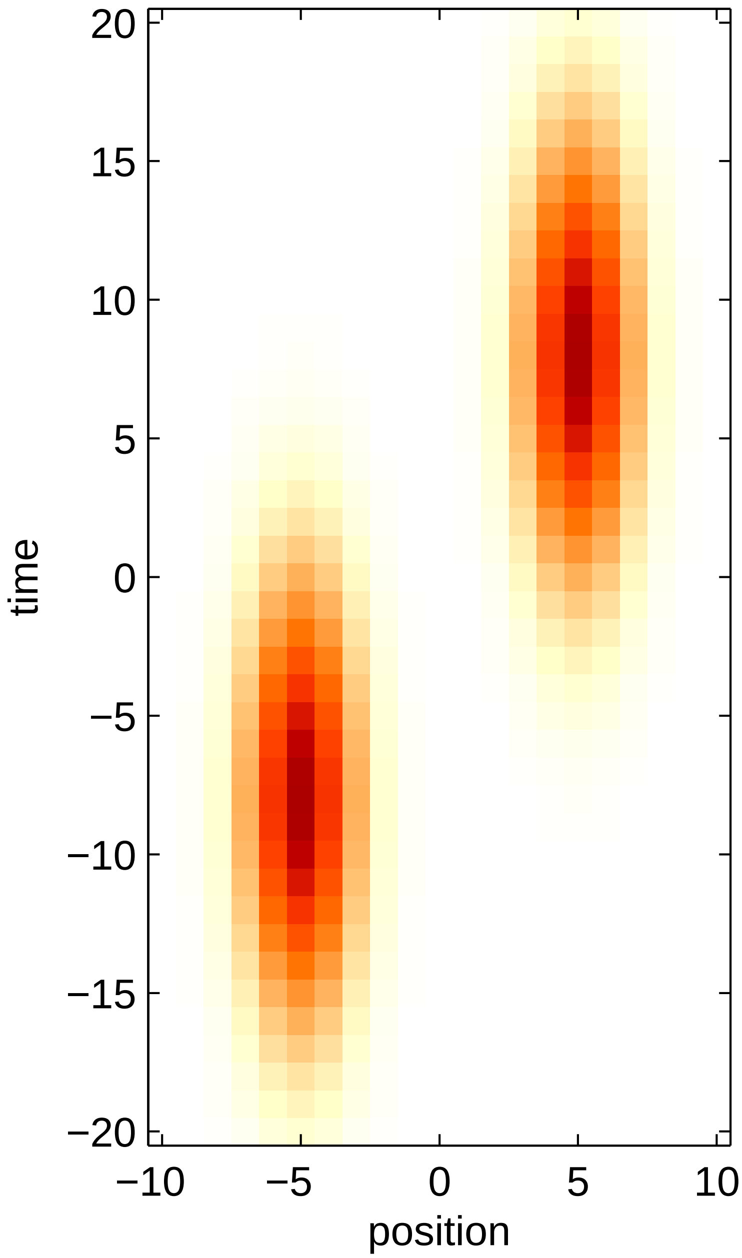
## Contents



## Receptive field example

Shows an example of the kind of receptive fields that we're trying to estimate. Here, we're looking at peri-saccadic receptive fields. A stimulus is flashed right before a saccade. If it's in the LFP's receptive field, there's a corresponding change in the LFP amplitude after the visual lag. However, and very interestingly, if the stimulus is in the spot where the receptive field will be **after** the saccade, there's a later activation: this is the so-called future field. It's not a purely visual signal; it's mostly an attentional thing, as it goes away as soon as you show more than a couple of targets. Our objective: tease out the two (or more) components on every electrode, and characterize them.

