Machine Learning 2021 Homework 3 電機4C 洪愷尹 0710851

1 Support Vector Machine (SVM)

[1] Use the principal component analysis (PCA) to reduce the dimension of images to d = 2

```
class PCA_2D:
  def __init__(self, dim, x_train):
    pca_2d = PCA(n_components=dim)
    self.transformed_data = pca_2d.fit_transform(x_train)
    self.normalization = (self.transformed_data - np.mean(self.transformed_data)) / np.std(self.transformed_data)
    self.components = pca_2d.components_

  def get_transformed_data(self):
    return self.transformed_data
  def get_transformed_data_with_normalization(self):
    return self.normalization
  def get_components(self):
    return self.components
```

Loading...

```
[47] pca_2d = PCA_2D(2, x_train)
     x_data = pca_2d.get_transformed_data_with_normalization()
     t_data = t_train
```

[2] Analyze the difference between two decision approaches (one-versus-the-rest and one- versus-one). Decide which one you want to choose and explain why you choose this approach.

Answer

1. **One-versus-the-rest** 跟 **One-versus-one** 都是二元分類器應用在多類別問題上的策略。

---

2. **One-versus-the-rest**：選取一類當作+1類，其餘當作-1類，利用二元分類器得到一個decision hyperplane，然後輪流將每一類別都當過+1類。比如：本題有三類，所以在train階段會得到3個分類器，在test階段時，選取三個分類器中分數最高的那一類當作決策。

---

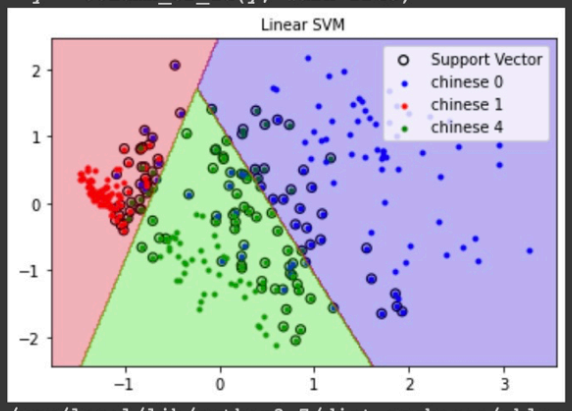
3. **One-versus-one**：一次只做兩個類別的分類，兩兩pk，直到兩兩都配對分類完成。train階段會得到3個分類器，在test階段時，選取得到最多票數的那類當作判斷決策。

---

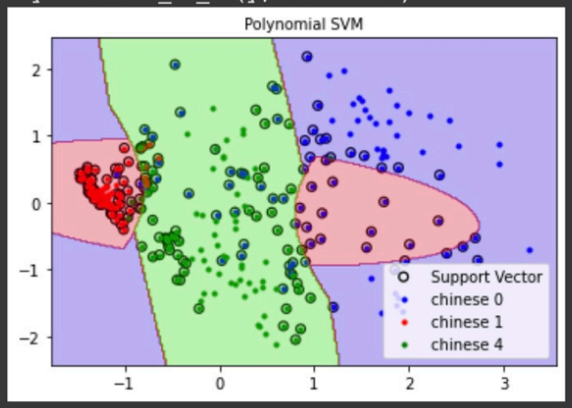
4. 本次使用One-versus-one的策略，因為SVM在這個策略下比較不會犧牲掉樣本數比較少的類別。

[3-4] Use the principle values projected to top two eigenvectors obtained from PCA, and build a SVM with linear kernel to do multi-class classification. Then, plot the corresponding decision boundary and show the support vector.

```
/usr/local/lib/python3.7/dist-packages/sklearn/utils/validation.py:985: DataConversionWarning: A column-ve
  y = column_or_1d(y, warn=True)
```




```
/usr/local/lib/python3.7/dist-packages/sklearn/utils/validation.py:985: DataConversionWarning: A column-ve
  y = column_or_1d(y, warn=True)
```



[5] Please discuss the difference between (3) (4).

Discussion

1. 由Linear SVM 跟 Polynomial SVM 結果來觀察，可以發現線性的Kernel對於這次的挑選的3個中文字分類表現比較好.

