Assignment II

March 6, 2022

1 Generative Models, Naive Bayes Classifier

Based on the Table on page 13 of Lecture 6, compute the followings

Sky	Temp	Humid	Wind	Water	Forecast	Play
sunny	warm	normal	strong	warm	same	yes
sunny	warm	high	strong	warm	same	yes
rainy	cold	high	strong	warm	change	no
sunny	warm	high	strong	cool	change	yes

1. $p(\text{Water} = \text{warm}|\text{Play} = \text{yes}), p(\text{Water} = \text{warm}|\text{Play} = \text{no})$

   From above table, we can easily see that:

$$p(\text{Water} = \text{warm}|\text{Play} = \text{yes}) = \frac{2}{3} \quad p(\text{Water} \quad = \text{warm}|\text{Play} = \text{no}) = 1$$

2. $p(\text{Play} = \text{yes}|\text{Water} = \text{warm}), p(\text{Play} = \text{no}|\text{Water} = \text{warm})$

   From above table, we can easily see that:

$$\begin{aligned} p(\text{Play} = \text{yes}) &= \frac{3}{4} \quad p(\text{Play} = \text{no}) = \frac{1}{4} \\ p(\text{Water} = \text{warm}) &= \frac{3}{4} \quad p(\text{Water} = \text{cool}) = \frac{1}{4} \\ p(\text{Water} = \text{warm}|\text{Play} = \text{yes}) &= \frac{2}{3} \quad p(\text{Water} = \text{warm}|\text{Play} = \text{no}) = 1 \end{aligned}$$

   Hence by Bayes rule we have:

$$\begin{aligned} p(\text{Play} = \text{yes}|\text{Water} = \text{warm}) &= \frac{p(\text{Water} = \text{warm}|\text{Play} = \text{yes})p(\text{Play} = \text{yes})}{p(\text{Water} = \text{warm})} = \frac{2}{3} \\ p(\text{Play} = \text{no}|\text{Water} = \text{warm}) &= \frac{p(\text{Water} = \text{warm}|\text{Play} = \text{no})p(\text{Play} = \text{no})}{p(\text{Water} = \text{warm})} = \frac{1}{3} \end{aligned}$$

3. $p(\text{Play} = \text{yes}|\text{Forecast} = \text{same}), p(\text{Play} = \text{yes}|\text{Forecast} = \text{change})$

   From above table, we can easily see that:

$$\begin{aligned} p(\text{Play} = \text{yes}) &= \frac{3}{4} \quad p(\text{Play} = \text{no}) = \frac{1}{4} \\ p(\text{Forecast} = \text{same}) &= \frac{1}{2} \quad p(\text{Forecast} = \text{change}) = \frac{1}{2} \\ p(\text{Forecast} = \text{same}|\text{Play} = \text{yes}) &= \frac{2}{3} \quad p(\text{Forecast} = \text{change}|\text{Play} = \text{yes}) = \frac{1}{3} \end{aligned}$$

   Hence by Bayes rule we have:

$$\begin{aligned} p(\text{Play} = \text{yes}|\text{Forecast} = \text{same}) &= \frac{p(\text{Forecast} = \text{same}|\text{Play} = \text{yes})p(\text{Play} = \text{yes})}{p(\text{Forecast} = \text{same})} = 1 \\ p(\text{Play} = \text{yes}|\text{Forecast} = \text{change}) &= \frac{p(\text{Forecast} = \text{change}|\text{Play} = \text{yes})p(\text{Play} = \text{yes})}{p(\text{Forecast} = \text{change})} = \frac{1}{2} \end{aligned}$$

4. $p(\text{Water} = \text{warm}|\text{Play} = \text{yes}), p(\text{Water} = \text{warm}|\text{Play} = \text{no})$ with Laplace smoothing

$$p(\text{Water} = \text{warm}|\text{Play} = \text{yes}) = \frac{3}{5} \quad p(\text{Water} \quad = \text{warm}|\text{Play} = \text{no}) = \frac{2}{3}$$

