Pattern Recognition Practical 5

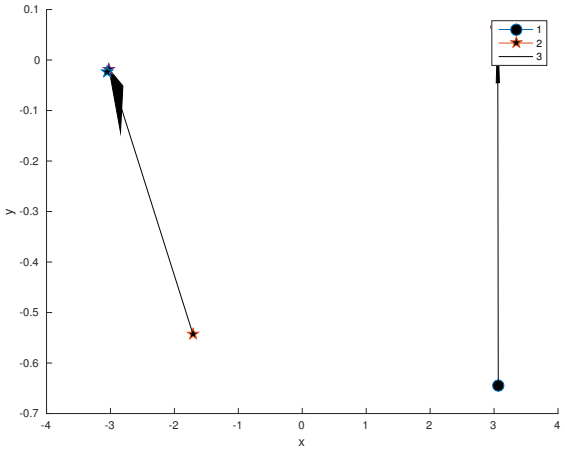
Group 24: Maikel Withagen (s1867733) Steven Bosch (s1861948)

October 15, 2015

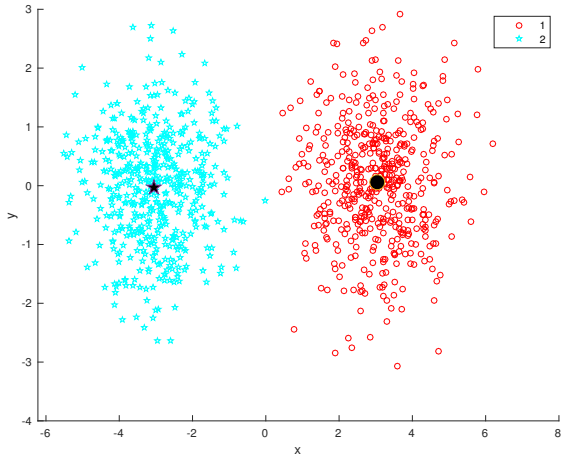
Assignment 1 k-means clustering, quantization error, gap statistic

Using the code given in the Appendix(Assignment 2Assignment 2), we created the plots shown below.

Figure 1: Results for k=2

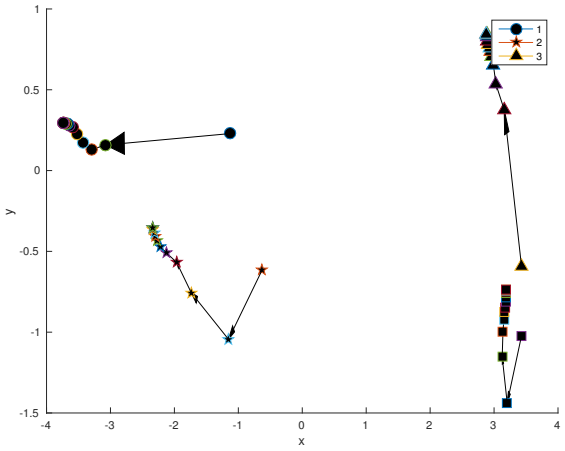


(a) intermediate positions of the cluster means, with their progress indicated by the arrows.

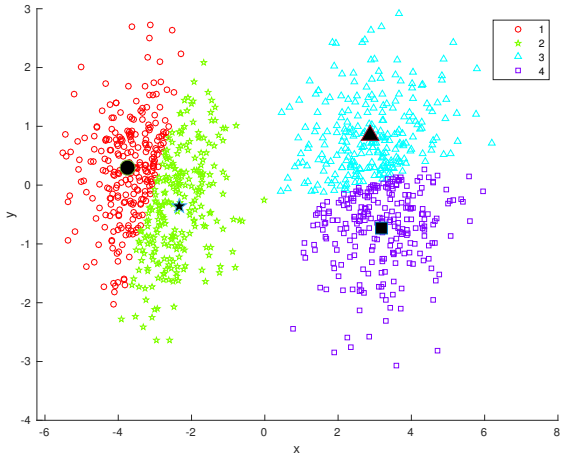


(b) The final cluster means with their assigned data points)

Figure 2: Results for k=4

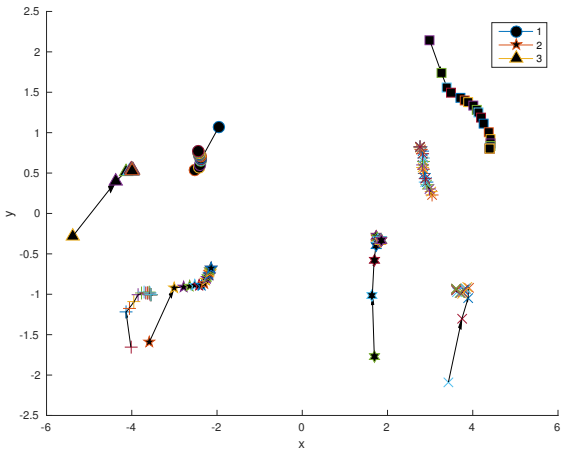


(a) intermediate positions of the cluster means, with their progress indicated by the arrows.