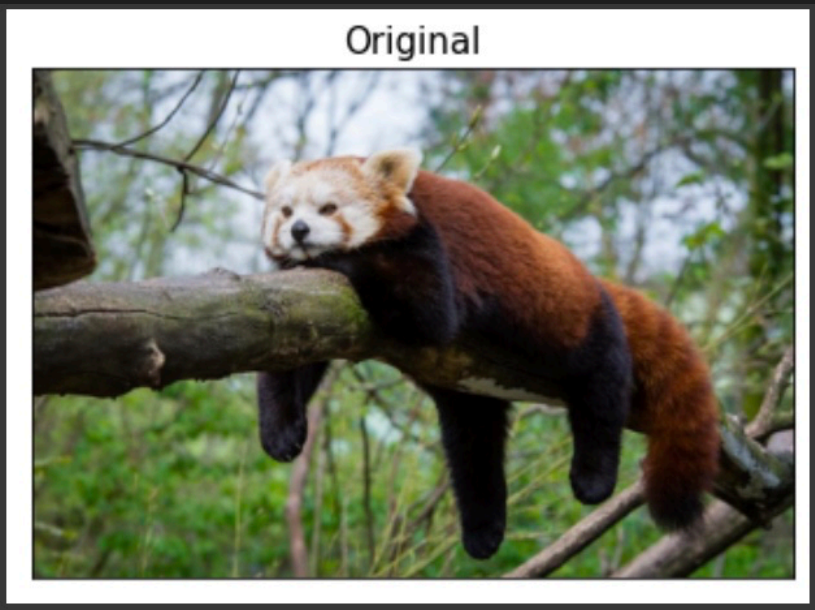
---

2. Polynomial Kernel 跟 Linear Kernel 投影的維度分別是(3, 2)，由結果論這次的分類器利用比較低的維度，也就是2維來分類效果較佳!

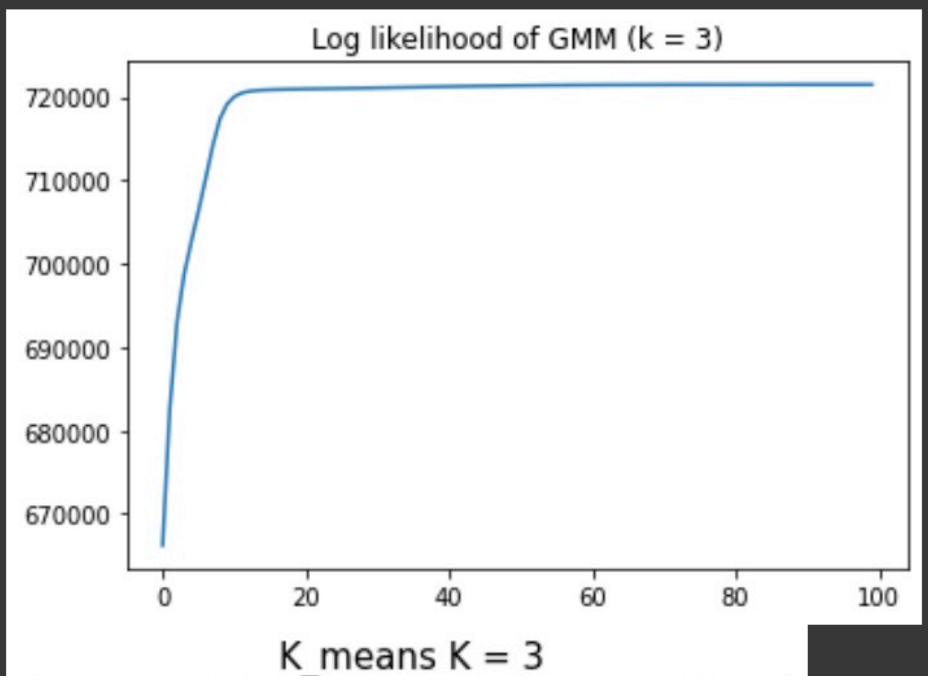
2 Gaussian Mixture Model (GMM)

