

MATH444 HW2

Jiasen Zhang

1 Problem 2

The results of k-means and k-medoids algorithms clustering WineData are shown in Figure 1 and 2. Each color represent one of the three clusters. For this set of data the results of the two algorithms are really close. Compared with the given distribution, my algorithms separate the blue cluster really well, while the red and green clusters are more difficult to distinguish. So based on the record attributes, one type of wine is easy to cluster while the other two types are not easy to distinguish.

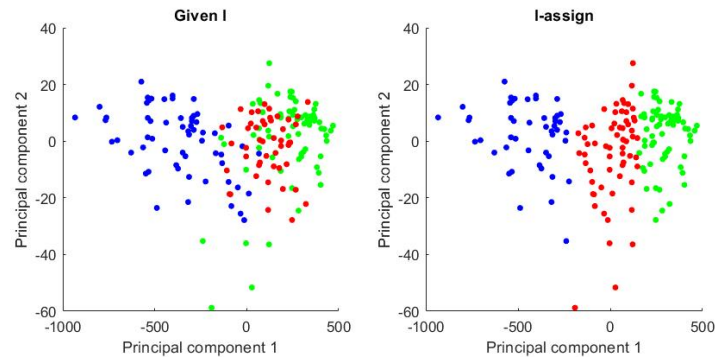


Figure 1: WineData: k-means

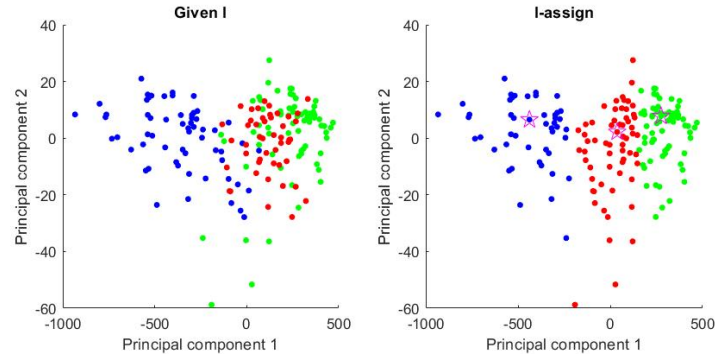


Figure 2: WineData: k-medoids. (Purple stars represent medoids.)

2 Problem 3

The result of k-medoids algorithm clustering CardiacSPECT is shown in Figure 3. The corresponding matrix C is $\begin{bmatrix} 44 & 40 \\ 28 & 75 \end{bmatrix}$, which means:

$$\begin{aligned} c_{11} &= 44 = \text{number of 1's in my cluster 1(A)} \\ c_{12} &= 40 = \text{number of 1's in my cluster 2(B)} \\ c_{21} &= 28 = \text{number of 0's in my cluster 1(A)} \\ c_{22} &= 75 = \text{number of 0's in my cluster 2(B)} \end{aligned}$$

$$\begin{aligned} c_{11} + c_{12} &= 84 = \text{total number of 1's in } I \\ c_{21} + c_{22} &= 103 = \text{total number of 0's in } I \end{aligned}$$

My result shows that $44/84=0.5238$ of 1's in the origin distribution I are distributed into my cluster 1, while $75/103=0.7282$ of 0's in I are distributed into my cluster 2. The larger c_{11} and c_{22} are, the better my result is. So my algorithm can classify the two categories to some extent.

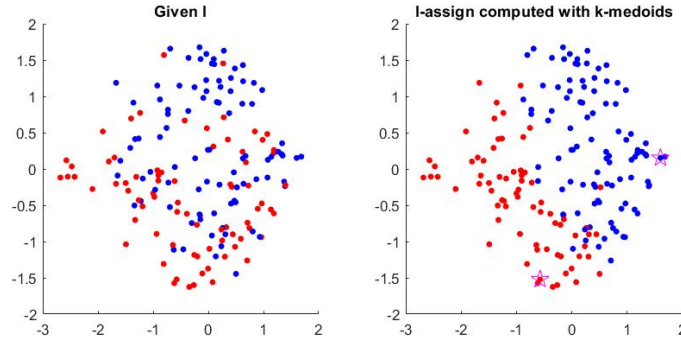


Figure 3: CardiacSPECT: k-medoids. (Purple stars represent medoids.)

