Homework 3

Nate Mankovich
Math Modeling

March 20, 2018

Part A: Theory and Foundations

Problem 1/3

Show that the optimization problem

$$\text{maximize } u^T v$$

subject to $u \in \mathcal{R}(X)$, $v \in \mathcal{R}(Y)$ givens the vectors u, v which subtend the smallest angles between the subspaces $\mathcal{R}(X)$ and $\mathcal{R}(Y)$. Subject to the constraints $||u|| = ||v|| = 1$.

This is the same as problem 3: In class we showed that the solution to the angles between subspaces optimization problem led to

$$Q_X^T Q_Y = \Phi \Sigma \Psi^T$$

Show that the elements of Σ are the cosines of the angles between the vector pairs $u^{(i)}$, $v^{(i)}$.
This is the same as showing $Q_X^{-1} Q_Y = U\Sigma$.

We know $\cos(\theta) = \langle u, v\rangle / \sqrt{\langle u, u\rangle \langle v, v\rangle}$ where $u = Xa$ and $v = Yb$.
Since $||u|| = ||v|| = 1$ we have $\langle u, u\rangle = \langle v, v\rangle = 1$ so

$$\cos(\theta_i) = \langle u^{(i)}, v^{(i)}\rangle = {u^{(i)}}^T v^{(i)}$$

$$\begin{aligned}
\cos(\theta_i) &= {u^{(i)}}^T v^{(i)} \\
&= (Xa)^T (Yb) \\
&= a^T X^T Y b \\
&= a^T (Q_x R_x)^T (Q_y R_y) b \\
&= a^T R_x^T Q_x^T Q_y R_y b \\
&= (R_x a)^T Q_x^T Q_y (R_y b) \text{ from problem 2} \qquad = \phi^T Q_x^T Q_y \psi
\end{aligned}$$

If we call Σ the matrix with a diagonal with $\cos(\theta_i)$ we have

$$\Phi \Sigma \Psi^T = Q_x^T Q_y \text{ because } \Phi \Phi^T = \Psi^T \Psi = I$$

Problem 2.

Using the notation from lecture, show that

$$a = R_X^\dagger \phi,\ b = R_Y^\dagger \psi$$

as solutions for the transformation vectors a, b in CCA.

From problem 1/3 we know the following optimization problems are equivalent

$$\max_{||a||=||b||=1} a^T X^T Y b \text{ and } \max_{||\phi||=||\psi||=1} \psi^T Q_x^T Q_y \psi$$

because $a^T X^T Y b = \phi^T Q_x^T Q_y \psi$.
Therefore $R_x a = \phi$ and $R_y b = \psi$. Now we can take the apply the pseudo-inverse of R_x and pseudo-inverse of R_y to equation respectively which gives us:

$$a = R_X^\dagger \phi,\ b = R_Y^\dagger \psi$$

Problem 4.

The GSVD equation can be written

$$\alpha_j X^T X \psi^{(j)} = \beta_j Q Q^T \psi^{(j)}$$

Show that the solutions $\psi^{(j)}$ to the generalized singular value problem are orthogonal in either of the following senses:

$$(\phi^{(i)})^T X^T X \phi^{(j)} = \lambda_i \delta_{ij}$$

and

$$(\phi^{(i)})^T Q^T Q \phi^{(j)} = \lambda_i \delta_{ij}$$

What can you conclude about the maximum noise fraction basis from this?

We know the GSVD on X gives us the following:

$$X = UCM^T \text{ and } Q = VSM^T$$

with the following properties:

- C, S are diagonal
- $C^2 = S^2 = I$
- U, V are unitary and $U^T U = V^T V = I$
- M is invertible so M^T is also invertible
- From class we know $\Phi = (M^T)^{-1}$ so $\phi^{(i)}$ is the ith column of $(M^T)^{-1}$

Using the invertibility of M^T and the GSVD we can see $X(M^T)^{-1} = UC$. Then we can look at the columns of UC.

