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 seperate 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.

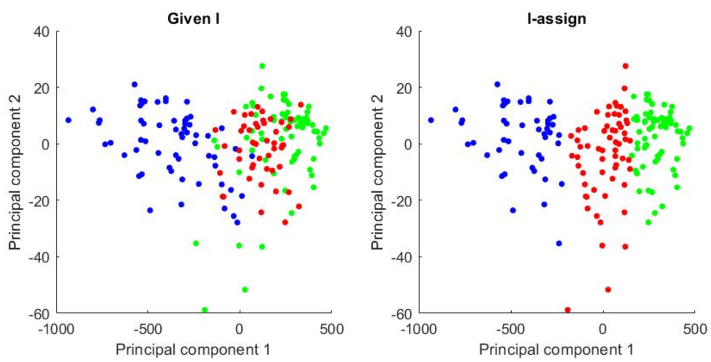

Figure 1: WineData: k-means

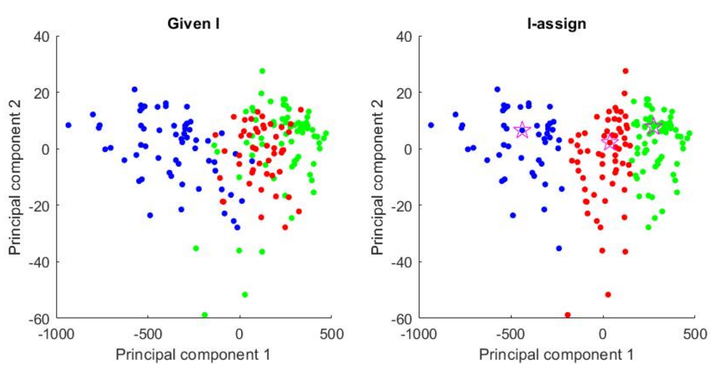

Figure 2: WineData: k-medoids. (Purple stars represent medoids.)

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:

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

$c_{11} + c_{12} = 84$	= total number of 1's in I
$c_{21} + c_{22} = 103$	= total number of 0's in I

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.

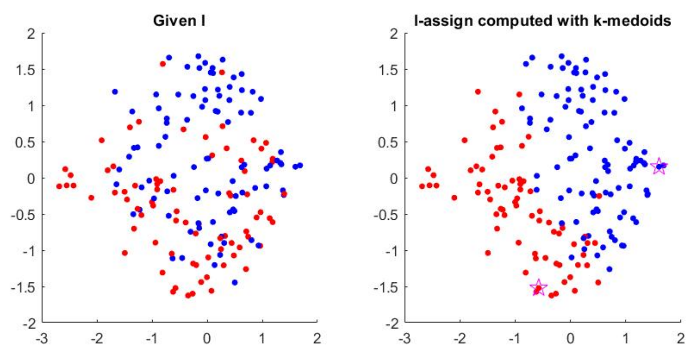

Figure 3: CardiacSPECT: k-medoids. (Purple stars represent medoids.)

For the second definition of matrix C in the problem, we interchenge 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.

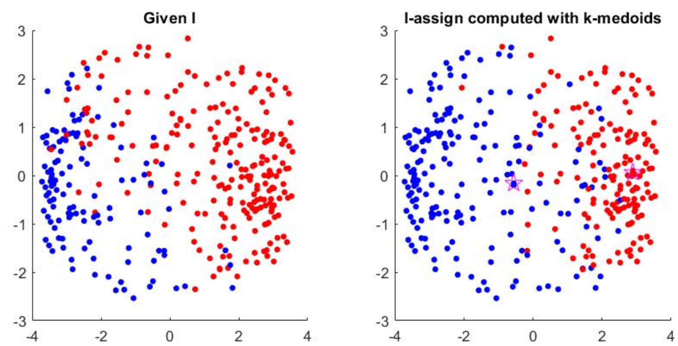

Figure 4: CongressionalVoteData: k-medoids. (Purple stars represent medoids.)

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 disgree, 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 shoule 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.

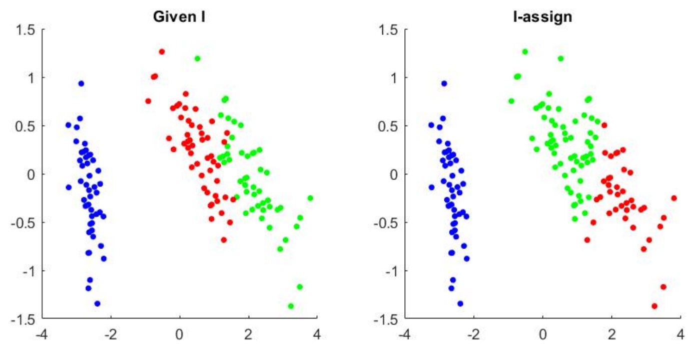

Figure 5: IrisData: k-means

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.

