Paper: Sparse multidimensional exponential analysis with an application to radar imaging.

Annie Cuyt, Yuan Hou, Ferre Knaepkens, Wen-shin Lee

Sparse multidimensional exponential analysis with an application to radar imaging

6. Full scale ISAR illustration

Contents

- Script environment
- 6. Full scale ISAR illustration

Script environment

This script depends on the random number generator state.

```
clear
close all
```

6. Full scale ISAR illustration

When returning to the example in Section 4.3, we take the radar parameters, the signal-to-noise ratio and the vectors $\Delta_i, i = 1, 2, 3$ as specified there. We collect 30000 samples $F_s = f(s\Delta_1)$ and 30000 samples $F_{si} = f(s\Delta_1 + \Delta_i), i = 2, 3$ along each of the shifts, so a total of 90000 samples in total. These samples are now reorganized as follows for use with the technique described in Section 5.

With the total of 90000 samples we perform the following analyses. For each of the analyses we take $N = 6000, \nu = 2000, \eta = 1500, \mathrm{n} = 6000$. The remaining parameters for the sub-Nyquist sampling in the direction Δ_1 are:

- $m = 2, \mu = 1, \kappa = 6, \mathrm{p} = 0.3$;
- $m = 3, \mu = 1, \kappa = 4, \mathrm{p} = 2/9$;
- $m = 4, \mu = 1, \kappa = 3, \mathrm{p} = 0.125$.

In each of the above analyses, the sampling in the direction Δ_1 starts with F_0 and continues with $F_m, F_{2m}, \ldots$ The shifted samples, that serve the purpose of repairing the possible sub-Nyquist aliasing effect, start with F_1 and continue with $F_{m+1}, F_{2m+1}, \ldots$ In order to make good use of the samples inbetween, the procedure can be repeated $m-1$ times with the same m and μ but now starting the sampling, instead of at F_0, at F_1 and then at F_2 and so on till F_{m-1}. In this way a choice of m produces $m\kappa$ estimates for the $\exp(\Phi_i), i = 1, \ldots, \eta$ instead of

κ, and thus provides a sound basis for validation since $m\kappa$ is usually sufficiently large.

For our choices above, we have $m\kappa = 12$ for $m = 2, 3, 4$ and so we can take, for instance, $m_\delta = (5/6)m\kappa = 10$. In Figure 10 left and right, we show how accurate the scattering centers are reconstructed, under SNR = 20 noise: with every scattering center in the original data we associate the $\log_{10}$ of the Euclidean distance to the nearest reconstructed scattering center (in meter on the x-axis) and then accumulate these (tally is on the y-axis). The distinction between the two figures is that Figure 10 (left) is the result for $m = 1$ (516 scatterers reliably identified), without the use of the enhancement given in Section 5, and Figure 10 (right) is the result for $m = 4$ (696 scatterers detected and validated).

The improvement from $m = 1$ to $m = 4$ may not seem impressive at first sight. But note that the accurately reconstructed scattering centers (say $\log_{10}(\cdot) \leq -1$) from $m = 1$, need not be the same as the accurately reconstructed ones from the use of $m = 4$. Therefore the combination of both results, merely joining the 516 reconstructions from $m = 1$ with the 696 reconstructions from $m = 4$, immediately leads to the improved distance graph shown in Figure 11.

