به نام خدا

مباحث ویژه در مخابرات – جداسازی کور منابع (دکتر اخوان)

تمرین سری **دوازدهم**

نیم‌سال دوم ۱۴۰۱-۱۴۰۰

عرفـان پـنـــاهی ۸۱۰۱۹۸۳۶۹

سوال ۱.

ابتدا طبق توضیحات صورت تمرین ، ماتریس منابع (S) و مخلوط کننده (A) و سپس ماتریس مشاهدات (X) را با اضافه کردن نویز به صورت زیر تعریف می‌کنیم.

```
T = 1000;
M = 3;
N = 6;
%--- D
while (1)
    D = randn(M,N);
    D = D ./ sqrt(sum(D.^2));
    DDT = (D'*D) - (D'*D) .* eye(6);
    u_D = max(max(abs(DDT)));
    if (u_D < 0.9)
        break
    end
end
%--- S
I = randi([1,N],[1,T]);
v = unifrnd(-5,5,1,T);
S = zeros(N,T);
for i = 1:T
    S(I(i),i) = v(i);
end
%--- noise
Noise = 0.1*randn(3,T);
%--- Observations
X = D*S + Noise;
```

همچنین در این سوال برای *sparse recovery* از الگوریتم *MP* با استفاده از تابع زیر استفاده می‌کنیم:

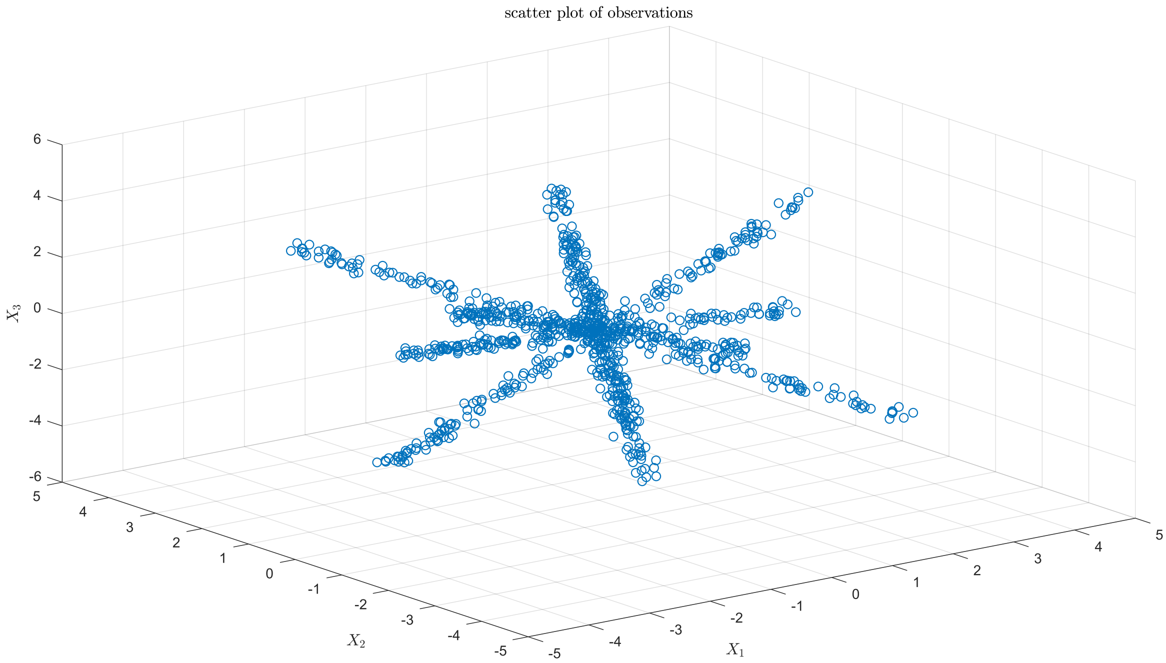
```
function [sMP] = MP(x,D,N)
    % N0 = 1;
    sMP = zeros(N,1);
    ro = x'*D;
    [~,posMP] = max(abs(ro));
    sMP(posMP) = ro(posMP);
end
```