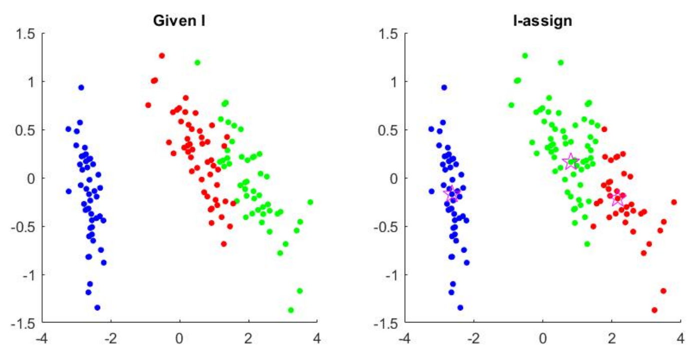

Figure 6: IrisData: k-medoids. (Purple stars represent medoids.)

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

```
function [I_assign] = my_k_means(k, X, tau, ninit)
% Jiasen Zhang

% input
% k = number of clusters
% X = data
% tau = tolerance
% ninit = times of initialization

% output
% I_assign = assignment vector indicating the cluster of each data vector

[n,p] = size(X);

% Initialize
Q = 0;
c = zeros(n,k);
for num=1:ninit
    partitionc = unidrnd(k,p,1);
    c_c = zeros(n,k); % current centroids
    ql = zeros(1,k); % corresponding within-cluster thightness
    for L=1:k
        % compute centroids
        c_c(:,L)=mean(X(:,partitionc==L),2);
    end
        % within-cluster tightness
    for j=1:p
        cluster = partitionc(j);
        ql(cluster)=ql(cluster)+norm(X(:,j)-c_c(:,cluster),2);
    end
    % compute currently overall tightness
    % choose smaller Q
    Q_c = sum(ql);
    if Q_c<Q || num==1
        Q=Q_c;
        c=c_c;
        partition=partitionc;
    end
end

% Begin iteration
diff = 2*tau; % make sure diff>tau at first
while(diff≥tau)
    % find closest cluster for x(j), update partition
    for j=1:p
        temp = norm(X(:,j)-c(:,1),2);
        partition(j)=1; % temporary cluster 1
        for L=2:k
            if norm(X(:,j)-c(:,L),2)<temp
                temp = norm(X(:,j)-c(:,L),2);
                % reassign x(j)
                partition(j) = L;
            end
        end
    end

    % update c(L) for each cluster and computed new ql
    for L=1:k
        c(:,L)=mean(X(:,partition==L),2);
        if max(isnan(c(:,L)))≠0
            c(:,L)=0;
        end
```

```
    end
    ql = zeros(k,1);
    for j=1:p
        cluster = partition(j);
        ql(cluster)=ql(cluster)+norm(X(:,j)-c(:,cluster),2);
    end

    % get new Q
    newQ = sum(ql);
    diff = abs(newQ-Q);
    Q = newQ;

end
% result
I_assign = partition;
end
```

5.2 k-medoids

```
function [I_assign, I_bar] = my_k_medoids(k, D, tau, ninit)
% Jiasen Zhang

% input
% k = number of clusters
% D = distance matrix
% tau = tolerance
% ninit = times of initialization

% output
% I_assign = assignment vector indicating the cluster of each data vector
% I_bar = medoids

[¬,p]=size(D);
% Initialize
Q = 0;
for num = 1:ninit
    % choose k data vectors randomly
    partitionc=ones(p,1);
    % get initial medoids randomly
    temp = randperm(p);
    c_c=temp(1:k);
    % for each x(j), find the nearest medoid
    for j=1:p
        Dm = D(j,c_c);
        cluster = find(Dm==min(Dm));
        partitionc(j) = cluster(1);
%         temp = D(j,c_c(1)); % temporary cluster 1
%         for L=2:k
%             if D(j,c_c(L))<temp % choose smaller one
%                 temp = D(j,c_c(L));
%                 partitionc(j) = L; % reassign x(j)
%             end
%         end
    end
    % get initial within-cluster tightness
    ql = zeros(k,1);
    for j=1:p
        cluster = partitionc(j);
        ql(cluster)=ql(cluster)+D(j,c_c(cluster));
    end
    Qc=sum(ql);
    if Qc<Q || num==1
        Q=Qc;
        c=c_c;
        partition=partitionc;
```

```
    end
end

% Begin iteration
diff = 2*tau; % make sure diff>tau at first
while(diff≥tau)
    % for each x(j), find the nearest medoid, update partition
    for j=1:p
        Dm = D(j,c);
        cluster = find(Dm==min(Dm));
        partition(j) = cluster(1);
    end

    % for each cluster, select a medoid so that the within-cluster
    % tightness is the smallest
    for L=1:k
        % get local distance matrix of cluster L
        index = find(partition==L);
        DL=D(index,index);
        sumDL=sum(DL,1);
        c0=index((sumDL==min(sumDL)));
        c(L)=c0(1);
    end

    % Get overall tightness newQ
    ql = zeros(k,1);
    for j=1:p % search all the points
        cluster = partition(j);
        ql(cluster)=ql(cluster)+D(j,c(cluster));
    end
    newQ = sum(ql);
    diff = abs(newQ-Q);
    Q = newQ;
end
I_assign = partition;
I_bar = c;
end
```

