MATH444 HW5

Jiasen Zhang

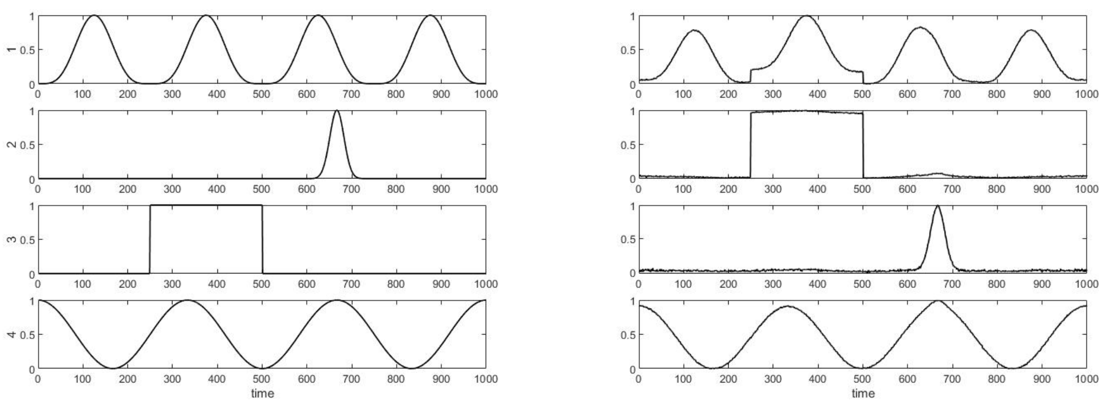
1 Problem 1

1.1 (b)

In this problem m represents the rank of W and H where $X = WH$. In the given signals F there are four signals. So first I set the rank of H as $m = 4$ and run NMF.

The true signals F and signals H seperated by NMF are shown in Figure 1. The 1st, 2nd, 3rd and 4th signals in H correspond to the 1st, 3rd, 2nd, 4th signals in F.

Obviously, the 1st signal in H is affected by the 2nd signal in H. But we can easily identify the types of the signals and correspond them with those in F.



A : True signals F

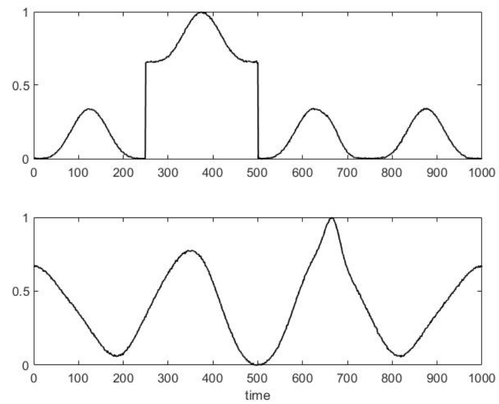
B : Computed sigmals H

Figure 1: m=4

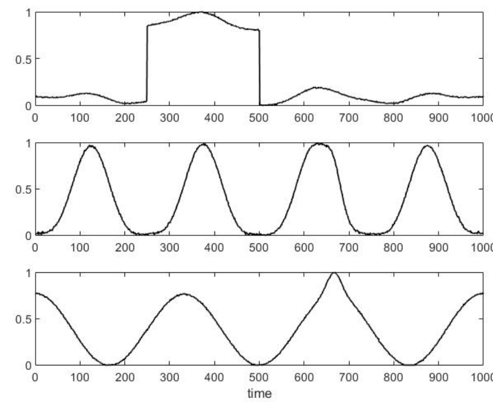
1.2 (c)

First I try $m = 2$ and 3, shown in Figure 2. When m is smaller, the reuslts are less random. When $m = 3$, the signals correspond to 3rd, 1st and 4th signals in F. When $m = 2$, the 1st signal looks like a combination of the 1st and 3rd signals in F. And the 2nd signal looks like a combination of the 2nd and 4th signals in F.

Another phenomenon is that when m is smaller than 4, the quality value $||X - WH||_F$ is larger, which means the NMF result is worse. The quality results are shown in Table 1 in the end.