2 Kernels

In this problem, we consider constructing new kernels by combining existing kernels. Recall that for some function $k(x, z)$ to be a kernel, we need to be able to write it as a dot product of vectors in some high-dimensional feature space defined by ϕ:

$$k(x, z) = \phi^T(x)\phi(z)$$

Mercer's theorem gives a necessary and sufficient condition for a function k to be a kernel function: its corresponding kernel matrix K has to be symmetric and positive semidefinite.
Suppose that $k_1(x, z)$ and $k_2(x, z)$ are two valid kernels. For each of the cases below, state whether k is also a valid kernel. If it is, prove it. If it is not, give a counterexample. You can use either Mercer's theorem, or the definition of a kernel as needed to prove it (If you use any properties on page 10 of Lecture 8, we need to prove them first).

1. $k(x, z) = a_1k_1(x, z) - a_2k_2(x, z)$, where $a_1, a_2 > 0$ are real numbers

   Disproof:

   If we consider $k_1(x, z)$ and $k_2(x, z)$ are same kernels and $a_1 = 1$ and $a_2 = 2$,then:

$$\begin{aligned} k(x, z) &= (a_1 - a_2)k_1(x, z) \\ &= -k_1(x, z) \end{aligned}$$

   the kernel matrix K of $k(x, z)$ is the negative of the kernel matrix of $k_1(x, z)$ and hence it is not positive semidefinite. then by Mercer's theorem, $k(x, z)$ is not a vaild kernel.

2. $k(x, z) = \sqrt{k_1(x, z)k_2(x, z)}$

   Proof:

   I will take this proof as two parts: 1) A square root of a valid kernel is also a kernel; 2) A product of two valid kernel k_1, k_2 is also a kernel.

   1) A square root of a valid kernel is also a kernel: suppose we have $k_1(x, z) = \phi_1^T(x)\phi_1(z)$, then we define $k(x, z)$ as:

$$k(x, z) = \sqrt{k_1(x, z)} = \sqrt{\phi_1^T(x)\phi_1(z)}$$

   Then we can define the feature mapping associated with $k(x, z)$ as $\phi(x) = \sqrt{\phi_1(x)}$.

   2) A product of two valid kernel k_1, k_2 is also a kernel: since k_1 and k_2 are valid kernels, we can say the kernel matrix K_1 of k_1 and K_2 of k_2 are positive semidefinite. Then we can write the following eigenvalue decomposition.

$$\begin{aligned} K_1 &= UDU^T = \sum_k d_k u_k u_k^T \\ K_2 &= VBV^T = \sum_k b_k v_k v_k^T \end{aligned} \tag{1}$$

   Where D, B are diagonal matrices with nonnegative entries $d_i, b_i \geq 0$ and , U and V are orthogonal matrices.Then we can write:

$$\begin{aligned} [K_1]_{ij} &= \sum_k d_k u_{ki} u_{kj} \\ [K_2]_{ij} &= \sum_k b_k v_{ki} v_{kj} \end{aligned} \tag{2}$$

   Then we need to prove $K = Z\Sigma Z^T$ is positive semidefinite.

$$\begin{aligned} [K]_{ij} &= \sum_{i,j} z_i z_j [K_1]_{ij}[K_2]_{ij} \\ &= \sum_{i,j} z_i z_j \sum_k d_k u_{ik} u_{jk} \sum_l b_l v_{il} v_{jl} \\ &= \sum_k \sum_l d_k b_l (\sum_i z_i u_{ik} v_{il})^2 \geq 0 \end{aligned} \tag{3}$$

   Hence, the kernel matrix is positive semidefinite and symmetric. the product of two kernel is also a kernel. Therefore $k(x, z) = \sqrt{k_1(x, z)k_2(x, z)}$ is a kernel.