$$(UC)^{(i)} = (X(M^T)^{-1})^{(i)} = (X\Phi)^{(i)} = X\phi^{(i)}$$

So we have

$$\left(X\phi^{(i)}\right)^T X\phi^{(j)} = (\phi^{(i)})^T X^T X \phi^{(j)} = ((UC)^{(i)})^T ((UC)^{(j)}) \text{ which is a scalar.}$$

Now we will show $((UC)^{(i)})^T((UC)^{(j)}) = \lambda_i \delta_{ij}$. We know $((UC)^{(i)})^T((UC)^{(j)}) = ((UC)^T(UC))_{ij}$. Now calculate

$$\begin{aligned} (UC)^T(UC) &= C^T U^T U C \\ &= C^2 \text{ because } C \text{ is diagonal and } U^T U = I. \end{aligned}$$

C^2 is also diagonal because C is diagonal. Therefore

$$(\phi^{(i)})^T X^T X \phi^{(j)} = ((UC)^T(UC))_{ij} = (C^2)_{ij} = \lambda_i \delta_{ij}$$

where λ_i is the squared generalized singular value of X associated with the ith column of U.

We can make a similar argument for $Q^T Q$.
Using the invertibility of M^T and the GSVD we can see $Q(M^T)^{-1} = VS$. Then we can look at the columns of VS. We find the ith column of VS is $Q\phi_i$. So we have

$$\left(Q\phi^{(i)}\right)^T Q\phi^{(j)} = (\phi^{(i)})^T Q^T Q \phi^{(j)} = ((VS)^{(i)})^T((VS)^{(j)}) = ((VS)^T(VS))_{ij} \text{ which is a scalar.}$$

$$\begin{aligned}(VS)^T(VS) &= S^TV^TVS \\ &= S^2 \text{ because } S \text{ is diagonal and } V^TV = I.\end{aligned}$$

S^2 is also diagonal because S is diagonal. Therefore

$$(\phi^{(i)})^T Q^T Q \phi^{(j)} = ((VS)^T(VS))_{ij} = (S^2)_{ij} = \lambda_i \delta_{ij}$$

where λ_i is the squared generalized singular value of Q associated with the ith column of V.

Problem 5.

Show that we can approximate the covariance matrix of the noise by

$$N^T N = \frac{1}{2} dX^T dX$$

Keep careful track of your assumptions.

We need to notice a few facts before beginning this proof.

1. $S_s \approx S$ because signal is smooth.
2. $S_s^T S \approx S^T S_s \approx S^T S$ by fact 1.
3. N_s is still a noise matrix, but it is different from N_s.
4. $N_s^T N \approx N^T N_s \approx 0$ because fact 3 and the columns of two different noise matrices will most likely be orthogonal.
5. $A_s B_s = AB$ follows from the definition of matrix multiplication.
6. $N^T S = S^T N = N_s^T S_s = S_s^T N_s = 0$ because noise and signal are orthogonal and fact 5.
7. $S_s^T S_s = S^T S$ and $N_s^T N_s = N^T N$ by fact 5.
8. $N_s^T S \approx N^T S_s \approx S_s^T N \approx S^T N_s \approx N^T S \approx N^T S_s \approx S_s^T N_s \approx S^T N_s \approx 0$ because of fact 3.
9. $(N + S)_s = N_s + S_s$ follows by the definition of matrix addition.

Now we can proceed by calculating $\frac{1}{2} dX^T dX$:

$$\begin{aligned}
\frac{1}{2} dX^T dX &= \frac{1}{2}(X - X_s)^T (X - X_s) \\
&= \frac{1}{2}(X^T - X_s^T)(X - X_s) \\
&= \frac{1}{2}(X^T X - X_s^T X + X_s^T X - X_s^T X_s) \\
&= \frac{1}{2}[(N^T N + N^T S + S^T N + S^T S) - (N_s^T N + N_s^T S + S_s^T N + S_s^T S) \\
&- (N^T N_s + N^T S_s + S^T N_s + S^T S_s) + (N_s^T N_s + N_s^T S_s + S_s^T N_s + S_s^T S_s)] \text{ by fact 9.} \\
&= \frac{1}{2}[(N^T N + 0 + S^T S) - (0 + 0 + S^T S) \\
&- (0 + 0 + S^T S) + (N^T N + 0 + S^T S)] \text{ by facts 2, 4, 6, 7, and 8.} \\
&= \frac{1}{2}[N^T N + N^T N] \\
&= N^T N
\end{aligned}$$

Part B: Computing

Download the open book/closed book data set from canvas. Take X to be the $P \times 2$ matrix of closed book exam results and Y to be the $P \times 3$ matrix of open book results.