Eventually, all runs executed with $m = 1, 2, 3, 4$ can be combined, merely joining all the computed scatterer reconstructions: 516 from $m = 1$, 667 from $m = 2$, 673 from $m = 3$ and 696 from $m = 4$, adding up to 2552 in total, with many of them (almost) duplicates. This then leads to highly acurate results for most of the scatterers. In Figure 12 we see that in this combined output 81 percent of the scatterers is reconstructed within an error of at most 10 cm and 95 percent is found within a distance of 30 cm! Only 3 scatterers are not reconstructed within a distance of 1 m. The most inaccurately reconstructed scatterer in Figure 13 is near the engine outlet, where one can note that one reconstruction is slightly off. In Figure 13 the 2552 reconstructions are displayed altogether. Note that, thanks to the validation technique, there are no false results, as also pointed out for Figure 8 where the sub-Nyquist subdivision of the data samples is not yet put to work.

```
load('fighter_1000')

object_parms.loc = [fighter_x; fighter_y; fighter_z];
object_parms.ampl = s_i;
c = physconst('LightSpeed');

ISAR_parms.df = 0.0015e9;
ISAR_parms.dtheta = 3.75e-04;
ISAR_parms.dphi = 3.75e-04;

ISAR_parms.f0 = 7.9e9;
ISAR_parms.theta0 =  -0.024;
```

```
ISAR_parms.phi0 = -0.024;
ISAR_parms.fc = 8.4e9;

delta = [ -8, 1, 3 ;
       2,  -2, 6 ;
       6, 27 , 7];

for k = 1:3
    bound = 1.1*max(4/c*abs(ISAR_parms.df*...
                object_parms.loc(1,:)*delta(k,1)...
            + ISAR_parms.fc*ISAR_parms.dtheta*...
                object_parms.loc(2,:)*delta(k,2)...
            + ISAR_parms.fc*ISAR_parms.dphi*...
                object_parms.loc(3,:)*delta(k,3)));
    delta(k,:) = delta(k,:)/bound;
end

ISAR_parms.delta = delta;

signal_parms.nsamples = 30000;
signal_parms.SNR = 20;
signal_parms.overlap = 0;
signal_parms.nwindow = 1;

ESPRIT_parms.ncols = 2000;
ESPRIT_parms.nrows = 4001;
ESPRIT_parms.nterms = 1500;
ESPRIT_parms.shift = 1;

clust_parms.MinPts = 9;
clust_parms.epsvec = 2.^(0:10)*1e-5;
clust_parms.disc_eps = 0.5;
clust_parms.lb = 0.7;
clust_parms.ub = 1.3;

signal = cell2mat(create_signal(object_parms, ISAR_parms,...
         signal_parms));

loc = cell(1,4);

signal_parms.nsamples = 6000;
signal_parms.overlap = 0.6;
ESPRIT_parms.overlap = 0.6;
signal_parms.nwindow = 11;
data = cell(1,signal_parms.nwindow);
step = round((1-signal_parms.overlap)*signal_parms.nsamples);
```

```
for k = 1:signal_parms.nwindow
    data{k} = signal(:,(1:signal_parms.nsamples)+(k-1)*step);
end
loc{1} = ISARsolver(data,ISAR_parms,ESPRIT_parms,clust_parms,...
                    false);


ESPRIT_parms.rate = 2;
signal_parms.nsamples = 12000;
signal_parms.overlap = 0.7;
signal_parms.nwindow = 6;
data = cell(1,signal_parms.nwindow);
step = round((1-signal_parms.overlap)*signal_parms.nsamples);
for k = 1:signal_parms.nwindow
    data{k} = signal(:,(1:signal_parms.nsamples)+(k-1)*step);
end
loc{2} = ISARsolver_subsampled(data,ISAR_parms,ESPRIT_parms,...
                               clust_parms);


ESPRIT_parms.rate = 3;
signal_parms.nsamples = 18000;
signal_parms.overlap = 7/9;
signal_parms.nwindow = 4;
data = cell(1,signal_parms.nwindow);
step = round((1-signal_parms.overlap)*signal_parms.nsamples);
for k = 1:signal_parms.nwindow
    data{k} = signal(:,(1:signal_parms.nsamples)+(k-1)*step);
end
loc{3} = ISARsolver_subsampled(data,ISAR_parms,ESPRIT_parms,...
                               clust_parms);


ESPRIT_parms.rate = 4;
signal_parms.nsamples = 24000;
signal_parms.overlap = 0.875;
signal_parms.nwindow = 3;
data = cell(1,signal_parms.nwindow);
step = round((1-signal_parms.overlap)*signal_parms.nsamples);
for k = 1:signal_parms.nwindow
    data{k} = signal(:,(1:signal_parms.nsamples)+(k-1)*step);
end
loc{4} = ISARsolver_subsampled(data,ISAR_parms,ESPRIT_parms,...
                               clust_parms);
```

```
exp_idx = 1;
N = 100;
n = zeros(1,N);
distvec = 10.^linspace(-3,0,N);
for k = 1:N
    dist = distvec(k);
    n(k) = n_combined_fun(loc,dist,exp_idx);
end
figure
plot(log10(distvec),n,'b-o')
grid on
ylim([0,1000])
title({['Fig. 10: Accuracy of the reconstructed scatterers ',...
        'for m=1 (left) versus the data'],['(log_{10} of ',...
        'the Euclidean distance)']})


exp_idx = 4;
N = 100;
n = zeros(1,N);
distvec = 10.^linspace(-3,0,N);
for k = 1:N
    dist = distvec(k);
    n(k) = n_combined_fun(loc,dist,exp_idx);
end
figure
plot(log10(distvec),n,'b-o')
grid on
ylim([0,1000])
title({['Fig. 10: Accuracy of the reconstructed scatterers ',...
        'for m=4 (right) versus the data'],['(log_{10} of ',...
        'the Euclidean distance)']})


exp_idx = [1,4];
N = 100;
n = zeros(1,N);
distvec = 10.^linspace(-3,0,N);
for k = 1:N
    dist = distvec(k);
    n(k) = n_combined_fun(loc,dist,exp_idx);
end
figure
plot(log10(distvec),n,'b-o')
grid on
```

```
ylim([0,1000])
title({['Fig. 11: Accuracy of the reconstructed scatterers ',...
        '(m=1 and m=4 combined) versus'], ['the data ',...
        '(log_{10} of the Euclidean distance)']})


exp_idx = [1,2,3,4];
N = 100;
n = zeros(1,N);
distvec = 10.^linspace(-3,0,N);
for k = 1:N
    dist = distvec(k);
    n(k) = n_combined_fun(loc,dist,exp_idx);
end
figure
plot(log10(distvec),n,'b-o')
grid on
ylim([0,1000])
title({['Fig. 12: Accuracy of the reconstructed scatterers ',...
        '(m=1,2,3,4 combined) versus the'], ['data ',...
        '(log_{10} of the Euclidean distance)']})


figure;
scatter3(loc{1}(1,:), loc{1}(2,:), loc{1}(3,:), 7,'filled',...
        'MarkerFaceColor','r')
hold on
scatter3(loc{2}(1,:), loc{2}(2,:), loc{2}(3,:), 7,'filled',...
        'MarkerFaceColor','r')
scatter3(loc{3}(1,:), loc{3}(2,:), loc{3}(3,:), 7,'filled',...
        'MarkerFaceColor','r')
scatter3(loc{4}(1,:), loc{4}(2,:), loc{4}(3,:), 7,'filled',...
        'MarkerFaceColor','r')
xlabel('X'); ylabel('Y'); zlabel('Z');
grid on
view([60,20])
h = plot(shp);
h.FaceColor = [96,96,96]/255;
h.EdgeColor = [96,96,96]/255;
h.FaceAlpha = 0.1;
h.EdgeAlpha = 0.1;
axis([-15 15 -15 15 -5 10]);
title(['Fig. 13: Fighter jet reconstruction of 934 out ',...
       'of 1000 scatterers'])
```

```
function [n,idx,idxloc]=found_scatterers(X,Y,Z,loc1,loc2,loc3,eps)

X = X(:);
Y = Y(:);
Z = Z(:);
loc1 = loc1(:);
loc2 = loc2(:);
loc3 = loc3(:);

N = numel(X);
idx = false(1,N);
idxloc = true(1,numel(loc1));
for j = 1:N
    [d,i] = min(sqrt((X(j)-loc1(idxloc)).^2 + ...
                     (Y(j)-loc2(idxloc)).^2 + ...
                     (Z(j)-loc3(idxloc)).^2));
    if d < eps
        idx(j) = true;
        vi = true(1,sum(idxloc));
        vi(i) = false;
        idxloc(idxloc) = vi;
    end
end
n = sum(idx);
idxloc = ~idxloc;

end

function n = n_combined_fun(loc,dist,exp_idx)

load('fighter_1000')

exp_num = length(exp_idx);

idx = false(1,1000);
n = zeros(1,exp_num);
for k = 1:exp_num
    [n(k),idx_tmp,~] = found_scatterers(fighter_x(~idx),...
                fighter_y(~idx),fighter_z(~idx),...
                loc{exp_idx(k)}(1,:),loc{exp_idx(k)}(2,:),...
                loc{exp_idx(k)}(3,:),dist);
    idx(~idx) = idx_tmp;
end
n = sum(n);
end
```