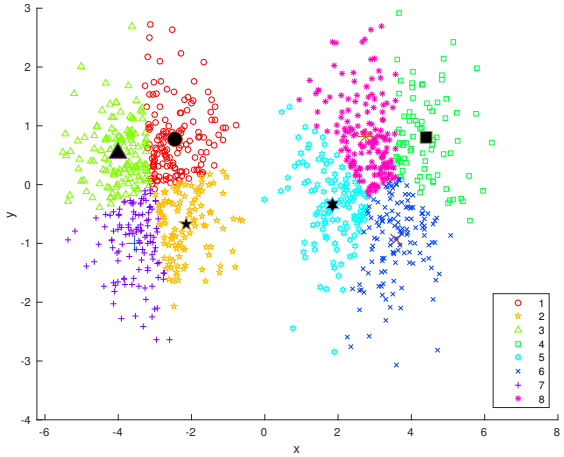


(b) The final cluster means with their assigned data points)

Figure 3: Results for k=8



(a) intermediate positions of the cluster means, with their progress indicated by the arrows.



(b) The final cluster means with their assigned data points)

Assignment 2 Batch Neural gas vs k-means

Appendix

../Code/kmeans.m

```
1 function [qError] = kmeans(dat, k, writeOutput)
2 % K-means clustering algorithm
3 close all;
4 shapes = 'op^shx+*dv<>.';
5
6 % Init the prototypes to a random point
```

```
7  prototypes = zeros(k,ndims(dat));
8  for i = 1:k
9      newPoint = dat(randi(length(dat)),1:2);
10     while (sum(pdist2(prototypes, newPoint) == 0) ~= 0)
11         newPoint = dat(randi(length(dat)),1:2);
12     end
13     prototypes(i,:) = newPoint;
14 end
15
16 % Init the first figure
17 figure(1)
18 hold on;
19 xlabel('x');
20 ylabel('y');
21
22 for i = 1 : size(prototypes, 1)
23     plot(prototypes(i,1),prototypes(i,2),'Marker', shapes(i), 'MarkerSize', 10, '
       MarkerFaceColor', 'black')
24 end
25
26
27 % Perform k-means
28 loop = 1;
29 loopCounter = 0;
30 while(loop == 1)
31     loop = 0;
32     loopCounter = loopCounter + 1
33
34     for point = 1 : length(dat)
35         dat(point,3) = find(pdist2(dat(point,1:2), prototypes) == min(pdist2(dat(point,1:2),
           prototypes)),1);
36     end
37
38     for prototype = 1 : size(prototypes, 1)
39         newMean = mean(dat(dat(:,3) == prototype,1:2));
40         if newMean ~= prototypes(prototype,:)
41             loop = 1;
42         end
43         plot_arrow( prototypes(prototype,1), prototypes(prototype,2), newMean(:,1), newMean
           (:, 2));
44         prototypes(prototype,:) = newMean;
45         plot(newMean(1),newMean(2),'Marker', shapes(prototype), 'MarkerSize', 10, '
           MarkerFaceColor', 'black')
46     end
47
48
49 end
50
51 % Calculate the quantization error
52 qError = 0;
53 for i = 1 : size(prototypes, 1)
54     qError = qError + sum(pdist2(prototypes(i,:), dat(dat(:,3) == i,1:2)));
55 end
56 % More figure stuff
57 legend('1','2','3')
58 if writeOutput == 1
59     print(sprintf('../Report/Fig1_k%d', k), '-depsc');
60 end
61 %sprintf('Fig1_k%d', k)
62 figure(2)
63 hold on;
64 gscatter(dat(:,1),dat(:,2),dat(:,3),[],shapes, 5)
65
66
67 for i = 1 : size(prototypes, 1)
```

```
68      plot(prototypes(i,1),prototypes(i,2),'Marker', shapes(i), 'MarkerSize', 13, '
            MarkerFaceColor', 'black')
69  end
70
71  xlabel('x');
72  ylabel('y');
73  if writeOutput == 1
74      print(sprintf('../Report/Fig2_k%d', k), '-depsc');
75  end
```

../Code/runKMeans.m

```
1  load('kmeans1.mat', 'kmeans1');
2  meanError = zeros(1,10);
3  k = 2;
4  for k = [2 3 4]
5      meanError(k) = kmeans(kmeans1,1,0);
6  end
7  mean(meanError)
```