CS 598: Machine Learning for Signal Processing

Fall 2017, Homework 1

Yinrui Li

Problem 1:

1. We denote 'cancer' by 'C', and 'mammogram' by 'M'. If a person has cancer, we set $C = 1$, and otherwise $C = 0$. Likewise, if a person has positive mammogram, we set $M = 1$ and otherwise $M = 0$.

   According to the question:

$$\begin{aligned} P(C = 1) &= 0.8\% \\ P(M = 1|C = 1) &= 90\% \\ P(M = 1|C = 0) &= 7\% \\ P(C = 1) &= 1 - 0.8\% = 99.2\% \end{aligned}$$

   By Bayesian Theory:

$$\begin{aligned} P(C = 1|M = 1) &= \frac{P(C = 1, M = 1)}{P(M = 1)} \\ &= \frac{P(M = 1|C = 1)P(C = 1)}{P(M = 1|C = 1)P(C = 1) + P(M = 1|C = 0)P(C = 0)} \\ &= \frac{90\% \times 0.8\%}{90\% \times 0.8\% + 7\% \times 99.2\%} \\ &\approx 9.39\% \end{aligned}$$

   So the probability that a woman has breast cancer given that she has positive mammogram is 9.39%.

Problem 2:

1. (a) We denote the $(M \times N) \times K$ matrix of the K gray scale images as A:

$$A_{\text{Image}} = \begin{bmatrix} x_{11} & x_{12} & \cdots & x_{1K} \\ x_{21} & x_{22} & \cdots & x_{2K} \\ \vdots & \vdots & \ddots & \vdots \\ x_{(M*N)1} & x_{(M*N)2} & \cdots & x_{(M*N)K} \end{bmatrix}$$

The mean matrix M can be computed as the matrix product of A and a **K-dimensional** column T. The value of each element in T is (1/K):

$$A_{\text{mean}} = A_{\text{Image}} \cdot T = \begin{bmatrix} x_{11} & x_{12} & \dots & x_{1K} \\ x_{21} & x_{22} & \dots & x_{2K} \\ \vdots & \vdots & \ddots & \vdots \\ x_{(M*N)1} & x_{(M*N)2} & \dots & x_{(M*N)K} \end{bmatrix} \begin{bmatrix} \frac{1}{K} \\ \frac{1}{K} \\ \vdots \\ \frac{1}{K} \end{bmatrix} = \begin{bmatrix} \frac{1}{K}\sum_1^K x_{1i} \\ \frac{1}{K}\sum_1^K x_{2i} \\ \vdots \\ \frac{1}{K}\sum_1^K x_{(M*N)i} \end{bmatrix}$$

The mean $(M \times N)$ matrix M can be derived by the vec–transpose of the mean column:

$$M = \text{Vec}(A_{\text{mean}})^{(M)}$$

(b) Let W be a 2*(M*N) matrix. We denote its entries by $W_{i,j}$.

Let $k = 0, 2, ..., N-1$,

$$W_{i,j} = \begin{cases} \frac{2}{M*N}, \text{ when } i = 1, j = M*k+1, M*k+2, ..., M*k+\frac{M}{2} \\ \frac{2}{M*N}, \text{ when } i = 0, j = M*k+\frac{M}{2}+1, M*k+\frac{M}{2}+2, ..., M*k+M \\ 0, \text{ when } i = 1, j = M*k+\frac{M}{2}+1, M*k+\frac{M}{2}+2, ..., M*k+M \\ 0, \text{ when } i = 0, j = M*k+1, M*k+2, ..., M*k+\frac{M}{2} \end{cases}$$

$$W = \begin{bmatrix} (\frac{2}{M*N})_1 & 0 \\ \vdots & \vdots \\ (\frac{2}{M*N})_{\frac{M}{2}} & 0 \\ 0 & (\frac{2}{M*N})_{\frac{M}{2}+1} \\ \vdots & \vdots \\ 0 & (\frac{2}{M*N})_M \\ , & , \\ \vdots & \vdots \\ , & , \\ (\frac{2}{M*N})_{M*(N-1)+1} & 0 \\ \vdots & \vdots \\ (\frac{2}{M*N})_{M*(N-1)+\frac{M}{2}} & 0 \\ 0 & (\frac{2}{M*N})_{M*(N-1)+\frac{M}{2}+1} \\ \vdots & \vdots \\ 0 & (\frac{2}{M*N})_{M*N} \end{bmatrix}^T$$

The Image matrix:

$$A_{\text{Image}} = \begin{bmatrix} x_{11} & x_{12} & \dots & x_{1K} \\ x_{21} & x_{22} & \dots & x_{2K} \\ \vdots & \vdots & \ddots & \vdots \\ x_{(M*N)1} & x_{(M*N)2} & \dots & x_{(M*N)K} \end{bmatrix}$$

The mean of the upper half and the bottom half of each of the images can be computed as:

$$M = W \cdot A_{\text{Image}} = \begin{bmatrix} u_1 & u_2 & \dots & u_K \\ b_1 & b_2 & \dots & b_K \end{bmatrix}$$