Fig. 10: Accuracy of the reconstructed scatterers for m=1 (left) versus the data ($\log_{10}$ of the Euclidean distance)

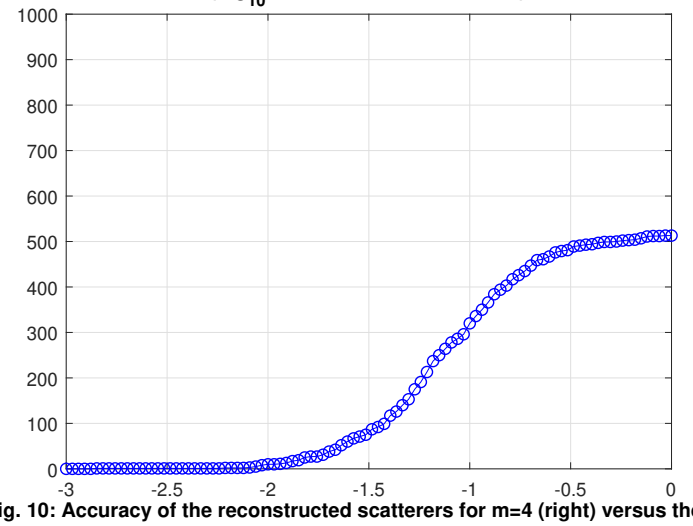

Fig. 10: Accuracy of the reconstructed scatterers for m=4 (right) versus the data ($\log_{10}$ of the Euclidean distance)

