

Appendix

Details of the E-M algorithm

Let $f(\cdot)$ denote probability density functions in general. We aim to obtain the set of parameters $\theta = \{\{\mathbf{A}_t\}_{t=1}^T, \mathbf{Q}_0, \mathbf{Q}, \{\sigma_i^2\}_{i=0}^p\}$ that maximizes the objective function $\log f(\{\mathbf{y}_t^{(r)}\}_{t=0, r=1}^{T, q}; \theta) + \log f(\theta)$, where $f(\theta)$ denotes the prior on θ . In our case $\log f(\{\mathbf{y}_t^{(r)}\}_{t=0, r=1}^{T, q}, \{\mathbf{u}_t^{(r)}\}_{t=0, r=1}^{T, q}; \theta)$ is much easier to compute than $\log f(\{\mathbf{y}_t^{(r)}\}_{t=0, r=1}^{T, q}; \theta)$, and the E-M algorithm (1) utilizes this property. Below, we briefly introduce how it works. Let $\tilde{\theta}$ denote an estimate of θ . For a more succinct notation, let $\mathbf{u}^\dagger \stackrel{\text{def}}{=} \{\mathbf{u}_t^{(r)}\}_{t=0, r=1}^{T, q}$ and $\mathbf{y}^\dagger \stackrel{\text{def}}{=} \{\mathbf{y}_t^{(r)}\}_{t=0, r=1}^{T, q}$. Let $\tilde{f}(\mathbf{u}^\dagger) = f(\mathbf{u}^\dagger | \mathbf{y}^\dagger; \tilde{\theta})$ be the posterior distribution of \mathbf{u}^\dagger conditioned on observations \mathbf{y}^\dagger , based on the current estimate $\tilde{\theta}$.

$$\begin{aligned} & \log f(\mathbf{y}^\dagger; \theta) + \log f(\theta) \\ &= \int \tilde{f}(\mathbf{u}^\dagger) \log f(\mathbf{y}^\dagger; \theta) d\mathbf{u}^\dagger + \log f(\theta) = \mathbb{E}_{\tilde{f}}(\log f(\mathbf{y}^\dagger; \theta)) + \log f(\theta) \\ &= \mathbb{E}_{\tilde{f}}(\log \frac{f(\mathbf{y}^\dagger, \mathbf{u}^\dagger; \theta)}{f(\mathbf{u}^\dagger | \mathbf{y}^\dagger; \theta)}) + \log f(\theta) \\ &= \mathbb{E}_{\tilde{f}}(\log f(\mathbf{y}^\dagger, \mathbf{u}^\dagger; \theta)) + \log f(\theta) - \mathbb{E}_{\tilde{f}}(\log f(\mathbf{u}^\dagger | \mathbf{y}^\dagger; \theta)) \end{aligned}$$

This also holds for $\theta = \tilde{\theta}$.

$$\log f(\mathbf{y}^\dagger; \tilde{\theta}) + \log f(\tilde{\theta}) = \mathbb{E}_{\tilde{f}}(\log f(\mathbf{y}^\dagger, \mathbf{u}^\dagger; \tilde{\theta})) + \log f(\tilde{\theta}) - \mathbb{E}_{\tilde{f}}(\log \tilde{f}(\mathbf{u}^\dagger))$$

Now consider the difference

$$(\log f(\mathbf{y}^\dagger; \theta) + \log f(\theta)) - (\log f(\mathbf{y}^\dagger; \tilde{\theta}) + \log f(\tilde{\theta})) \quad (1)$$

$$= \mathbb{E}_{\tilde{f}}(\log f(\mathbf{y}^\dagger, \mathbf{u}^\dagger; \theta)) + \log f(\theta) \quad (2)$$

$$- \mathbb{E}_{\tilde{f}}(\log f(\mathbf{y}^\dagger, \mathbf{u}^\dagger; \tilde{\theta})) - \log f(\tilde{\theta}) \quad (3)$$

$$+ \mathbb{E}_{\tilde{f}}(\log \frac{\tilde{f}(\mathbf{u}^\dagger)}{f(\mathbf{u}^\dagger | \mathbf{y}^\dagger; \theta)}) \quad (4)$$

In each iteration, given $\tilde{\theta}$, we select the new θ that maximize the term (2). Because term (3) is fixed given $\tilde{\theta}$, and term (4) is the non-negative Kullback-Leibler distance between $\tilde{f}(\mathbf{u}^\dagger)$ and $f(\mathbf{u}^\dagger | \mathbf{y}^\dagger; \theta)$, we are essentially maximizing a lower bound of the difference term (1). Therefore, when it converges, term (4) goes to zero, and we reach a local maximum of the objective function.