Problem 1.

a) Compute the canonical vectors $\{a_1, a_2\}$ and $\{b_1, b_2\}$ for X and Y respectively.

$$\{a_1, a_2\} = \left\{ \begin{bmatrix} 0.0028, \\ 0.0055 \end{bmatrix}, \begin{bmatrix} 0.0068 \\ -0.0081 \end{bmatrix} \right\} \text{ and } \{b_1, b_2\} = \left\{ \begin{bmatrix} 0.0088 \\ 0.0009 \\ 0.0004 \end{bmatrix}, \begin{bmatrix} 0.0097 \\ -0.0105 \\ 0.0015 \end{bmatrix} \right\}$$

The code for this is:

```
load('MardiaExamData.mat')

X1= EXAMS(:,1:2);
Y1 = EXAMS(:,3:5);

%mean subtract X and Y
mX = mean(X1);
mY = mean(Y1);
X(:,1) = X1(:,1) - mX(1,1)*ones(88,1);
X(:,2) = X1(:,2) - mX(1,2)*ones(88,1);
Y(:,1) = Y1(:,1) - mY(1,1)*ones(88,1);
Y(:,2) = Y1(:,2) - mY(1,2)*ones(88,1);
Y(:,3) = Y1(:,3) - mY(1,3)*ones(88,1);

%problem1: (computing canonical vectors, etc)

%QR decomposition
[Qx,Rx] = qr(X,0);
[Qy,Ry] = qr(Y,0);

%svd Q
[U1,S1,V1] = svd((Qx')*(Qy));

%find A and B
A = pinv(Rx)*U1;
B = pinv(Ry)*V1;

%part a
%a1,2 b1,2
a1 = A(:,1);
a2 = A(:,2);

b1 = B(:,1);
b2 = B(:,2);
```

b) Plot $\alpha_i = a_1^T x^{(i)}$ and $\beta_i = b_1^T y^{(i)}$. What conclusions can you draw?

