Mathematical equations underlying numerosity population receptive field models

Joram Soch

soch@cbs.mpg.de

Abstract

We review the mathematical formulation of the statistical model used in "Topographic Representation of Numerosity in the Human Parietal Cortex" (Harvey et al., *Science*, 2013; vol. 341, pp. 1123-1126; DOI: 10.1126/science.1239052) and provide actual equations for the entire model, with notation that is easy to follow and common in cognitive neuroscience. Since this study does not report any equations, we occassionally reference "Population receptive field estimates in human visual cortex" (Dumoulin & Wandell, *NeuroImage*, 2008; vol. 39, pp. 647-660; DOI: 10.1016/j.neuroimage.2007.09.034) which describes the modelling framework on which the analysis was based. In addition, we show how to simulate data from the generative model and how to extend the modelling approach with (i) inclusion of nuisance variables, (ii) accounting for hemodynamic variability and (iii) estimation via (restricted) maximum likelihood.

Contents

1 Introduction

Harvey and colleagues presented a computational model for numerosity processing in human cerebral cortex (Harvey et al., 2013), measured with functional magnetic resonance imaging (fMRI) and analyzed using a population receptive field (pRF) model for visual perception by Dumoulin and Wandell (Dumoulin & Wandell, 2008). Briefly, measured fMRI signals were assumed to originate from neuronal activity following presented numerosity that follows log-Gaussian-shaped tuning functions, convolved with some hemodynamic response to yield a hemodynamic signal. This approach was later re-applied and extended by Harvey and colleagues (e.g. Harvey et al., 2011; Harvey et al., 2015; Harvey & Dumoulin, 2017; Paul et al., 2022).
Here, we review the statistical model formulation behind numerosity pRF models to facilitate simulation of fMRI signals compatible with specific tuning parameters in the context of our own research project on the EMergence of PRecISE numerosity representations in the human brain (EMPRISE).

2 Model specification

2.1 Notation

An fMRI numerosity processing experiment has been performed. Let $x_t \in \mathcal{X}$ be the presented numerosity at continuous time t. Here, $\mathcal{X} = \{1, 2, \ldots, L, M\}$ is the stimulus space where L is the number of low numerosities levels (Harvey et al., 2013: $L = 7$; EMPRISE: $L = 5$) and M is the high numerosity presented in intermissions (Harvey & EMPRISE: $M = 20$; Harvey & Dumoulin, 2017: $L \in \{10, 20\}$?[1]).
Furthermore, let y_{ij} be the measured BOLD signal in a single voxel during the i-th scan (EMPRISE: $i = 1, \ldots, 145$) belonging the j-run (EMPRISE: $j = 1, \ldots, 8$). We denote as $\bar{y}$ the signal which results from averaging across all runs and we denote as $\bar{y}_j$ the signal which results from averaing across repeated "epochs" of identical stimulus presentation within a run (Harvey & EMPRISE: 4 epochs) and we denote as $\bar{\bar{y}}$ the signal which results from averaging across both, runs and epochs.

2.2 Neuronal model

We define a logarithmic Gaussian tuning function on stimulus x as

$$f_{\log}(x) = \exp\left[-\frac{1}{2}\frac{(\ln x - \mu_{\log})^2}{\sigma_{\log}^2}\right] \tag{1}$$

[1]Harvey & Dumoulin, 2017, p. 7: "The numerosities one to seven were first presented in ascending order, followed by 16.8 seconds showing 10 circles (sic!), followed (sic!) seven to one in descending order, followed by 16.8 seconds with 20 circles." However, this is incompatible with their Fig. 1 which shows a single high numerosity of 20.

and a linear Gaussian tuning function (cf. Dumoulin & Wandell, 2008, eq. 2[2]) as

$$f_{\text{lin}}(x) = \exp\left[-\frac{1}{2}\frac{(x-\mu_{\text{lin}})^2}{\sigma_{\text{lin}}^2}\right] \tag{2}$$

where $\exp(\mu_{\log})$ or μ_{lin} corresponds to the preferred numerosity of the tuned population (see Appendix A) and $\sigma_{\log}$ or σ_{lin} corresponds to the tuning width of the tuned population in logarithmic or linear stimulus space, respectively.
Instead of using the standard deviation σ, tuning functions are preferably parametrized in terms of the full width at half maximum (FWHM) which can be directly calculated from the standard deviation (see Appendix A):

$$\omega = \text{FWHM} = \exp\left[\mu_{\log} + \sqrt{2\ln 2}\,\sigma_{\log}\right] - \exp\left[\mu_{\log} - \sqrt{2\ln 2}\,\sigma_{\log}\right] \ . \tag{3}$$

Then, neuronal activity at time t is assumed to be (cf. Dumoulin & Wandell, 2008, eq. 3)

$$\begin{aligned} z_t &= \sum_{x\in\mathcal{X}} [x = x_t] \cdot f_{\log}(x) \\ &= f_{\log}(x_t) \end{aligned} \tag{4}$$

where the Iverson bracket notation $[x = x_t]$ indicates presence of stimulus x_t at time t.