For the second definition of matrix C in the problem, we interchange the two columns of C . In this case $C = \begin{bmatrix} 40 & 44 \\ 75 & 28 \end{bmatrix}$. The larger c_{12} and c_{21} are, the better my result is.

3 Problem 4

The result is shown in Figure 4, and the matrix C is: $\begin{bmatrix} 195 & 72 \\ 7 & 160 \end{bmatrix}$. That means $195/267=0.7303$ of 1's in the origin distribution I are distributed into my cluster 1, and $160/167=0.9581$ of 0's in I are distributed into my cluster 2.

From the figure we can see there are two areas on the left and right where the two groups are really dense, while the central area is sparse in which the two groups mix with each other. Although some red points in the central area are classified into blue points, my algorithm deals with the two dense area well, which means it generally corresponds to the party line.

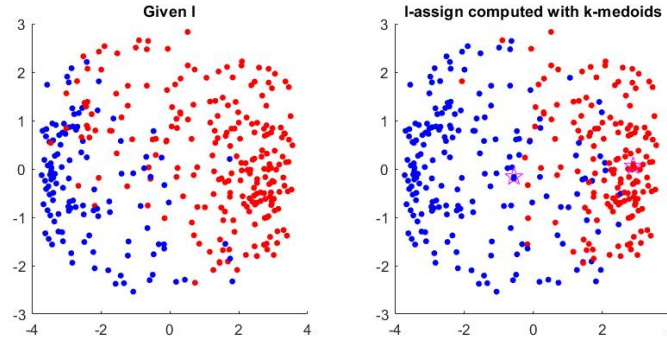


Figure 4: CongressionalVoteData: k-medoids. (Purple stars represent medoids.)

According to my cluster, the two dense areas are far from each other, and most of the votes are located in these two dense area. That means most of the votes are partisan to their parties. The red medoid is just in the red dense area, but the blue medoid is near the center. Although the number of points in the central area is not too large, they are difficult to classify because they are not so partisan.

Their absentee record is good. Most of the voters voted more than 14 issues. And only several pairs of voters have less than 10 simultaneous voting record.

Missing votes can be misleading. For example, two voters may vote the only one issue they disagree, but missing lots of issues they agree. In this case my algorithm will classified them into different parties, while in fact they should be in the same party. My suggestion is that if the simultaneous voting record is less than a number e.g. 5, then we can use a neutral value instead. If a voter votes too few issues, he or she should be discarded.

4 Problem 5

The result of k-means algorithm clustering IrisData is shown in Figure 5. The matrix C is $\begin{bmatrix} 50 & 0 & 0 \\ 0 & 14 & 36 \\ 0 & 47 & 3 \end{bmatrix}$. That means all of 1's in the origin distribution I are distributed into my cluster 1, $36/50=0.72$ of 2's in I are distributed into my cluster 3, and $47/50=0.94$ of 3's in I are distributed into my cluster 2.

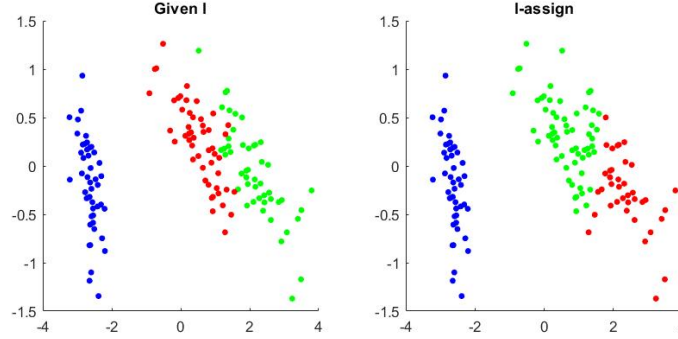


Figure 5: IrisData: k-means

The result of k-medoids algorithm clustering IrisData is shown in Figure 6. The matrix C is $\begin{bmatrix} 50 & 0 & 0 \\ 0 & 14 & 36 \\ 0 & 48 & 2 \end{bmatrix}$. That means all of 1's in the origin distribution I are distributed into my cluster 1, $36/50=0.72$ of 2's in I are distributed into my cluster 3, and $48/50=0.96$ of 3's in I are distributed into my cluster 2.