[1] Please build a K-means model by minimizing J loss.

```python
class K_Means:
  def __init__(self, data, K=3, iteration=100) -> None:
      self.K = K
      self.iteration = iteration
      self.eye = np.eye(K)
      #initialize means and rnk
      self.means = data[np.random.choice(len(data), self.K, replace=False)] # without replacement
      self.rnk = np.ones([len(data), self.K]) #(252150,3)

  def minimize_J_loss(self, data):
    for iter in range(self.iteration):
      error = np.sum((data[:, None] - self.means)**2, axis=2)
      rnk = self.eye[np.argmin(error, axis=1)]

      if np.array_equal(rnk, self.rnk):
        break
      else:
        self.rnk = rnk

      #re-calculate k means
      self.means = np.sum(rnk[:, :, None] * data[:, None], axis=0) / np.sum(rnk, axis=0)[:, None]
```



==========



GMM K = 3

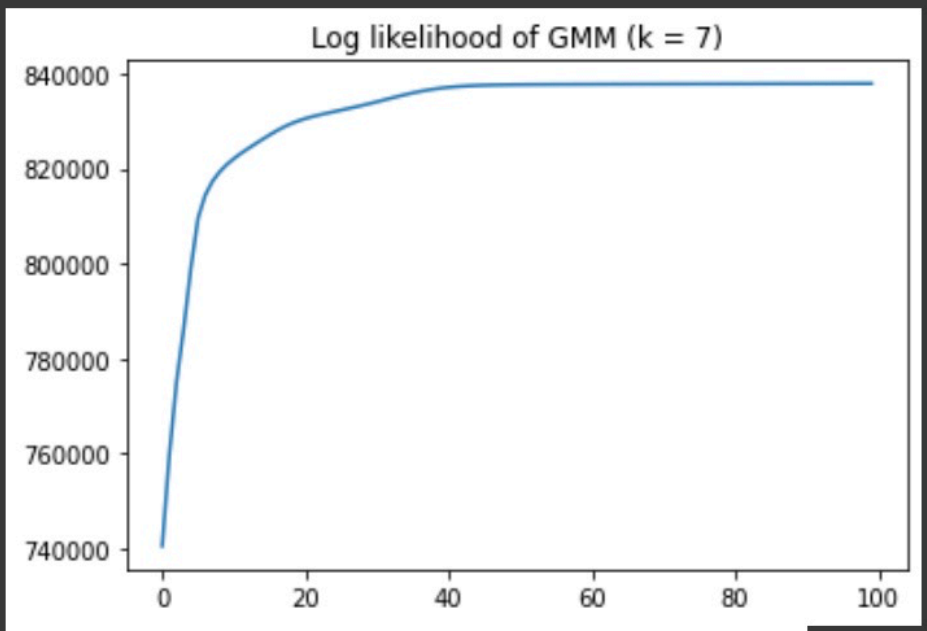
K_means K = 3

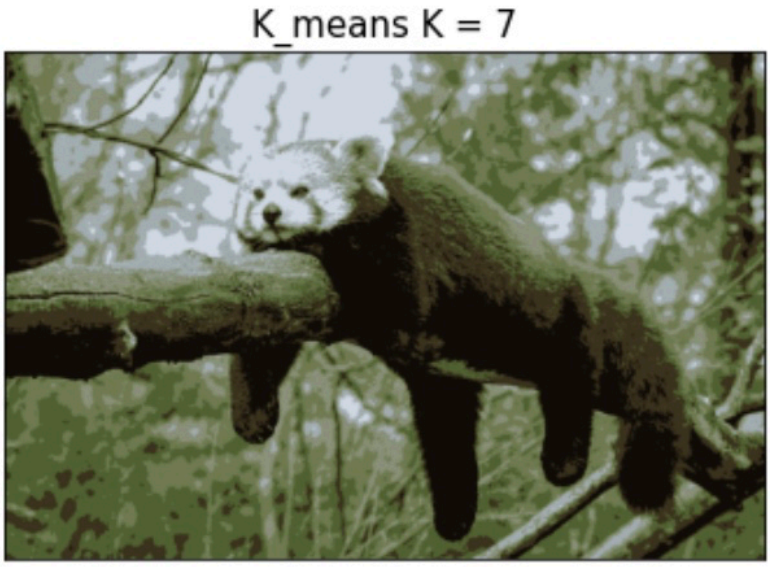
	R	G	B
0	177	186	189
1	35	31	18
2	106	118	77