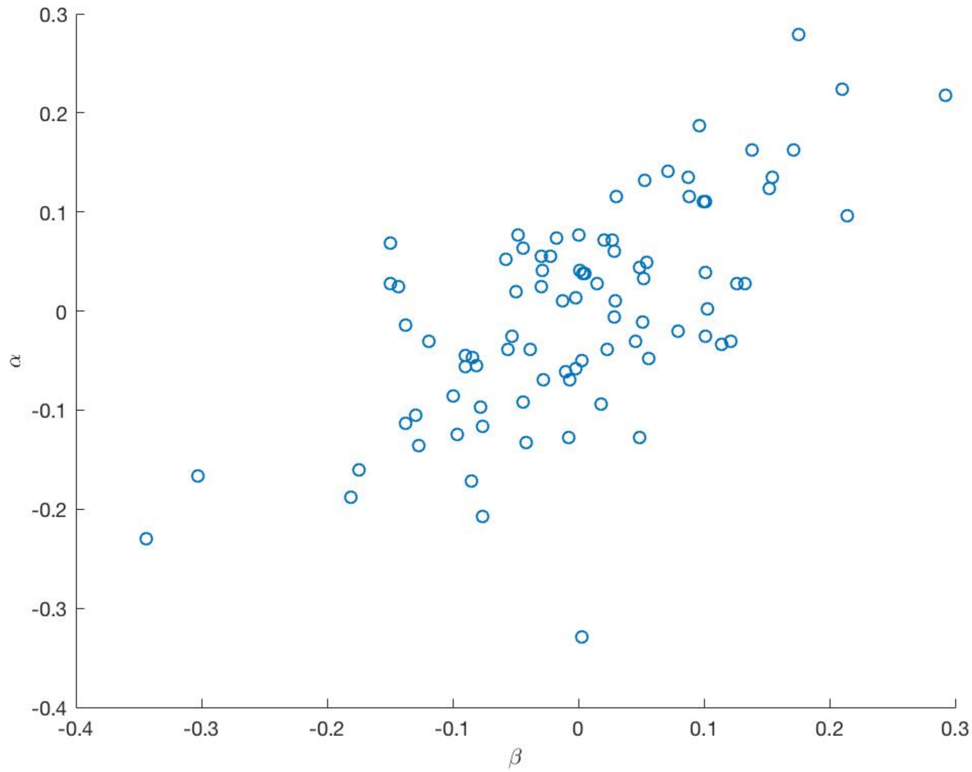

Figure 1: A plot of the β_i on x-axis and α_i y-axis for $i = 1, 2, ..., 88$.

These data seem to be clustering near the line $\alpha = \beta$ which suggests they are correlated. Below is the code for generating the figure.

```
%find alpha and beta
alpha = a1' * X';
beta = b1' * Y';

%scatter alpha and beta
scatter(beta,alpha); xlabel("\beta"); ylabel("\alpha") %beta vs alpha
```

c) Compute the vectors $u^{(1)}, u^{(2)}$ and $v^{(1)}, v^{(2)}$; we showed this explicitly for $u^{(1)}, v^{(1)}$ in class. Argue why the same equations produce $u^{(2)}, v^{(2)}$.

Since these vectors have 88 entries, it does not make sense to write them out in latex. However, here are plots of the coordinates of each vector:

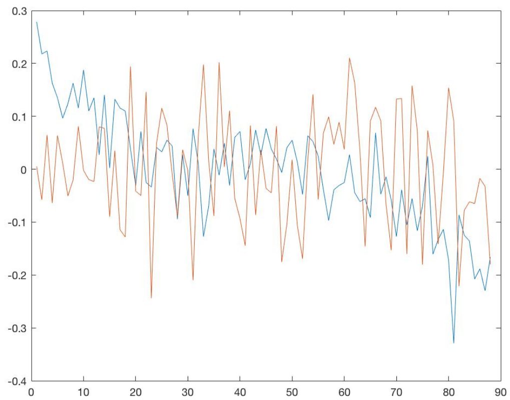

Figure 2: A plot of the coordinates of u_1 (blue) and u_2 (orange).

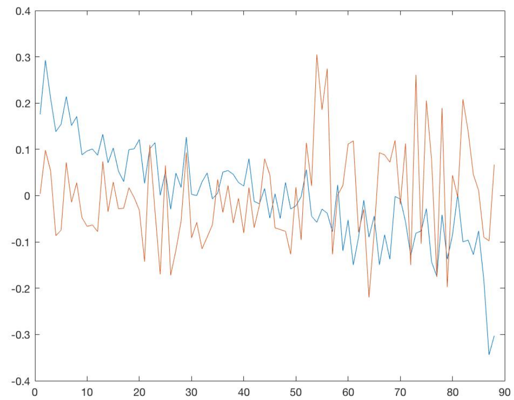

Figure 3: A plot of the coordinates of v_1 (blue) and v_2 (orange).

Notice the columns of Φ are $\phi^{(i)}$, and the columns of Ψ are $\psi^{(i)}$. So we can calculate all v_i and all u_i as the ith column of the respective matrix equations:

$$Q_x\Phi \text{ and } Q_y\Psi.$$

The code to do this is:

```
%find u and v
u = Qx*U1;
v = Qy*V1;

%u1,2 and v1,2
u1 = u(:,1);
u2 = u(:,2);
v1 = v(:,1);
v2 = v(:,2);

%generate plots
plot(u1); hold; plot(u2);
plot(v1); hold; plot(v2);
```

d) Verify that the angle between $u^{(1)}$ and $v^{(1)}$ is $\arccos \sigma_1$.

