Math 444: Homework 5

Iris Zhang

March 22, 2020

Problem 1

The Matlab code for NMF is attached in the appendix I.

Problem 2

For **binaryalphadigs.mat**, I pick the combination of 3 letters and 3 numerical digits: "zjh280". Then we organized them into data matrix X, randomizing the order in which they appear. I include a snippet of my MATLAB code here:

```
% organize the string in data matrix X
isZ = cellfun(@(x)isequal(x,'Z'),classlabels);
index_Z = find(isZ,2);
isJ = cellfun(@(x)isequal(x,'J'),classlabels);
index_J = find(isJ,2);
isH = cellfun(@(x)isequal(x,'H'),classlabels);
index_H = find(isH,2);
is2 = cellfun(@(x)isequal(x,'2'),classlabels);
index_2 = find(is2,2);
is8 = cellfun(@(x)isequal(x,'8'),classlabels);
index_8 = find(is8,2);
is0 = cellfun(@(x)isequal(x,'0'),classlabels);
index_0 = find(is0,2);
string_index = [index_Z index_J index_H index_2 index_8 index_0];
X = dat(string_index,:);
for i = 1:6
    for j = 1:39
        X{i,j} = reshape(X{i,j}',[],1);
    end
end
X = [X{:}];
% randomize the order
cols = size(X,2);
P = randperm(cols);
X = X(:,P);
```

Then, we apply NMF developed in problem 1 and plot the feature vectors for k = 6,9,12, respectively, see Figure 1.

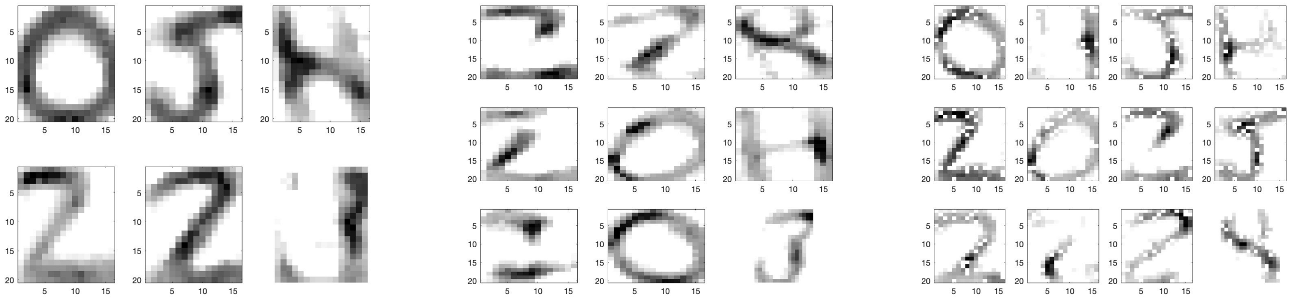

Figure 1: feature vectors in gray scale of X after appkying NMF for k=6(left),9(middle),12(right)

Moreover, we plot the coefficients of H for k =6,9,12 with different colors indicating different characters, see Figure 2.
The color used in those graphs and characters selected are paired in the following:
Red – 'Z'; Blue – 'J'; Yellow – 'H'; Green – '2'; Black – '8'; Coeruleus – '0'.

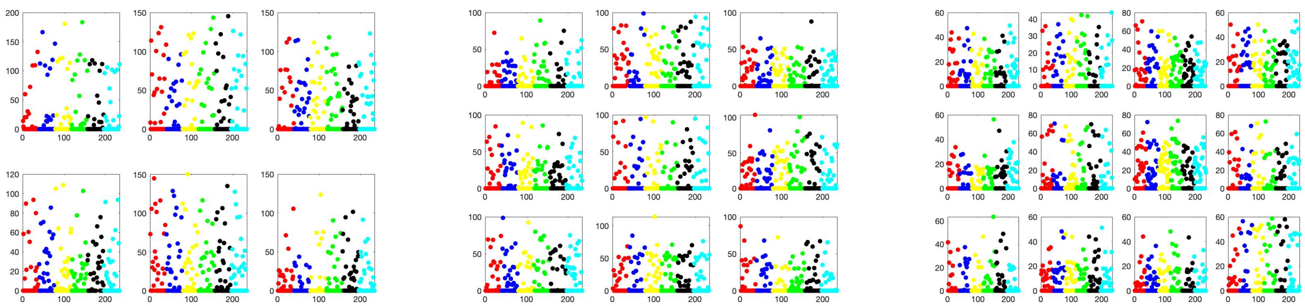

Figure 2: Coefficients of matrix H for k=6(left),9(middle),12(right)

