

The Matlab code for all experiments is in the **Appendix** section.

6. The goal of this problem is to investigate the influences of the ordering of the dataset in sequential clustering algorithm namely *Basic Sequential Algorithmic Scheme (BSAS)* and *Modified Basic Sequential Algorithmic Scheme (MBSAS)*. According to the literature, the main difference between BSAS and MB-SAS is that the latter separates the clustering procedure into two parts. Firstly, the class determination phase is served as cluster discovery stage to find out the possible number of clusters with the constraints  $\Theta$  : the distance threshold to subsume a data point to a cluster and  $q$  : the numbers of maximum allowed clusters. Secondly, the pattern classification stage where each point is being assigned to one of the already created clusters.

Plot of the vectors:

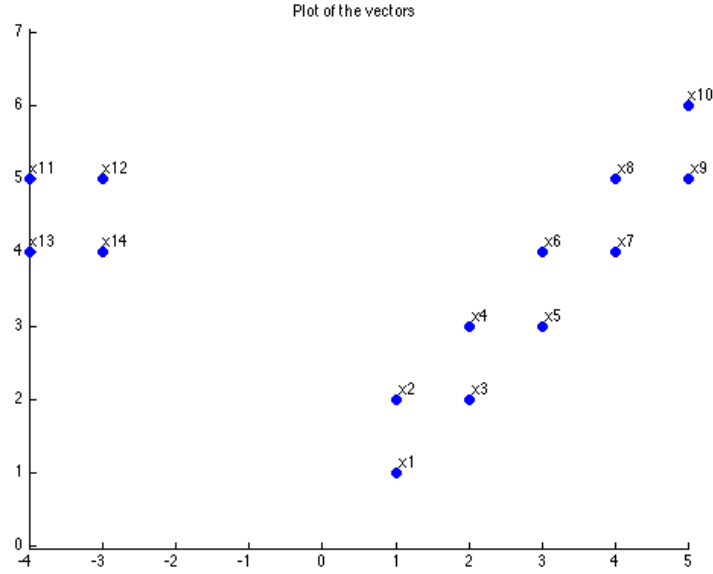


Figure 1: Plot of vectors

For case **a.** where the ordering of the dataset is  $\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_{14}$  we got:

$$BSAS = \{\{\mathbf{x}_1, \mathbf{x}_2, \mathbf{x}_3\}, \{\mathbf{x}_4, \mathbf{x}_5, \mathbf{x}_6\}, \{\mathbf{x}_7, \mathbf{x}_8, \mathbf{x}_9\}, \{\mathbf{x}_{10}\}, \{\mathbf{x}_{11}, \mathbf{x}_{12}, \mathbf{x}_{13}, \mathbf{x}_{14}\}\}$$

$$MBSAS = \{\{\mathbf{x}_1, \mathbf{x}_2\}, \{\mathbf{x}_4, \mathbf{x}_3, \mathbf{x}_5\}, \{\mathbf{x}_7, \mathbf{x}_6, \mathbf{x}_8\}, \{\mathbf{x}_{10}, \mathbf{x}_9\}, \{\mathbf{x}_{11}, \mathbf{x}_{12}, \mathbf{x}_{13}, \mathbf{x}_{14}\}\}$$

The reason of the difference being MBSAS is trying to balance out the cluster assignment decision until all possible clusters are identified. This solves the problem with BSAS which decided a vector  $\mathbf{x}$  to belong to an already created cluster or a new cluster prior to the final cluster formation.

For case **b.** where the ordering of the dataset is  $\mathbf{x}_1, \mathbf{x}_{10}, \mathbf{x}_2, \mathbf{x}_3, \mathbf{x}_4, \mathbf{x}_{11}, \mathbf{x}_{12}, \mathbf{x}_5, \mathbf{x}_6, \mathbf{x}_7, \mathbf{x}_{13}, \mathbf{x}_8, \mathbf{x}_{14}, \mathbf{x}_9$  and the clusters obtained:

$$BSAS = \{\{\mathbf{x}_1, \mathbf{x}_2, \mathbf{x}_3\}, \{\mathbf{x}_{10}, \mathbf{x}_9\}, \{\mathbf{x}_4, \mathbf{x}_5, \mathbf{x}_6\}, \{\mathbf{x}_{11}, \mathbf{x}_{12}, \mathbf{x}_{13}, \mathbf{x}_{14}\}, \{\mathbf{x}_7, \mathbf{x}_8\}\}$$

$$MBSAS = \{\{\mathbf{x}_1, \mathbf{x}_2\}, \{\mathbf{x}_{10}, \mathbf{x}_9\}, \{\mathbf{x}_4, \mathbf{x}_3, \mathbf{x}_5\}, \{\mathbf{x}_{11}, \mathbf{x}_{12}, \mathbf{x}_{13}, \mathbf{x}_{14}\}, \{\mathbf{x}_7, \mathbf{x}_6, \mathbf{x}_8\}\}$$