2.3 Hemodynamic model

We define $h(t)$ as the hemodynamic response function (HRF) at post-stimulus time t, e.g. obtained from SPM using HRF parameters θ:

$$h(t) \leftarrow \texttt{spm\_hrf}(\theta) \ . \tag{5}$$

Then, the hemodynamic signal at time t is assumed to be a convolution of the neural activity with the HRF (cf. Dumoulin & Wandell, 2008, eq. 4):

$$\begin{aligned} s_t &= z_t * h(t) \\ &= \int_0^t z_\tau \cdot h(t-\tau)\,\mathrm{d}\tau \ . \end{aligned} \tag{6}$$

Finally, the measured hemodynamic signal is assumed to be a linear combination of the underlying hemodynamic signal and zero-mean normally distributed measurement noise (cf. Dumoulin & Wandell, 2008, eq. 1[3])

$$y = \beta_s\, s + \varepsilon,\ \varepsilon_i \sim \mathcal{N}(0, \sigma^2),\ i = 1, \ldots, n \tag{7}$$

where y is an $n \times 1$ vector representing the measured BOLD signal (n = number of scans), s is an $n \times 1$ vector of the hemodynamic signal s_t, sampled at the acquisition times of y, β_s is a scaling factor and σ is the noise variance.

[2]There is a typesetting error in this equation: The minus sign needs to be inside the parantheses.
[3]To be precise, Dumoulin and Wandell make no assumptions about the noise distribution.

3 Model estimation

3.1 Estimation of tuning parameters

Let μ and ω be candidate tuning parameters (preferred numerosity and tuning width, i.e. FWHM) describing the potential neuronal response of a single voxel relative to presented stimuli x_t. We denote the resulting tuning function (see eq. 1) as $f_{\log}(x_t; \mu, \omega)$, the expected neuronal signal (see eq. 4) as $z_t(\mu, \omega)$ and the expected hemodynamic signal (cf. eq. 7) as $s(\mu, \omega)$.
Then, we can estimate the scaling factor by comparing the expected against the measured hemodynamic signal

$$\hat{\beta}_s(\mu, \omega) = \arg\min_{\beta_s} \sum_{i=1}^{n} \left[y_i - \beta_s s_i(\mu, \omega)\right]^2 \tag{8}$$

and calculate the residual sum of squares (RSS) as a function of the estimated scaling factor (cf. Dumoulin & Wandell, 2008, eq. 5):

$$\mathrm{RSS}(\mu, \omega) = \sum_{i=1}^{n} \left[y_i - \hat{\beta}_s(\mu, \omega) s_i(\mu, \omega)\right]^2 \ . \tag{9}$$

The estimated preferred numerosity $\hat{\mu}$ and estimated tuning width $\hat{\omega}$ are then obtained as those candidate tuning parameters that minimize the RSS:

$$(\hat{\mu}, \hat{\omega}) = \arg\min_{(\mu,\omega)} \mathrm{RSS}(\mu, \omega) \ . \tag{10}$$

3.2 Estimation of hemodynamic parameters

In order to account for different hemodynamic response functions between subjects, but not between regions or voxels, Harvey and colleagues report (Harvey et al., 2013, suppl. p. 9; Harvey & Dumoulin, 2017, p. 7) to have estimated subject-wise HRF parameters after an initial estimation of tuning parameters (which was based on default HRF parameters) and then re-estimating tuning parameters (using the updated HRF parameters). This estimation scheme can be summarized as follows[4]:

$$\begin{aligned} h_0 &\leftarrow \texttt{spm\_hrf}(\theta_0) \\ (\hat{\mu}_0, \hat{\omega}_0) &= \arg\min_{(\mu,\omega)} \mathrm{RSS}(\mu, \omega; h_0) \\ \hat{\theta} &\leftarrow \{y, x, \hat{\mu}_0, \hat{\omega}_0\} \\ h &\leftarrow \texttt{spm\_hrf}(\hat{\theta}) \\ (\hat{\mu}, \hat{\omega}) &= \arg\min_{(\mu,\omega)} \mathrm{RSS}(\mu, \omega; h) \end{aligned} \tag{11}$$

where θ_0 are the default HRF parameters and subscript 0 signifies initial estimates.

[4]It is obvious that the third line of these equations is dubious and needs clarification. For an implementation, see here: https://github.com/SkeideLab/EMPRISE/blob/akieslinger/data/code/harveymodel.py#L549 (function "estimate_hrf").

