 0.89    Hit: 134 / 150
```

```
In [4]: runcell(0, 'C:/Users/이충섭/Desktop/solution.py')
pi :             [0.35431522 0.33319345 0.31249134]
count / total : [0.35333333 0.33333333 0.31333333]
EM Accuracy: 0.82    Hit: 123 / 150
KM Accuracy: 0.89    Hit: 134 / 150
```

- Check the distribution of scores in both ways and analyze why
- A structural explanation (class, main) is also required along with the source code

- Bonus points, explain the reason based on what you learned in lecture

```
# Why are these two elements almost the same? additional 10 points
print(f'pi :             {EM_model.pi}')
print(f'count / total : {np.bincount(EM_pred) / 150}')
```