به نام خدا

دانشگاه تهران

دانشکده مهندسی برق و کامپیوتر
جداسازی کور منابع
گزارش پروژه ۱۲

نام و نام خانوادگی	فاطمه صالحی
شماره دانشجویی	۸۱۰۱۹۸۴۲۳

سوال ۱

تولید ماتریس دیکشنری :

```
%% Dictionary generation
while(true)
    D = randn(3,6);
    D = D./repmat(sqrt(sum(D.^2)),3,1);
    ItsOk = Mutual_Coherence(D);
    if ItsOk==1
        break
    end
end
```

```
function ItsOk = Mutual_Coherence(D)
    C = D'*D;
    S = size(C,1);
    Eye = eye(S);
    C = abs(C- C.*Eye);
    mutual_coherence = max(max(C));
    if mutual_coherence > 0.9
        ItsOk = 0;
    else
        ItsOk = 1;
    end
end
```

تولید ماتریس منابع :

```
%% Source generation
S = zeros(6,1000);
for i = 1:1000
    Source = randi([1 6]);
    x = -5+10*rand(1,1);
    S(Source,i) = x;
end
```

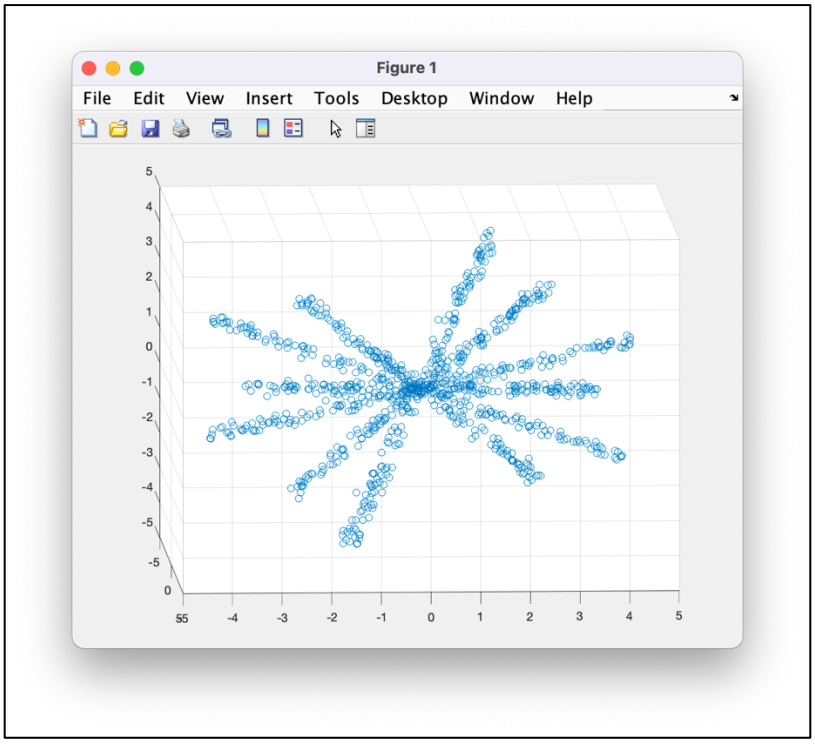
تولید ماتریس نویز :

```
%% Noise generation
Noise = randn(3,1000)*0.1;
```

تولید ماتریس مشاهدات :

```
%% Observation
X = D*S + Noise;
```

الف) Scatter plot مشاهدات :



همانطور که میبینیم داده ها در ۶ راستای متفاوت تشکیل شده اند؛ برای بدست آوردن ماتریس D مانند تمرین اول عمل میکنیم بدین معنا که داده های ابتدا و انتهای هر راستا را پیدا کرده و در یک آرایه ذخیره میکنیم:

$Data_{end}$, $Data_{beginnig}$

میانگین هر دسته را به عنوان نماینده هر آرایه بدست می آوریم :

$Mean_{end}$, $Mean_{beginnig}$

این ۲ نقطه ابتدا و انتهای هر راستا نشان میدهد. اگر از نقطه $Mean_{beginnig}$ به $Mean_{end}$ وصل کنیم و نرمالایز کنیم، خروجی این فرایند یکی از ستون های ماتریس D میباشد با این ابهام که جایگاه آن در ماتریس دیکشنری معلوم نیست (ابهام جایگشت) و همچنین ممکن است قرینه آن در ماتریس D وجود داشته باشد.
ابهام جایگشت از این نظر اهمیتی ندارد چون صرفا شماره منابع متفاوت خواهد بود؛ اما بسته به اینکه علامت مقدار هر منبع چقدر اهمیت دارد، ابهام دوم میتواند مشکل ساز باشد.