In each iteration, there are two steps: an E-step to compute \tilde{f} or the expression for term (2) given the current $\tilde{\theta}$, and an M-step to maximize the term (2) and update θ .

If we only have a prior on $\{\mathbf{A}_t\}_{t=1}^T$, (i.e., $f(\theta) = f(\{\mathbf{A}_t\}_{t=1}^T) \propto \exp(-(\lambda_0 \sum_{t=1}^T \|\mathbf{A}_t\|_F^2 + \lambda_1 \sum_{t=2}^T \|\mathbf{A}_t - \mathbf{A}_{t-1}\|_F^2))$), then term (3) has the following form up to some constant

$$\mathbb{E}_{\tilde{f}}(\log f(\mathbf{y}^\dagger, \mathbf{u}^\dagger; \theta)) + \log f(\theta) \quad (5)$$

$$= -\frac{1}{2}(q \log \det(\mathbf{Q}_0) + \mathbb{E}_{\tilde{f}}(\sum_{r=1}^q \mathbf{u}_0^{(r)'} \mathbf{Q}_0^{-1} \mathbf{u}_0^{(r)})) \quad (6)$$

$$- \frac{1}{2}(qT \log \det(\mathbf{Q}) + \mathbb{E}_{\tilde{f}}(\sum_{t=1}^T \sum_{r=1}^q (\mathbf{u}_t^{(r)} - \mathbf{A}_t \mathbf{u}_{t-1}^{(r)})' \mathbf{Q}^{-1} (\mathbf{u}_t^{(r)} - \mathbf{A}_t \mathbf{u}_{t-1}^{(r)})) + \log f(\{\mathbf{A}_t\}_{t=1}^T)) \quad (7)$$

$$- \frac{1}{2}(q(T+1) \log \det(\mathbf{R}) + \mathbb{E}_{\tilde{f}}(\sum_{t=0}^T \sum_{r=1}^q (\mathbf{y}_t^{(r)} - \mathbf{C} \mathbf{u}_t^{(r)})' \mathbf{R}^{-1} (\mathbf{y}_t^{(r)} - \mathbf{C} \mathbf{u}_t^{(r)}))). \quad (8)$$

where $'$ denotes the transpose of a column vector or a matrix, and $\det(\cdot)$ denotes the determinant of a square matrix. To evaluate (6), (7), and (8), we need to compute the posterior mean and covariance of

\mathbf{u}_t , as well as the cross covariance of \mathbf{u}_t and \mathbf{u}_{t-1} at each t for each trial ($r = 1, 2, \dots, q$) given $\tilde{\boldsymbol{\theta}}$:

$$\begin{aligned}\mathbf{u}_{t|T}^{(r)} &\stackrel{def}{=} \mathbb{E}(\mathbf{u}_t^{(r)} | \{\mathbf{y}_\tau^{(r)}\}_{\tau=0}^T), \\ \mathbf{P}_{t|T}^{(r)} &\stackrel{def}{=} \text{cov}(\mathbf{u}_t^{(r)} | \{\mathbf{y}_\tau^{(r)}\}_{\tau=0}^T), \\ \mathbf{P}_{(t,t-1)|T}^{(r)} &\stackrel{def}{=} \text{cov}(\mathbf{u}_t^{(r)}, \mathbf{u}_{t-1}^{(r)} | \{\mathbf{y}_\tau^{(r)}\}_{\tau=0}^T)\end{aligned}$$

To compute these values, we use the forward and backward steps in the Kalman smoothing algorithm (1). For simplicity, we drop the superscript (r) and the $\tilde{\cdot}$ symbol on $\tilde{\boldsymbol{\theta}}$. We define the following terms

$$\begin{aligned}\mathbf{u}_{t|s} &\stackrel{def}{=} \mathbb{E}(\mathbf{u}_t | \mathbf{y}_0, \mathbf{y}_1, \dots, \mathbf{y}_s), \\ \mathbf{P}_{t|s} &\stackrel{def}{=} \text{cov}(\mathbf{u}_t | \mathbf{y}_0, \mathbf{y}_1, \dots, \mathbf{y}_s), \\ \mathbf{P}_{(t,t-1)|s} &\stackrel{def}{=} \text{cov}(\mathbf{u}_t, \mathbf{u}_{t-1} | \mathbf{y}_0, \mathbf{y}_1, \dots, \mathbf{y}_s).\end{aligned}$$

