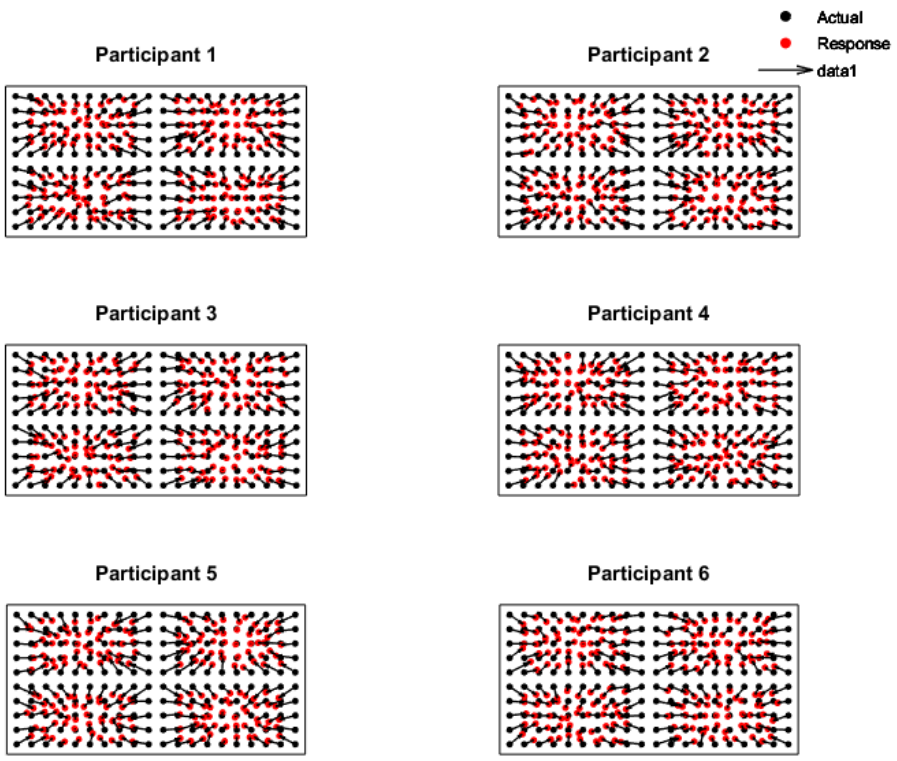
Table of Contents


```
addpath(genpath('..\..\..\prototypes_toolbox'))

clear; close all;

% load data
load('PrototypesData.mat', 'SubjectsData');

% data info
subjlist    = unique(SubjectsData.ParticipantID);
nsubj       = length(subjlist);

% compute error vectors
SubjectsData = prototypes_compute_errorVectors(SubjectsData);
```

plot errors

```
figure('Position', [827 310 819 632]);

% plot data for each participant
for s = 1:nsubj
    subplot(3, 2, s);

    % plot the actual dots and the responses
    prototypes_plot_dots(SubjectsData, s);

    % plot the error vectors
    hold on;prototypes_plot_errorVectors(SubjectsData, s);
    title(sprintf('Participant %d', s));
end
```



Compute cosine maps

```
% use 4 processor, if present
alphavalue              = 10;
nproc                   = 4;
SubjectsCosineMaps      = prototypes_compute_cosineMap(SubjectsData,
 alphavalue, nproc);

Computing cosine map for subject 1 using 4 processors...
Processor 4 of 4 processor has ended. Waiting for the other ones to
 finish...

Processor 2 of 4 processor has ended. Waiting for the other ones to
 finish...

Processor 1 of 4 processor has ended. Waiting for the other ones to
 finish...

Processor 3 of 4 processor has ended. Waiting for the other ones to
 finish...
Done
Computing cosine map for subject 2 using 4 processors...
Processor 4 of 4 processor has ended. Waiting for the other ones to
 finish...
```