ب) روش MOD :

```
%% MOD
d = randn(3,6);
d = d./repmat(sqrt(sum(d.^2)),3,1);
x1 = X;
for ok =1:600
    % source finder
    sMP=zeros(6,1000);
    for L = 1:1000
        sMP(:,L) = S_finder(x1(:,L),d,sMP(:,L));
    end
      % dictionary finder
    d = x1*pinv(sMP);
```

```
    d = d./repmat(sqrt(sum(d.^2)),3,1);
end
Saving_D = D;
count =0;
for i = 1:size(d,2)
    Max = [];
    for  j = 1:size(Saving_D,2)
        Max = [Max,abs((d(:,i))' * Saving_D(:,j))];
    end
    if(max(Max) > 0.99)
        count = count + 1;
        [~,indx] = max(Max);
        Saving_D(:,indx) = [];
    end
end
SRR = count/size(D,2)*100;
fprintf("\n Successful Recovery Rate: %.3f \n",SRR);
```

```
Successful Recovery Rate: 100.000
```

ب) روش $K-SVD$:

```
%% K-SVD
N0 = 3;
x1 = X;
[n,m] = size(D);
d = randn(n,m);
d = d./sqrt(sum(d.^2));
[N,M] = size(S);
K_SVD = zeros(1,100);
for ok =1:100
    % source finder
    posOMP=zeros(1,N0);
    sOMP=zeros(N,M);
    for L = 1:M
        sOMP(:,L) = S_finder(X(:,L),x1(:,L),d,sOMP(:,L),posOMP,N0);
    end
      % dictionary finder
    for i = 1:50
        index = find(sOMP(i,:) ~= 0);

        num = 1:50;
        num(i)=[];
        X_r = X - d(:,num)*sOMP(num,:);
        X_m = X_r(:,index);

        [U,LANDA,V] = svd(X_m);
```

```
        [m,n] = size(LANDA);
        L = min(m,n);
        [landa,pos]=sort(diag(LANDA(1:L,1:L)),'descend');
        d(:,i) = U(:,pos(1));
        sOMP(i,index) = landa(1)* (V(:,pos(1)))';
    end
    % Representation Error of K_SVD
    X_hat = d*sOMP;
    K_SVD(ok) = trace((X-X_hat)*(X-X_hat)')/trace(X*X');
end
figure(1)
hold on
plot(K_SVD)


Saving_D = D;
count =0;
for i = 1:size(d,2)
    Max = [];
    for  j = 1:size(Saving_D,2)
        Max = [Max,abs((d(:,i))' * Saving_D(:,j))];
    end
    if(max(Max) > 0.99)
        count = count + 1;
        [~,indx] = max(Max);
        Saving_D(:,indx) = [];
    end
end
SRR = count/size(D,2)*100;
fprintf("\n Successful Recovery Rate of K-SVD: %.3f \n",SRR);
```

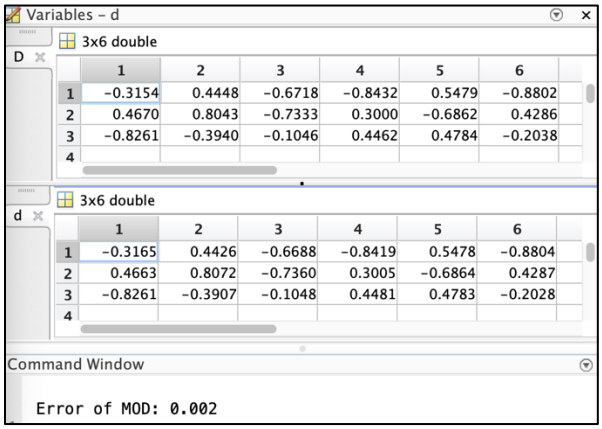
```
Successful Recovery Rate: 100.000
```

تابع S_finder:

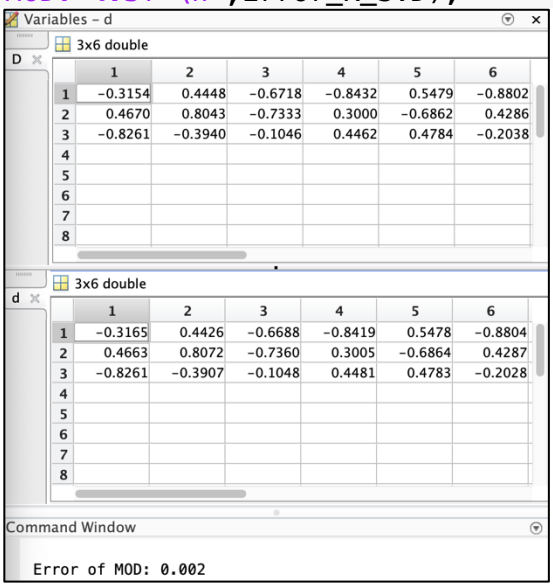
```
function [s] = S_finder(x1,D,sMP)
    ro=x1'*D;
    [~,posMP]=max(abs(ro));
    sMP(posMP)=ro(posMP);
    s = sMP;
end
```