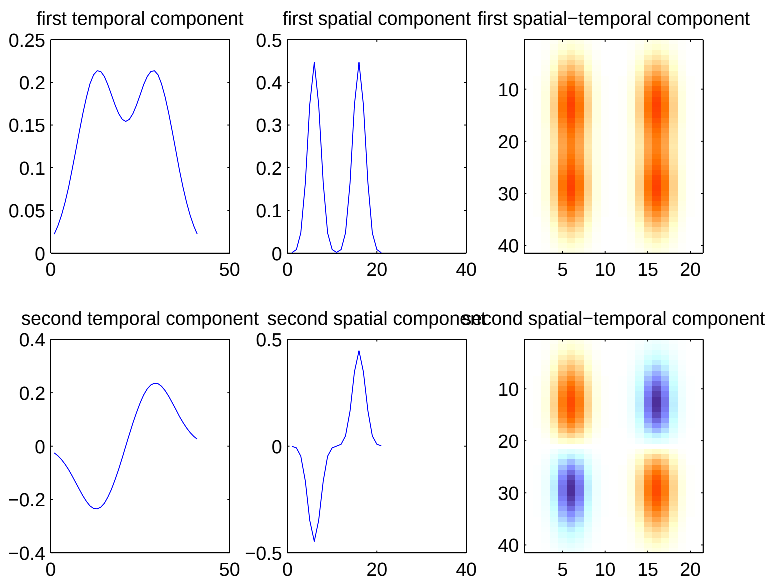
```
clf
% Create a "remapped receptive field"
[xi,yi] = meshgrid(-10:10,-20:20);

% early visual component
c1 = exp((-((xi+5).^2/2+(yi+8).^2/32)/2));
% late remapped attentional component
c2 = exp((-((xi-5).^2/2+(yi-8).^2/32)/2));

rf = c1+c2;
imagesc(xi(:),yi(:),rf,[-1,1]);
xlabel('position');
ylabel('time');
axis xy
axis equal;
axis tight
colormap(margecolor);
```




### SVD

Try and isolate with the SVD Look at the SVD: It's finding a mean and a difference component; we'd like to find two localized components

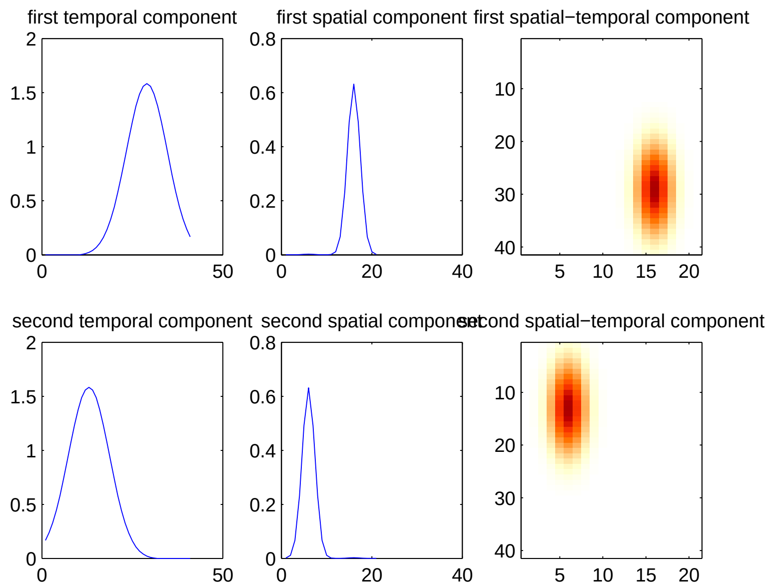
```
clf
[U,S,V] = svd(rf);

subplot(2,3,1);plot(U(:,1)*sign(mean(U(:,1))));title('first temporal component');
subplot(2,3,2);plot(V(:,1)*sign(mean(U(:,1))));title('first spatial component');
subplot(2,3,3);imagesc(S(1,1)*U(:,1)*V(:,1)',.7*[-1,1]);title('first spatial-temporal compor
subplot(2,3,4);plot(U(:,2)*sign(mean(U(:,2))));title('second temporal component');
subplot(2,3,5);plot(V(:,2)*sign(mean(U(:,2))));title('second spatial component');
subplot(2,3,6);imagesc(S(2,2)*U(:,2)*V(:,2)',.7*[-1,1]);title('second spatial-temporal compo
```