So the mean of the upper and bottom half is:

$$\begin{bmatrix} \bar{u} \\ \bar{b} \end{bmatrix} = \begin{bmatrix} u_1 & u_2 & \dots & u_K \\ b_1 & b_2 & \dots & b_K \end{bmatrix} \cdot \left.\begin{bmatrix} \frac{1}{K} & \frac{1}{K} \\ \frac{1}{K} & \frac{1}{K} \\ \vdots & \vdots \\ \frac{1}{K} & \frac{1}{K} \end{bmatrix}\right\} (2 \times \text{K}) \text{ matrix}$$

The covariance matrix is:

$$E[(u-\mu_u)(b-\mu_b)^T] \approx \frac{1}{K-1}\sum_{k=1}^{K}[(u_k-\bar{u})(b_k-\bar{b})^T]$$

$$= \begin{bmatrix} u_1-\bar{u} & u_2-\bar{u} & \dots & u_K-\bar{u} \\ b_1-\bar{b} & b_2-\bar{b} & \dots & b_K-\bar{b} \end{bmatrix} \cdot \begin{bmatrix} u_1-\bar{u} & u_2-\bar{u} & \dots & u_K-\bar{u} \\ b_1-\bar{b} & b_2-\bar{b} & \dots & b_K-\bar{b} \end{bmatrix}^T$$

2. (a) We denote the M*N*3*K tensor by X.

1. Apply vec operator and convert X to a M*N*3*K vector.

2. Do vec-transpose and convert the vector to a $(M*N)\times(3*K)$ matrix. Each column represents a single color channel of an image.

3. The mean can be derived by multiplying the matrix with a 3K-dimensional vector and then applying vec-transpose to convert it back to a M*N tensor:

$$\text{vec}(X) = \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_{M*N*3*K} \end{bmatrix}$$

$$\text{vec}(X)^{(M*N)} = \overbrace{\begin{bmatrix} x_{1,r,1} & x_{1,g,2} & \cdots & x_{1,b,K} \\ x_{2,r,1} & x_{2,g,2} & \cdots & x_{2,b,K} \\ \vdots & \vdots & \ddots & \vdots \\ x_{(M*N),r,1} & x_{(M*N),g,2} & \cdots & x_{(M*N),b,K} \end{bmatrix}}^{3K \text{ columns}} = \begin{bmatrix} x_{11} & x_{12} & \cdots & x_{1(3K)} \\ x_{21} & x_{22} & \cdots & x_{2(3K)} \\ \vdots & \vdots & \ddots & \vdots \\ x_{(M*N)1} & x_{(M*N)2} & \cdots & x_{(M*N)(3K)} \end{bmatrix}$$

$$X_{\text{mean vec}} = \text{vec}(X)^{(M*N)} \cdot T = \begin{bmatrix} x_{11} & x_{12} & \cdots & x_{1(3K)} \\ x_{21} & x_{22} & \cdots & x_{2(3K)} \\ \vdots & \vdots & \ddots & \vdots \\ x_{(M*N)1} & x_{(M*N)2} & \cdots & x_{(M*N)(3K)} \end{bmatrix} \left.\begin{bmatrix} \frac{1}{3K} \\ \frac{1}{3K} \\ \vdots \\ \frac{1}{3K} \end{bmatrix}\right\} \text{3K-dimontional vector}$$

$$= \begin{bmatrix} \frac{1}{3K}\sum_{i=1}^{3K} x_{1i} \\ \frac{1}{3K}\sum_{i=1}^{3K} x_{2i} \\ \vdots \\ \frac{1}{3K}\sum_{i=1}^{3K} x_{(M*N)i} \end{bmatrix}$$

The mean M*N matrix:

$$X_{\text{mean}} = \text{vec}(X_{\text{mean vec}})^{(M)} = \text{vec}(\text{vec}(X)^{(M*N)} \cdot T)^{(M)}$$

(b) We denote the M*N*3*K tensor by X.

1. Apply vec operator to convert X to a M*N*3*K vector.

2. Do vec-transpose to convert the vector to a $(M*N) \times (3*K)$ matrix, and multiply the matrix with a 3K-dimensional vector in the form (1/k, 0, 0, 1/k, 0, 0, ...), and then apply vec-transpose to convert it back to a M*N tensor:

$$\text{vec}(X) = \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_{M*N*3*K} \end{bmatrix}$$

$$X_{\text{mean vec}} = \text{vec}(X)^{(M*N)} \cdot T_{red} = \begin{bmatrix} x_{11} & x_{12} & \cdots & x_{1(3K)} \\ x_{21} & x_{22} & \cdots & x_{2(3K)} \\ \vdots & \vdots & \ddots & \vdots \\ x_{(M*N)1} & x_{(M*N)2} & \cdots & x_{(M*N)(3K)} \end{bmatrix} \left.\begin{bmatrix} \frac{1}{K} \\ 0 \\ 0 \\ \frac{1}{K} \\ 0 \\ 0 \\ \vdots \\ \frac{1}{K} \\ 0 \\ 0 \end{bmatrix}\right\} \text{3K-dimontional vector}$$

$$= \begin{bmatrix} \frac{1}{K}\sum_{i=1}^{K} x_{1(3i-2)} \\ \frac{1}{K}\sum_{i=1}^{K} x_{2(3i-2)} \\ \vdots \\ \frac{1}{K}\sum_{i=1}^{K} x_{(M*N)(3i-2)} \end{bmatrix}$$