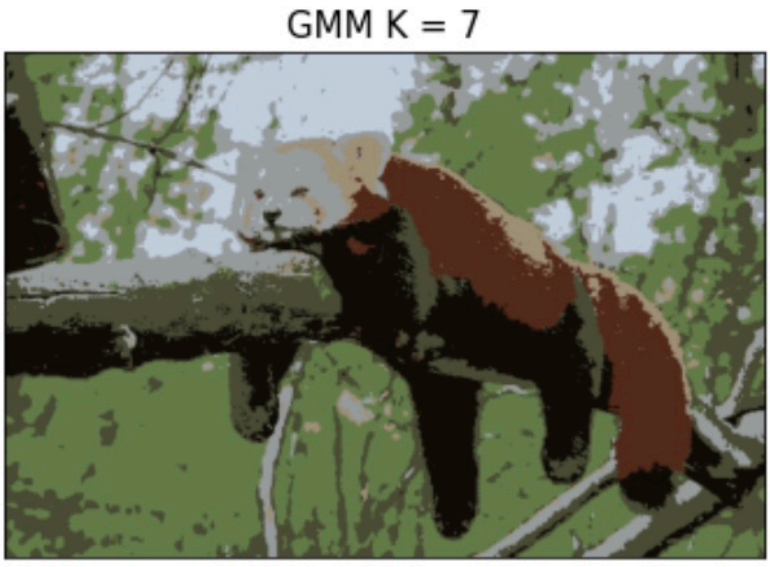
GMM K = 3

	R	G	B
0	175	188	198
1	20	16	12
2	101	109	73

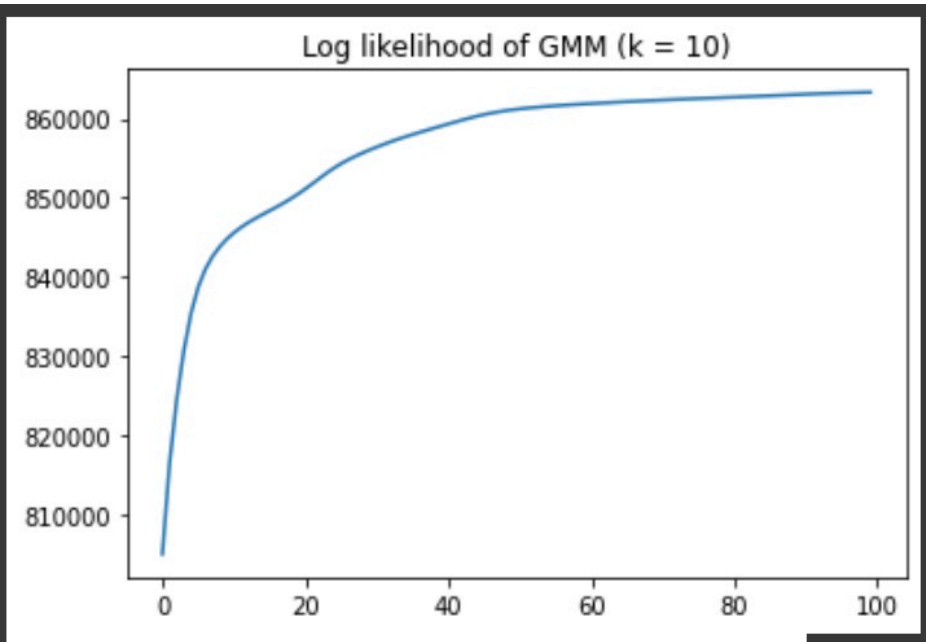
==========


Log likelihood of GMM (k = 7)
840000
820000
800000
780000
760000
740000
0
20
40
60
80
100


K_means K = 7


GMM K = 7

```
K_means K = 7
     R    G    B
0  112  124   81
1  163  174  177
2   82   99   51
3  208  217  227
4   18   14    9
5  138  145  122
6   60   52   31

GMM K = 7
     R    G    B
0  100  126   73
1  151  156  157
2   74   77   54
3  194  207  221
4   16   13    9
5  160  144  119
6   83   43   28
```