## NMF

The total RF is low-rank, U*S*V', but U>0 and V>0. Therefore use non-negative matrix factorization

```
[U,V] = mynnmf(rf,2);
V = V';

subplot(2,3,1);plot(U(:,1)*sign(mean(U(:,1))));title('first temporal component');
subplot(2,3,2);plot(V(:,1)*sign(mean(U(:,1))));title('first spatial component');
subplot(2,3,3);imagesc(U(:,1)*V(:,1)',[-1,1]);title('first spatial-temporal component');
subplot(2,3,4);plot(U(:,2)*sign(mean(U(:,2))));title('second temporal component');
subplot(2,3,5);plot(V(:,2)*sign(mean(U(:,2))));title('second spatial component');
subplot(2,3,6);imagesc(U(:,2)*V(:,2)',[-1,1]);title('second spatial-temporal component');
```

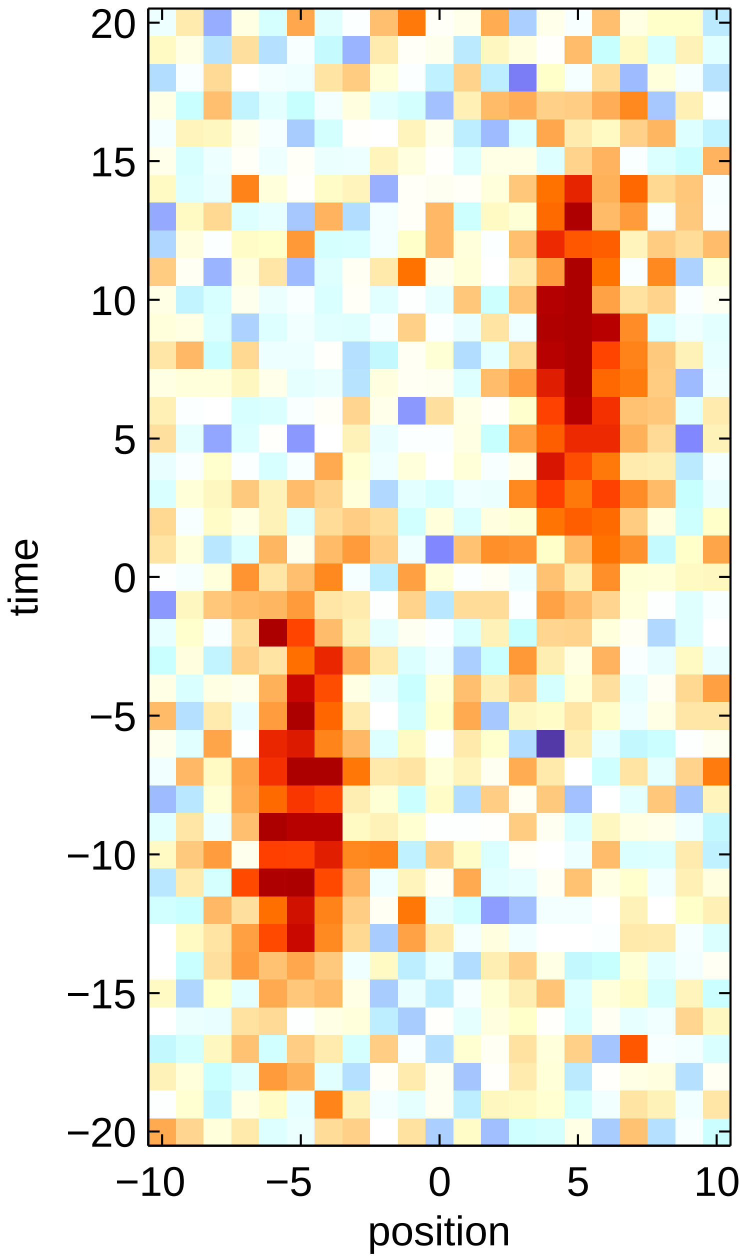



Bingo!