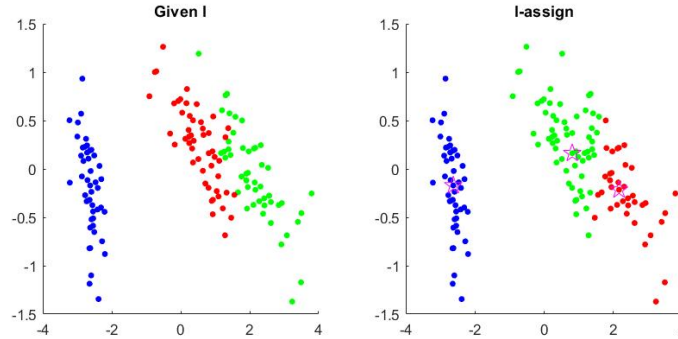


Figure 6: IrisData: k-medoids. (Purple stars represent medoids.)

The second type of flower (2 in I) may be more difficult to classify with the third type (3 in I). Both of the two algorithms perform really good for the first and third types and k-medoids is slightly better.

5 Matlab code

5.1 k-means

```
1 function [I_assign] = my_kmeans(k, X, tau, ninit)
2 % Jiasen Zhang
3
4 % input
5 % k = number of clusters
6 % X = data
7 % tau = tolerance
8 % ninit = times of initialization
9
10 % output
11 % I_assign = assignment vector indicating the cluster of each data vector
12
13 [n,p] = size(X);
14
15 % Initialize
16 Q = 0;
17 c = zeros(n,k);
18 for num=1:ninit
19     partitionc = unidrnd(k,p,1);
20     c_c = zeros(n,k); % current centroids
21     ql = zeros(1,k); % corresponding within-cluster tightness
22     for L=1:k
23         % compute centroids
24         c_c(:,L)=mean(X(:,partitionc==L),2);
25     end
26     % within-cluster tightness
27     for j=1:p
28         cluster = partitionc(j);
29         ql(cluster)=ql(cluster)+norm(X(:,j)-c_c(:,cluster),2);
30     end
31     % compute currently overall tightness
32     % choose smaller Q
33     Q_c = sum(ql);
34     if Q_c<Q || num==1
35         Q=Q_c;
36         c=c_c;
37         partition=partitionc;
38     end
39 end
40
41 % Begin iteration
42 diff = 2*tau; % make sure diff>tau at first
43 while(diff>tau)
44     % find closest cluster for x(j), update partition
45     for j=1:p
46         temp = norm(X(:,j)-c(:,1),2);
47         partition(j)=1; % temporary cluster 1
48         for L=2:k
49             if norm(X(:,j)-c(:,L),2)<temp
50                 temp = norm(X(:,j)-c(:,L),2);
51                 % reassign x(j)
52                 partition(j) = L;
53             end
54         end
55     end
56
57     % update c(L) for each cluster and computed new ql
58     for L=1:k
59         c(:,L)=mean(X(:,partition==L),2);
60         if max(isnan(c(:,L)))≠0
61             c(:,L)=0;
62         end
63     end
64 end
```

```

63     end
64     ql = zeros(k,1);
65     for j=1:p
66         cluster = partition(j);
67         ql(cluster)=ql(cluster)+norm(X(:,j)-c(:,cluster),2);
68     end
69
70     % get new Q
71     newQ = sum(ql);
72     diff = abs(newQ-Q);
73     Q = newQ;
74
75 end
76 % result
77 I_assign = partition;
78 end