We find this angle to be 0.8459 radians. The code is below:

```
%verify angle between u1,v1 is arccos(sigma1)
acos(u1'*v1) %angle u1v1
acos(S1(1,1)) %arccos(sigma1)
```

Problem 2.

Download the data set MNFdata.mat from canvas. This is a 629 x 4 data matrix X. This problem concerns the implementation of the maximum noise fraction algorithm and the SVD on this data set.

a) Compute the maximum noise fraction basis ϕ where $\phi = X\psi$ by solving the following problems

   i) the eigenvector problem
   $$(dX^T dX/2)^{-1} X^T X \psi = \lambda\psi.$$

```
%construct Xs
Xs(1,:) = X(629,:);
for i=1:628;
     Xs(i+1,:) = X(i,:);
end

%construct dX
dX = X-Xs;

%construct N^TN
NtN = (1/2)*dX'*dX;

[Psi0,ab0] = eig(pinv(NtN)*X'*X);
%compute basis
Phi0 = X*Psi0;
%Normalize
for i =1:4
    Phi00(:,i) = Phi0(:,i)/(sqrt((Phi0(:,i)')*Phi0(:,i)))
end
```

   ii) the generalized eigenvector problem
   $$C\psi = D\lambda\psi$$
   where $D = dXTdX/2$ and $C = XTX$.

```
%construct Xs
Xs(1,:) = X(629,:);
for i=1:628;
     Xs(i+1,:) = X(i,:);
end

%construct dX
dX = X-Xs;

%construct N^TN
NtN = (1/2)*dX'*dX;

[Psi1,ab1] = eig(NtN,X'*X);
%compute basis
Phi1 = X*Psi1;
%Normalize
for i =1:4
     Phi11(:,i) = Phi1(:,i)/(sqrt((Phi1(:,i)')*Phi1(:,i)))
end
```

iii) the GSVD of $A = dX$ and $B = X$.

```
%construct Xs
Xs(1,:) = X(629,:);
for i=1:628;
     Xs(i+1,:) = X(i,:);
end

%construct dX
dX = X-Xs;

%construct N^TN
NtN = (1/2)*dX'*dX;

[U2,V2,X2,C2,S2] = gsvd(dX/(sqrt(2)),X);
%find Psien
Psi2 = pinv(X2');
%compute basis
Phi2 = X*Psi2;
%Normalize
for i =1:4
     Phi22(:,i) = Phi2(:,i)/(sqrt((Phi2(:,i)')*Phi2(:,i)))
end
```

b) Compare your results in part a).