## With noise

Try again, this time with noise

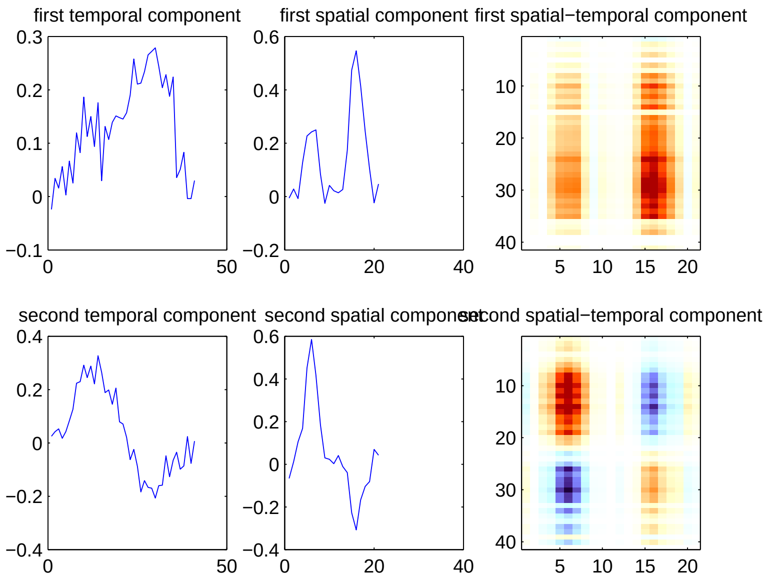
```
rf = c1+c2+randn(size(c1))*.2;

clf
imagesc(xi(:),yi(:),rf,[-1,1]);
xlabel('position');
ylabel('time');
axis xy
axis equal
axis tight
```




**Noise + SVD**

Look at the SVD:

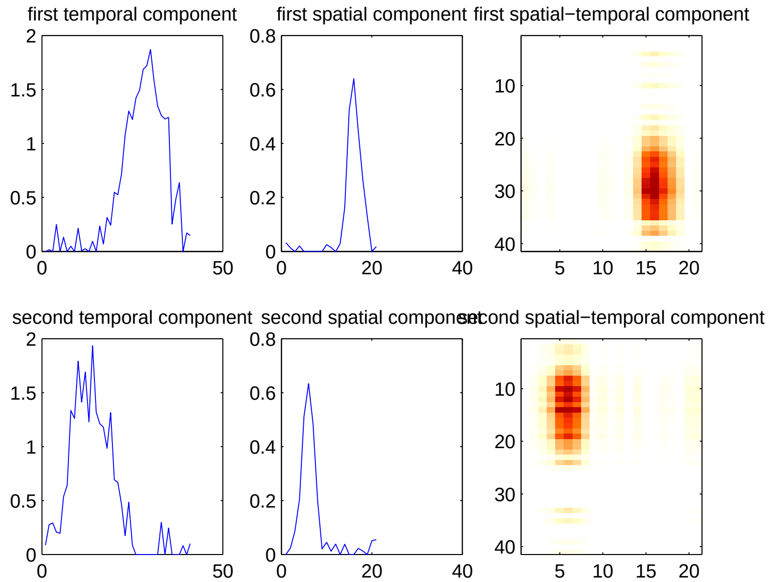
```
[U,S,V] = svd(rf);

subplot(2,3,1);plot(U(:,1)*sign(mean(U(:,1))));title('first temporal component');
subplot(2,3,2);plot(V(:,1)*sign(mean(U(:,1))));title('first spatial component');
subplot(2,3,3);imagesc(S(1,1)*U(:,1)*V(:,1)',.7*[-1,1]);title('first spatial-temporal compon
subplot(2,3,4);plot(U(:,2)*sign(mean(U(:,2))));title('second temporal component');
subplot(2,3,5);plot(V(:,2)*sign(mean(U(:,2))));title('second spatial component');
subplot(2,3,6);imagesc(S(2,2)*U(:,2)*V(:,2)',.7*[-1,1]);title('second spatial-temporal compo
```




This is not the decomposition we're looking for.