3.3 Inclusion of nuisance regressors

It is very unlikely that equation (7) is a realistic model of preprocessed, but otherwise unaltered fMRI signal. In particular, this model does not account for a baseline level of hemodynamic activity and does not allow for experimental conditions of no interest and nuisance variables, possibly influencing the preprocessed signal. We assume that these factors can be summarized as a design matrix

$$X(\mu,\omega) = \begin{bmatrix} s(\mu,\omega) & X_c & 1_n \end{bmatrix} \tag{12}$$

where $s(\mu,\omega)$ is identical to the s in equation (7), X_c is an $n \times c$ matrix of confounds and 1_n is an $n \times 1$ vector of ones, such that the model from (7) changes to

$$y = X\beta + \varepsilon,\ \varepsilon \sim \mathcal{N}(0, \sigma^2 I_n)\ . \tag{13}$$

Then, regression coefficients can be estimated using ordinary least squares (OLS)

$$\hat{\beta}(\mu,\omega) = (X^{\mathrm{T}}X)^{-1}X^{\mathrm{T}}y \tag{14}$$

and estimated tuning parameters would be given as minimizing the RSS:

$$(\hat{\mu},\hat{\omega}) = \underset{(\mu,\omega)}{\arg\min} \left[y - X\hat{\beta}(\mu,\omega)\right]^{\mathrm{T}} \left[y - X\hat{\beta}(\mu,\omega)\right]\ . \tag{15}$$

3.4 Inclusion of hemodynamic derivatives

Another way to account for variability in the hemodynamic response, also between voxels, is to include HRF derivatives into the model. Let $h'(t)$ be the time derivative and let $h''(t)$ be the dispersion deriative of the canonical HRF[5,6]:

$$\{h(t),\ h'(t),\ h''(t)\} \leftarrow \texttt{spm\_get\_bf}(\theta)\ . \tag{16}$$

Then, we obtain three hemodynamic signals (cf. eq. 6)

$$\begin{aligned} s_t &= z_t * h(t) \\ s'_t &= z_t * h'(t) \\ s''_t &= z_t * h''(t) \end{aligned} \tag{17}$$

which can be combined in the predictive linear model (cf. eq. 7)

$$y = S\beta_S + \varepsilon,\ \varepsilon_i \sim \mathcal{N}(0, \sigma^2),\ i = 1, \ldots, n \tag{18}$$

[5]Note: Whereas the time derivative does in fact correspond to the first derivative, the dispersion derivative is not equal to the second deriative of the canonical HRF, but corresponds to the derivative with respect to the dispersion parameter of the canonical HRF; see: https://github.com/spm/spm12/blob/master/spm_get_bf.m#L144 (section "add dispersion deriative").

[6]For an implementation, see here: https://github.com/SkeideLab/EMPRISE/blob/JoramSoch/code/Python/PySPMs.py#L186 (function "spm_get_bf").

or, when including nuisance regressors, the design matrix is (cf. eq. 12)

$$X(\mu,\omega) = \begin{bmatrix} S(\mu,\omega) & X_c & 1_n \end{bmatrix} \quad \text{where} \quad S(\mu,\omega) = \begin{bmatrix} s & s' & s'' \end{bmatrix} \tag{19}$$

while s' and s'' are temporally downsampled $n \times 1$ vector representations of s'_t and s''_t. Due to their nature of being deriatives with respect to time and dispersion, s' and s'' allow to capture voxel-by-voxel modulations in hemodynamic response time and resonse dispersion – which are the most prominent forms of hemodynamic variability observed in empirical data (Zeidman et al., 2018, Fig. 3).

3.5 Maximum likelihood estimation

Given that stimulus signals for fixed μ and ω, nuisance regressors and implicit baseline are available in the form of a design matrix X (cf. eq. 19), tuning parameters – and possibly, hemodynamic parameters – can also be obtained via maximum likelihood estimation (MLE). Equation (13) implies the following log-likelihood function:

$$\begin{aligned} \mathrm{LL}(X,\beta,\sigma^2) &= \log p(y|X,\beta,\sigma^2) = \log \mathcal{N}(y; X\beta, \sigma^2 I_n) \\ &= \log\left(\sqrt{\frac{1}{(2\pi)^n |\sigma^2 I_n|}} \cdot \exp\left[-\frac{1}{2}(y-X\beta)^{\mathrm{T}}(\sigma^2 I_n)^{-1}(y-X\beta)\right]\right) \\ &= -\frac{n}{2}\log(2\pi) - \frac{n}{2}\log(\sigma^2) - \frac{1}{2\sigma^2}(y-X\beta)^{\mathrm{T}}(y-X\beta)\ . \end{aligned} \tag{20}$$