In the forward step, we set $\mathbf{u}_{0|0} = \mathbf{Q}_0 \mathbf{C}' (\mathbf{C} \mathbf{Q}_0 \mathbf{C}' + \mathbf{R})^{-1} \mathbf{y}_0$ and $\mathbf{P}_{0|0} = \mathbf{Q}_0 - \mathbf{Q}_0 \mathbf{C}' (\mathbf{C} \mathbf{Q}_0 \mathbf{C}' + \mathbf{R})^{-1} \mathbf{C} \mathbf{Q}_0$. For $t = 1, 2, \dots, T$, we have

$$\begin{aligned}\mathbf{u}_{t|(t-1)} &= \mathbf{A}_t \mathbf{u}_{(t-1)|(t-1)} \\ \mathbf{P}_{t|(t-1)} &= \mathbf{A}_t \mathbf{P}_{(t-1)|(t-1)} \mathbf{A}_t' + \mathbf{Q} \\ \mathbf{K}_t &\stackrel{def}{=} \mathbf{P}_{t|(t-1)} \mathbf{C}' (\mathbf{C} \mathbf{P}_{t|(t-1)} \mathbf{C}' + \mathbf{R})^{-1} \\ \mathbf{u}_{t|t} &= \mathbf{u}_{t|(t-1)} + \mathbf{K}_t (\mathbf{y}_t - \mathbf{C} \mathbf{u}_{t|(t-1)}) \\ \mathbf{P}_{t|t} &= \mathbf{P}_{t|(t-1)} - \mathbf{K}_t \mathbf{C} \mathbf{P}_{t|(t-1)}\end{aligned}$$

In the backward step, for $t = T, T-1, \dots, 1$

$$\begin{aligned}\mathbf{H}_{t-1} &\stackrel{def}{=} \mathbf{P}_{(t-1)|(t-1)} \mathbf{A}_t' \mathbf{P}_{t|(t-1)}^{-1} \\ \mathbf{u}_{(t-1)|T} &= \mathbf{u}_{(t-1)|(t-1)} + \mathbf{H}_{t-1} (\mathbf{u}_{t|T} - \mathbf{A}_t \mathbf{u}_{(t-1)|(t-1)}) \\ \mathbf{P}_{(t-1)|T} &= \mathbf{P}_{(t-1)|(t-1)} + \mathbf{H}_{t-1} (\mathbf{P}_{t|T} - \mathbf{P}_{t|(t-1)}) \mathbf{H}_{t-1}'\end{aligned}$$

and with $\mathbf{P}_{(T,T-1)|T} = (\mathbf{I} - \mathbf{K}_T \mathbf{C}) \mathbf{A}_T \mathbf{P}_{(T-1)|(T-1)}$, for $t = T-1, T-2, \dots, 2$, we have

$$\mathbf{P}_{(t-1,t-2)|T} = \mathbf{P}_{(t-1)|(t-1)} \mathbf{H}_{t-2}' + \mathbf{H}_{t-1} (\mathbf{P}_{(t,t-1)|T} - \mathbf{A}_t \mathbf{P}_{(t-1)|(t-1)}) \mathbf{H}_{t-2}'.$$

If we denote the terms (6), (7), and (8) with \mathcal{L}_1 , \mathcal{L}_2 and \mathcal{L}_3 respectively, then using the posterior statistics above, we have

$$\begin{aligned}\mathcal{L}_1 &= q \log \det(\mathbf{Q}_0) + \mathbb{E}_{\tilde{f}} \left(\sum_{r=1}^q (\mathbf{u}_0^{(r)'} \mathbf{Q}_0^{-1} \mathbf{u}_0^{(r)}) \right) = q \log \det(\mathbf{Q}_0) + \text{tr}(\mathbf{Q}_0^{-1} \mathbf{B}_0) \\ \mathcal{L}_2 &= qT \log \det(\mathbf{Q}) + \mathbb{E}_{\tilde{f}} \left(\sum_{t=1}^T \sum_{r=1}^q (\mathbf{u}_t^{(r)} - \mathbf{A}_t \mathbf{u}_{t-1}^{(r)})' \mathbf{Q}^{-1} (\mathbf{u}_t^{(r)} - \mathbf{A}_t \mathbf{u}_{t-1}^{(r)}) \right) - \log f(\{\mathbf{A}_t\}_{t=1}^T) \\ &= qT \log \det(\mathbf{Q}) + \text{tr}(\mathbf{Q}^{-1} \sum_{t=1}^T (\mathbf{B}_{1t} - \mathbf{A}_t \mathbf{B}_{2t}' - \mathbf{B}_{2t} \mathbf{A}_t' + \mathbf{A}_t \mathbf{B}_{3t} \mathbf{A}_t')) + \lambda_0 \sum_{t=1}^T \|\mathbf{A}_t\|_F^2 + \lambda_1 \sum_{t=2}^T \|\mathbf{A}_t - \mathbf{A}_{t-1}\|_F^2 \\ \mathcal{L}_3 &= q(T+1) \log \det(\mathbf{R}) + \mathbb{E}_{\tilde{f}} \left(\sum_{t=0}^T \sum_{r=1}^q (\mathbf{y}_t^{(r)} - \mathbf{C} \mathbf{u}_t^{(r)})' \mathbf{R}^{-1} (\mathbf{y}_t^{(r)} - \mathbf{C} \mathbf{u}_t^{(r)}) \right) \\ &= q(T+1) \log \det(\mathbf{R}) + \text{tr}(\mathbf{R}^{-1} \mathbf{B}_4)\end{aligned}$$

