ISETbio as a tool for studying information transmission in the early visual system

Nicolas Cottaris

June 11, 2016

Hyperspectral images (@scene)

Retinal images (@optics)

Inline code like this `sceneGet(scene, 'a')` works ok.

```
decoderParams = struct(...
 'type', 'optimalLinearFilter', ...
 'thresholdConeSeparationForInclusionInDecoder', 0, ...
 'spatialSamplingInRetinalMicrons', 3.0, ...
 'extraMicronsAroundSensorBorder', 0, ...
 'temporalSamplingInMilliseconds', 10, ...
 'latencyInMillseconds', -150, ...
 'memoryInMilliseconds', 600 ...
    );

sensorParams = struct(...
  'coneApertureInMicrons', 3.0, ...
  'LMSdensities', [0.6 0.3 0.1], ...
  'spatialGrid', [18 26], ...
  'samplingIntervalInMilliseconds', sensorTimeStepInMilliseconds, ...
  'integrationTimeInMilliseconds', integrationTimeInMilliseconds, ...
  'randomSeed',  1552784, ...
    'eyeMovementScanningParams', struct(...
    'samplingIntervalInMilliseconds', sensorTimeStepInMilliseconds, ...
    'meanFixationDurationInMilliseconds', 200, ...
    'stDevFixationDurationInMilliseconds', 20, ...
    'meanFixationDurationInMillisecondsForAdaptingField', 400, ...
    'stDevFixationDurationInMillisecondsForAdaptingField', 20, ...
    'fixationOverlapFactor', 0.6, ...
    'saccadicScanMode',  'randomized'...
   ) ...
);
```

Eye movements (@sensor)

Photoisomerizations (@sensor)

Photocurrents (@outersegment)

The linear decoding filter

Seek filter, $\vec{w}_{xy}$, whose inner product with the vector of outer segment responses, r_t^i, assembled from:

- cones, $[1 \dots n]$
- at time delays, $t = lat + k + [0 \dots m-1]$

tracks the input stimulus at position $(x, y)^*$ and time delay k:

$$\left[\begin{array}{c|c|c|c|c} 1 & \underbrace{r^1_{t(1)} r^1_{t(2)} \dots r^1_{t(m)}}_{cone\ \#1\ response} & \underbrace{r^2_{t(1)} r^2_{t(2)} \dots r^2_{t(m)}}_{cone\ \#2\ response} & \cdots & \underbrace{r^n_{t(1)} r^n_{t(2)} \dots r^n_{t(m)}}_{cone\ \#n\ response} \end{array}\right] \cdot \underbrace{\left[\begin{array}{c} w_{o,xy} \\ \hline w^1_{1,xy} \\ \vdots \\ w^1_{m,xy} \\ \hline w^2_{1,xy} \\ \vdots \\ w^2_{m,xy} \\ \hline \vdots \\ w^n_{1,xy} \\ \vdots \\ w^n_{m,xy} \end{array}\right]}_{\vec{w}_{xy}} \cong c_{xy}(k)$$

- lat: filter latency,
- m: filter memory,
- $(x, y)^*$: scene position projected on the retina

The linear decoding filter

The filter must track the stimulus across all time points, $t = [1 \dots T]$:

$$\underbrace{\left[\begin{array}{c|cccc|c|cccc} 1 & r^1_{lat+1} & r^1_{lat+2} & \dots & r^1_{lat+1+m-1} & \dots & r^n_{lat+1} & r^n_{lat+2} & \dots & r^n_{lat+1+m-1} \\ 1 & r^1_{lat+2} & r^1_{lat+3} & \dots & r^1_{lat+2+m-1} & \dots & r^n_{lat+2} & r^n_{lat+3} & \dots & r^n_{lat+2+m-1} \\ \vdots & \vdots & \vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \vdots \\ 1 & r^1_{lat+k} & r^1_{lat+k+1} & \dots & r^1_{lat+k+m-1} & \dots & r^n_{lat+k} & r^n_{lat+k+1} & \dots & r^n_{lat+k+m-1} \\ \vdots & \vdots & \vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \vdots \\ 1 & r^1_{lat+T} & r^1_{lat+T+1} & \dots & r^1_{lat+T+m-1} & \dots & r^n_{lat+T} & r^n_{lat+T+1} & \dots & r^n_{lat+T+m-1} \end{array}\right]}_{\mathbf{X} \in \mathcal{R}^{T \times (1+nm)}} \underbrace{\left[\begin{array}{c} w_{o,xy} \\ \hline w^1_{1,xy} \\ \vdots \\ w^1_{m,xy} \\ \hline \vdots \\ \hline w^n_{1,xy} \\ \vdots \\ w^n_{m,xy} \end{array}\right]}_{\vec{w}_{xy}} \cong \underbrace{\left[\begin{array}{c} c_{xy}(1) \\ c_{xy}(2) \\ \vdots \\ c_{xy}(k) \\ \vdots \\ c_{xy}(T) \end{array}\right]}_{\vec{c}_{xy}}$$

