MATH444 HW4

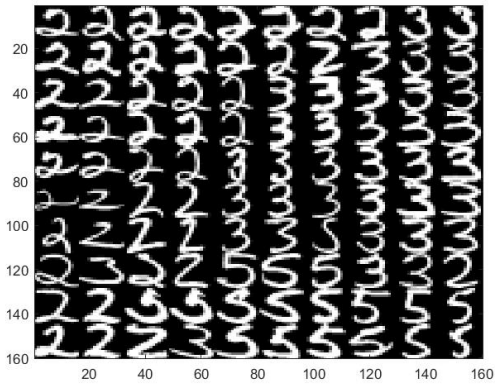
Jiasen Zhang

1 Problem 2

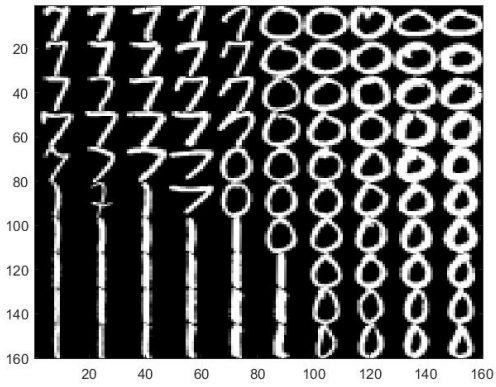
In this problem I choose three digits from the ten handwritten digits. By running SOM algorithm, I can find some prototypes of the three digits, and they have the same topological structure with a 2D lattice Q. If two prototypes are similar to each other, they will be also close in terms of distance in the lattice.

I set $K = 100$, I set $T_{max} = 600 \times K = 60000$, the length of the learning phase $T_0 = 2000$, other parameters are the same as the text. Figure 1 shows two examples. In A I chose 2, 3 and 5 and in B I chose 1, 7 and 0.

In Figure 1, rather than using prototypes, I find the closest data points of each prototype, and put them in the 2D lattice. As we expect, each of the three digits are organized together. And the digits with similar feathers are closer in distance. For example, in the upper right of Figure 1(B), the digit 0 are wider, while in the lower right, the digit 0 are narrower. If the font of digit 0 are narrow, they will be closer to digit 1.



A : 2, 3 and 5



B : 1, 7 and 0

Figure 1: Problem 2: HandwrittenDigits.

2 Problem 3

The breast cancer data contains two types of diagnosis: benign and malignant. Like problem 2, I find some prototypes having the same topological structure with a 2D lattice Q. If two prototypes are similar, they will be closer in the lattice. So it can reflect whether the two diagnosis are seperated by the attributes.

I set $K = 100$, I set $T_{max} = 600 \times K = 60000$, the length of the learning phase $T_0 = 2000$, other parameters are the same as the text.

The result is shown in Figure 2. Red and blue points represent the two different types of diagnosis. The data are divided in 100 small groups according to the closest prototypes. The algorithm performs well and shows that the attributes can seperate the two diagnosis: The two types of data form two groups in the lattice, which means if two data or prototypes are similar, they tend to represent the same diagnosis. However there are some circles(prototypes) containing both diagnosis, which means these data cannot be seperatec well by the attributes.

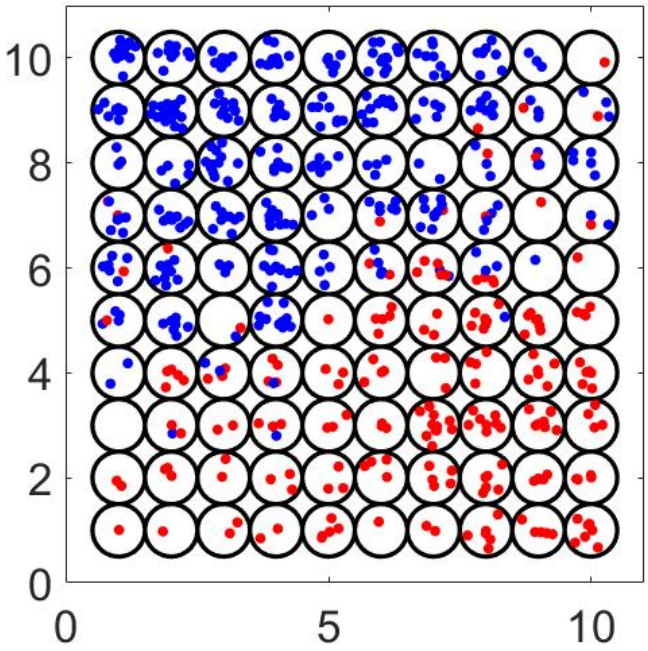


Figure 2: Problem 3: WisconsinBreastCancerData.