ج) رفع ابهام منابع و بدست آوردن خطا :

```
%% Error for MOD
d = [d(:,4),-d(:,6),-d(:,3),-d(:,1),-d(:,2),d(:,5)];
sMP=zeros(6,1000);
for L = 1:1000
        sMP(:,L) = S_finder(x1(:,L),d,sMP(:,L));
end
Error_MOD = trace((sMP-S)*(sMP-S)')/trace(S*S');
fprintf("\n Error of MOD: %.3f \n",Error_MOD);
```




```
%% Error of K-SVD
d = [-d(:,2) -d(:,1) d(:,6) -d(:,4) d(:,5) d(:,3)];
sMP=zeros(6,1000);
for L = 1:1000
        sMP(:,L) = S_finder(x1(:,L),d,sMP(:,L));
end
Error_K_SVD = trace((sMP-S)*(sMP-S)')/trace(S*S');
fprintf("\n Error of MOD: %.3f \n",Error_K_SVD);
```



همانطور که مشاهده میشود به ازای $Successful\ Recovery\ Rate$ برابر ۱۰۰، هر دوی روش ها خطایی برابر ۰.۰۰۲ دارند.

سوال ۲

پیاده سازی روش MOD :

```
%% MOD
N0 = 3;
x1 = X;
[n,m] = size(D);
d = randn(n,m);
d = d./sqrt(sum(d.^2));
[n,m] = size(S);
MOD = zeros(1,100);
for ok =1:100
    % source finder
    posOMP=zeros(1,N0);
    sOMP=zeros(n,m);
    for L = 1:m
        sOMP(:,L) = S_finder(X(:,L),x1(:,L),d,sOMP(:,L),posOMP,N0);
    end
      % dictionary finder
    d = x1*pinv(sOMP);
    d = d./sqrt(sum(d.^2));
    % Representation Error of MOD
    X_hat = d*sOMP;
    MOD(ok) = trace((X-X_hat)*(X-X_hat)')/trace(X*X');
end
Saving_D = D;
count =0;
for i = 1:size(d,2)
    Max = [];
    for  j = 1:size(Saving_D,2)
        Max = [Max,abs((d(:,i))' * Saving_D(:,j))];
    end
    if(max(Max) > 0.99)
        count = count + 1;
        [~,indx] = max(Max);
        Saving_D(:,indx) = [];
    end
end
SRR = count/size(D,2)*100;
fprintf("\n Successful Recovery Rate of MOD: %.3f \n",SRR);
figure(1)
plot(MOD)
xlabel('Iterations')
xlabel('Representation Error of MOD')
```

پیاده سازی روش $K-SVD$:

```
%% K-SVD
N0 = 3;
x1 = X;
[n,m] = size(D);
d = randn(n,m);
d = d./sqrt(sum(d.^2));
[N,M] = size(S);
K_SVD = zeros(1,100);
for ok =1:100
    % source finder
    posOMP=zeros(1,N0);
    sOMP=zeros(N,M);
    for L = 1:M
        sOMP(:,L) = S_finder(X(:,L),x1(:,L),d,sOMP(:,L),posOMP,N0);
    end
      % dictionary finder
    for i = 1:50
        index = find(sOMP(i,:) ~= 0);

        num = 1:50;
        num(i)=[];
        X_r = X - d(:,num)*sOMP(num,:);
        X_m = X_r(:,index);

        [U,LANDA,V] = svd(X_m);

        [m,n] = size(LANDA);
        L = min(m,n);
        [landa,pos]=sort(diag(LANDA(1:L,1:L)),'descend');
        d(:,i) = U(:,pos(1));
        sOMP(i,index) = landa(1)* (V(:,pos(1)))';
    end
    % Representation Error of K_SVD
    X_hat = d*sOMP;
    K_SVD(ok) = trace((X-X_hat)*(X-X_hat)')/trace(X*X');
end
figure(1)
hold on
plot(K_SVD)


Saving_D = D;
count =0;
for i = 1:size(d,2)
    Max = [];
    for  j = 1:size(Saving_D,2)
        Max = [Max,abs((d(:,i))' * Saving_D(:,j))];
    end
```

```
    if(max(Max) > 0.99)
        count = count + 1;
        [~,indx] = max(Max);
        Saving_D(:,indx) = [];
    end
end
SRR = count/size(D,2)*100;
fprintf("\n Successful Recovery Rate of K-SVD: %.3f \n",SRR);
```

