MAT 167 WQ 2022 CA 01 Report

Randy Li

February 4, 2024

Contents

1   Introduction and Motivation

Linear Algebra can be used widely in our real applications, such as machine learning and pattern recognition, data mining and search engine, and signal and image processing. All these application use matrix transformation to process the data. This report will explore specifically about signal processing. Signal processing is a the process of converting the compression or encoded audio into the form closer to the original. In this project, we will do signal processing by compressing each signal matrix to differnt extents and accomplish that by using matlab.

2   Data Set

The data set for signal processing is a matrix file called `CA_01.mat`, which is file downloaded from canvas.
Since, we have load the matrix, x will be the signal matrix data we used. U will be the orthonormal matrix of the original matrix.
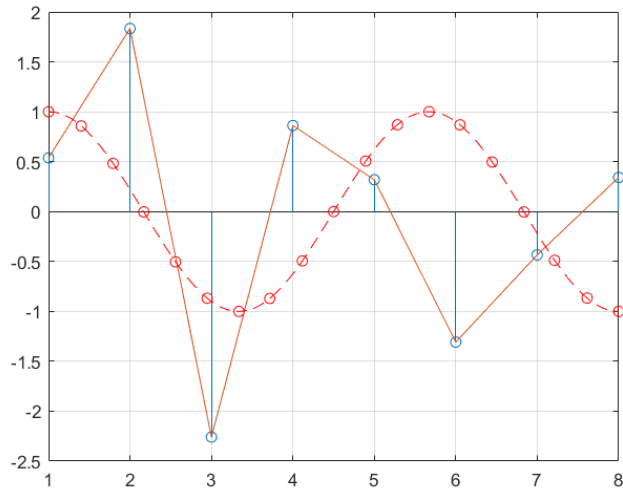
3   Procedure

3.1   Loading and Understanding the Data

Part a)
Step1: Use "load `CA_01.mat`" to load the matrix to the project.
Step2: Draw the signal x
Here is the original figure:

3.2   Finding Orthonormal Basis Vectors

b) Find the orthonormal basis vectors by using the Discrete Cosine Transform formula:

$$f_k(x) = \cos\left(\frac{k\pi(x-1)}{7}\right), \quad k \in \{0, 1, 2, 3, 4, 5, 6, 7\} \tag{1}$$

Step2: Create a matrix A of orthonormal basis vectors by using the following code:

```
A=[A arrayfun(@(x) cos(k*pi*(x-1)/7), [1:8]')];
```

Each column vectors will be:

$$\begin{bmatrix} \cos\left(\frac{2\pi(1-1)}{7}\right) \\ \cos\left(\frac{2\pi(2-1)}{7}\right) \\ \cos\left(\frac{2\pi(3-1)}{7}\right) \\ \cos\left(\frac{2\pi(4-1)}{7}\right) \\ \cos\left(\frac{2\pi(5-1)}{7}\right) \\ \cos\left(\frac{2\pi(6-1)}{7}\right) \\ \cos\left(\frac{2\pi(7-1)}{7}\right) \\ \cos\left(\frac{2\pi(8-1)}{7}\right) \end{bmatrix} \tag{2}$$

Here is the Matrix A:

```
A=
    1.0000    1.0000    1.0000    1.0000    1.0000    1.0000    1.0000    1.0000
    1.0000    0.9010    0.6235    0.2225   -0.2225   -0.6235   -0.9010   -1.0000
    1.0000    0.6235   -0.2225   -0.9010   -0.9010   -0.2225    0.6235    1.0000
    1.0000    0.2225   -0.9010   -0.6235    0.6235    0.9010   -0.2225   -1.0000
    1.0000   -0.2225   -0.9010    0.6235    0.6235   -0.9010   -0.2225    1.0000
    1.0000   -0.6235   -0.2225    0.9010   -0.9010    0.2225    0.6235   -1.0000
    1.0000   -0.9010    0.6235   -0.2225   -0.2225    0.6235   -0.9010    1.0000
    1.0000   -1.0000    1.0000   -1.0000    1.0000   -1.0000    1.0000   -1.0000
```

Step 3: Determine whether A is orthonormal or not by using the formula $A^T A$.

Here is the result of $A^T A$:

```
A^TA =8.0000         0    1.0000    0.0000    1.0000    0.0000    1.0000         0
           0    4.5000         0    1.0000    0.0000    1.0000   -0.0000    1.0000
      1.0000         0    4.5000    0.0000    1.0000    0.0000    1.0000   -0.0000
      0.0000    1.0000    0.0000    4.5000   -0.0000    1.0000    0.0000    1.0000
      1.0000    0.0000    1.0000   -0.0000    4.5000    0.0000    1.0000   -0.0000
      0.0000    1.0000    0.0000    1.0000    0.0000    4.5000   -0.0000    1.0000
      1.0000   -0.0000    1.0000    0.0000    1.0000   -0.0000    4.5000   -0.0000
           0    1.0000   -0.0000    1.0000   -0.0000    1.0000   -0.0000    8.0000
```

According to the result part of $A^T A$, since it is not an identity matrix, A is not an orthonormal matrix.

