EE5175: Image Signal Processing - Lab 9 report

Ruban Vishnu Pandian V (EE19B138)

April 2, 2023

Contents

1 DCT, Walsh-Hadamard transform and SVD

Aim:

- To obtain the energy packing efficiency and data de-correlation efficiency of the DCT and Walsh-Hadamard transforms when applied on a 8X8 covariance matrix of a Markov-1 process with $\rho = 0.91$.
- To find $\beta^2 R^{-1}$, check whether it has the given tri-diagonal structure and diagonalize it using the DCT matrix.
- To compute SVD for a given image **g** by finding the eigenvalue decompositions of matrices **gg'** and **g'g**. Reconstruct **g** using SVD and compare with the original image **g**.
- To reconstruct **g** by removing one singular value at a time and compare the error obtained with sum of squares of first **k** singular values.

2 Discrete Cosine and Walsh-Hadamard transforms:

2.1 Discrete Cosine transform (DCT):

Similar to the Discrete Fourier transform (DFT), Discrete Cosine transform (DCT) is a unitary transform with cosines as the basis functions. Let us have a 1-D signal $x[n]$ for $n = 0, 1, ...N-1$. Then the DCT coefficients X[k] are given as:

$$X[k] = \alpha(k) \sum_{n=0}^{N-1} x[n] cos(\frac{2\pi(2n+1)k}{2N})$$

$$alpha(k) = \begin{cases} \frac{1}{\sqrt{N}} & n = 0 \\ \frac{2}{\sqrt{N}} & n \neq 0 \end{cases}$$

DCT can be related to the DFT of an extended 1-D signal $y[n]$ which is given as:

$$y[n] = \begin{cases} x[n] & 0 \leq n \leq N-1 \\ x[2N-1-n] & N \leq n \leq 2N-1 \end{cases}$$

We know that DFT computes the frequency components of the periodic extension of the base N-length input signal. And fewer coefficients are needed if there are no sudden jumps in the signal, i.e., high frequency components are very insignificant. So for a general $x[n]$, the DCT will be a better transform for data compression purpose since irrespective of what $x[n]$ is, the sudden jumps due to periodic extension is avoided in $y[n]$. This purely comes from the way in which $y[n]$ is constructed from $x[n]$.

2.2 Walsh-Hadamard transform (WHT):

WHT is a unitary transform whose basis functions are square waves (made up of +1 and -1) unlike the DFT where the basis functions are complex exponentials. Similar to DFT, WHT can also be used as a spectral analysis tool to analyze signals. However, it's best suited for signals with sudden changes since we require more Walsh functions to fit a smooth curve (Similar to the high frequency sinusoids required to model a sudden change).

Mathematically, it is given by a recursive model:

$$H_1 = \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix}$$

$$H_n = \begin{bmatrix} H_{n-1} & H_{n-1} \\ H_{n-1} & -H_{n-1} \end{bmatrix}$$

In a more compact form, we can write it as $H_n = H_1 \otimes H_{n-1}$ where $\otimes$ denotes **Tensor/Kronecker** product. From this recursion and using properties of the Kronecker product, we can clearly say that:

$$H_n = H_1 \otimes H_1 \otimes ...H_1$$

(n times)

In the given assignment, $n = 3$ since we are dealing with 8X8 matrices.

3 Singular value decomposition (SVD):

Let **S** be a MxN matrix. Then its singular value decomposition is given as:

$$S = U\Sigma V^H$$

where U (MxM),V (NxN) are unitary matrices, Σ (MXN) is a diagonal matrix with the singular values of S as its diagonal elements and V^H denotes Hermitian/Conjugate transpose of V. U is made up of the left eigenvectors of S as its columns and V using the right eigenvectors of S as its columns.

Singular values are guaranteed to be non-negative. Hence, the decomposition is usually done in such a way that Σ is in the descending order of singular values and U, V are obtained accordingly. These matrices are obtained from the eigenvalue decomposition of the transpose product matrices as follows:

$$gg^H = A\Sigma\Sigma^H A^H$$

$$g^H g = B\Sigma^H \Sigma B^H$$

The singular values will simply be the square roots of the eigenvalues of gg^H (or $g^H g$). However, the issue with this method is that it has sign ambiguity. That is A and B individually can have any sign for their eigenvectors since they multiply with themselves in the individual eigenvalue decomposition. Hence, when they are multiplied together along with Σ to produce g, the result may not be always correct.