In matrix T_{red}, $T_i = 1/K$ when (i-1) is divisible by 3 (i=1, 4, 7, ..., 3K-2), otherwise $T_i = 0$. T_{red} can be derived by:

$$T_{red} = vec([1, 0, 0]^T \overbrace{[1, 1, ..., 1]}^{Kdimentional})$$

The mean M*N matrix for red channel:

$$X_{\text{mean,red}} = \text{vec}(X_{\text{mean vec}})^{(M)} = \text{vec}(\text{vec}(X)^{(M*N)} \cdot T_{red})^{(M)}$$

Problem 3:

1. Assume x is a N-dimensional vector.

   The 1024×1024 Fourier transformation matrix is:

$$F = \frac{1}{n} \begin{bmatrix} 1 & 1 & 1 & \cdots & 1 \\ 1 & \omega & \omega^2 & \cdots & \omega^n - 1 \\ 1 & \omega^2 & \omega^4 & \cdots & \omega^{2(n-1)} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & \omega^{n-1} & \omega^{2(n-1)} & \cdots & \omega^{(n-1)(n-1)} \end{bmatrix} n = 1024, \omega = e^{-2\pi i/n}$$

The Hann function is:

$$h(n) = \frac{1}{2}(1 - cos(\frac{2\pi n}{1024 - 1})), n = 0, 1, ..., 1023$$

So we can design a 1024×1024 hann window matrix H, and multiply it with the Fourier transformation matrix, in order to perform both fourier and hann window transformation:

$$H = \begin{bmatrix} h(0) & 0 & 0 & \dots & 0 \\ 0 & h(1) & 0 & \dots & 0 \\ 0 & 0 & h(2) & \dots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \dots & h(n-1) \end{bmatrix}$$

$$F \cdot H = \begin{bmatrix} h(0) & h(1) & h(2) & \dots & h(1023) \\ h(0) & h(1)\omega & h(2)\omega^2 & \dots & h(1023)\omega^n - 1 \\ h(0) & h(1)\omega^2 & h(2)\omega^4 & \dots & h(1023)\omega^{2(n-1)} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ h(0) & h(1)\omega^{n-1} & h(2)\omega^{2(n-1)} & \dots & h(1023)\omega^{(n-1)(n-1)} \end{bmatrix}$$

Considering a hop size of 512, the matrix to perform DFT and hann window for the whole vector X is:

$$W = \begin{bmatrix} [\quad F \cdot H \quad] & \dots & \dots & \dots & \dots \\ \dots & [\quad F \cdot H \quad] & \dots & \dots & \dots \\ \dots & \dots & [\quad F \cdot H \quad] & \dots & \dots \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ \dots & \dots & \dots & \dots & [\quad F \cdot H \quad] \end{bmatrix}$$ W is a $(1024 \times k) \times (512 + 512 \times k)$ matrix.

The number of transformation matrices $F \cdot H$ in W is computed as k:

$$k = \lceil \frac{N - 1024 + 512}{512} \rceil$$ (k is **rounded up** to the nearest integer)

The first $F \cdot H$ matrix lies in $W_{i,j}, i, j = 1, 2, ..., 1024$.

The second $F \cdot H$ matrix lies in $W_{i,j}, i = 1025, ..., 2048, j = 513, ..., 1536$.

The third $F \cdot H$ matrix lies in $W_{i,j}, i = 2049, ..., 3072, j = 1025, ..., 2048$.

...

The last $F \cdot H$ matrix lies in $W_{i,j}, i = 1024 * (k-1) + 1, ..., 1024 * k, j = 512 * (k-1) + 1, ..., 512 + 512 * k$.

Because k is rounded-up to the nearest integer, we also need to do zero-padding if N is not divisible by 512. The transformation matrix Z s is:

$$Z = \begin{bmatrix} 1 & 0 & 0 & \dots & 0 \\ 0 & 1 & 0 & \dots & 0 \\ 0 & 0 & 1 & \dots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \dots & 1 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \dots & 0 \\ 0 & 0 & 0 & \dots & 0 \\ 0 & 0 & 0 & \dots & 0 \end{bmatrix} \text{Z is a (512+512} \times \text{k)} \times \text{N matrix}$$

Therefore, we have:

$$A = W \cdot Z$$

Ax is a $(1024 \times k)$ dimensional vector.