3 Matlab Code

3.1 SOM function

```
function M = SOM(K, X, Tmax, T0)
% Jiasen Zhang: SOM with 2D lattice

% input
% K = number of prototype vectors
% X[n,p] = Data
% Tmax = maximum number of iteration
% T0 = length of learning phase

% output
% M[n,K] = updated prototype vectors

[n,p] = size(X);
N=round(sqrt(K));
Q=zeros(2,K); % 2D lattice
for i=1:N
    for j=1:N
        L = N*(i-1)+j;
        Q(:,L)=[i;j];
    end
end

D = zeros(K,K); % distanse matrix
for i=1:K
    for j=1:i
        D(i,j)=norm((Q(:,i)-Q(:,j)),2);
        D(j,i)=D(i,j);
    end
end

% draw initial M (prototype vectors)
Xbar = mean(X,2);
M = zeros(n,K);
for L=1:K
    M(:,L)=Xbar + 2*rand(n,1)-1;
end

% begin iteration
gamma0 = N/3;
gamma1 = 0.5;
alpha0 = 0.9;
alpha1 = 0.01;
for time=1:Tmax
    if mod(time,5000)==0
        fprintf('%d\n',time);
    end
    %for t=1:p
    % draw a data vector of X0
    t = unidrnd(p);
    Xt = X(:,t);
    % find BMU of each Xt and get jt
    jt = 1; % set jt=1 first
    for L=2:K
        if norm((M(:,L)-Xt),2)<norm((M(:,jt)-Xt),2)
            jt = L;
        end
    end
    % get jt

    % update M
    alphat = max(alpha0*(1-time/T0),alpha1);
    gammat = max(gamma0*(1-time/T0),gamma1);
```

```
    Hjt = exp(-(D(jt,:).^2)/(2*gammat*gammat));
    for L=1:K
        M(:,L) = M(:,L)+alphat*Hjt(L)*(Xt-M(:,L));
    end
    %end
end
% get M

end
```

3.2 Problem 2

```
clear all;clc;
tic
load HandwrittenDigits;
% choose 3 handwritten numbers, get X0
i0=1;
for i=1:length(I)
    if I(i)==1 || I(i)==7 || I(i)==0
        I0(i0)=I(i);
        X0(:,i0)=X(:,i);
        i0 = i0+1;
    end
end

% SOM
K=100;
Tmax = 600*K;
T0 = 2000;
M = SOM(K, X0, Tmax, T0);

% visualize the images
N = round(sqrt(K));
V = zeros(16*N,16*N);
for i=1:N
    for j=1:N
        L = N*(i-1)+j;
        % find data closest to M(:,L)
        numForL = 1;
        for tt=1:length(I0)
            if norm((M(:,L)-X0(:,tt)),2)<norm((M(:,L)-X0(:,numForL)),2)
                numForL = tt;
            end
        end

        % store X(:,numForL) in V
        V((i-1)*16+1:i*16,(j-1)*16+1:j*16)=reshape(X0(:,numForL),16,16)';
        % we can also store M in V and plot
        %V((i-1)*16+1:i*16,(j-1)*16+1:j*16)=reshape(M(:,L),16,16)';
    end
end
imagesc(V);colormap(gray);
toc
```

3.3 Problem 3

```
clear all;clc;
tic
load WisconsinBreastCancerData;
X = Data_WCD_Matrix;
I = I_Label;
```

```
6
7  K=100;
8  N = round(sqrt(K));
9  Tmax=600*K;
10 T0 = 2000;
11 M = SOM(K,X,Tmax, T0);
12
13 %-------------------------------------------------------------------------
14 % Defining the map and corresponding prototype vectors. The map Q consists
15 % of 10x10 points in a rectangular grid in two-dimensional space.
16 %-------------------------------------------------------------------------
17 % Map
18 q1 = [1:N]'*ones(1,N);
19 q2 = ones(N,1)*[1:N];
20 Q = [q1(:) q2(:)];
21 % Define the distance squared matrix
22 D2 = zeros(K,K);
23 for j = 1:K
24     for k = 1:j
25         D2(j,k) = norm(Q(j,:) - Q(k,:))^2;
26         D2(k,j) = D2(j,k);
27     end
28 end
29 % Plotting the buttons on the map
30 thplot = linspace(0,2*pi,K);
31 cc = cos(thplot);
32 ss = sin(thplot);
33 for j = 1:K
34     plot(Q(j,1)*ones(1,K) + 0.5*cc,Q(j,2)*ones(1,K)+0.5*ss,'k-','LineWidth',2);
35     hold on;
36 end
37 axis('square');
38 axis([0,N+1,0,N+1]);
39 set(gca,'FontSize',20);
40
41 % for each label
42 for k=1:length(I)
43     % for X(k), find the closest prototype and its index i
44     i=1;
45     for ni=2:K
46         if norm((M(:,ni)-X(:,k)),2)<norm((M(:,i)-X(:,k)),2)
47             i=ni;
48         end
49     end
50     r = 0.4*rand;
51     th = 2*pi*rand;
52     if I(k)==1
53         plot((Q(i,1)+r*cos(th)),(Q(i,2)+r*sin(th)),'r.','MarkerSize',16);hold on;
54     else
55         plot((Q(i,1)+r*cos(th)),(Q(i,2)+r*sin(th)),'b.','MarkerSize',16);hold on;
56     end
57
58 end
59 toc
```