One way in which this can be resolved is to check how the vector gb_i relates to $s_i a_i$ where a_i, b_i, s_i are i^{th} column of A, i^{th} column vector of B and i^{th} singular value respectively. If they are equal, then the column vector b_i is already correct. But if they differ by a sign change, then the new column vector of B is obtained by sign-inverting b_i. Note that this new matrix B is still in accordance with the eigenvalue decomposition of $g^H g$. This is only done to ensure that $g = A\Sigma B^H$.

4 Low rank approximation:

We can re-write the SVD in summation form as:

$$g = \sum_{k=1}^{P} s_k a_k b_k^H$$

where P is the rank of g. Often times, most of the significant information will be captured by the first few singular values, i.e., they will be comparatively very large than the remaining singular values. Let's assume only L (¡P) singular values are used to construct g. Let this reconstructed matrix be $g(L)$. Then, the we have the following formulae:

$$g(L) = \sum_{k=1}^{L} s_k a_k b_k^H$$

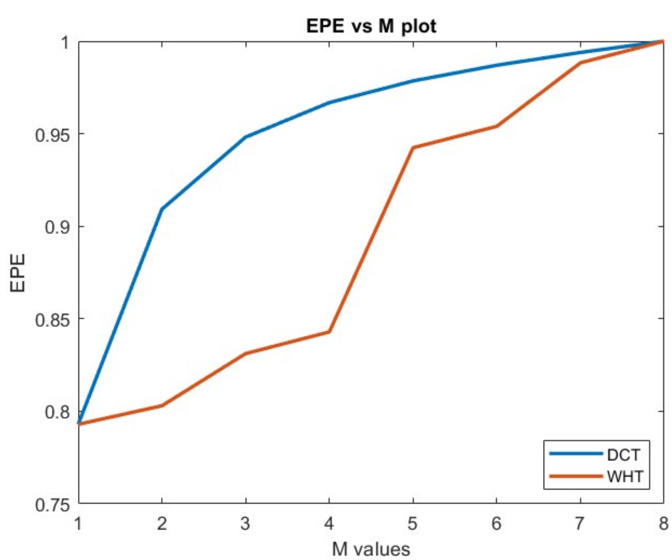
$$||g - g(L)||^2 = \sum_{k=L+1}^{P} s_k^2$$

where $||A||$ denotes Frobenius norm of the matrix A. Hence, depending on the accepted error tolerance, we can use a low rank approximation instead of the original matrix since computation required and storage would be lower for a low rank approximation (Fewer computations to make and fewer eigenvectors to store).

5 Results:

5.1 Question 1:

The covariance matrix is transformed using both the DCT and WHT transforms and the plot for energy packing efficiency is given below:



DCT seems to have worked better than WHT in terms of the EPE metric. Most of the energy is stored in the first few coefficients as compared to WHT.

The data decorrelation efficiency (DDE) value for DCT is 0.9805 and for WHT, it is 0.9486. Again, DCT has better DDE than WHT but only by a very small margin.
The eigenvectors of R and the basis vectors of DCT matrix are quite similar. They have certain differences but the diagonal elements obtained when the DCT transform is applied are almost equal to the eigenvalues of R.

5.2 Question 2:

The matrix $\beta^2 R^{-1}$ is given below:

```
Q_est =

    0.5470   -0.4978   -0.0000   -0.0000   -0.0000   -0.0000    0.0000   -0.0000
   -0.4978    1.0000   -0.4978   -0.0000    0.0000    0.0000   -0.0000    0.0000
   -0.0000   -0.4978    1.0000   -0.4978    0.0000   -0.0000    0.0000   -0.0000
   -0.0000   -0.0000   -0.4978    1.0000   -0.4978    0.0000   -0.0000    0.0000
   -0.0000    0.0000    0.0000   -0.4978    1.0000   -0.4978   -0.0000    0.0000
   -0.0000    0.0000   -0.0000    0.0000   -0.4978    1.0000   -0.4978   -0.0000
    0.0000   -0.0000    0.0000   -0.0000   -0.0000   -0.4978    1.0000   -0.4978
   -0.0000    0.0000   -0.0000    0.0000    0.0000   -0.0000   -0.4978    0.5470
```

And the tri-diagonal matrix Q given in the assignment is given below:

```
Q_act =

    0.5022   -0.4978         0         0         0         0         0         0
   -0.4978    1.0000   -0.4978         0         0         0         0         0
         0   -0.4978    1.0000   -0.4978         0         0         0         0
         0         0   -0.4978    1.0000   -0.4978         0         0         0
         0         0         0   -0.4978    1.0000   -0.4978         0         0
         0         0         0         0   -0.4978    1.0000   -0.4978         0
         0         0         0         0         0   -0.4978    1.0000   -0.4978
         0         0         0         0         0         0   -0.4978    0.5022
```