```

5.2 k-medoids

```

1 function [I_assign, I_bar] = my_k_medoids(k, D, tau, ninit)
2 % Jiasen Zhang
3
4 % input
5 % k = number of clusters
6 % D = distance matrix
7 % tau = tolerance
8 % ninit = times of initialization
9
10 % output
11 % I_assign = assignment vector indicating the cluster of each data vector
12 % I_bar = medoids
13
14 [n,p]=size(D);
15 % Initialize
16 Q = 0;
17 for num = 1:ninit
18     % choose k data vectors randomly
19     partitionc=ones(p,1);
20     % get initial medoids randomly
21     temp = randperm(p);
22     c=c(temp(1:k));
23     % for each x(j), find the nearest medoid
24     for j=1:p
25         Dm = D(j,c);
26         cluster = find(Dm==min(Dm));
27         partitionc(j) = cluster(1);
28     %     temp = D(j,c(1)); % temporary cluster 1
29     %     for L=2:k
30     %         if D(j,c(L))<temp % choose smaller one
31     %             temp = D(j,c(L));
32     %             partitionc(j) = L; % reassign x(j)
33     %         end
34     %     end
35     end
36     % get initial within-cluster tightness
37     ql = zeros(k,1);
38     for j=1:p
39         cluster = partitionc(j);
40         ql(cluster)=ql(cluster)+D(j,c(cluster));
41     end
42     Qc=sum(ql);
43     if Qc<Q || num==1
44         Q=Qc;
45         c=c;
46         partition=partitionc;

```

```

47     end
48 end
49
50 % Begin iteration
51 diff = 2*tau; % make sure diff>tau at first
52 while(diff>=tau)
53     % for each x(j), find the nearest medoid, update partition
54     for j=1:p
55         Dm = D(j,c);
56         cluster = find(Dm==min(Dm));
57         partition(j) = cluster(1);
58     end
59
60     % for each cluster, select a medoid so that the within-cluster
61     % tightness is the smallest
62     for L=1:k
63         % get local distance matrix of cluster L
64         index = find(partition==L);
65         DL=D(index,index);
66         sumDL=sum(DL,1);
67         c0=index((sumDL==min(sumDL)));
68         c(L)=c0(1);
69     end
70
71     % Get overall tightness newQ
72     ql = zeros(k,1);
73     for j=1:p % search all the points
74         cluster = partition(j);
75         ql(cluster)=ql(cluster)+D(j,c(cluster));
76     end
77     newQ = sum(ql);
78     diff = abs(newQ-Q);
79     Q = newQ;
80 end
81 I_assign = partition;
82 I_bar = c;
83 end

```

5.3 Problem 2

```

1 clear all;clc;
2 % parameters
3 k=3;
4 tau=1e-4; % tolerance
5 ninit = 20; % times of initialization
6
7 load WineData;
8 [n,p]=size(X);
9 % Get distance matrix D
10 D = zeros(p,p);
11 for i=1:p
12     for j=1:p
13         D(i,j)=norm(X(:,i)-X(:,j),2);
14     end
15 end
16
17 % choose one and comment the other one
18 %[I_assign, I_bar] = my_k_medoids(k, D, tau, ninit); % k-medoids
19 I_assign = my_k_means(k, X, tau, ninit); % k-means
20
21 tic
22 % % plot taking several seconds
23 % subplot(1,2,1);plot(I);
24 % subplot(1,2,2);bar(I_assign);
25 figure();

```

```

26 Xbar = mean(X,2);
27 Xc = X - Xbar;
28 [u,d,v] = svd(Xc);
29 z = u(:,1:2)'*Xc;
30 % plot the 3 groups
31 subplot(1,2,1);
32 for j=1:length(I)
33     if I(j)==1
34         scatter(z(1,j),z(2,j),'b','k.','SizeData',200);hold on;
35     elseif I(j)==2
36         scatter(z(1,j),z(2,j),'g','k.','SizeData',200);hold on;
37     else
38         scatter(z(1,j),z(2,j),'r','k.','SizeData',200);hold on;
39     end
40 end
41 title('Given I')
42 subplot(1,2,2);
43 for j=1:length(I_assign)
44     if I_assign(j)==1
45         scatter(z(1,j),z(2,j),'b','k.','SizeData',200);hold on;
46     elseif I_assign(j)==2
47         scatter(z(1,j),z(2,j),'g','k.','SizeData',200);hold on;
48     else
49         scatter(z(1,j),z(2,j),'r','k.','SizeData',200);hold on;
50     end
51 end
52 % plot medoids
53 % for L=1:k
54 %     scatter(z(1,I_bar(L)),z(2,I_bar(L)),'m','p','SizeData',170);
55 % end
56 title('I-assign')
57 toc