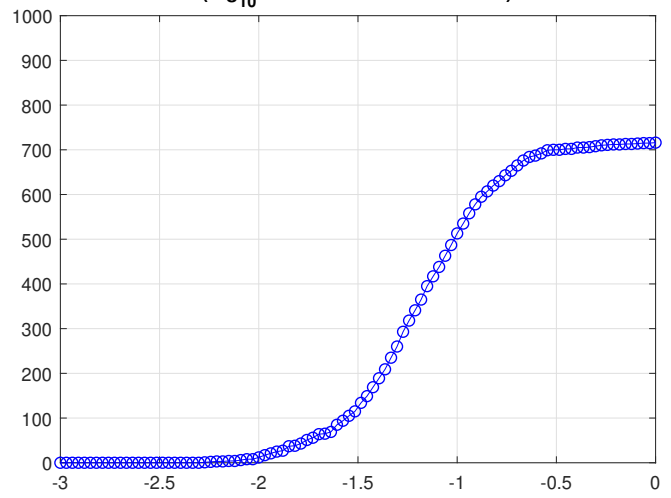

Fig. 11: Accuracy of the reconstructed scatterers (m=1 and m=4 combined) versus the data ($\log_{10}$ of the Euclidean distance)

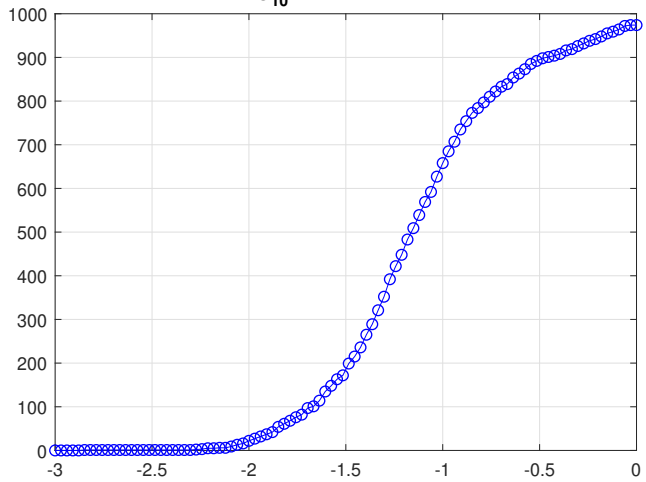

Fig. 12: Accuracy of the reconstructed scatterers (m=1,2,3,4 combined) versus the data ($\log_{10}$ of the Euclidean distance)

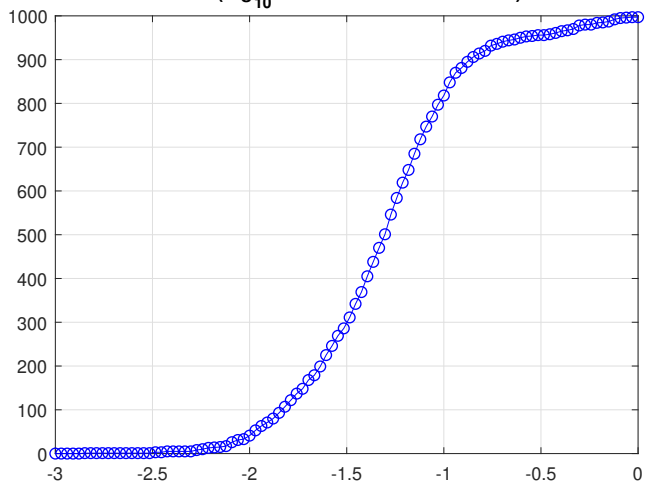
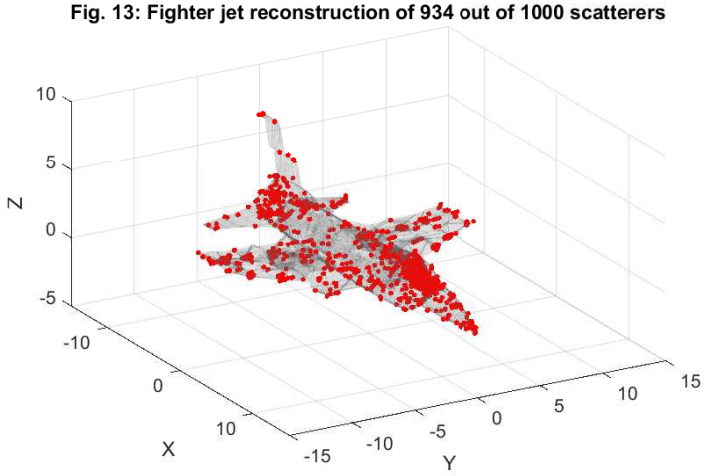
Fig. 13: Fighter jet reconstruction of 934 out of 1000 scatterers