Maximizing this function with respect to β and σ^2 gives rise to the ML estimates

$$\begin{aligned} \hat{\beta} &= \arg\max_{\beta} \mathrm{LL}(X,\beta,\sigma^2) = (X^{\mathrm{T}}X)^{-1}X^{\mathrm{T}}y \\ \hat{\sigma}^2 &= \arg\max_{\sigma^2} \mathrm{LL}(X,\hat{\beta},\sigma^2) = \frac{1}{n}(y-X\hat{\beta})^{\mathrm{T}}(y-X\hat{\beta}) \end{aligned} \tag{21}$$

and the maximum log-likelihood for given tuning parameters μ and ω

$$\mathrm{MLL}(\mu,\omega) = \mathrm{LL}(X[\mu,\omega],\hat{\beta},\hat{\sigma}^2) = -\frac{n}{2}\log\left(\frac{\mathrm{RSS}\,[\mu,\omega]}{n}\right) - \frac{n}{2}\log(2\pi) - \frac{n}{2}\ , \tag{22}$$

such that ML estimates of the tuning parameters would be given by

$$(\hat{\mu},\hat{\omega}) = \arg\max_{(\mu,\omega)} \mathrm{MLL}(\mu,\omega)\ , \tag{23}$$

i.e. as those values of μ and ω which maximize the maximum log-likelihood. Since MLL is a monotonically decreasing function of RSS (see eq. 22), maximizing the log-likelihood is equivalent to minimizing the residual sum of squares (see Appendix B).

3.6 Restricted maximum likelihood

Another restriction of (7) consists in the fact that it assumes independent and identically distributed (i.i.d.) error terms ε_i and thus does not allow for error covariance which typically exists in the form of serial correlations in fMRI. We assume that such serial correlations are given in the form of a temporal covariance matrix V, e.g. obtained using SPM's restricted maximum likelihood (ReML) approach:

$$V \leftarrow \texttt{spm\_reml}(Y, X) \tag{24}$$

where Y is an $n \times v$ matrix of fMRI signals across all voxels inside a region of interest (ROI) or in the whole brain, such that the model from (7) changes to

$$y = X\beta + \varepsilon,\ \varepsilon \sim \mathcal{N}(0, \sigma^2 V)\ . \tag{25}$$

Then, regression coefficients can be estimated using weighted least squares (WLS)

$$\hat{\beta}(\mu, \omega) = (X^{\mathrm{T}} V^{-1} X)^{-1} X^{\mathrm{T}} V^{-1} y \tag{26}$$

and estimated tuning parameters would be given as minimizing the weighted RSS:

$$(\hat{\mu}, \hat{\omega}) = \underset{(\mu,\omega)}{\arg\min} \left[y - X\hat{\beta}(\mu, \omega)\right]^{\mathrm{T}} V^{-1} \left[y - X\hat{\beta}(\mu, \omega)\right]\ . \tag{27}$$

Maximum likelihood estimation under such correlated observations would proceed by maximizing the log-likelihood function $\mathrm{LL}(X, V, \beta, \sigma^2)$ – now also a function of the temporal non-sphericity matrix V – which gives rise to the same parameter estimates

$$\hat{\beta} = \underset{\beta}{\arg\max}\, \mathrm{LL}(X, V, \beta, \sigma^2) = (X^{\mathrm{T}} V^{-1} X)^{-1} X^{\mathrm{T}} V^{-1} y \tag{28}$$

and the maximum log-likelihood as a function of the tuning parameters

$$\mathrm{MLL}(\mu, \omega) = -\frac{n}{2} \log \left(\frac{\mathrm{wRSS}\left[\mu, \omega\right]}{n} \right) - \frac{n}{2} \log(2\pi) - \frac{n}{2} - \frac{1}{2} \log |V|\ . \tag{29}$$

Once again, since the MLL is uniquely related to the weighted RSS (see Appendix B), maximum likelihood estimation will give the same tuning parameter estimates as estimation that minimizes the weighted sum of squares – as long as both are accounting for the same correlation structure V.

3.7 Grid-based parameter search

Although maximum log-likelihood (see eqs. 22/29) and residual sum of squares (see eqs. 15/27) are both functions of the tuning parameters μ and ω, there is no simple functional form of the former in terms of the latter. This due to the transformations that happen between the neuronal signal z_t (see eq. 4) and the hemodynamic regressor s (see eq. 7), i.e. discrete convolution and temporal downsampling.
For this reason, we cannot directly calculate estimates of μ and ω from the measured data y. Instead, the strategy is to perform an exhaustive parameter search within a plausible grid of reasonable values at a sufficient precision:

$$\begin{aligned} \mu \in \mathrm{M} &= \{\mu_{\min},\, \mu_{\min} + \Delta\mu,\, \ldots,\, \mu_{\max} - \Delta\mu,\, \mu_{\max}\} \\ \omega \in \Omega &= \{\omega_{\min},\, \omega_{\min} + \Delta\omega,\, \ldots,\, \omega_{\max} - \Delta\omega,\, \omega_{\max}\} \ . \end{aligned} \tag{30}$$