Step4: Rectify A with Gram-Schmidt Orthonormalization method and use U to represent the result.
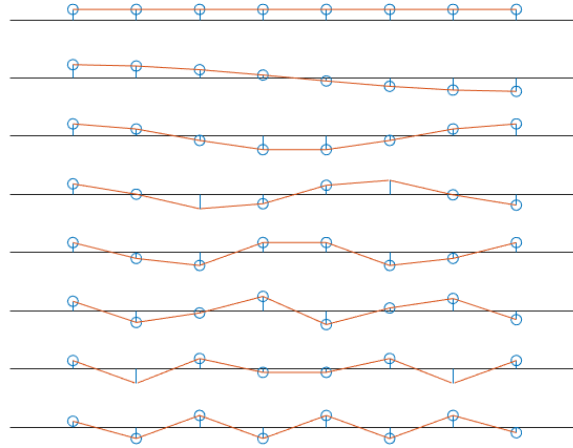
Here is the result of U:

```
U =

    0.3536    0.4714    0.4183    0.3761    0.3416    0.3129    0.2887    0.1961
    0.3536    0.4247    0.2383    0.0108   -0.2182   -0.4070   -0.5202   -0.3922
    0.3536    0.2939   -0.1661   -0.5026   -0.4667   -0.0846    0.3600    0.3922
    0.3536    0.1049   -0.4905   -0.3254    0.3434    0.4788   -0.1285   -0.3922
    0.3536   -0.1049   -0.4905    0.3254    0.3434   -0.4788   -0.1285    0.3922
    0.3536   -0.2939   -0.1661    0.5026   -0.4667    0.0846    0.3600   -0.3922
    0.3536   -0.4247    0.2383   -0.0108   -0.2182    0.4070   -0.5202    0.3922
    0.3536   -0.4714    0.4183   -0.3761    0.3416   -0.3129    0.2887   -0.1961
```

Part e): Draw the 8 basis vectors, which are the column vectors of U.

Here is the figure of column vectors of U:

3.3   Signal Compression

Part f): Compute the expansion coefficients a = [a1, a2, . . . , a8]T

```
Expansion coefficients a:
   -0.0371
    0.8324
    0.7153
    0.3991
    2.0651
   -0.5212
   -1.9098
   -1.4373
```

The result a is the projection of x on the matrix U. x is the linear combination of the basis of U where a is the vector representing each coefficients.

Part g): examine the (relative) size of the coefficients in the vector a and create a new vector a2 of length 8 whose only nonzero entries are the two largest entries (in absolute value) in a.
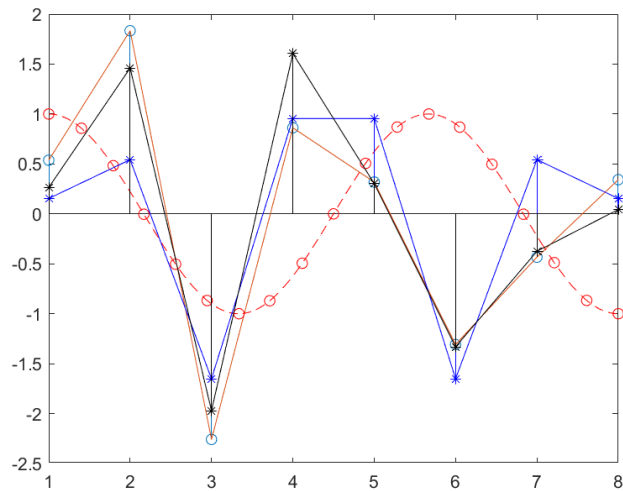
Part h) Construct an approximation x2 of x from a2 and plot x2 over Figure 1.

Part i) Construct an approximation x4 of x from a4 and plot x4 over Figure 1.

part k)compute the relative error of x8 with respect to the original signal x as follows:

```
sqrt(sum((x-x8).^2)/sum(x.^2))
```

Here is the final result of Figure 1:

3.4 Signal Compression

4 Results

Relative error for each compressed signal.

```
Relative error for x8: 0.000000
Relative error for x4: 0.285065
Relative error for x2: 0.564616
```

What numerical results were obtained? What do they represent? Briefly summarize this section.

5 Summary and Conclusion

For this project, I first load the signal as a matrix file; then use matlab to compress the matrix to different extent using largest value method, use x2 to represent the matrix with 6 compressed signal in each column vector and x4 to represent the matrix with 4 compressed signal in each column vector; finally, calculate the relative error relative to the original matrix x.

According to the final result of relative error, the more the signal is compressed, the less like the sound to the original, which is represented using relative error. From the result part, the relative error of x2 is 0.56; that of x4 is 0.29; and that of

x8 is 0. X2 has only 25 percent in each column vector that were not compressed, and x4 has 50 percent in each column vector that were not compressed, while x8 restore the original signal. So no double that relative error of x8 will be nearly 0. Because x4 has twice as much as that of x2 whose signal is not compressed. The relative error is nearly half as that of x2, as my expected.

6   References

What I did for chatgpt: For part g, the whole question, I asked the chatgpt because I don't know what the function to write using matlab.
Here is the list of functions:
1, finding absolute value-abs();
2, finding largest two values-maxk(, 2);
3, assign a to a2-a2(indices) = a(indices);
I also ask about some latex grammar:
Using textt to write the file name
how to insert a equation in the report