5.3 Problem 2

```
clear all;clc;
% parameters
k=3;
tau=1e-4; % tolerance
ninit = 20; % times of initialization

load WineData;
[n,p]=size(X);
% Get distance matrix D
D = zeros(p,p);
for i=1:p
    for j=1:p
        D(i,j)=norm(X(:,i)-X(:,j),2);
    end
end

% choose one and comment the other one
%[I_assign, I_bar] = my_k_medoids(k, D, tau, ninit); % k-medoids
I_assign = my_k_means(k, X, tau, ninit); % k-means

tic
% % plot taking several seconds
% subplot(1,2,1);plot(I);
% subplot(1,2,2);bar(I_assign);
figure();
```

```
Xbar = mean(X,2);
Xc = X - Xbar;
[u,d,v] = svd(Xc);
z = u(:,1:2)'*Xc;
% plot the 3 groups
subplot(1,2,1);
for j=1:length(I)
    if I(j)==1
        scatter(z(1,j),z(2,j),'b','k.','SizeData',200);hold on;
    elseif I(j)==2
        scatter(z(1,j),z(2,j),'g','k.','SizeData',200);hold on;
    else
        scatter(z(1,j),z(2,j),'r','k.','SizeData',200);hold on;
    end
end
title('Given I')
subplot(1,2,2);
for j=1:length(I_assign)
    if I_assign(j)==1
        scatter(z(1,j),z(2,j),'b','k.','SizeData',200);hold on;
    elseif I_assign(j)==2
        scatter(z(1,j),z(2,j),'g','k.','SizeData',200);hold on;
    else
        scatter(z(1,j),z(2,j),'r','k.','SizeData',200);hold on;
    end
end
% plot medoids
% for L=1:k
%     scatter(z(1,I_bar(L)),z(2,I_bar(L)),'m','p','SizeData',170);
% end
title('I-assign')
toc
```

5.4 Problem 3

```
clear all;clc;
% parameters
k=2;
tau=1e-4;
ninit = 20;

load CardiacSPECT.mat
[n,p]=size(X);
% Get distance matrix D
D = zeros(p,p);
for i=1:p
    for j=1:p
        n11_n00 = length(find(X(:,i)==X(:,j))); % n11+n00
        n10_n01 = length(find(X(:,i)≠X(:,j))); % n10+n01
        D(i,j) = n10_n01/(n10_n01+n11_n00);
    end
end

[I_assign, I_bar] = my_k_medoids(k, D, tau, ninit);

% get C
C=zeros(2,2);
for j=1:p
    cluster=I_assign(j);
    if cluster==2 && I(j)==1 % # of 1 in cluster 2(B)
        C(1,2)=C(1,2)+1;
    elseif cluster==2 && I(j)==0 % # of 0 in cluster 2(B)
        C(2,2)=C(2,2)+1;
    elseif cluster==1 && I(j)==1 % # of 1 in cluster 1(A)
        C(1,1)=C(1,1)+1;
```

```
    elseif cluster==1 && I(j)==0 % # of 0 in cluster 1(A)
        C(2,1)=C(2,1)+1;
    end
end
C

tic
figure();
Xbar = mean(X,2);
Xc = X - Xbar;
[u,d,v] = svd(Xc);
sigma=diag(d);
z = u(:,1:3)'*Xc;
% plot the groups
subplot(1,2,1);
for j=1:length(I)
    if I(j)==1
        scatter(z(1,j),z(2,j),'r','k.','SizeData',200);hold on;
    elseif I(j)==0
        scatter(z(1,j),z(2,j),'b','k.','SizeData',200);hold on;
    end
end
title('Given I')
subplot(1,2,2);
for j=1:length(I_assign)
    if I_assign(j)==2
        scatter(z(1,j),z(2,j),'r','k.','SizeData',200);hold on;
    elseif I_assign(j)==1
        scatter(z(1,j),z(2,j),'b','k.','SizeData',200);hold on;
    end
end
% plot medoids
for L=1:k
    scatter(z(1,I_bar(L)),z(2,I_bar(L)),'m','p','SizeData',170);
end
title('I-assign computed with k-medoids')
toc
```