From above graphs of feature vectors, we can see that they can roughly represent different characters. However, since 2 and Z are pretty similar, we cannot tell exactly which one is represented. Besides, 8 is not perfectly presented.
From above graphs of coefficients in H, we can see that the values of entries in H for different characters do not have completely clear separations. Especially red and green, which represents Z and 2, respectively, have basically the same distribution. Besides, the distributions of black and coeruleus, representing 8 and 0, have some in common for some directions of feature vectors. However, we can still see that the Blue and Yellow, representing J and H, have relatively different distributions compared to others.
Moreover, I have tried to normalize matrix H and expect to see more separated dots for different characters, but the separations are still not clear as I expect, see Figure 3.

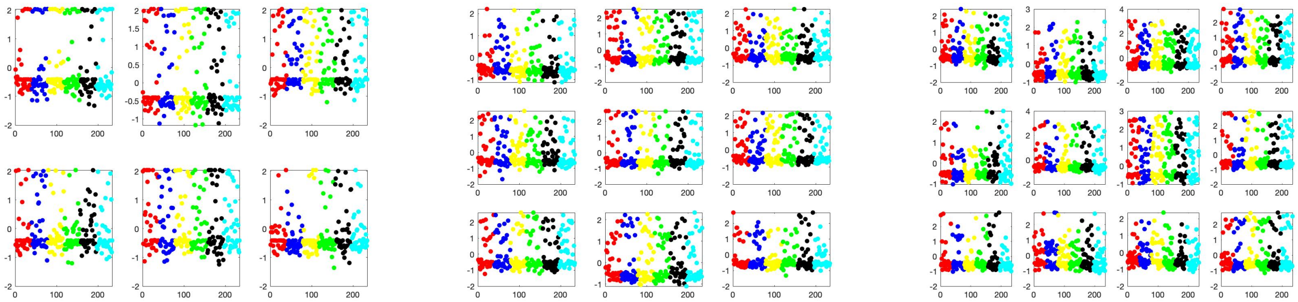

Figure 3: Coefficients of normalized H for k=6(left),9(middle),12(right)

Problem 3

For this problem, we repeat the procedure in problem 2 for two letters and two numbers. Here we choose the new string "WY16". Using only half of the realization to be the training set, after applying NMF, we first plot the feature vectors for k =6,9,12, respectively, see Figure 4.

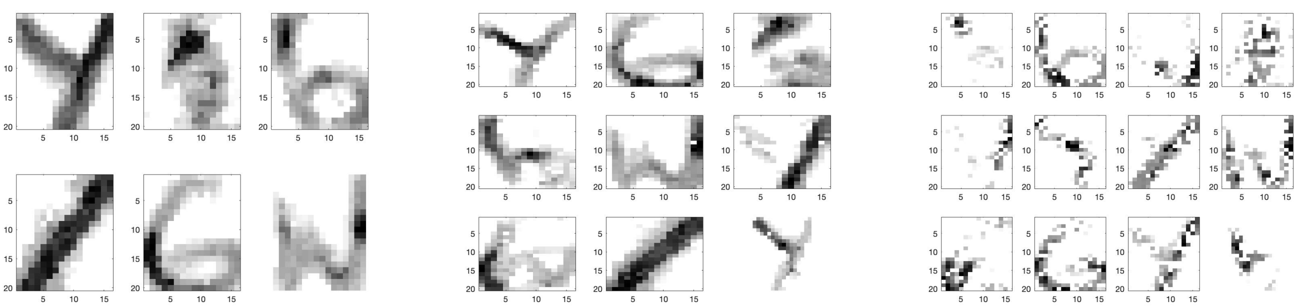

Figure 4: feature vectors of new selected data matrix with training set for k=6(left),9(middle),12(right)