i.e.: $\mathbf{X} \cdot \vec{w}_{xy} \cong \vec{c}_{xy}$

Estimating the filter coefficients

We are seeking $\vec{w}_{xy}$ such that: $\mathbf{X} \cdot \vec{w}_{xy} = \widetilde{c}_{xy} \approx \vec{c}_{xy}$. In other words, we want to minimize the in-sample error

$$E_{in}(\vec{w}_{xy}) = \frac{1}{T} \left\| \mathbf{X} \cdot \vec{w}_{xy} - \vec{c}_{xy} \right\|^2$$

To minimize E_{in}, one has to solve for w that satisfies:

$$(\mathbf{X}^{\mathrm{T}} \mathbf{X}) \cdot \vec{w}_{xy} = \mathbf{X}^{\mathrm{T}} \vec{c}_{xy}$$

which leads to:

optimal decoding filter

$$\vec{w}_{xy} = (\mathbf{X}^{\mathrm{T}} \cdot \mathbf{X})^{-1} \cdot \mathbf{X}^{\mathrm{T}} \cdot \vec{c}_{xy}$$

The matrix: $(\mathbf{X}^{\mathrm{T}} \cdot \mathbf{X})^{-1} \cdot \mathbf{X}^{\mathrm{T}}$ is called the pseudo inverse of $\mathbf{X}$, *pinv* $(\mathbf{X})$

Estimating the filter coefficients

$$pinv\,(\mathbf{X}) = (\mathbf{X}' \cdot \mathbf{X})^{-1} \cdot \mathbf{X}'$$

For $\mathbf{X}'\mathbf{X}$ to be invertible: T (# of rows) must be $\geq 1 + nm$ (#of cols). row rank ? col rank?

- the k-th row of the $\mathbf{X}$ corresponds to one observation of the multi-unit response, corresponding to the stimulus at time k.
- if the T observations are linearly independent on each other, then only $1 + n \times m$ of them would be required to estimate the $1 + n \times m$ filter coefficients.
- the more linearly dependent the observations are, the higher their number has to be, so as to capture the dimensionality of the filter.

NOTE

If $\mathbf{X}'\mathbf{X}$ is not invertible, $pinv\,(\mathbf{X})$ does not exist.

Estimating the filter coefficients

If $\mathbf{X}^\mathsf{T}\mathbf{X}$ is not invertible, $\vec{w}_{xy}$ cannot be computed as: $\vec{w}_{xy} = (\mathbf{X}^\mathsf{T}\mathbf{X})^{-1}\mathbf{X}^\mathsf{T}\vec{c}_{xy}$
Moreover, there are many $\vec{w}_{xy}$ that satisfy

$$\mathbf{X}^\mathsf{T}\mathbf{X} \cdot \vec{w}_{xy} = \mathbf{X}^\mathsf{T} \cdot \vec{c}_{xy}$$

and therefore minimize E_{in}. One solution,

minimum-norm decoding filter

$$\vec{w}_{xy} = \left(V \cdot S^{-1} \cdot U^\mathsf{T}\right) \cdot \vec{c}_{xy}$$

is special as it has the minimal $\left\|\vec{w}_{xy}\right\|$ of all the solutions. In the above equation, matrices V, U, and S come from the SVD of $\mathbf{X}$:

$$\mathbf{X} = U \cdot S \cdot V^\mathsf{T}$$

Estimating the filter coefficients

The minimum norm solution is not the only solution. It is the solution with the minimum L2 norm, which is results from having the maximum number of non-zero coefficients, w. Does this lead to the larger RFs? Check.
As an alternative, we can select the number of SVD components, i.e.,

$$\tilde{\mathbf{X}} = U \cdot \tilde{S} \cdot V^{\mathrm{T}}$$

where

$$\tilde{S} = [\sigma_1 \; \sigma_2 \; \ldots \; \sigma_k \; 0 \; 0 \; \ldots 0 \,]$$