5.5 Problem 4

```
clear all;clc;
% parameters
k=2;
tau=1e-4;
ninit = 20;

load CongressionalVoteData;

% discard the one who didn't vote
sumColumn=sum(abs(X),1); % find the 0
vote = find(sumColumn~=0); % discard the 0
X=X(:,vote);
I=I(vote);
[n,p]=size(X); % n=16, p=434

% distance matrix
D=zeros(p,p);
for i=1:p
    for j=1:p
        numerator = 0;
        denominator = 0;
        for s=1:n
            if X(s,i)==X(s,j) && X(s,i)~=0 %agree, both are 1 or -1
                denominator = denominator+1;
            elseif X(s,i)+X(s,j)==0 && X(s,i)~=0 %disagree, 1 and -1
```

```
                denominator = denominator+1;
                numerator = numerator+1;
            end
        end
        if denominator==0
            D(i,j)=1/2;
        else
            D(i,j)=numerator/denominator;
        end
    end
end

[I_assign, I_bar] = my_k_medoids(k, D, tau, ninit);

% get C
C=zeros(2,2);
for j=1:p
    cluster=I_assign(j);
    if cluster==2 && I(j)==1 % # of 1 in cluster 2(B)
        C(1,2)=C(1,2)+1;
    elseif cluster==2 && I(j)==0 % # of 0 in cluster 2(B)
        C(2,2)=C(2,2)+1;
    elseif cluster==1 && I(j)==1 % # of 1 in cluster 1(A)
        C(1,1)=C(1,1)+1;
    elseif cluster==1 && I(j)==0 % # of 0 in cluster 1(A)
        C(2,1)=C(2,1)+1;
    end
end
C

tic
figure();
Xbar = mean(X,2);
Xc = X - Xbar;
[u,d,v] = svd(Xc);
z = u(:,1:2)'*Xc;
% plot the groups
subplot(1,2,1);
for j=1:length(I)
    if I(j)==1
        scatter(z(1,j),z(2,j),'r','k.','SizeData',200);hold on;
    else
        scatter(z(1,j),z(2,j),'b','k.','SizeData',200);hold on;
    end
end
title('Given I')
subplot(1,2,2);
for j=1:length(I_assign)
    if I_assign(j)==2
        scatter(z(1,j),z(2,j),'r','k.','SizeData',200);hold on;
    else
        scatter(z(1,j),z(2,j),'b','k.','SizeData',200);hold on;
    end
end
% plot medoids
for L=1:k
    scatter(z(1,I_bar(L)),z(2,I_bar(L)),'m','p','SizeData',170);
end
title('I-assign computed with k-medoids')
toc
```

5.6 Problem 5

```
clear all;clc;
% parameters
```

```
k=3;
tau=1e-4;
ninit = 20;

load IrisData;
[n,p]=size(X);
% No annotation included, so I got it from the website of UCI
I = zeros(p,1);
I(1:50)=1;
I(51:100)=2;
I(101:p)=3;

% Get distance matrix D
D = zeros(p,p);
for i=1:p
    for j=1:p
        D(i,j)=norm(X(:,i)-X(:,j),2);
    end
end

% choose one and comment the other one
[I_assign, I_bar] = my_k_medoids(k, D, tau, ninit); % k-medoids
%I_assign = my_k_means(k, X, tau, ninit); % k-means

% get C
C=zeros(3,3);
for j=1:p
    C(I(j),I_assign(j)) = C(I(j),I_assign(j))+1;
end
C

tic
figure();
Xbar = mean(X,2);
Xc = X - Xbar;
[u,d,v] = svd(Xc);
z = u(:,1:2)'*Xc;
subplot(1,2,1);
for j=1:length(I)
    if I(j)==1
        scatter(z(1,j),z(2,j),'b','k.','SizeData',200);hold on;
    elseif I(j)==2
        scatter(z(1,j),z(2,j),'g','k.','SizeData',200);hold on;
    else
        scatter(z(1,j),z(2,j),'r','k.','SizeData',200);hold on;
    end
end
title('Given I')
subplot(1,2,2);
for j=1:length(I_assign)
    if I_assign(j)==1
        scatter(z(1,j),z(2,j),'b','k.','SizeData',200);hold on;
    elseif I_assign(j)==2
        scatter(z(1,j),z(2,j),'g','k.','SizeData',200);hold on;
    else
        scatter(z(1,j),z(2,j),'r','k.','SizeData',200);hold on;
    end
end
% plot medoids
for L=1:k
    scatter(z(1,I_bar(L)),z(2,I_bar(L)),'m','p','SizeData',170);
end
title('I-assign')
toc
```