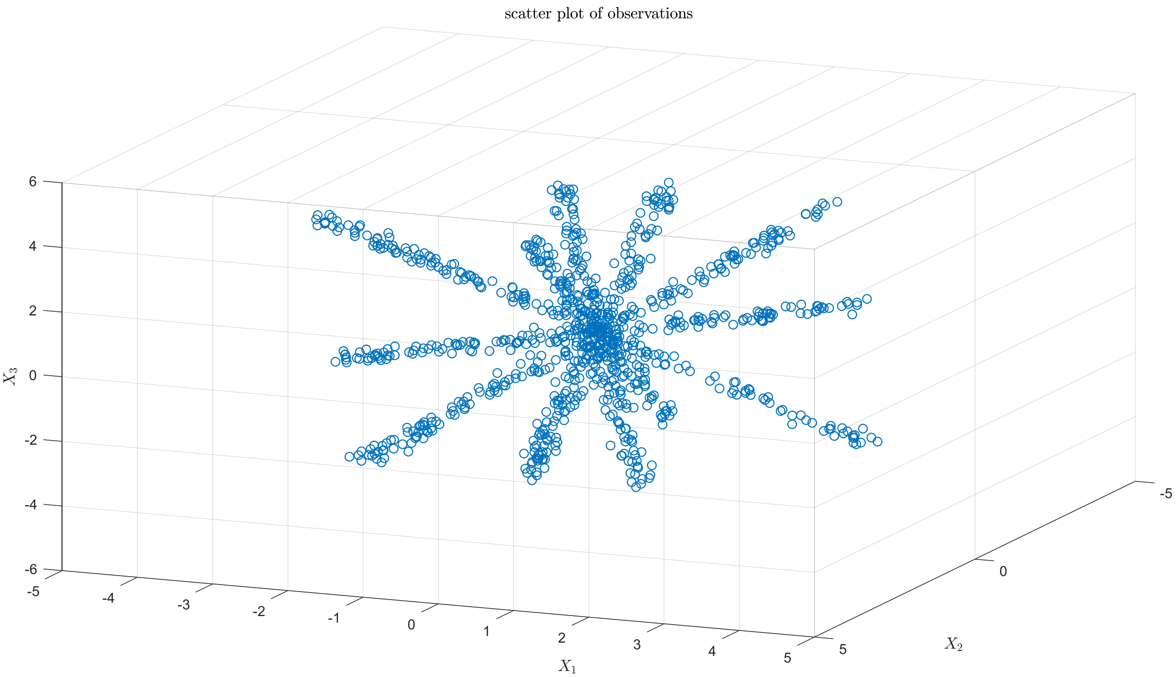
قسمت الف. رسم *scatter plot* مشاهدات

تصویر ۱ ، *scatter plot* مشاهدات را برحسب هر سطر مشاهدات نشان می‌دهد. همانطور که مشاهده می‌شود ، با توجه به اینکه منابع اسپارس با *sparsity level* برابر $N_0 = 1$ هستند ، سطر های مشاهدات در راستای بردار های مکانی انتشار یافته اند (نویز به قدری کوچک است که تقریباً بردار های مکانی قابل تشخیص اند.) می‌توانیم با استفاده از این نمودار ۳ بعدی ، بردار های مکانی را تشخیص دهیم اما چند ابهام سر راه وجود دارد.

ابهام ۱ (جایگشت) : در حقیقت نمی‌دانیم که هرکدام از بردار های مکانی مربوط به کدام یک از منابع هستند که البته این موضوع برای ما اهمیتی ندارد و صرفاً شماره منبع تغییر می کند.

ابهام ۲ (علامت و مقدار) : در حالت کلی اگر بردار های مکانی یا زمانی نرمالیزه نباشند ابهام مقدار (ضریب) وجود دارد. در حالتی که یکی از این دو بردار ها نرمالیزه باشند ، نمی‌توانیم از روی *scatter plot* تشخیص دهیم که راستای بردار مکانی مثبت است یا منفی.