تابع S_finder:

```
function sOMP = S_finder(x,x1,D,sOMP,posOMP,N0)
   for i=1:N0
        ro=x1'*D;
        [~,posOMP(i)]=max(abs(ro));
        if i>1
             Dsub=D(:,posOMP(1:i));
             sOMP(posOMP(1:i),:)=pinv(Dsub)*x;
             x1=x-D*sOMP;
        else
             sOMP(posOMP(1))=ro(posOMP(1));
            x1=x1-sOMP(posOMP(1))*D(:,posOMP(1));
        end
   end
end
```

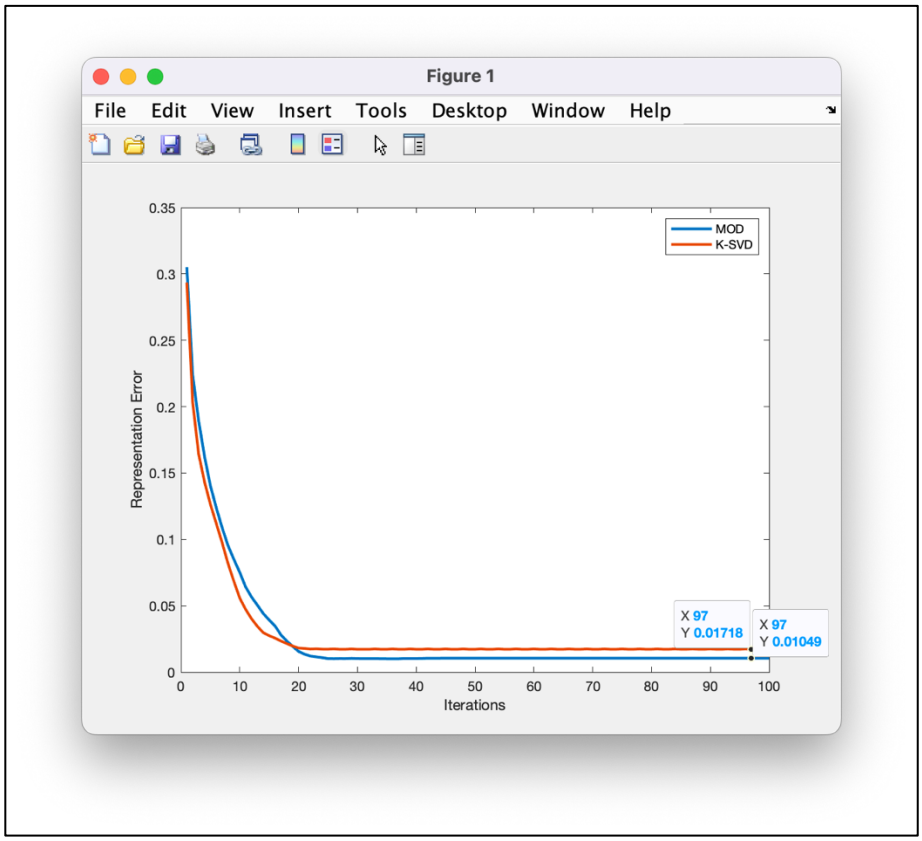
مقایسه $Successful\ Recovery\ Rate$ دو روش نام برده شده:

```
Successful Recovery Rate of MOD: 80.000

Successful Recovery Rate of K-SVD: 84.000
```

با ذکر این نکته که با ۱۰۰ بار تکرار هر یک از روش ها به نرخ های فوق رسیده ایم، نتیجه میشود که $K-SVD$ عملکرد بهتری دارد اما، طبق حدس و آزمایش انجام شده، همیشه روش $K-SVD$ بهتر نیست و به نظر بنده هر دوی این روش ها قوی هستند و تمایز فاحشی بین آنها نمیتوان قائل شد. برای اثبات این ادعا یک بار دیگر کد اجرا شده و نتایج زیر حاصل شد.

```
Successful Recovery Rate of MOD: 92.000

Successful Recovery Rate of K-SVD: 86.000
```

نمودار همگرایی روش $K-SVD$ و MOD :



همانطور که میبینیم روش $K-SVD$ زودتر از روش MOD همگرا شده است ولی در نهایت روش MOD خطای کمتری را دارا میباشد.