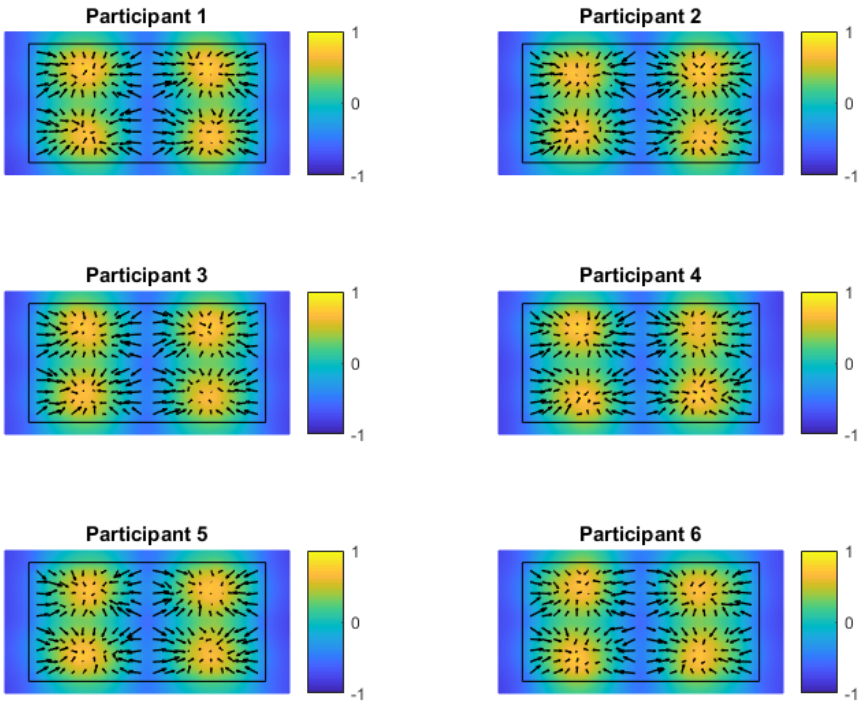
```
Processor 2 of 4 processor has ended. Waiting for the other ones to
 finish...

Processor 1 of 4 processor has ended. Waiting for the other ones to
 finish...

Processor 3 of 4 processor has ended. Waiting for the other ones to
 finish...
Done
Computing cosine map for subject 3 using 4 processors...
Processor 4 of 4 processor has ended. Waiting for the other ones to
 finish...

Processor 2 of 4 processor has ended. Waiting for the other ones to
 finish...

Processor 1 of 4 processor has ended. Waiting for the other ones to
 finish...

Processor 3 of 4 processor has ended. Waiting for the other ones to
 finish...
Done
Computing cosine map for subject 4 using 4 processors...
Processor 4 of 4 processor has ended. Waiting for the other ones to
 finish...

Processor 2 of 4 processor has ended. Waiting for the other ones to
 finish...

Processor 1 of 4 processor has ended. Waiting for the other ones to
 finish...

Processor 3 of 4 processor has ended. Waiting for the other ones to
 finish...
Done
Computing cosine map for subject 5 using 4 processors...
Processor 4 of 4 processor has ended. Waiting for the other ones to
 finish...

Processor 2 of 4 processor has ended. Waiting for the other ones to
 finish...

Processor 1 of 4 processor has ended. Waiting for the other ones to
 finish...

Processor 3 of 4 processor has ended. Waiting for the other ones to
 finish...
Done
Computing cosine map for subject 6 using 4 processors...
Processor 4 of 4 processor has ended. Waiting for the other ones to
 finish...

Processor 2 of 4 processor has ended. Waiting for the other ones to
 finish...
```

```
Processor 1 of 4 processor has ended. Waiting for the other ones to
 finish...

Processor 3 of 4 processor has ended. Waiting for the other ones to
 finish...
Done
```