For functional MRI data measured within EMPRISE, we assume a parameter grid characterized by preferred numerosities between 0 and 6 in steps of 0.05 (120 values) and FWHM tuning widths between 0 and 20 in steps of 0.25 (80 values):

$$\begin{aligned} \mu_{\min} = 0.05, \mu_{\max} = \ \ 6, \Delta\mu = 0.05 \\ \omega_{\min} = 0.25, \omega_{\max} = 20, \Delta\omega = 0.25 \ . \end{aligned} \tag{31}$$

The total parameter space then is the cartesian product of these two sets

$$\begin{aligned} (\mu, \omega) \in \Theta \ &= \mathrm{M} \times \Omega \\ |\Theta| &= \text{number of possible tuning parameters} \\ &\quad \text{(here: } 120 \times 80 = 9{,}600 \text{ values)} \end{aligned} \tag{32}$$

and all optimization (minimizing or maximizing) will always address this set.

4 Data simulation

4.1 Specification of simulation parameters

In order to simulate voxel-wise multi-run fMRI signals compatible with a numerosity experimental design such as EMPRISE, the following parameters need to be specified:

- preferred numerosity μ_k and FWHM tuning width ω_k for each voxel k where $k = 1, \ldots, v$ and v is the number of voxels;
- mean signal coefficients μ_s for signal regressors (numerosity signal and implicit baseline) and mean signal coefficients μ_c for confound regressors (cf. eq. 12);
- between-voxel variance σ_k^2;
- between-run variance σ_j^2;
- within-run variance σ_i^2;
- time constant τ.

Furthermore, we assume that the following quantities are available:

- a hemodynamic response function $h(t)$ (see eq. 5);
- a set of presented numerosities $x_{t,j}$ (see Section 2.1) at time t in each run j where $j = 1, \ldots, r$ and r is the number of runs;
- an $n \times c$ matrix of confound regressors $X_{c,j}$ for each run where n is the number of scans and c is the number of variables.

4.2 Sampling of simulated signals

Based on these choices, synthetic data are then generated as follows: First, presented numerosities x_t, preferred numerosity μ and FWHM tuning width ω from a single voxel are used to generate neuronal signals according to equation (4). These neuronal signals are then convolved with the hemodynamic response function $h(t)$ according to equation (6) and, with the confound regressors $X_{c,j}$, combined into the design matrix X_j for a single run according to equation (12):

$$X_j = \begin{bmatrix} s_j(\mu, \omega) & X_{c,j} & 1_n \end{bmatrix} \tag{33}$$

Second, voxel-wise regression coefficients are sampled as

$$\begin{aligned} \beta_s &\sim \mathcal{N}(\mu_s, \sigma_k^2) && \text{for stimulus regressor and implicit baseline} \\ \beta_c &\sim \mathcal{N}(\mu_c, \sigma_k^2) && \text{for each confound regressor} \end{aligned} \tag{34}$$

and run-wise regression coefficients are sampled as

$$\begin{aligned} \beta_{s,j} &\sim \mathcal{N}(\beta_s, \sigma_j^2) && \text{for each run } j = 1, \ldots, r \\ \beta_{c,j} &\sim \mathcal{N}(\beta_c, \sigma_j^2) && \text{for each run } j = 1, \ldots, r \end{aligned} \tag{35}$$

and $\beta_{s,j}$ and $\beta_{c,j}$ are then combined into an $(1 + c + 1) \times 1$ vector (cf. eq. 33) of regression coefficients β_j for a single run.

Third, a scan-by-scan covariance matrix implying a temporal auto-correlation structure is generated as

$$V = \begin{bmatrix} \tau^0 & \tau^1 & \cdots & \tau^{n-2} & \tau^{n-1} \\ \tau^1 & \tau^0 & \tau^1 & \ddots & \tau^{n-2} \\ \vdots & \tau^1 & \tau^0 & \tau^1 & \vdots \\ \tau^{n-2} & \ddots & \tau^1 & \tau^0 & \tau^1 \\ \tau^{n-1} & \tau^{n-2} & \cdots & \tau^1 & \tau^0 \end{bmatrix} \quad \text{where} \quad 0 < \tau < 1 \tag{36}$$

and noise terms for each run are sampled from the multivariate normal distribution with mean zero and covariance matrix as a product of the within-run variance and the temporal correlation matrix:

$$\varepsilon_j \sim \mathcal{N}(0, \sigma_i^2 \, V) \tag{37}$$

Note that this is equivalent to sampling each error term from the univariate normal distribution with mean zero and the noise variance, if τ is very close to zero, such that V becomes close to the identity matrix:

$$\varepsilon_{ij} \sim \mathcal{N}(0, \sigma_i^2), \; i = 1, \ldots, n, \; \text{if } \tau \to 0, \; \text{s.t. } V \approx I_n \; . \tag{38}$$

Finally, the run-wise design matrix X_j from equation (33), regression coefficients β_j from equation (35) and noise terms ε_j from equation (37) are used to generate simulated signals y_j for a single run according to equation (25):

$$y_j = X_j \beta_j + \varepsilon_j \; . \tag{39}$$

This procedure is repeated for each voxel and each run to generate an $n \times v \times r$ (scan-by-voxel-by-run) array of simulated fMRI signals.