where

$$\begin{aligned}
B_0 &= \sum_{r=1}^q (\mathbf{P}_{0|T}^{(r)} + \mathbf{u}_{0|T}^{(r)} (\mathbf{u}_{0|T}^{(r)})') \\
B_{1t} &= \sum_{r=1}^q (\mathbf{P}_{t|T}^{(r)} + \mathbf{u}_{t|T}^{(r)} (\mathbf{u}_{t|T}^{(r)})') \\
B_{2t} &= \sum_{r=1}^q (\mathbf{P}_{(t,t-1)|T}^{(r)} + \mathbf{u}_{t|T}^{(r)} (\mathbf{u}_{(t-1)|T}^{(r)})') \\
B_{3t} &= \sum_{r=1}^q (\mathbf{P}_{(t-1)|T}^{(r)} + \mathbf{u}_{(t-1)|T}^{(r)} (\mathbf{u}_{(t-1)|T}^{(r)})') \\
B_4 &= \sum_{r=1}^q \sum_{t=0}^T [(\mathbf{y}_t^{(r)} - \mathbf{C} \mathbf{u}_{t|T}^{(r)}) (\mathbf{y}_t^{(r)} - \mathbf{C} \mathbf{u}_{t|T}^{(r)})' + \mathbf{C} \mathbf{P}_{t|T}^{(r)} \mathbf{C}']
\end{aligned}$$

In the M-step, we optimize for the three terms separately. When an analytical solution is difficult to compute, we use gradient descent with back-tracking line search (2) (e.g., minimizing $g = \mathcal{L}_2$ and $g = \mathcal{L}_3$, where $\mathbf{x} = \{\mathbf{A}_t\}_{t=1}^T$ or $\mathbf{x} = \{\sigma_i\}_{i=0}^p$.

Data: function g , initial value \mathbf{x}_0 , $\beta \in (0, 1)$
Result: \mathbf{x} that minimizes $g(\mathbf{x})$
initialization: $\mathbf{x} \leftarrow \mathbf{x}_0$;
while the difference of \mathbf{x} in two consecutive iterations is not small enough **do**
 $\tau \leftarrow 1$;
 compute the gradient $\nabla g(\mathbf{x})$;
 while $g(\mathbf{x} - \tau \nabla g(\mathbf{x})) > g(\mathbf{x}) - \tau/2 \|\nabla g(\mathbf{x})\|_2^2$ **do**
 $\tau \leftarrow \beta \tau$;
 end
 $\mathbf{x} \leftarrow \mathbf{x} - \tau \nabla g(\mathbf{x})$;
end

Algorithm 1: Backtracking with line-search

Solving for dynamic connectivity parameters when state-variables are observed

Given $\{\mathbf{u}_t^{(r)}\}_{t=0, r=1}^{T, q}$, we solve for \mathbf{Q}_0 , \mathbf{Q} and $\{\mathbf{A}_t\}_{t=1}^T$ by maximizing the log-likelihood plus the logarithm of the prior, which is equivalent to minimizing

$$\begin{aligned}
& -\log f(\{\mathbf{u}_t^{(r)}\}_{t=0, r=1}^{T, q}) - \log f(\{\mathbf{A}_t\}_{t=1}^T) \\
& \propto q \log \det(\mathbf{Q}_0) + \text{tr}(\mathbf{Q}_0^{-1} \sum_{r=1}^q \mathbf{u}_0^{(r)} \mathbf{u}_0^{(r)})' \\
& + qT \log \det(\mathbf{Q}) + \text{tr}(\mathbf{Q}^{-1} \sum_{t=1}^T \sum_{r=1}^q (\mathbf{u}_t^{(r)} - \mathbf{A}_t \mathbf{u}_{t-1}^{(r)}) (\mathbf{u}_t^{(r)} - \mathbf{A}_t \mathbf{u}_{t-1}^{(r)})') \\
& + \lambda_0 \sum_{t=1}^T \|\mathbf{A}_t\|_F^2 + \lambda_1 \sum_{t=2}^T \|\mathbf{A}_t - \mathbf{A}_{t-1}\|_F^2
\end{aligned}$$