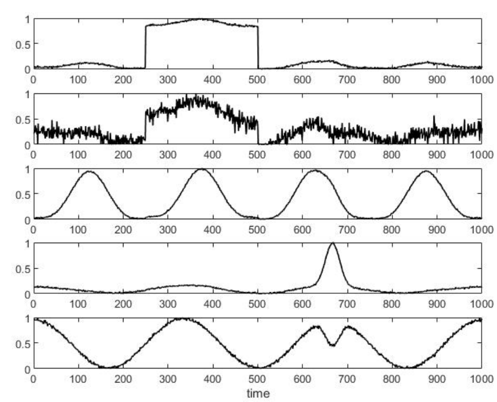
A : $m = 2$



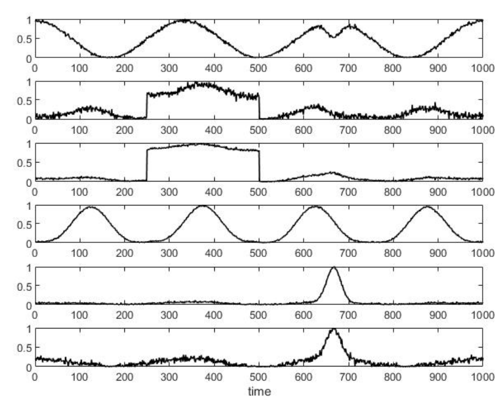
B : $m = 3$

Figure 2

Then I try $m = 5$ and 6, shown in Figure 3. The signals still correspond to the true signals in F, but some signals look noisy because they are seperated from the true signals randomly. For example, in Figure 3 A the first two signals are similar but the second one has noise. So both of them belong to the 3rd signal in F.



A : $m = 5$



B : $m = 6$

Figure 3

To show how to find the optimal number of sources, I list a table with $m = 1$ to 7. In the second column, I try each value of m for ten times and compute their average qualities. The third column shows whether there are noise(yes or no). We can see when $m \leq 4$, the quality increase quickly with m and there is no noise. When $m > 4$, the quality is smaller and almost unchanged, but there is noise.

In conclusion, to find the optimal m, we can make m really small first, and increase m until the quality doesn't decrease quickly and the results look noisy. Or we can start with a big m, then decrease m until there is no noise while the quality is still small.

m	Quality	Noise
1	49.34	N
2	32.03	N
3	8.52	N
4	2.54	N
5	2.49	Y
6	2.38	Y
7	2.29	Y

Table 1

2 Matlab Code

```
clear all;clc;

load SoundSourceData;
[n,p] = size(X);
% n = number of microphones
% p = number of time steps

m = 4; % number of sound sources

[W,H] = NMF(X,m);
quality = norm(X-W*H,'fro')


% plot true signals
figure();
for i=1:4
    subplot(4,1,i);
    plot(F(i,:),'k','LineWidth',1);
    ylabel(num2str(i));
end
xlabel('time');

% plot H
figure();
for i=1:m
    subplot(m,1,i);
    plot(H(i,:),'k','LineWidth',1);
end
xlabel('time');





%%%%%%%%%%%%%%%% NMF function %%%%%%%%%%%%%%%%
function [W, H] = NMF(X, k)
```

```
%Jiasen Zhang: NMF

% input:
% k = rank of W and H
% X = nonnegative matrix

% output: W[n,k], H[k,p]

tau = 1e-6; % tolerance
tmax = 50000;
[n,p]=size(X);

% initialization
W = rand(n,k);
H = rand(k,p);
Hn = H;
Wn = W;
for j=1:k
    %H(j,:)=H(j,:)/norm(H(j,:),'inf');
    H(j,:)=H(j,:)/max(H(j,:));
end

% iteration
for t=1:tmax
    % update H
    Xc = W*H;
    temp1 = W'*X;
    temp2 = W'*Xc;
    for i=1:k
        for j=1:p
            if temp2(i,j)==0
                Hn(i,j)=H(i,j);
            else
                Hn(i,j)=H(i,j)*temp1(i,j)/temp2(i,j);
            end
        end
    end
    %Hn = (W'*X)./(W'*Xc).*H; % new H
    for j=1:k
        Hn(j,:)=Hn(j,:)/norm(Hn(j,:),'inf');
        %Hn(j,:)=Hn(j,:)/max(Hn(j,:));
    end

    Xc = W*Hn; % new Xc

    % update W
    temp1 = X*Hn';
    temp2 = Xc*Hn';
    for i=1:n
        for j=1:k
            if temp2(i,j)==0
                Wn(i,j)=W(i,j);
            else
                Wn(i,j)=W(i,j)*temp1(i,j)/temp2(i,j);
            end
        end
    end
    %Wn = (X*Hn')./(Xc*Hn').*W; % new W


    dWH = norm(Wn-W,'fro')/norm(W,'fro')+norm(Hn-H,'fro')/norm(H,'fro');
    W=Wn;
    H=Hn;
    quality = norm(X-W*H,'fro');

    if mod(t,1000)==0
        fprintf('%d, %e  %e\n',t,dWH,quality);
    end
```

```
    % stopping condition
    if dWH<tau
        break;
    end
end
end
```