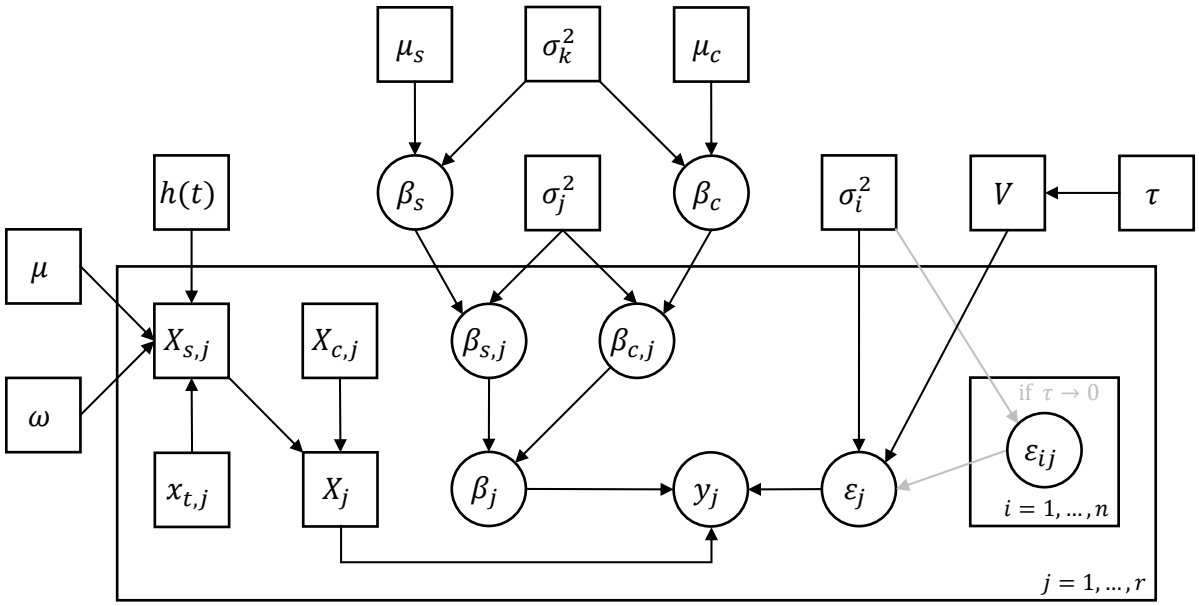

Figure 1: *Graphical model illustrating the generative model.* This figure illustrates the statistical model underlying data simulation in a single voxel. Squares correspond to fixed quantities and circles correspond to random variables. Arrows denote functional dependence (e.g. $\tau \to V$, cf. eq. 36) or that the probability distribution of the target quantity is parametrized in terms of the source quantities (e.g. $\{\sigma_i^2, V\} \to \varepsilon_j$, cf. eq. 37). Everything inside the large box is run-dependent and everything inside the small box is additionally scan-dependent.