## Noise + NMF

```
[U,V] = mynnmf(rf,2,'replicates',5);
V = V';

subplot(2,3,1);plot(U(:,1)*sign(mean(U(:,1))));title('first temporal component');
subplot(2,3,2);plot(V(:,1)*sign(mean(U(:,1))));title('first spatial component');
subplot(2,3,3);imagesc(U(:,1)*V(:,1)',[-1,1]);title('first spatial-temporal component');
subplot(2,3,4);plot(U(:,2)*sign(mean(U(:,2))));title('second temporal component');
subplot(2,3,5);plot(V(:,2)*sign(mean(U(:,2))));title('second spatial component');
subplot(2,3,6);imagesc(U(:,2)*V(:,2)',[-1,1]);title('second spatial-temporal component');
```




Not half bad, but suffers from lack of regularization on spatiotemporal components

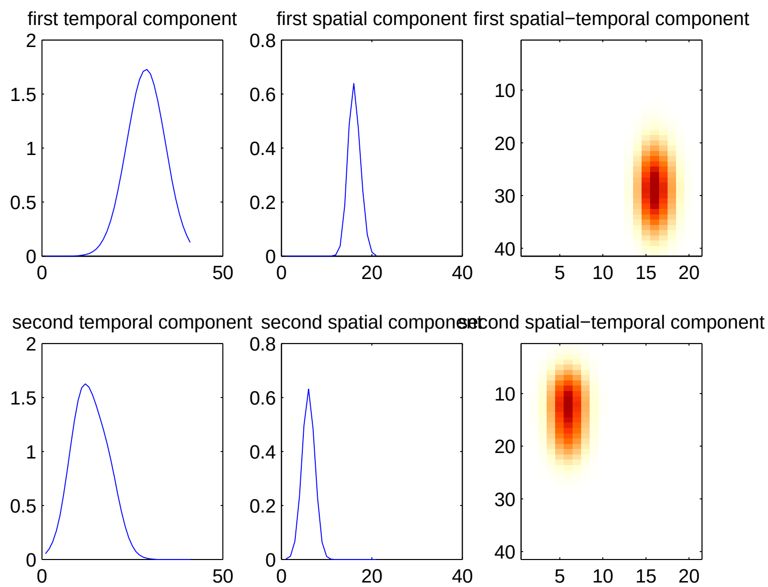
## ALD proof-of-concept

Instead of saying that rf = U*S*V', U>0, V>0, say that U(:,1),U(:,2),V(:,1),V(:,2) are localized. Right now, does this via algo:

1. res <- rf - U(:,1)*V(:,1)'
2. design_mat <- blkdiag(U(:,2),U(:,2),U(:,2),...)
3. V(:,2) <- ald(res(:),design_mat)
4. design_mat <- blkdiag(V(:,2),V(:,2),V(:,2),...)
5. U(:,2) <- ald(res'(:),design_mat)
6. lather, rinse, repeat for other component until convergence

```
niter = 3;
ncomps = size(V,2);

% Start with U,V from NMF
sserror = zeros(niter,1);

for ii = 1:niter
    for kk = 1:ncomps
        res = rf - U*V' + U(:,kk)*V(:,kk)';
        Xmat = [];
```

```
        for nn = 1:size(V,1)
            Xmat = blkdiag(Xmat,U(:,kk));
        end
        %reestimate spatial filters
        y = res(:);
        results = runALD(Xmat,y,size(V,1),1);
        V(:,kk) = results.khatSF;

        Xmat = [];
        for nn = 1:size(U,1)
            Xmat = blkdiag(Xmat,V(:,kk));
        end
        %reestimate spatial filters

        y = res';
        y = y(:);
        results = runALD(Xmat,y,size(U,1),1);
        U(:,kk) = results.khatSF;
    end
    yp = U*V';
    sserror(ii) = .5*sum((rf(:)-yp(:)).^2);
end


subplot(2,3,1);plot(U(:,1)*sign(mean(U(:,1))));title('first temporal component');
subplot(2,3,2);plot(V(:,1)*sign(mean(U(:,1))));title('first spatial component');
subplot(2,3,3);imagesc(U(:,1)*V(:,1)',[-1,1]);title('first spatial-temporal component');
subplot(2,3,4);plot(U(:,2)*sign(mean(U(:,2))));title('second temporal component');
subplot(2,3,5);plot(V(:,2)*sign(mean(U(:,2))));title('second spatial component');
subplot(2,3,6);imagesc(U(:,2)*V(:,2)',[-1,1]);title('second spatial-temporal component');
```


first temporal component
first spatial component
first spatial−temporal component
second temporal component
second spatial component
second spatial−temporal component
0
0.5
1
1.5
2
50
0.2
0.4
0.6
0.8
20
40
5
10
15
20
30

Bingo!