Reconstructing the input stimulus

Note that the decoding filter is given by:

$$\begin{aligned}
\vec{w}_{xy} &= pinv\left(\mathbf{X}\right) \cdot \vec{c}_{xy} \\
&= \left((\mathbf{X}' \cdot \mathbf{X})^{-1} \cdot \mathbf{X}' \right) \cdot \vec{c}_{xy} \\
&= \underbrace{(\mathbf{X}' \cdot \mathbf{X})^{-1}}_{\text{corr. b/n responses}} \cdot \underbrace{\mathbf{X}' \cdot \vec{c}_{xy}}_{\text{corr. b/n resp. \& stim}}
\end{aligned}$$

Reconstructing multiple components of the input stimulus

We can construct multiple decoding filters, for reconstructing different components of the input, such as L-, M-, S-cone contrast, at a set of spatial positions (x_i, y_j):

$$\vec{w}^{L}_{x_i,y_j} = \underbrace{(\mathbf{X}' \cdot \mathbf{X})^{-1}}_{\text{corr. b/n responses}} \cdot \underbrace{\mathbf{X}' \cdot \vec{c}^{L}_{x_i,y_j}}_{\text{corr. b/n resp. \& L-cone contrast at } (x_i,\, y_j)}$$

$$\vec{w}^{M}_{x_i,y_j} = \underbrace{(\mathbf{X}' \cdot \mathbf{X})^{-1}}_{\text{corr. b/n responses}} \cdot \underbrace{\mathbf{X}' \cdot \vec{c}^{M}_{x_i,y_j}}_{\text{corr. b/n resp. \& M-cone contrast at } (x_i,\, y_j)}$$

$$\vec{w}^{S}_{x_i,y_j} = \underbrace{(\mathbf{X}' \cdot \mathbf{X})^{-1}}_{\text{corr. b/n responses}} \cdot \underbrace{\mathbf{X}' \cdot \vec{c}^{S}_{x_i,y_j}}_{\text{corr. b/n resp. \& S-cone contrast at } (x_i,\, y_j)}$$

Reconstructing multiple components of the input stimulus

Expand the filter so as to track the scene's L-, M-, and S-cone contrasts at different positions (x_i, y_j):

$$\underbrace{\begin{bmatrix} 1 \; r^1_{lat+1} & \cdots & r^n_{lat+m} \\ \vdots \;\; \vdots & \ddots & \vdots \\ 1 \; r^1_{lat+k} & \cdots & r^n_{lat+k+m-1} \\ \vdots \;\; \vdots & \ddots & \vdots \\ 1 \; r^1_{lat+T} & \cdots & r^n_{lat+T+m-1} \end{bmatrix}}_{\mathbf{X}} \begin{bmatrix} & | & | & | & \\ \cdots & \vec{w}^L_{x_i,y_j} & \vec{w}^M_{x_i,y_j} & \vec{w}^S_{x_i,y_j} & \cdots \\ & | & | & | & \end{bmatrix} = \begin{bmatrix} \cdots & c^L_{x_i,y_j}(1) & c^M_{x_i,y_j}(1) & c^S_{x_i,y_j}(1) & \cdots \\ \cdots & \vdots & \vdots & \vdots & \cdots \\ \cdots & c^L_{x_i,y_j}(k) & c^M_{x_i,y_j}(k) & c^S_{x_i,y_j}(k) & \cdots \\ \cdots & \vdots & \vdots & \vdots & \cdots \\ \cdots & c^L_{x_i,y_j}(T) & c^M_{x_i,y_j}(T) & c^S_{x_i,y_j}(T) & \cdots \end{bmatrix}$$

Our simulations

For a filter with memory m = 200 bins and n = 400 cones, we would need at least 80,000 multi-unit observations, so it would beed at least 23 GB of RAM for its storage (at single float precision). In our simulations, we used m = 40 bins (each bin being 5 msec wide) for a decoding filter memory of 200 milliseconds.

Multiple ilinear decoders for L-, M-, S-cone contrast decoding

Title Block 1

- item 1
- item 2

Title Block 2

Text.

Using the above formulation, we can estimate multiple independent filters that will track the scene L-, M-, and S-cone contrast at different retinal positions (x_i, y_j):

$$
\begin{aligned}
\mathbf{X} &* \vec{w}_{x_i,y_j,L} \cong \vec{c}_{x_i,y_j,L}(t) \\
\mathbf{X} &* \vec{w}_{x_i,y_j,M} \cong \vec{c}_{x_i,y_j,M}(t) \\
\mathbf{X} &* \vec{w}_{x_i,y_j,S} \cong \vec{c}_{x_i,y_j,S}(t)
\end{aligned}
$$

Something else

- Memory.

Something else

- Memory.
- Fixation.

Something else

- Memory.
- Fixation.
- Cones.