

# Math 444: Homework 4

Iris Zhang

March 6, 2020

## Problem 1

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

## Problem 2

For **WisconsinBreastCancerData**, we apply SOM developed in problem 1. After we get 100 prototypes of the data, we construct a set of 10 by 10 circles to see whether there is a separation in the two different types of diagnosis, see Figure 1.

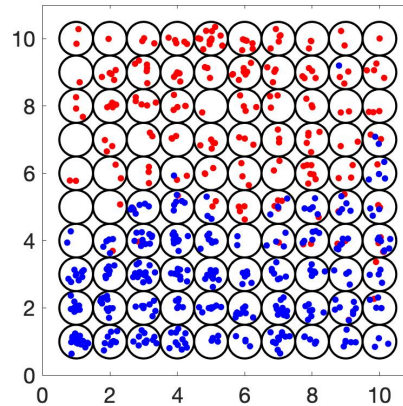


Figure 1: separation of SOM for Wisconsin Breast Cancer Data

From the above graph, we can see that there is clear separation between two different types of diagnosis after we do the SOM.

### Problem 3

For **ProstateCancerData**, we apply SOM developed in problem 1. First, we choose to take 100 prototypes of the data, we construct a set of 10 by 10 circles to see whether there is a separation in the two different types of diagnosis, see Figure 2.

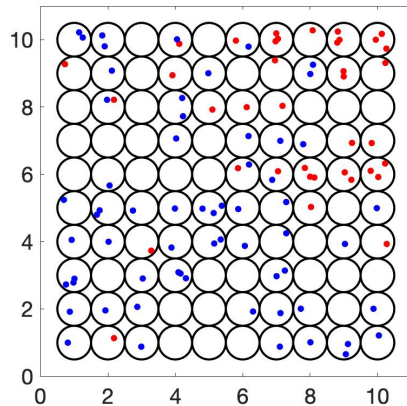


Figure 2: separation of SOM for Prostate Cancer Data with 100 prototypes

Since we only get 100 data points in the original data set, we try to do the SOM of 16 prototypes and plot the set of 4 by 4 circles, see Figure 3.

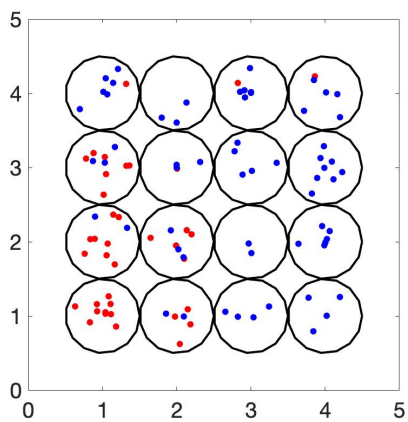


Figure 3: separation of SOM for Prostate Cancer Data with 16 prototypes

From above graph, especially in the second graph, we can see that the data is roughly separated, although there are some mixed overlaps. And, when we reduce the number of prototypes due to the number of data points we have, the clusters are much clearer.

## Problem 4

For **alcohol**, we apply SOM developed in problem 1. Since we only get 77 data points, we first choose to take 64 prototypes of the data, we construct a set of 8 by 8 circles to see whether there is a separation in the three different types of alcohol, see Figure 4.

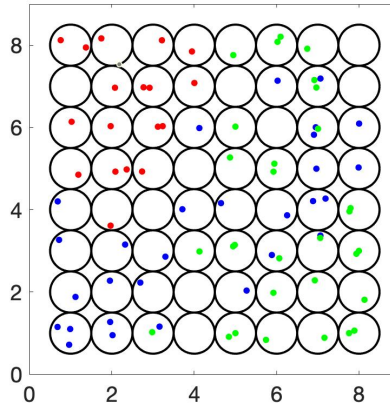


Figure 4: separation of SOM for alcohol with 64 prototypes

Then, we try to reduce the number of prototypes since based on problem 3, we expect the separation will be more clear.

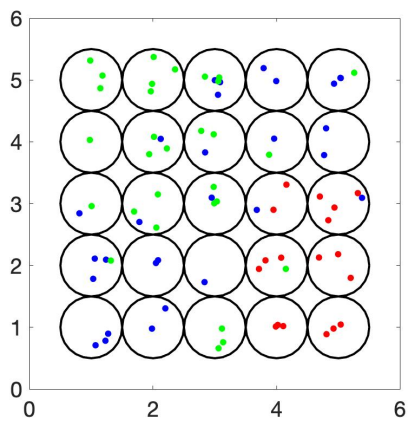


Figure 5: separation of SOM for alcohol with 25 prototypes

From the above graphs, we can see that the separation between red and the other two is clear but the green and blue are mixed.

## Appendix I : Matlab code for SOM

```
function [Prototype,best_match_index] = SOM(X,k)
[n,p]=size(X);
% Define a 2-d lattice Q.
lattice_width = sqrt(k);
q1 = [1:lattice_width]'*ones(1,lattice_width);
q2 = ones(lattice_width,1)*[1:lattice_width];
Q = [q1(:) q2(:)];
% Define the distance squared matrix
D2 = zeros(k,k);
for j = 1:k
    for l = 1:k
        D2(j,l) = (norm(Q(j,:) - Q(l,:)))^2;
    end
end
% Set Tmax to be the maximum number of iterations
Tmax = 1000 *k;
% initialization
t = 0;
T_0 = 1000;
alpha_0 = 0.9;
alpha_1 = 0.01;
gamma_0 = lattice_width/3;
gamma_1 = 0.5;
best_match_index = zeros(1,p);
% random prototypes
Prototype = rand(n,k);
% iteration
while t<=Tmax
    % Find the best matching unit among the current prototypes
    i = randi([1 p]);
    x_vector = X(:,i);
    match = zeros(1,k);
    for j = 1:k
        match(1,j)= (norm(Prototype(:,j)-x_vector))^2;
    end
    [~,match_t_index] = min(match);
    best_match_index(1,i) = match_t_index;
    % Compute the current changing parameters
    alpha_t = max(alpha_0.*(1-t/T_0),alpha_1);
    gamma_t = max(gamma_0.*(1-t/T_0),gamma_1);
    % Compute the neighborhood matrix
    H = zeros(k);
    for l = 1:k
        for j = 1:k
            H(l,j)= exp((-1/(2*(gamma_t)^2))* D2(l,j));
        end
    end
    % update all prototype
    for l = 1:k
        for j = 1:k
            Prototype(:,j) = Prototype(:,j)+ alpha_t * H(j,match_t_index)*(x_vector-Prototype(:,j));
        end
    end
end
```

```
        end
    end
    t = t+1;
end
end
```