It is clearly evident that they are almost the same, i.e., $\beta^2 R^{-1}$ has a tri-diagonal structure given by Q. The DCT transform is applied on both the matrices and the results are given below:

```
Q_est_diag =

    0.0156    0.0000    0.0146   -0.0000    0.0112    0.0000    0.0061    0.0000
    0.0000    0.1018   -0.0000    0.0183    0.0000    0.0122    0.0000    0.0043
    0.0146    0.0000    0.3151    0.0000    0.0146    0.0000    0.0079    0.0000
   -0.0000    0.0183    0.0000    0.6345    0.0000    0.0103    0.0000    0.0036
    0.0112    0.0000    0.0146    0.0000    1.0112    0.0000    0.0061    0.0000
    0.0000    0.0122    0.0000    0.0103    0.0000    1.3879    0.0000    0.0024
    0.0061    0.0000    0.0079    0.0000    0.0061    0.0000    1.7073    0.0000
   -0.0000    0.0043    0.0000    0.0036    0.0000    0.0024    0.0000    1.9206
```

Diagonalization of $\beta^2 R^{-1}$

```
Q_act_diag =

    0.0044   -0.0000    0.0000   -0.0000    0.0000    0.0000   -0.0000    0.0000
    0.0000    0.0802   -0.0000    0.0000   -0.0000   -0.0000    0.0000   -0.0000
    0.0000   -0.0000    0.2960    0.0000   -0.0000    0.0000   -0.0000    0.0000
   -0.0000    0.0000    0.0000    0.6190    0.0000   -0.0000    0.0000   -0.0000
    0.0000   -0.0000   -0.0000    0.0000    1.0000    0.0000   -0.0000   -0.0000
    0.0000   -0.0000    0.0000   -0.0000    0.0000    1.3810    0.0000    0.0000
   -0.0000    0.0000   -0.0000    0.0000   -0.0000    0.0000    1.7040   -0.0000
    0.0000   -0.0000    0.0000   -0.0000   -0.0000    0.0000   -0.0000    1.9198
```

Diagonalization of Q

It seems like DCT has perfectly diagonalized the actual Q matrix (The result of the transformation is a diagonal matrix). But it hasn't diagonalized $\beta^2 R^{-1}$. It has closely diagonalized it, not exactly. Most of the diagonal values match with the actual values but there are some non-zero off-diagonal elements too.

5.3 Question 3:

SVD of the input image is computed and it is used to reconstruct the image. The results are given below:

```
g =

   255   255   255   255   255   255   255   255
   255   255   255   100   100   100   255   255
   255   255   100   150   150   150   100   255
   255   255   100   150   200   150   100   255
   255   255   100   150   150   150   100   255
   255   255   255   100   100   100   255   255
   255   255   255   255    50   255   255   255
    50    50    50    50   255   255   255   255
```

Actual input image

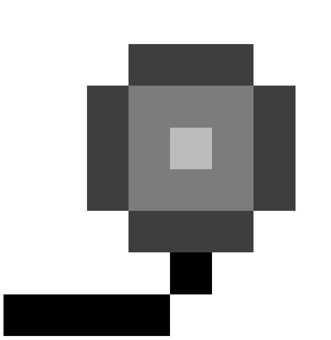
```
g_rec =

  255.0000  255.0000  255.0000  255.0000  255.0000  255.0000  255.0000  255.0000
  255.0000  255.0000  255.0000  100.0000  100.0000  100.0000  255.0000  255.0000
  255.0000  255.0000  100.0000  150.0000  150.0000  150.0000  100.0000  255.0000
  255.0000  255.0000  100.0000  150.0000  200.0000  150.0000  100.0000  255.0000
  255.0000  255.0000  100.0000  150.0000  150.0000  150.0000  100.0000  255.0000
  255.0000  255.0000  255.0000  100.0000  100.0000  100.0000  255.0000  255.0000
  255.0000  255.0000  255.0000  255.0000   50.0000  255.0000  255.0000  255.0000
   50.0000   50.0000   50.0000   50.0000  255.0000  255.0000  255.0000  255.0000
```

Reconstructed image

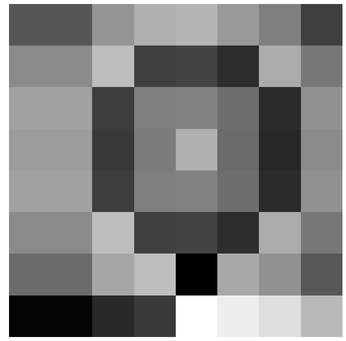
The reconstructed image perfectly matches the original image since SVD is an exact decomposition, not an approximation.