تصویر ۱ : *scatter plot* مشاهدات

قسمت ب. روش *MOD*

در این روش مطابق با توضیحات جلسات از رویکرد *Alternation Minimization* استفاده می‌کنیم و به این منظور یک بار D را ثابت در نظر گرفته و S را تخمین می‌زنیم و در مرحله بعد S را ثابت گرفته و D را تخمین می‌زنیم. در هر تکرار D با استفاده از *pseudo inverse* استفاده می‌کنیم.

قطعه کد مربوط به این روش به صورت زیر است.

```
N0 = 1;
Dhat = randn(M,N);
Dhat = Dhat ./ sqrt(sum(Dhat.^2));
for i = 1:500
% D is fixed
    shat = zeros(N,T);
    for j = 1:T
        shat(:,j) = MP(X(:,j),Dhat,N);
    end
% S is fixed
    Dhat = X * pinv(shat);
    Dhat = Dhat ./ sqrt(sum(Dhat.^2));
end
```

همچنین *Succesfully Recovery Rate* مطابق با تعریف داده شده با استفاده از قطعه کد زیر بدست می‌آید:

```
corr = Dhat' * D;
SPR = sum(sum(abs(corr) >= 0.99) >= 1) / N;
```

در نهایت مقدار *SPR* به صورت زیر بدست می‌آید:

```
MOD : Successful Recovery Rate: 1.000

D =

    0.1932   -0.5454   -0.5446    0.3013    0.8726    0.0488
    0.8824    0.1860   -0.8381   -0.9290    0.3134   -0.1317
   -0.4290   -0.8173    0.0308   -0.2151   -0.3746   -0.9901

>> Dhat

Dhat =

    0.1932    0.2991   -0.8703    0.5462    0.0483    0.5425
    0.8832   -0.9291   -0.3128   -0.1832   -0.1313    0.8393
   -0.4273   -0.2174    0.3805    0.8174   -0.9902   -0.0347
```

حال می‌خواهیم با استفاده از $\widehat{D}$ و D ابهام جایگشت بردار های مکانی را برطرف کنیم و سپس خطای این روش را بدست آوریم (گزارش در قسمت د)

```
Order = [1,4,6,2,3,5];
Neg = [2,3,5];
Shat_MOD = shat(Order,:);
Shat_MOD(Neg,:) = -Shat_MOD(Neg,:);
E_MOD = trace((S - Shat_MOD)*(S - Shat_MOD)')/trace(S*S');
```

قسمت ج. روش K-SVD

در این روش مانند روش قبلی از رویکرد *Alternation Minimization* استفاده می‌کنیم و به این منظور یک بار D را ثابت در نظر گرفته و S را تخمین می‌زنیم و در مرحله بعد S را ثابت گرفته و D را تخمین می‌زنیم. در این روش در هر تکرار برای تخمین D از رویکرد *SVD* استفاده می‌کنیم.

قطعه کد مربوط به این روش به صورت زیر است.

```
N0 = 1;
Dhat = randn(M,N);
Dhat = Dhat ./ sqrt(sum(Dhat.^2));
for i = 1:500
% D is fixed
    shat = zeros(N,T);
    for j = 1:T
        shat(:,j) = MP(X(:,j),Dhat,N);
    end
% S is fixed
    for j = 1:N
        n = 1:N;
        n(j) = [];
        Xr = X - Dhat(:,n) * shat(n,:);
        k = find(shat(j,:) ~= 0);
        mXr = Xr(:,k);
        [U,G,V] = svd(mXr);
        [m,n] = size(G);
        L = min(m,n);
        [~,index] = sort(diag(G(1:L,1:L)));
        Dhat(:,j) = U(:,index(end));
        Dhat = Dhat ./ sqrt(sum(Dhat.^2));
        shat(j,k) = G(index(end),index(end)) * (V(:,index(end)))';
    end
end
```