```

5.4 Problem 3

```

1 clear all;clc;
2 % parameters
3 k=2;
4 tau=1e-4;
5 ninit = 20;
6
7 load CardiacSPECT.mat
8 [n,p]=size(X);
9 % Get distance matrix D
10 D = zeros(p,p);
11 for i=1:p
12     for j=1:p
13         n11_n00 = length(find(X(:,i)==X(:,j))); % n11+n00
14         n10_n01 = length(find(X(:,i)~=X(:,j))); % n10+n01
15         D(i,j) = n10_n01/(n10_n01+n11_n00);
16     end
17 end
18
19 [I_assign, I_bar] = my_k_medoids(k, D, tau, ninit);
20
21 % get C
22 C=zeros(2,2);
23 for j=1:p
24     cluster=I_assign(j);
25     if cluster==2 && I(j)==1 % # of 1 in cluster 2(B)
26         C(1,2)=C(1,2)+1;
27     elseif cluster==2 && I(j)==0 % # of 0 in cluster 2(B)
28         C(2,2)=C(2,2)+1;
29     elseif cluster==1 && I(j)==1 % # of 1 in cluster 1(A)
30         C(1,1)=C(1,1)+1;

```



```

31     elseif cluster==1 && I(j)==0 % # of 0 in cluster 1(A)
32         C(2,1)=C(2,1)+1;
33     end
34 end
35 C
36
37 tic
38 figure();
39 Xbar = mean(X,2);
40 Xc = X - Xbar;
41 [u,d,v] = svd(Xc);
42 sigma=diag(d);
43 z = u(:,1:3)'*Xc;
44 % plot the groups
45 subplot(1,2,1);
46 for j=1:length(I)
47     if I(j)==1
48         scatter(z(1,j),z(2,j),'r','k.','SizeData',200);hold on;
49     elseif I(j)==0
50         scatter(z(1,j),z(2,j),'b','k.','SizeData',200);hold on;
51     end
52 end
53 title('Given I')
54 subplot(1,2,2);
55 for j=1:length(I_assign)
56     if I_assign(j)==2
57         scatter(z(1,j),z(2,j),'r','k.','SizeData',200);hold on;
58     elseif I_assign(j)==1
59         scatter(z(1,j),z(2,j),'b','k.','SizeData',200);hold on;
60     end
61 end
62 % plot medoids
63 for L=1:k
64     scatter(z(1,I_bar(L)),z(2,I_bar(L)),'m','p','SizeData',170);
65 end
66 title('I-assign computed with k-medoids')
67 toc

```

5.5 Problem 4

```

1 clear all;clc;
2 % parameters
3 k=2;
4 tau=1e-4;
5 ninit = 20;
6
7 load CongressionalVoteData;
8
9 % discard the one who didn't vote
10 sumColumn=sum(abs(X),1); % find the 0
11 vote = find(sumColumn~=0); % discard the 0
12 X=X(:,vote);
13 I=I(vote);
14 [n,p]=size(X); % n=16, p=434
15
16 % distance matrix
17 D=zeros(p,p);
18 for i=1:p
19     for j=1:p
20         numerator = 0;
21         denominator = 0;
22         for s=1:n
23             if X(s,i)==X(s,j) && X(s,i)~=0 %agree, both are 1 or -1
24                 denominator = denominator+1;
25             elseif X(s,i)+X(s,j)==0 && X(s,i)~=0 %disagree, 1 and -1