Then, we randomly choose 3 realizations in the remaining realizations (testing set) for each selected characters. In order to see how well they can be represented in terms of the nonnegative feature vectors, we use MATLAB to solve the linear system $W_k * Z = x^{(j)}, j = 1, 2, 3$. Then we plot the coefficients of Z in the same manner as we have done for matrix H in problem 2, see Figure 5.
The color used in those graphs and characters selected are paired in the following:
Red – 'W'; Green – 'Y'; Blue – '1'; Black – '6';

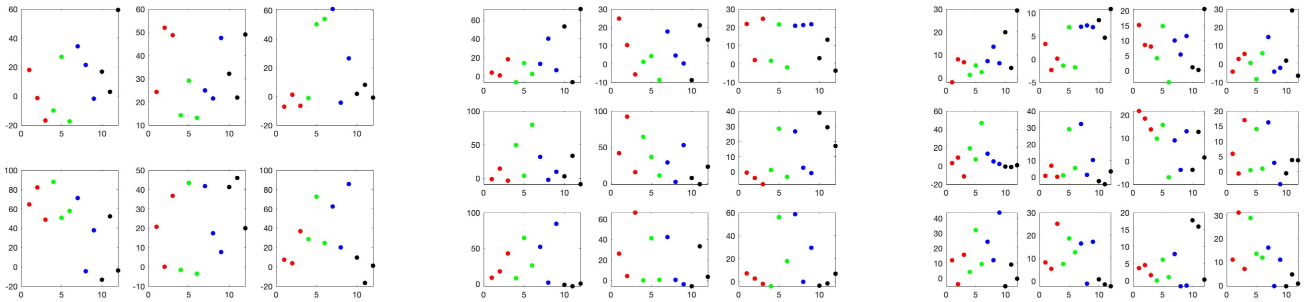

Figure 5: coefficients of Z with testing set for k=6(left),9(middle),12(right)

To be much clearer, we plot the difference $x^{(j)} - W_k * z^{(j)}$ using colormap hot, see Figure 6.

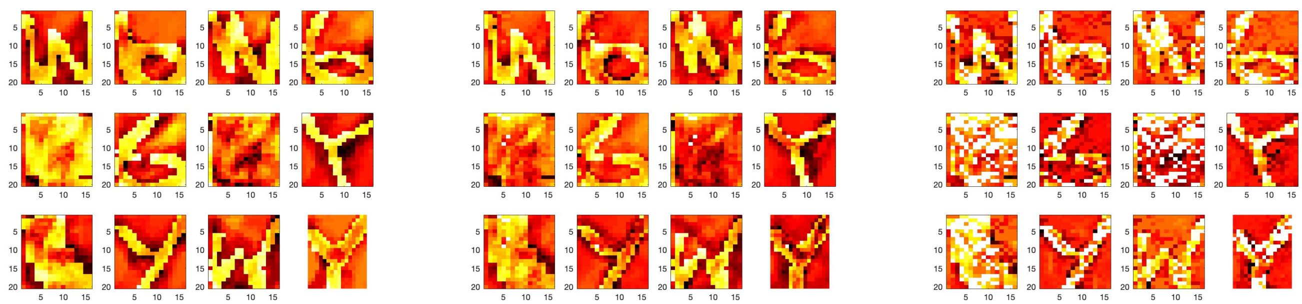

Figure 6: Differences between original value and approximation with testing set for k=6(left),9(middle),12(right)

Now, we randomly take three realizations in the training set for each selected character. Then we plot the coefficients of Z, see Figure 7.