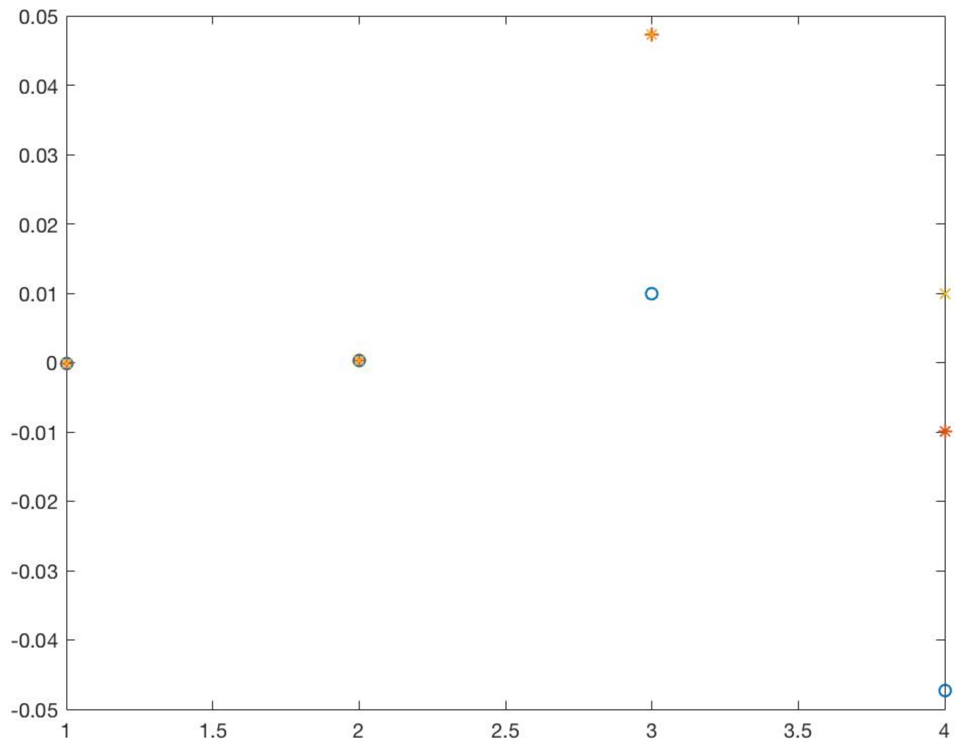

Figure 4: A plot of the normalized first coordinate of each phi vector for each of each of the bases. Each different symbol is the first phi vector for each basis.

Notice all we need for equivalence between these bases is just to switch some of the columns of each Φ and to multiply some columns of Φ by negative one. Notice some first coordinates of the phi vectors agree completely between the four different methods.

```
%setup and plot the first
%phi vector
plot(Phi00(1,:)', 'o');
hold
plot(Phi11(1,:)','*');
plot(Phi22(1,:)','x');
```

c) Compute the SVD of this data matrix and compare the resulting basis vectors with MNF.

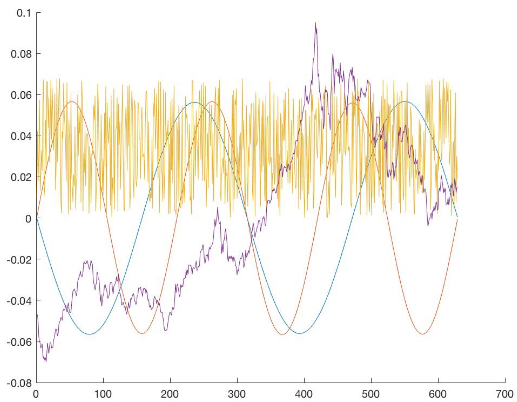

Figure 5: A plot of the four basis vectors from MNF.

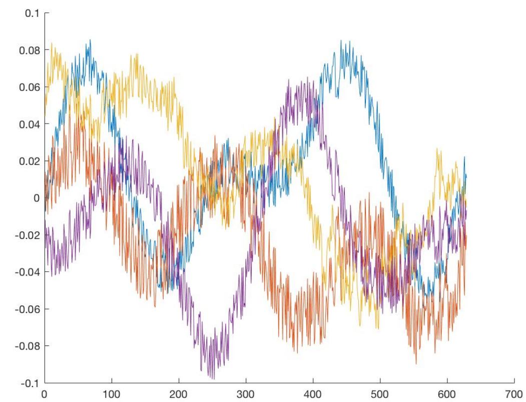

Figure 6: A plot of the four basis vectors from SVD.

These plots help us visualize the fact MNF algorithm separates the noise basis vectors from the signal basis. The noise basis vectors are the jagged lines whereas the signal basis vectors are smooth. Notice all the basis vectors from the SVD are jagged. So the SVD does not separate noise from signal.

```
%mnf plot basis vectors
hold
for i=1:4
    plot(Phi00(:,i))
end

%compute svd
[U,S,V] = svd(X);
%svd plot basis vectors
hold
for i=1:4
    plot(U(:,i))
end
```

d) Project the data (columns of X) onto the first 2 MNF basis vectors and the first 2 SVD left singular vectors and compare.

