CS 598 PS
Machine Learning for Signal Processing
Problem Set 1

Christian Howard
howard28@illinois.edu

Contents

1 Problem 1 - A Probability Problem

For this first problem, the goal is to find the probability a woman has breast cancer given she has a positive mammograph. To present the solution to this problem, let me first define a few things that will prove useful. First, below is the notation for the various events:

R_1 = women has breast cancer	R_2 = women does not have breast cancer
M^+ = mammogram is positive	M^- = mammogram is negative

In the problem statement, we are given the following information:

- The probability that a woman has breast cancer is 0.8%
- If a woman has breast cancer, there is a probability of 90% that the mammogram will be positive
- If she has no breast cancer, the probability of a positive mammogram is 7%

Based on the above information, we can find the following probabilities:

$$\begin{aligned} P(M^+|R_1) &= 0.9 \\ P(M^+|R_2) &= 0.07 \\ P(R_1) &= 0.008 \\ P(R_2) &= 0.992 \end{aligned}$$

Given the above information and definitions, our goal is to find $P(R_1|M^+)$. Using Baye's Rule, this becomes straight forward. The following is how we can arrive to the desired probability:

$$\begin{aligned} P(R_1|M^+) &= \frac{P(M^+|R_1)P(R_1)}{P(M^+)} \\ &= \frac{P(M^+|R_1)P(R_1)}{P(M^+|R_1)P(R_1) + P(M^+|R_2)P(R_2)} \\ &= \frac{(0.9)(0.008)}{(0.9)(0.008) + (0.07)(0.992)} \\ &= 0.0939 \\ &= 9.39\% \end{aligned}$$

Thus, we find that the probability of a woman with breast cancer getting a positive mammogram is 9.39%.

2 Problem 2 - Manipulating Data using Linear Algebra

2.1

I am given K grayscale images such that each image has dimension $M \times N$. For use as a data set, each image is vectorized and stuffed into a large data set matrix, D, that has dimension $(MN) \times K$.

2.1.a

The goal for this part is to return the mean image, let us call it μ_I, for the above data set and return it as an $M \times N$ matrix.

Given the data set is represented by the matrix D, one simple way to approach this would be to multiply this matrix by a vector that will do the averaging for the vectorized form of the images and then invert the vectorization of the result. This final operation can be defined as the following:

$$\mu_I = (D\boldsymbol{v})^{(M)}$$

where $\boldsymbol{v} = (v_1, v_2, \cdots, v_T)$ is defined such that $v_i = \frac{1}{K} \forall i$ and $T = MN$ represents the total number of pixels in one image in the data set.

2.1.b

The goal for this part is to compute a 2×2 covariance matrix where this covariance is based on the mean color for the top and bottom half of a given image for all K images in the data set.

Based on looking at patterns with simple D values, it was found the following could be defined as the matrix needed to take the full dataset D and return a new data set where each column has the desired average color values:

$$H_{(M,N)} = \mathbf{1}^T_{(N)} \otimes I \otimes \left(\frac{2}{M} \mathbf{1}^T_{(M/2)} \right)$$

where $\mathbf{1}_{(p)} \in \{1\}^{p \times 1}$ (i.e. a column vector of 1s of dimension p). Using this matrix, and using the same $\boldsymbol{v}$ from the last problem, we can compute the covariance Σ using the following:

$$\begin{aligned} D_\mu &= H_{(M,N)} D \\ \boldsymbol{\mu} &= D_\mu \boldsymbol{v} \\ \boldsymbol{\delta} &= D_\mu - \mathbf{1}^T_{(K)} \otimes \boldsymbol{\mu} \\ \Sigma &= \frac{1}{K} \boldsymbol{\delta} \boldsymbol{\delta}^T \end{aligned}$$

2.2

To make things more interesting, now we are given a set of K colored images where this whole data set is represented as a $M \times N \times 3 \times K$ tensor called D, where the dimension of size 3 is the color channel dimension.

2.2.a

For this part we are asked to compute the average color image, which I will define as μ_c. First, for notational simplicity, let us define I_p as an identity matrix of size $p \times p$. With this, we can find an operator to find the average image in its vectorized form using the following operation:

$$\mu_c = \left(\frac{1}{K} \mathbf{1}_{(K)}^T \otimes I_3 \otimes I_N \otimes I_M \right) \text{vec}(D)$$

Working left to right in the Kronecker product sequence, the first term represents an averaging row vector. This vector then undergoes a Kronecker product with a sequence of identity matrices of dimensions 3×3, $N \times N$, and $M \times M$ going from left to right. The resulting matrix, after all the Kronecker products, will take the vectorized data set and produce a vectorized average color image, μ_c.

2.2.b

This next part is to instead compute the average image within just the red channel of the matrix. We will keep in mind that since the image is RGB, red will be the first channel in the third tensor dimension. With this defined, we can find the average vectorized red channel image, μ_r, using the below operation:

$$\mu_r = \left(\frac{1}{K} \mathbf{1}_{(K)}^T \otimes [1, 0, 0] \otimes I_N \otimes I_M \right) \text{vec}(D)$$

Using the above operation, the second term $[1, 0, 0]$ effectively zeros out contributions from the green and blue channels, producing an average vectorized $M \times N$ image which represents the average red channel image. If we wanted to go further and find the full $M \times N$ red image, called it R_μ, we can obtain it simply using the vector transpose like so:

$$R_\mu = (\mu_r)^{(M)}$$

3 Problem 3 - Signal Processing is Linear Algebra