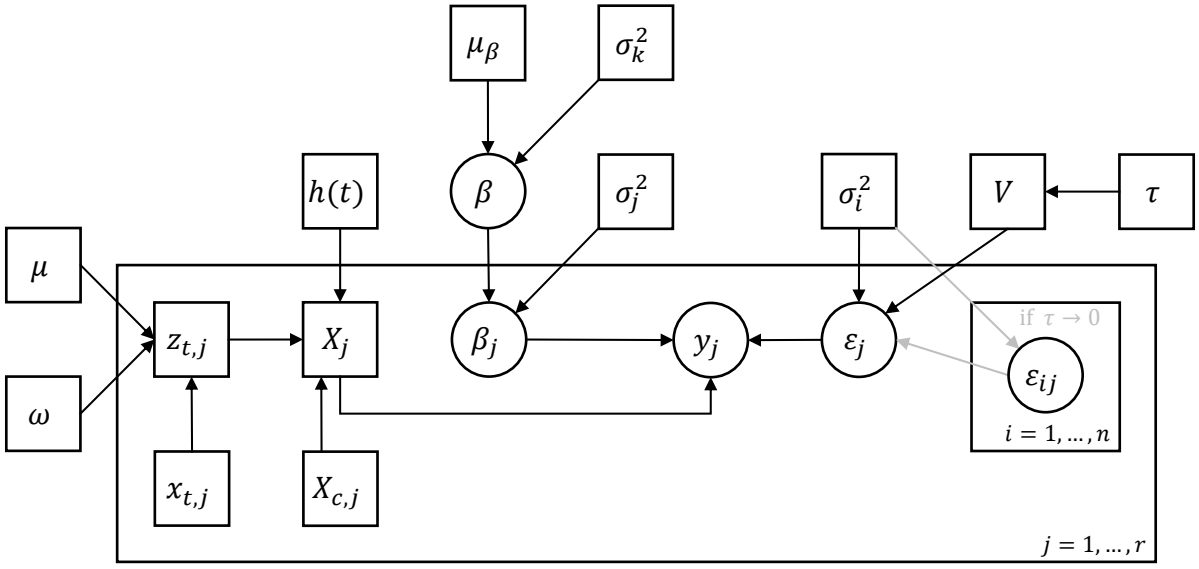

Figure 2: *Graphical model illustrating the generative model.* This figure is a simplified version of Figure 1 in which (i) regression coefficients β_j are not differentiated between signal and confound regressors and (ii) creation of the run-wise design matrix X_j is explained more transparently.

5 Appendix

A Maximum and width of logarithmic tuning function

Let $f_{\log}(x; \mu_{\log}, \sigma_{\log})$ be the Gaussian tuning function to a linear stimulus x in logarithmic space (see eq. 1). Given tuning parameters $\mu_{\log}$ and $\sigma_{\log}$, we are interested in the preferred stimulus μ (= the value at which $f_{\log}(x)$ reaches its maximum) and the tuning width ω (= the FWHM of $f_{\log}(x)$).
The first two derivatives of the logarithmic tuning function are:

$$f'_{\log}(x) = \left[-\frac{1}{x\,\sigma^2_{\log}} (\ln x - \mu_{\log}) \right] \cdot \exp\left[-\frac{1}{2} \frac{(\ln x - \mu_{\log})^2}{\sigma^2_{\log}} \right] \tag{A.1}$$

$$f''_{\log}(x) = \left[-\frac{1}{x^2\,\sigma^2_{\log}} \left(-\frac{1}{\sigma^2_{\log}} (\ln x - \mu_{\log})^2 + (\ln x - \mu_{\log} + 1) \right) \right] \cdot \exp\left[-\frac{1}{2} \frac{(\ln x - \mu_{\log})^2}{\sigma^2_{\log}} \right] . \tag{A.2}$$

Setting the first derivative to zero, we obtain:

$$\begin{aligned} f'_{\log}(\mu) &= 0 \\ 0 &= \left[-\frac{1}{\mu\,\sigma^2_{\log}} (\ln \mu - \mu_{\log}) \right] \cdot \exp\left[-\frac{1}{2} \frac{(\ln \mu - \mu_{\log})^2}{\sigma^2_{\log}} \right] \\ 0 &= -\frac{1}{\mu\,\sigma^2_{\log}} (\ln \mu - \mu_{\log}) \\ 0 &= \ln \mu - \mu_{\log} \\ \mu &= \exp\left[\mu_{\log}\right] \ . \end{aligned} \tag{A.3}$$

Since the second deriative at this value is negative

$$f''_{\log}(\mu) = -\frac{1}{\left(\exp\left[\mu_{\log}\right] \sigma_{\log}\right)^2} < 0 \ , \tag{A.4}$$

there is a maximum at μ. The function value at μ is

$$\begin{aligned} f_{\log}(\mu) &= \exp\left[-\frac{1}{2} \frac{(\ln \exp\left[\mu_{\log}\right] - \mu_{\log})^2}{\sigma^2_{\log}} \right] \\ &= \exp\left[0\right] = 1 \ . \end{aligned} \tag{A.5}$$

Thus, we are searching for the values of x that satisfy

$$\begin{aligned} f_{\log}(x_{1/2}) &= \frac{1}{2} \\ \frac{1}{2} &= \exp\left[-\frac{1}{2}\frac{(\ln x_{1/2} - \mu_{\log})^2}{\sigma_{\log}^2}\right] \\ 2\ln 2 &= \frac{(\ln x_{1/2} - \mu_{\log})^2}{\sigma_{\log}^2} \\ \sqrt{2\ln 2}\cdot\sigma_{\log} &= \pm(\ln x_{1/2} - \mu_{\log}) \\ x_1 &= \exp\left[\mu_{\log} + \sqrt{2\ln 2}\,\sigma_{\log}\right] \\ x_2 &= \exp\left[\mu_{\log} - \sqrt{2\ln 2}\,\sigma_{\log}\right] \ . \end{aligned} \tag{A.6}$$

With that, the FWHM tuning width is given by[7]

$$\begin{aligned} \omega &= x_1 - x_2 \\ \omega &= \exp\left[\mu_{\log} + \sqrt{2\ln 2}\,\sigma_{\log}\right] - \exp\left[\mu_{\log} - \sqrt{2\ln 2}\,\sigma_{\log}\right] \ . \end{aligned} \tag{A.7}$$