Below are the plots for the projections of the data onto the first two basis vectors of the MNF basis and the first 2 SVD left singular vectors. Notice that the projection onto the first two vectors of the MNF basis does not appear to include any noise because the first two MNF basis vectors are the signal basis. In contrast, the projection onto the first 2 SVD left singular vectors are quite jagged (contains a bit of noise).
The code for this is:

```
for i=1:4
    SVDpX(:,i) = U(:,1)'*X(:,i)*U(:,1) +  U(:,2)'*X(:,i)*U(:,2)
end

%project X onto the first two basis vectors
for i=1:4
    MNFpX(:,i) = Phi00(:,1)'*X(:,i)*Phi00(:,1) +  Phi00(:,2)'*X(:,i)*Phi00(:,2);
end

%compare
plot(SVDpX(:,1)); hold; plot(MNFpX(:,1)); %SVD in blue, do for 1,2
```

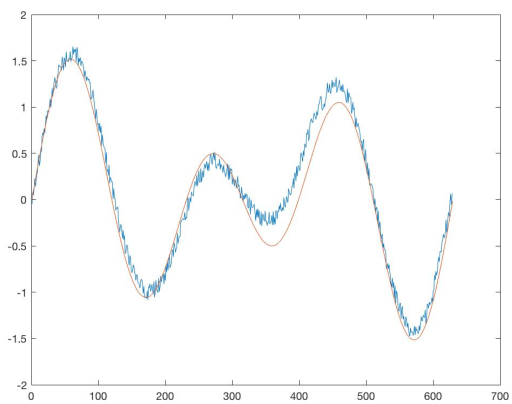

Figure 7: The projection of the first data vector. The projection onto the left singular vectors of the SVD is blue. The projection onto the basis of the MNF is red.

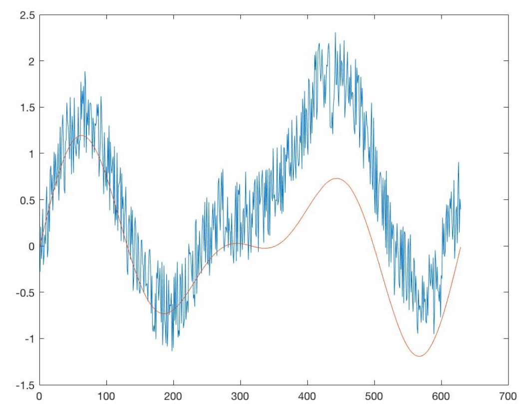

Figure 8: The projection of the second data vector. The projection onto the left singular vectors of the SVD is blue. The projection onto the basis of the MNF is red.

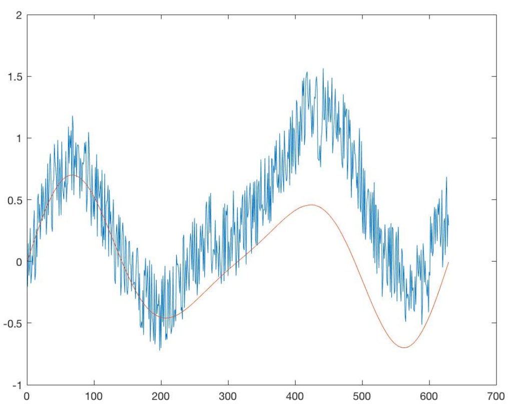

Figure 9: The projection of the third data vector. The projection onto the left singular vectors of the SVD is blue. The projection onto the basis of the MNF is red.

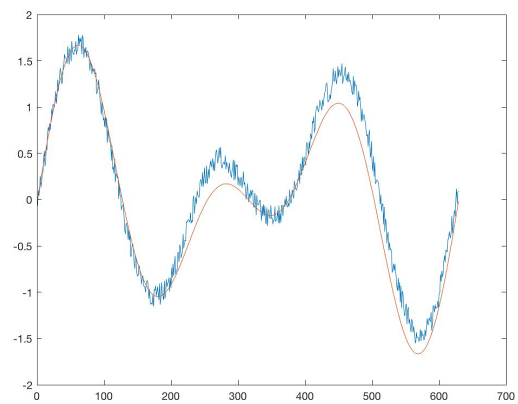

Figure 10: The projection of the fourth data vector. The projection onto the left singular vectors of the SVD is blue. The projection onto the basis of the MNF is red.