For case **c.** where the ordering of the dataset is  $\mathbf{x}_1, \mathbf{x}_{10}, \mathbf{x}_5, \mathbf{x}_2, \mathbf{x}_3, \mathbf{x}_{11}, \mathbf{x}_{12}, \mathbf{x}_4, \mathbf{x}_6, \mathbf{x}_7, \mathbf{x}_{13}, \mathbf{x}_{14}, \mathbf{x}_8, \mathbf{x}_9$  and the clusters obtained:

$$BSAS = \{\{\mathbf{x}_1, \mathbf{x}_2, \mathbf{x}_3\}, \{\mathbf{x}_{10}, \mathbf{x}_9\}, \{\mathbf{x}_5, \mathbf{x}_4, \mathbf{x}_6\}, \{\mathbf{x}_{11}, \mathbf{x}_{12}, \mathbf{x}_{13}, \mathbf{x}_{14}\}, \{\mathbf{x}_7, \mathbf{x}_8\}\}$$

$$MBSAS = \{\{\mathbf{x}_1, \mathbf{x}_2, \mathbf{x}_3\}, \{\mathbf{x}_{10}, \mathbf{x}_8, \mathbf{x}_9\}, \{\mathbf{x}_5, \mathbf{x}_4, \mathbf{x}_6, \mathbf{x}_7\}, \{\mathbf{x}_{11}, \mathbf{x}_{12}, \mathbf{x}_{13}, \mathbf{x}_{14}\}\}$$

In the last case MBSAS looked through the data and suggested only four clusters conversely BSAS purely depended on the ordering the data set and formed a total of five clusters.

## Appendix:

assignment\_2.m

```
clc;
clear all;
close all;
```

```
problem_6
problem_7
```

problem\_6.m

```
% load data
X = [[1 1]; [1 2]; [2 2]; [2 3]; [3 3]; [3 4]; [4 4]
     [4 5]; [5 5]; [5 6]; [-4 5]; [-3 5]; [-4 4]; [-3 4]];

% clustering parameters
q = 14;
Theta = sqrt(2);

% run clustering algorithm
% a.
seq = 1:14;
A = X(seq,:);
A_bsas = bsas(A, Theta, q);
A_mbsas = mbsas(A, Theta, q);
display(seq);
print_cluster(A_bsas, seq);
print_cluster(A_mbsas, seq);

% b.
seq = [1 10 2 3 4 11 12 5 6 7 13 8 14 9];
B = X(seq,:);
B_bsas = bsas(B, Theta, q);
B_mbsas = mbsas(B, Theta, q);
display(seq);
print_cluster(B_bsas, seq);
print_cluster(B_mbsas, seq);

% c.
seq = [1 10 5 2 3 11 12 4 6 7 13 14 8 9];
C = X(seq,:);
C_bsas = bsas(C, Theta, q);
C_mbsas = mbsas(C, Theta, q);
display(seq);
print_cluster(C_bsas, seq);
print_cluster(C_mbsas, seq);

% d. Plot the vectors
figure;
scatter(X(:,1), X(:,2), 'filled');
axis equal;
title('Plot of the vectors');

labels = num2str((1:size(X,1))', 'x%d');
text(X(:,1), X(:,2), labels, 'horizontal', 'left', 'vertical', 'bottom');
```

bsas.m

```
function [ C ] = bsas( X, Theta, q )
% bsas - Basic Sequential Clustering Algorithm
%
%      X      : sequential dataset
%      Theta   : clustering threshold
%      q       : maximum number of allowed clusters

[N,~] = size(X);

% numbers of cluster
m = 1;
C = {1};
% clusters representative
```

```

R = X(C{m} ,:);
for i=2:N
    [d, ck] = min_distance(X(i,:), R);
    if d > Theta && m < q
        m = m + 1;
        C{m} = i;
        R(m,:) = X(i,:);
    else
        nc = length(C{ck});
        C{ck} = [C{ck} i];
        % update representative
        % mc.new = (nc*mc+x) / (nc+1)
        R(ck,:) = (nc*R(ck,:) + X(i,:)) / (nc+1);
    end
end
end

```

#### mbsas.m

```

function [ C ] = mbsas( X, Theta, q )
% mbsas - Modified Basic Sequential Clustering Algorithm
%
%      X      : sequential dataset
%      Theta   : clustering threshold
%      q       : maximum number of allowed clusters

[N,~] = size(X);

% numbers of cluster
m = 1;
C = {1};
% clusters representative
R = X(C{m} ,:);

% class determination
for i=2:N
    [d,~] = min_distance(X(i,:), R);
    if d > Theta && m < q
        m = m + 1;
        C{m} = i;
        R(m,:) = X(i,:);
    end
end

% pattern classification
classified = cell2mat(C);
for i=1:N
    if ~nnz(classified == i)
        [~, ck] = min_distance(X(i,:), R);
        nc = length(C{ck});
        C{ck} = [C{ck} i];
        % update representative
        % mc.new = (nc*mc+x) / (nc+1)
        R(ck,:) = (nc*R(ck,:) + X(i,:)) / (nc+1);
    end
end
end

```

#### min\_distance.m

```

function [ d, k ] = min_distance( x, C )
% distance - Compute min distance from x to all clusters = min d(x, C)
% C - clusters representative

[m,~] = size(C);
D = zeros(1, m);
for i=1:m
    D(i) = d2(x, C(i,:));
end
[d, k] = min(D);
end

% Euclidean distance between x and y
function [ d ] = d2(x, y)
    x_tilde = x - y;

```

```

    d = sqrt(x_tilte * x_tilte ');
end

```

print\_cluster.m

```

function print_cluster(C, seq)
% print_cluster - print cluster values
% C - set of clusters (cell array)
% seq - sequence of x

disp([inputname(1) ' '=']);
disp(' ');
m = length(C);
for i=1:m
    fprintf([' %d' sprintf(' %x%d', seq(C{i})) ' ']);
end
disp(' ');
disp(' ');

```