Note that μ can be inverted into $\mu_{\log}$ without knowing ω

$$\mu = \exp\left[\mu_{\log}\right] \quad \Leftrightarrow \quad \mu_{\log} = \ln\mu \ , \tag{A.8}$$

but ω cannot be inverted into $\sigma_{\log}$ without knowing μ:

$$\begin{aligned} \sigma_{\log} &= x(\mu,\omega), \quad \text{such that} \\ \omega &= \exp\left[\ln\mu + \sqrt{2\ln 2}\,x\right] - \exp\left[\ln\mu - \sqrt{2\ln 2}\,x\right] \ . \end{aligned} \tag{A.9}$$

[7]This means that tuning width is transformed from logarithmic space to linear space in a different way than preferred numerosity. For an implementation, see here: https://github.com/SkeideLab/EMPRISE/blob/akieslinger/data/code/harveymodel.py#L418 (function "sigma2fhwm").

B Equivalence of least squares and maximum likelihood

Consider the multiple linear regression model with correlated observations[8]:

$$y = X\beta + \varepsilon,\ \varepsilon \sim \mathcal{N}(0, \sigma^2 V)\ . \tag{B.1}$$

After parameter estimated using weighted least squares

$$\hat{\beta} = (X^{\mathrm{T}} V^{-1} X)^{-1} X^{\mathrm{T}} V^{-1} y\ , \tag{B.2}$$

the weighted residual sum of squares is:

$$\mathrm{wRSS} = (y - X\hat{\beta})^{\mathrm{T}} V^{-1} (y - X\hat{\beta})\ . \tag{B.3}$$

For maximum likelihood estimation, equation (B.1) entails the likelihood function

$$\begin{aligned} p(y|X, V, \beta, \sigma^2) &= \mathcal{N}(y; X\beta, \sigma^2 V) \\ &= \sqrt{\frac{1}{(2\pi)^n |\sigma^2 V|}} \cdot \exp\left[-\frac{1}{2}(y - X\beta)^{\mathrm{T}} (\sigma^2 V)^{-1} (y - X\beta)\right] \end{aligned} \tag{B.4}$$

which leads to the following log-likelihood function:

$$\begin{aligned} \mathrm{LL}(X, V, \beta, \sigma^2) &= \log p(y|X, V, \beta, \sigma^2) \\ &= -\frac{n}{2}\log(2\pi) - \frac{n}{2}\log(\sigma^2) - \frac{1}{2}\log|V| - \frac{1}{2\sigma^2}(y - X\beta)^{\mathrm{T}} V^{-1} (y - X\beta)\ . \end{aligned} \tag{B.5}$$

Plugging the maximum likelihood estimates[9]

$$\begin{aligned} \hat{\beta} &= (X^{\mathrm{T}} V^{-1} X)^{-1} X^{\mathrm{T}} V^{-1} y \\ \hat{\sigma}^2 &= \frac{1}{n}(y - X\hat{\beta})^{\mathrm{T}} V^{-1} (y - X\hat{\beta}) \end{aligned} \tag{B.6}$$

into the log-likelihood function, we obtain[10]:

$$\begin{aligned} \mathrm{MLL}(X, V) &= \mathrm{LL}(X, V, \hat{\beta}, \hat{\sigma}^2) \\ &= -\frac{n}{2}\log(2\pi) - \frac{n}{2}\log\left(\frac{1}{n}(y - X\hat{\beta})^{\mathrm{T}} V^{-1} (y - X\hat{\beta})\right) - \frac{1}{2}\log|V| - \frac{n\,(y - X\hat{\beta})^{\mathrm{T}} V^{-1} (y - X\hat{\beta})}{2\,(y - X\hat{\beta})^{\mathrm{T}} V^{-1} (y - X\hat{\beta})} \\ &\overset{(B.3)}{=} -\frac{n}{2}\log(2\pi) - \frac{n}{2}\log\left(\mathrm{wRSS}/n\right) - \frac{1}{2}\log|V| - \frac{n\,\mathrm{wRSS}}{2\,\mathrm{wRSS}} \\ &= -\frac{n}{2}\log(2\pi) - \frac{n}{2}\log\left(\mathrm{wRSS}/n\right) - \frac{n}{2} - \frac{1}{2}\log|V|\ . \end{aligned} \tag{B.7}$$

As one can see, $\mathrm{MLL}(X, V) \propto \log(\mathrm{wRSS})$. Thus, RSS-based estimation gives the same results as ML-based estimation.

[8]Note: For simplicity, we will proof the equivalence for the general case that the covariance of the errors is $\sigma^2 V$ (requiring weighted least squares, WLS) which includes the special case that $V = I_n$ (allowing for ordinary least squares, OLS).

[9]For a derivation, see: https://statproofbook.github.io/P/mlr-mle.

[10]For the derivation, see: https://statproofbook.github.io/P/mlr-mll.