Plot cosine maps

plot data for each participant

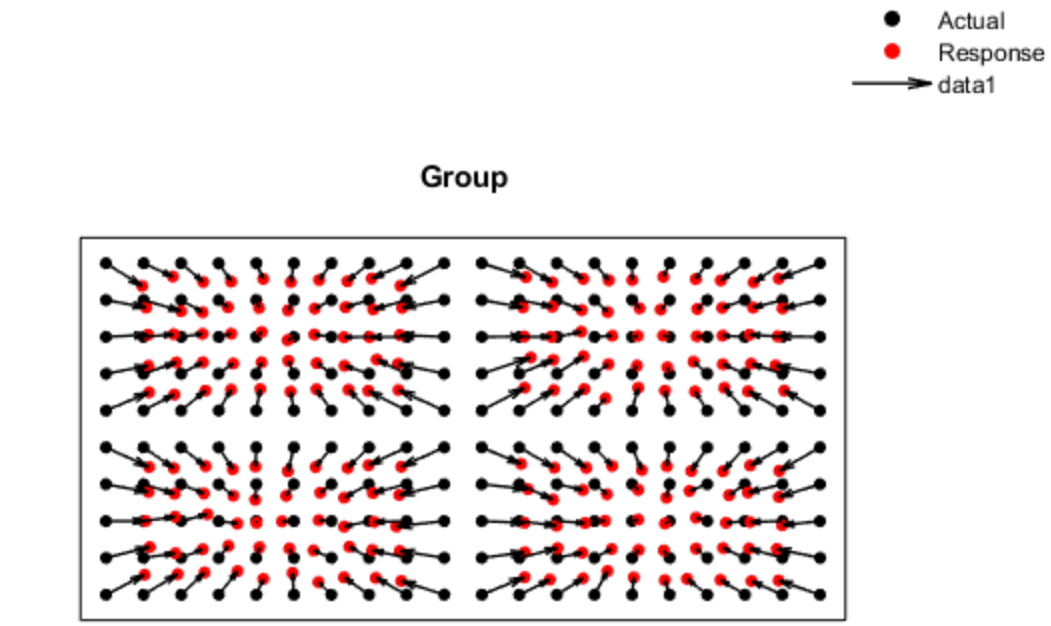
```
figure('Position', [827 310 819 632]);


for s = 1:nsubj
    subplot(3, 2, s);

    % plot the actual dots and the responses
    prototypes_plot_cosineMap(SubjectsCosineMaps, s);

    % plot the error vectors
    hold on;prototypes_plot_errorVectors(SubjectsData, s);
    title(sprintf('Participant %d', s));
end
```