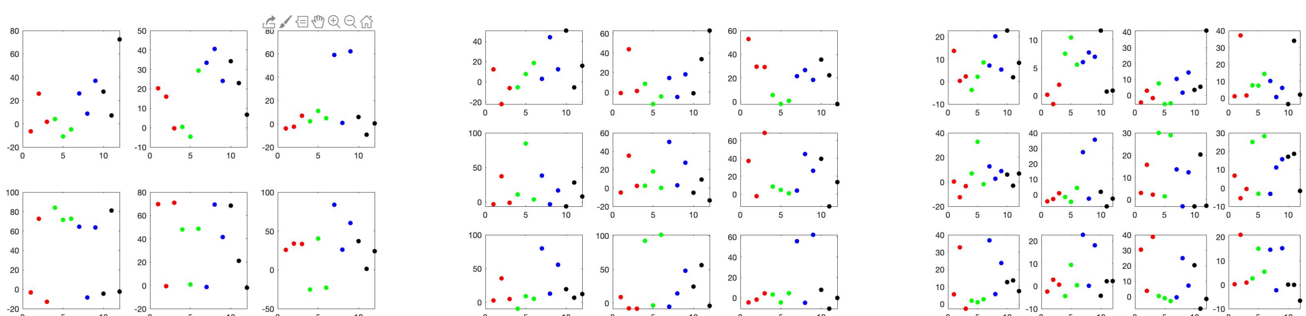

Figure 7: coefficients of Z with training set for k=6(left),9(middle),12(right)

Then, we plot the difference $x^{(j)} - W_k * z^{(j)}$ with the chosen training data using colormap hot, see Figure 8.

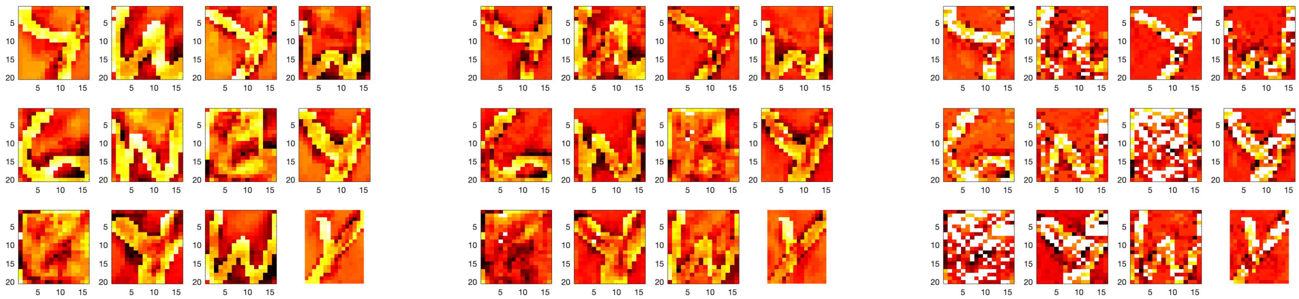

Figure 8: Differences between original value and approximation with training set for k=6(left),9(middle),12(right)

Overall, since this time the characters that we choose do not have much in common, the separation should be much better than the separation of the six chosen characters in problem 2.
For both the testing set and training set, as k changes from 6 to 9 then to 12, the separations among those characters in the coefficients of Z are getting clearer. In the meantime, the plots of difference suggests this change as well.
For the training and testing set, with the same value of k, there is not much difference in terms of the separation, which indicates that our NMF is relatively stable.

Appendix I : Matlab code for NMF

```
function [W,H] = NMF(X,k)
[n,p] = size(X);
eps = 2.22*10^-16;
tau = 0.01;
Tmax = 10^4;
test = 10^5;
% Initialize W and H
Wc = rand(n,k);
W = Wc;
Hc = rand(k,p);
H = Hc;
t = 0;
while t < Tmax && test > tau
% Update W
W_sub = W*H+eps;
W_prestep = zeros(n,p);
for i =1:n
    for j = 1:p
       W_prestep(i,j)= X(i,j)./ W_sub(i,j);
    end
end
% W_prestep;
W_step = W_prestep * H';
for i = 1: n
    for j = 1:k
        W(i,j) = W(i,j).* W_step(i,j);
    end
end
% Normalize the column sums of W
D = sum(W);
D_eps_part = 1./(D+eps);
D_eps = diag(D_eps_part);
W = W * D_eps;

% Update H
H_sub = W*H+eps;
H_prestep = zeros(n,p);
for i =1:n
    for j = 1:p
       H_prestep(i,j)= X(i,j)./ H_sub(i,j);
    end
end
H_step = W'* H_prestep;
for i = 1:k
    for j = 1:p
        H(i,j) = H(i,j).* H_step(i,j);
    end
end
test_1 = norm(W-Wc,'fro')./norm(Wc,'fro');
test_2 = norm(H-Hc,'fro')./norm(Hc,'fro');
test = test_1+test_2;
```

```
Wc = W;
Hc = H;
t = t+1;
end
end
```