3. if $k(x,z) = e^{\frac{x^T z}{\sigma^2}}$ is a valid kernel, prove that the Gussian kernel $k_G(x,z) = e^{-\frac{\|x-z\|_2^2}{2\sigma^2}}$ is also a valid kernel

   Proof:

   We can write $k_G(x,z) = e^{-\frac{\|x-z\|_2^2}{2\sigma^2}}$ as:

$$\begin{aligned} k_G(x,z) &= e^{-\frac{(x-z)^T(x-z)}{2\sigma^2}} \\ &= e^{-\frac{x^T x - 2x^T z + z^T z}{2\sigma^2}} \\ &= (k(-\frac{x}{\sqrt{2}}, \frac{x}{\sqrt{2}}))k(x,z)(k(\frac{z}{\sqrt{2}}, -\frac{z}{\sqrt{2}})) \end{aligned}$$

   Since k are valid kernels, there exist a feature mapping such that $k(x,z) = \phi^T(x)\phi(z)$, then:

$$k_G(x,z) = \phi^T(-\frac{x}{\sqrt{2}})\phi(\frac{x}{\sqrt{2}})\phi^T(x)\phi(z)\phi^T(\frac{z}{\sqrt{2}})\phi(-\frac{z}{\sqrt{2}})$$

   Hence we can define the feature mapping associated with $k(x,z)$ as $\phi_G(x) = \phi(x)\phi^T(\frac{x}{\sqrt{2}})\phi(-\frac{x}{\sqrt{2}})$.

3 PCA and Eigenface

For this exercise, you will use principal component analysis (PCA) to analyze face images in any programming language of your choice (e.g., Python/Matlab/R). The data set faces.dat; each row represents an image (400 images), and each column represents a pixel ($64 \times 64 = 4096$pixels).

1. Display the 200th image (since the original image was green so i applied gray to each image)

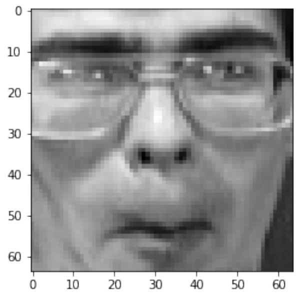


Figure 1: 200th image

2. Remove the mean of the images, and then display the 100th image.

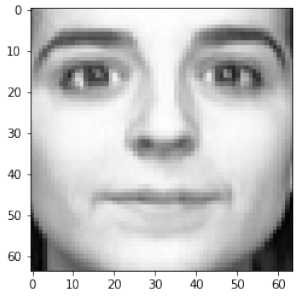
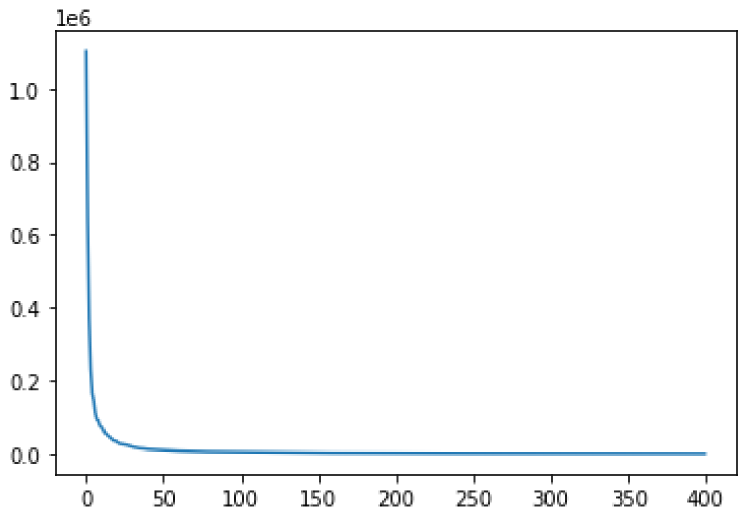


Figure 2: 100th image

3. Perform PCA on the mean-centered data matrix. You can either implement PCA by yourself using eigenvalue decomposition over the sample covariance matrix, or use a existing machine learning toolbox. Sort the eigenvalues in a descending order and plot them.