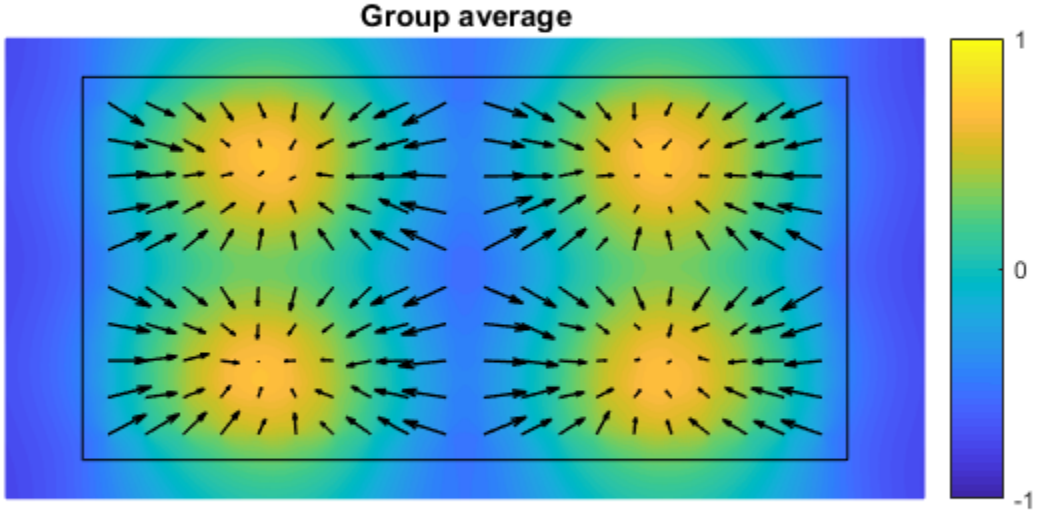


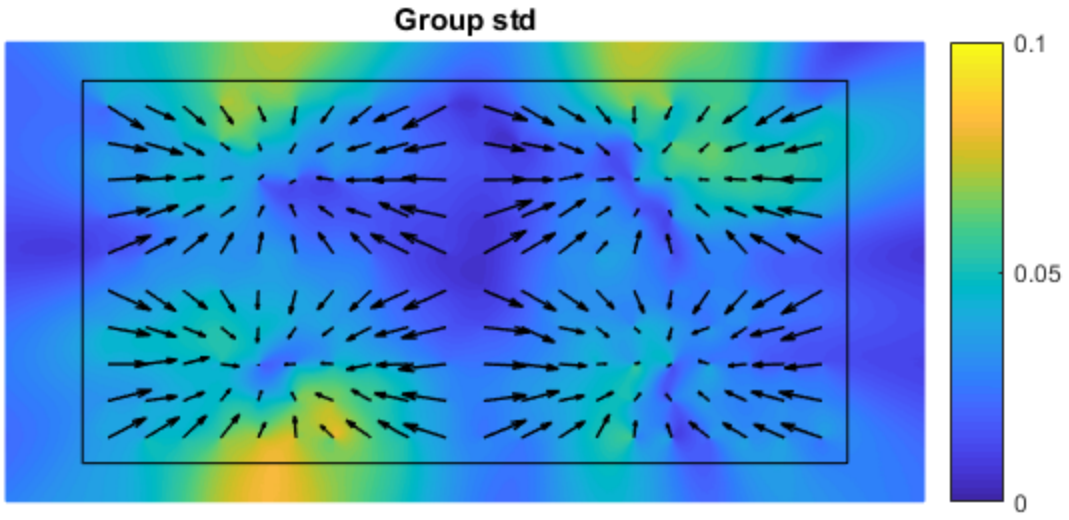
Stats (descriptive)

==========
average the error vectors across participants
==========
Remember that this can (should) be done only when the actual dots are the same

```
GroupData = prototypes_mean(SubjectsData);

% plot data
figure;prototypes_plot_dots(GroupData);
hold on;prototypes_plot_errorVectors(GroupData);
title('Group');

%
 ========================================================================
% average the cosine maps across participants
%
 ========================================================================
GroupCosineMaps = prototypes_mean(SubjectsCosineMaps);

% plot mean
figure;prototypes_plot_cosineMap(GroupCosineMaps);
hold on;prototypes_plot_errorVectors(GroupData);
title('Group average');

% plot standard deviation
figure;prototypes_plot_cosineMap(GroupCosineMaps, [], [0
 0.1], 'W_CosineMap_sd');
hold on;prototypes_plot_errorVectors(GroupData);
title('Group std');
```