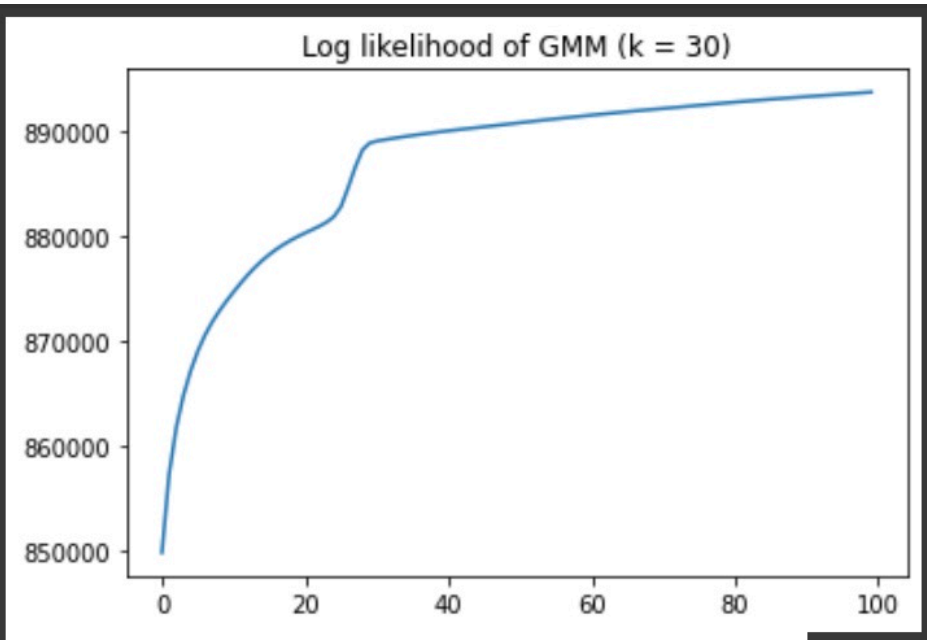


K_means K = 10

GMM K = 10

```
K_means K = 10
     R    G    B
0  212  220  230
1   72   93   44
2   97  117   69
3  144  155  146
4   50   44   28
5  110   60   40
6  172  183  189
7  123  136  101
8  173  115   81
9   17   12    8

GMM K = 10
     R    G    B
0  207  219  232
1   74   76   55
2   90  120   58
3  158  175  185
4   24   19   14
5   82   43   27
6  155  149  146
7  123  140  113
8  175  123   88
9    7    5    4
```

6 分鐘



K_means K = 30

GMM K = 30

分鐘

K_means K = 30

	R	G	B
0	106	112	81
1	22	16	10
2	228	235	244
3	88	97	71
4	176	166	150
5	92	47	30
6	99	114	105
7	124	128	103
8	161	100	69
9	209	217	228
10	46	48	32
11	124	70	46
12	62	62	48
13	75	82	56
14	153	169	175
15	144	151	126
16	186	201	214
17	32	29	22
18	188	136	99
19	94	127	55
20	8	6	4
21	133	153	95
22	208	194	184
23	168	184	194
24	112	139	75
25	120	136	131
26	57	91	18
27	139	154	155
28	76	109	40
29	61	28	12

```
GMM K = 30
      R    G    B
0   105  106   79
1    14   10    5
2   219  229  240
3    85   99   61
4   139  139  140
5    78   42   32
6    95  114  107
7   120  127  101
8   173  105   65
9   219  216  216
10   43   42   29
11  122   67   43
12   45   43   40
13   74   78   53
14  154  175  173
15  139  147  127
16  192  206  220
17   25   20   15
18  170  129   94
19   92  128   57
20    6    5    4
21  129  149   97
22  189  175  167
23  158  175  188
24  110  137   75
25  123  139  130
26   54   80   17
27  142  164  151
28   78  109   47
29   44   18    7
```

[4] Make some discussion about what is crucial factor to affect the output image between K-means and Gaussian mixture model (GMM), and explain the reason.

Discussion

1. K Means 跟 GMM 都是cluster model.

---

2. 最大的差別是，GMM是有covariance的考量在裡面，其中decision boundary是橢圓形的; K Means的boundary則是圓形的。

---

3. 從上述(2)跟結果來看，可以發現GMM的結果在將浣熊與背景分離上效果比較好，但是在細節毛紋上K means的結果比較好，這跟GMM是soft cluster 利用高斯分佈有關。