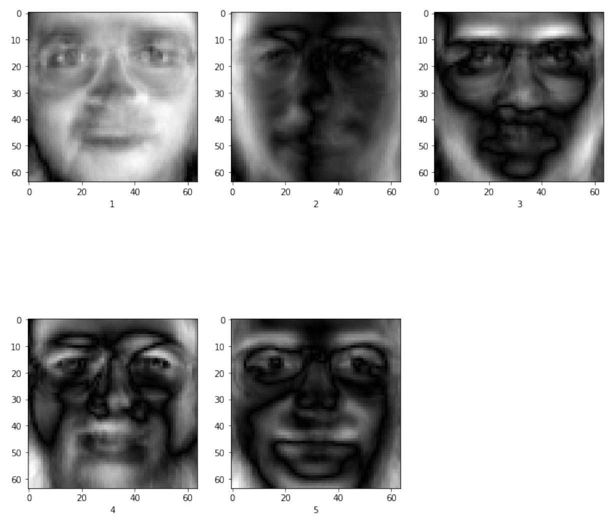
4. You will find the last (i.e., 400th) eigenvalue is 0. Explain why.

I think eigen(Cov(data)) is equal to eigen(Cov(data.T)) but this two terms has different shape, so the 400th eigenvalue is 0.

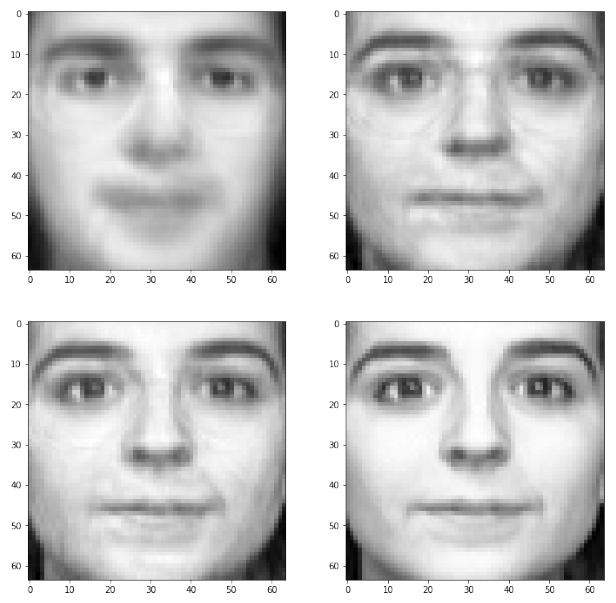
5. Based on the eigenvalues, determine the dimensionality of the data you want to keep (i.e., how many principal components you want to keep), which accounts for most of the variance. Explain your reason.

   I will keep the principal components as 100, PCA will bring a certain loss of accuracy when reducing the dimension, but it can roughly retain the main components. Keep the principal components as 100 will ensure we that the data distribution after dimensionality reduction is as large as possible, thus losing less details.

6. Display the top-5 leading eigenvectors (corresponding to the top-5 largest eigenvalues) in 5 figures.



7. Display, respectively, the reconstructed 100th images using 10, 100, 200, and 399 principal components. (Hint: In Lecture 9 (page 19), we have learned that $\hat{x} = vv^T x$ if we project x into 1-dimensional space using the 1st principal component. Reconstructed $\hat{x}$ using top-K principal components is a straightforward extension: $\hat{x} = \sum_{k=1}^{K} v_k {v_k}^T x$)



   Note: 10 principal components (top left), 100 principal components (top right), 200 principal components (bottom left) and 399 principal components (bottom right).

Hint: while each vector is 4096-dimensional, in order to display the images and principal components (i.e., 3(a), 3(b), 3(f), 3(g)), you should first reshape the 4096-dimensional vector into a 64×64 matrix.