Actual
Response
data1
Group


Group average
1
0
-1



stats (inferential) - permutation analysis

NOTE: You need to have cosmomvpa in the path (http://cosmomvpa.org/download.html)

```
opt                  = [];
opt.runPermutation   = 1;
opt.niter            = 500;
groupStat            = prototypes_stat_secondLevel(SubjectsCosineMaps,
 opt);

% plot the mean
figure;prototypes_plot_cosineMap(groupStat, [], [-0.5
 0.5], 'W_SimixSubject_avg');  % same as from prototypes_mean

% plot the t scores (no mask)
figure;prototypes_plot_cosineMap(groupStat, [], [-15
 15], 'W_SimixSubject_T');      % no masked

% plot the t scores (only shows the surviving pixels)
figure;prototypes_plot_cosineMap(groupStat, [], [-15
 15], 'W_SimixSubject_Tcorr');  % corrected (masked)
```

```
Undefined function 'cosmo_flatten' for input arguments of type 'cell'.

Error in prototypes_simMap2cosmo (line 23)
    ds{s}=cosmo_flatten(data, dim_labels, dim_values,2);
```

```
Error in
 prototypes_stat_secondLevel>prototypes_stat_secondLevel_oneSampleT
 (line 54)
ds              = prototypes_simMap2cosmo(groupData.(opt.dataType));

Error in prototypes_stat_secondLevel (line 28)
        groupStat = prototypes_stat_secondLevel_oneSampleT(groupData,
 opt);

Error in tutorial01_CosineMaps (line 92)
groupStat           = prototypes_stat_secondLevel(SubjectsCosineMaps,
 opt);
```

model fit: find prototypes and w

```
% initial parameters for fitting the data
param0              = [];

% how much participants used the prototypes?
% - 0 indicates they used maximally the prototypes (max bias)
% - 1 indicates they did not use the prototypes
param0.w            = 0.5;

% we assume the prototypes are at the centre of the 4 subquadrants
ShapeDim            =
 prototypes_get_metadata(SubjectsData, 'ShapeDim');
param0.prototypes   = [0.2 0.25; 0.3 0.75; 0.75 0.25; 0.75
 0.75].*ShapeDim;


opt                 = [];
% uncomment this if you want to visualize the fitting procedure
% opt.figure          = 100;
opt.DisplayParam     = 1;

subjlist = unique(SubjectsData.ParticipantID);
param_best = [];
for s = 1:nsubj

    % select a participant
    aSubject           = SubjectsData(SubjectsData.ParticipantID ==
 subjlist(s),:);

    % fit data for this participant
    param_best{s}      = prototypes_fit_model(aSubject,
 @prototypes_model_CAM, param0, opt);
end
```

model fit: visualize parameters

```
param_w             = zeros(nsubj, 1);
param_prototypes    = zeros(4, 2, nsubj);
```

```
param_R2_adj        = zeros(nsubj, 1);
for s = 1:nsubj
    param_w(s)              = param_best{s}.w;
    param_prototypes(:, :, s)  = param_best{s}.prototypes;
    param_R2_adj(s)         = param_best{s}.R2_adj;
end

%
 ========================================================================
% plot the weights
%
 ========================================================================
figure; plot(1,
 param_w, 'Marker', 'o', 'MarkerFaceColor', 'r', 'MarkerEdgeColor', 'k');
ax = gca; ax.YTick=[0.7 0.75 0.8];ax.XTick=[];
hold on;plot([0 2], [mean(param_w), mean(param_w)], 'k--');
title('memory weight');

%
 ========================================================================
% plot the prototypes
%
 ========================================================================
figure;
for s = 1:nsubj
    hold on; scatter(param_prototypes(:, 1,s) , param_prototypes(:,
 2,s), 'Marker', 'o', 'MarkerFaceColor', 'r', 'MarkerEdgeColor', 'k');
end

param_prototypes_avg = mean(param_prototypes, 3);
hold on;scatter(param_prototypes_avg(:, 1) , param_prototypes_avg(:,
 2), 'Marker', 'o', 'MarkerFaceColor', 'g', 'MarkerEdgeColor', 'k');
title('prototypes locations');
ax = gca; ax.YLim = [0 ShapeDim(2)];ax.XLim=[0
 ShapeDim(1)];axis image;

rect = prototypes_get_metadata(SubjectsData, 'ShapeRect');
rectangle('Position', rect);

figure; prototypes_plot_cosineMap(GroupCosineMaps);
hold on;scatter(param_prototypes_avg(:, 1) , param_prototypes_avg(:,
 2), 'Marker', 'o', 'MarkerFaceColor', 'g', 'MarkerEdgeColor', 'k');
title('prototypes locations');
```

Published with MATLAB® R2017a