5.4 Question 4:

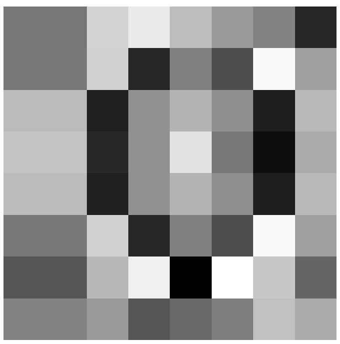
Using SVD, the image was reconstructed by removing one singular value at a time. The results are given below:



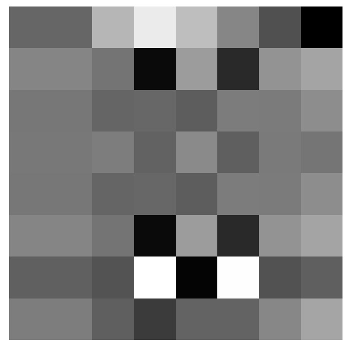
Original image



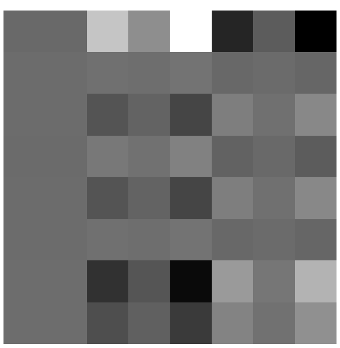
One singular value removed



Two singular value removed



Three singular values removed



Four singular values removed

All singular values removed

It is evident that most of the information was present in the first two singular values since after that, the images don't look convincing. The squared error between the original and reconstructed images match the sum of squares of first k singular values too.

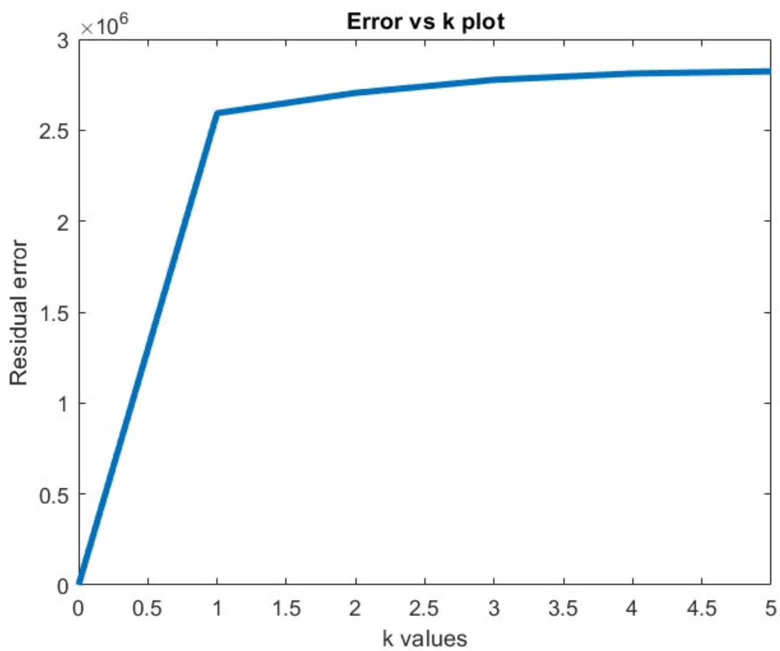
```
error_obs =

   1.0e+06 *

         0    2.5934    2.7050    2.7768    2.8116    2.8234
```

Error observed from the reconstructed images

```
error_calc =

   1.0e+06 *

         0    2.5934    2.7050    2.7768    2.8116    2.8234
```

Error calculated using singular values

The plot of errors is also given below:



Error vs k plot

6 Observations and Conclusions

1. DCT performs better than WHT in terms of both EPE and DDE for a Markov-1 process. It is to be noted that this wokrs if ρ is close to 1 (0.91 in our case).

2. Eigenvectors of the R matrix are very similar to the DCT matrix basis columns.

3. $\beta^2 R^{-1}$ matrix almost has a similar structure given by Q. The DCT matrix almost diagonalizes it.

4. SVD exactly reconstructs the input image since it is an exact decomposition of the input matrix.

5. Low rank approximations of SVD work well if the singular values considered in the approximation are high (Since they contribute more weight to the approximation). The error plots too reflect it.