همچنین *Succesfully Recovery Rate* مثل قسمت قبل بدست می‌آید:

```
K-SVD : Successful Recovery Rate: 1.000

D =

    0.1932   -0.5454   -0.5446    0.3013    0.8726    0.0488
    0.8824    0.1860   -0.8381   -0.9290    0.3134   -0.1317
   -0.4290   -0.8173    0.0308   -0.2151   -0.3746   -0.9901

>> Dhat

Dhat =

    0.8703   -0.5462   -0.1932    0.0483    0.2991   -0.5425
    0.3128    0.1832   -0.8832   -0.1313   -0.9291   -0.8393
   -0.3805   -0.8174    0.4273   -0.9902   -0.2174    0.0347
```

حال می‌خواهیم با استفاده از $\hat{D}$ و D ابهام جایگشت بردار های مکانی را برطرف کنیم و سپس خطای این روش را بدست آوریم (گزارش در قسمت د)

```
Order = [3,2,6,5,1,4];
```

```
Neg = [1];
Shat_MOD = shat(Order,:);
Shat_MOD(Neg,:) = -Shat_MOD(Neg,:);
E_MOD = trace((S - Shat_MOD)*(S - Shat_MOD)')/trace(S*S');
```

قسمت د. محاسبه خطای هر دو روش

ابهامی که در این قسمت وجود دارد مانند دو ابهام قسمت الف است. برای رفع این دو ابهام به صورت دستی عمل می‌کنیم. به طوری که با مقایسه $\widehat{D}$ و D ، جایگشت منابع و همچنین علامت آنها را اصلاح می‌کنیم.

در نهایت برای محاسبه خطا از تعریف نُرم به صورت زیر استفاده می‌کنیم.

$$\| A \|_2^2 = trace(AA^T)$$

در نهایت خطای هر دو روش به صورت زیر بدست می‌آید:

```
MOD: E = 0.00153116725930525039
K-SVD: E = 0.00153116725930524974
```

سوال ۲.

در این قسمت قالب کلی کد شبیه به سوال ۱ خواهد بود با فرق این که زمانی که می‌خواهیم در هر تکرار S را تخمین بزنیم ، از الگوریتم *OMP* استفاده می‌کنیم. تابع مربوط به این الگوریتم به صورت زیر است (از تمرین ۱۰)

```
function [sOMP] = OMP(x,D,N,N0)
    x1 = x;
    posOMP = zeros(1,N0);
    sOMP = zeros(N,1);
    tic
    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
```

* **روش MOD :** کد مربوط به این قسمت به صورت زیر است:

```
Dhat = randn(M,N);
Dhat = Dhat ./ sqrt(sum(Dhat.^2));
EOR_MOD = zeros(1,100);
for i = 1:100
% D is fixed
    shat = zeros(N,T);
    for j = 1:T
        shat(:,j) = OMP(X(:,j),Dhat,N,N0);
    end
% S is fixed
    Dhat = X * pinv(shat);
    Dhat = Dhat ./ sqrt(sum(Dhat.^2));
    EOR_MOD(i) = trace((X - Dhat*shat)*(X - Dhat*shat)')/trace(X*X');
end
corr = Dhat' * D;
SPR = sum(sum(abs(corr) >= 0.99) >= 1) / N;
fprintf("MOD : Successful Recovery Rate: %.3f \n",SPR);
```

مقدار *SPR* به دست آمده بصورت زیر است:

```
MOD : Successful Recovery Rate: 0.880
```

* **روش K-SVD :** کد مربوط به این روش به صورت زیر است:

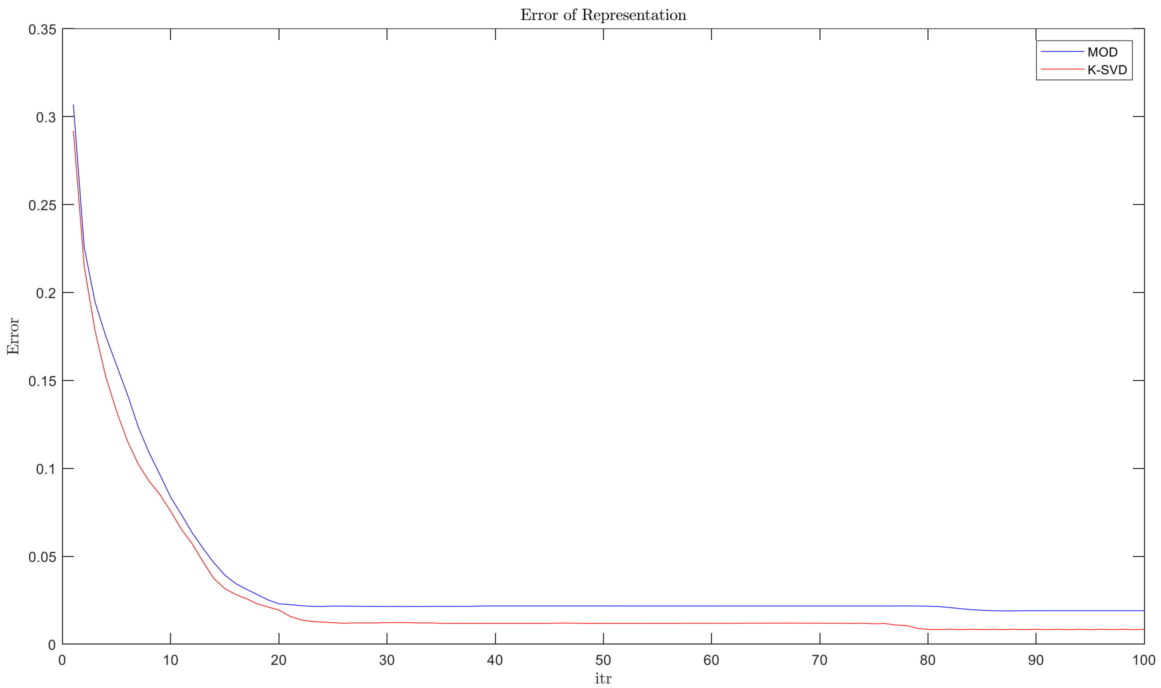
```
Dhat = randn(M,N);
Dhat = Dhat ./ sqrt(sum(Dhat.^2));
EOR_KSVD = zeros(1,100);
for i = 1:100
% D is fixed
    shat = zeros(N,T);
    for j = 1:T
        shat(:,j) = OMP(X(:,j),Dhat,N,N0);
    end
```

```
% S is fixed
    for j = 1:N
        n = 1:N;
        n(j) = [];
        Xr = X - Dhat(:,n) * shat(n,:);
        k = find(shat(j,:) ~= 0);
        mXr = Xr(:,k);
        [U,G,V] = svd(mXr);
        [m,n] = size(G);
        L = min(m,n);
        [~,index] = sort(diag(G(1:L,1:L)));
        Dhat(:,j) = U(:,index(end));
        Dhat = Dhat ./ sqrt(sum(Dhat.^2));
        shat(j,k) = G(index(end),index(end)) * (V(:,index(end)))';
    end
    EOR_KSVD(i) = trace((X - Dhat*shat)*(X - Dhat*shat)')/trace(X*X');
end
corr = Dhat' * D;
SPR = sum(sum(abs(corr) >= 0.99) >= 1) / N;
fprintf("K-SVD : Successful Recovery Rate: %.3f \n",SPR);
```

مقدار *SPR* به دست آمده بصورت زیر است:

```
K-SVD : Successful Recovery Rate: 0.940
```

نمودار همگرایی خطا برای هر دو روش مطابق با تصویر ۲ خواهد بود.



تصویر ۲ : نمودار همگرایی خطا

همانطور که در تصویر ۲ مشاهده می‌شود ، خطای هر دو روش تقریباً مشابه با هم خواهد بود.

این خطا به صورت رندوم در هر روش ممکن است کمتر یا بیشتر باشد اما در حالت کلی خطای هر دو روش در یک حد خواهند بود.