```

```

26         denominator = denominator+1;
27         numerator = numerator+1;
28     end
29 end
30 if denominator==0
31     D(i,j)=1/2;
32 else
33     D(i,j)=numerator/denominator;
34 end
35 end
36 end
37
38 [Iassign, Ibar] = my_k_medoids(k, D, tau, ninit);
39
40 % get C
41 C=zeros(2,2);
42 for j=1:p
43     cluster=Iassign(j);
44     if cluster==2 && I(j)==1 % # of 1 in cluster 2(B)
45         C(1,2)=C(1,2)+1;
46     elseif cluster==2 && I(j)==0 % # of 0 in cluster 2(B)
47         C(2,2)=C(2,2)+1;
48     elseif cluster==1 && I(j)==1 % # of 1 in cluster 1(A)
49         C(1,1)=C(1,1)+1;
50     elseif cluster==1 && I(j)==0 % # of 0 in cluster 1(A)
51         C(2,1)=C(2,1)+1;
52     end
53 end
54 C
55
56 tic
57 figure();
58 Xbar = mean(X,2);
59 Xc = X - Xbar;
60 [u,d,v] = svd(Xc);
61 z = u(:,1:2)*Xc;
62 % plot the groups
63 subplot(1,2,1);
64 for j=1:length(I)
65     if I(j)==1
66         scatter(z(1,j),z(2,j),'r','k.','SizeData',200);hold on;
67     else
68         scatter(z(1,j),z(2,j),'b','k.','SizeData',200);hold on;
69     end
70 end
71 title('Given I')
72 subplot(1,2,2);
73 for j=1:length(Iassign)
74     if Iassign(j)==2
75         scatter(z(1,j),z(2,j),'r','k.','SizeData',200);hold on;
76     else
77         scatter(z(1,j),z(2,j),'b','k.','SizeData',200);hold on;
78     end
79 end
80 % plot medoids
81 for L=1:k
82     scatter(z(1,Ibar(L)),z(2,Ibar(L)),'m','p','SizeData',170);
83 end
84 title('I-assign computed with k-medoids')
85 toc

```

5.6 Problem 5

```

1 clear all;clc;
2 % parameters

```

```

3 k=3;
4 tau=1e-4;
5 ninit = 20;
6
7 load IrisData;
8 [n,p]=size(X);
9 % No annotation included, so I got it from the website of UCI
10 I = zeros(p,1);
11 I(1:50)=1;
12 I(51:100)=2;
13 I(101:p)=3;
14
15 % Get distance matrix D
16 D = zeros(p,p);
17 for i=1:p
18     for j=1:p
19         D(i,j)=norm(X(:,i)-X(:,j),2);
20     end
21 end
22
23 % choose one and comment the other one
24 [Iassign, Ibar] = my_k_medoids(k, D, tau, ninit); % k-medoids
25 %Iassign = my_k_means(k, X, tau, ninit); % k-means
26
27 % get C
28 C=zeros(3,3);
29 for j=1:p
30     C(I(j),Iassign(j)) = C(I(j),Iassign(j))+1;
31 end
32 C
33
34 tic
35 figure();
36 Xbar = mean(X,2);
37 Xc = X - Xbar;
38 [u,d,v] = svd(Xc);
39 z = u(:,1:2)'*Xc;
40 subplot(1,2,1);
41 for j=1:length(I)
42     if I(j)==1
43         scatter(z(1,j),z(2,j),'b','k.','SizeData',200);hold on;
44     elseif I(j)==2
45         scatter(z(1,j),z(2,j),'g','k.','SizeData',200);hold on;
46     else
47         scatter(z(1,j),z(2,j),'r','k.','SizeData',200);hold on;
48     end
49 end
50 title('Given I')
51 subplot(1,2,2);
52 for j=1:length(Iassign)
53     if Iassign(j)==1
54         scatter(z(1,j),z(2,j),'b','k.','SizeData',200);hold on;
55     elseif Iassign(j)==2
56         scatter(z(1,j),z(2,j),'g','k.','SizeData',200);hold on;
57     else
58         scatter(z(1,j),z(2,j),'r','k.','SizeData',200);hold on;
59     end
60 end
61 % plot medoids
62 for L=1:k
63     scatter(z(1,Ibar(L)),z(2,Ibar(L)),'m','p','SizeData',170);
64 end
65 title('I-assign')
66 toc

```