

---

# Deep Learning for Computer Vision (2018 Spring)

## HW1

---

B03611026 郭冠軒

### Problem 1

The minimum  $P_e$  decision scheme with  $P(\omega_1) = 3/4$  is:

$$x = \begin{cases} \omega_1, & \text{if } x < 5. \\ \omega_2, & \text{otherwise.} \end{cases}$$

The decision regions are:

$$R_1 = \{x \mid 0 < x < 5\},$$

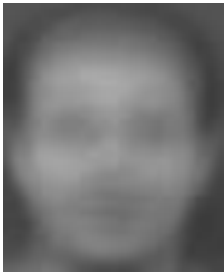



$$R_2 = \{x \mid 5 \leq x < 6\}.$$

The resulting  $P_e$  is:





$$P_e = \frac{1 - P(\omega_1)}{6 - 3} \times (5 - 3) = \frac{1}{6}$$

### Problem 2

(a) Plot the mean face and the first three eigenfaces.

Face	Mean Face	1 <sup>st</sup> Eigenface	2 <sup>nd</sup> Eigenface	3 <sup>rd</sup> Eigenface
Image				

(b) Plot the reconstructed images with corresponding MSE values.

Use first n eigenfaces	n = 3	n = 50	n = 100	n = 239
Image				
MSE ( round off to 2 <sup>nd</sup> decimal place )	659.41	213.26	81.95	0.00

(c) Apply the k-nearest neighbors classifier.

Validation results (accuracy)

Hyperparameters choice	Fold 1 accuracy	Fold 2 accuracy	Fold 3 accuracy	Average accuracy
k = 1, n = 3	0.7875	0.7375	0.6875	0.7375
k = 1, n = 50	0.9500	0.8875	0.9625	0.9333
k = 1, n = 159	0.9500	0.9000	0.9500	0.9333
k = 3, n = 3	0.6375	0.6250	0.5250	0.5958
k = 3, n = 50	0.8750	0.8250	0.8125	0.8375
k = 3, n = 159	0.8750	0.8250	0.8125	0.8375
k = 5, n = 3	0.5625	0.5500	0.4750	0.5292
k = 5, n = 50	0.8125	0.7625	0.7500	0.7750
k = 5, n = 159	0.8000	0.7500	0.7125	0.7542

Since there are two choices of hypermeters resulting in the same average accuracy, I use both of the hyperparameters choices to report the recognition rate on the test set.

Test results (recognition rate)

Hyperparameters choice	Recognition rate (accuracy)
k = 1, n = 50	0.96250
k = 1, n = 159	0.94375

### Problem 3

In order to derive the first eigenvector of a  $d \times d$  positive semi-definite symmetric matrix (denoted as  $A$ ) with  $d$  distinct eigenvalues, we can use the method called “power iteration”.

Here is the procedure to achieve our goal via power iteration:

First, let a vector  $x_0$  be a random vector, which has a non-zero component in the direction of an eigenvector associated with the first eigenvalue.

Second, multiply the vector  $x_k$  ( $k = 0, 1, \dots$ ) with the matrix  $A$  and normalize.

$$x_{k+1} = \frac{Ax_k}{\|Ax_k\|}$$

The subsequence  $(x_k)$  will converge if  $A$  has a unique greatest eigenvalue, and we can repeat the second step many times to obtain the approximation of the first eigenvector.

If the symmetric matrix  $A$  is not constrained to be positive semi-definite, the vector  $x_k$  will converge to the eigenvector with the greatest value of absolute eigenvalue, not the greatest eigenvalue. This is why we need to have the symmetric matrix to be positive semi-definite.

[ Reference ]: [https://en.wikipedia.org/wiki/Power\\_iteration](https://en.wikipedia.org/wiki/Power_iteration)