where the optimization procedure is similar to that in the M-step. \mathbf{Q}_0 has an analytical solution $\mathbf{Q}_0 \leftarrow (1/q) \sum_{r=1}^q \mathbf{u}_0^{(r)} (\mathbf{u}_0^{(r)})'$, and $\{\mathbf{A}_t\}_{t=1}^T$ and \mathbf{Q} can be updated in alternations. Given $\{\mathbf{A}_t\}_{t=1}^T$, \mathbf{Q} has an analytical solution $\mathbf{Q} \leftarrow 1/(qT) \sum_{t=1}^T \sum_{r=1}^q (\mathbf{u}_t^{(r)} - \mathbf{A}_t \mathbf{u}_{t-1}^{(r)}) (\mathbf{u}_t^{(r)} - \mathbf{A}_t \mathbf{u}_{t-1}^{(r)})'$, and given \mathbf{Q} , $\{\mathbf{A}_t\}_{t=1}^T$ can be solved by gradient descent with backtracking line search, where the gradient is

$$2\mathbf{Q}^{-1} \left(-\sum_{r=1}^q \mathbf{u}_t^{(r)} (\mathbf{u}_{t-1}^{(r)})' + \mathbf{A}_t \sum_{r=1}^q \mathbf{u}_{t-1}^{(r)} (\mathbf{u}_{t-1}^{(r)})' \right) + 2\mathbf{D}_t$$

and

$$D_t = \begin{cases} \lambda_1(2\mathbf{A}_t - \mathbf{A}_{t+1} - \mathbf{A}_{t-1}) + \lambda_0\mathbf{A}_t & \text{for } t = 2, \dots, T-1; \\ \lambda_1(\mathbf{A}_1 - \mathbf{A}_2) + \lambda_0\mathbf{A}_1 & \text{for } t = 1; \\ \lambda_1(\mathbf{A}_T - \mathbf{A}_{T-1}) + \lambda_0\mathbf{A}_T & \text{for } t = T; \end{cases}$$

Computing the spatio-temporal covariance of the ROI mean activity and evaluating the marginal log-likelihood of sensor data

According to the auto-regressive model, given $\{\mathbf{A}_t\}_{t=1}^T, \mathbf{Q}_0, \mathbf{Q}$, we have

$$\mathbf{u}_t = \mathbf{A}_t \mathbf{u}_{t-1} + \boldsymbol{\epsilon}_t = \left(\prod_{\tau=t}^1 \mathbf{A}_\tau \right) \mathbf{u}_0 + \boldsymbol{\epsilon}_t + \sum_{j=1}^{t-1} \left(\prod_{\tau=t}^{t-j+1} \mathbf{A}_\tau \right) \boldsymbol{\epsilon}_{t-j}$$

and the marginal covariance of \mathbf{u}_t for $t = 1, \dots, T$ is

$$\text{cov}(\mathbf{u}_t) = \left(\prod_{\tau=t}^1 \mathbf{A}_\tau \right) \mathbf{Q}_0 \left(\prod_{\tau=t}^1 \mathbf{A}_\tau \right)' + \mathbf{Q} + \sum_{i=1}^{t-1} \left(\prod_{\tau=t}^{t-j+1} \mathbf{A}_\tau \right) \mathbf{Q} \left(\prod_{\tau=t}^{t-j+1} \mathbf{A}_\tau \right)'$$

where the in the product $\prod_{\tau=t}^{t-j+1}$, τ decreases from t to $t-j+1$. To compute the marginal covariance, it is convenient to first compute matrices $\tilde{\mathbf{A}}_{j,k} = \prod_{\tau=k}^j \mathbf{A}_\tau, j \leq k$, and then we have

$$\text{cov}(\mathbf{u}_t) = \tilde{\mathbf{A}}_{1,t} \mathbf{Q}_0 \tilde{\mathbf{A}}_{1,t}' + \mathbf{Q} + \sum_{j=1}^{t-1} \tilde{\mathbf{A}}_{(t-j+1),t} \mathbf{Q} \tilde{\mathbf{A}}_{(t-j+1),t}'.$$

Let $\mathbf{U}_{p \times (T+1)} = [\mathbf{u}_0, \mathbf{u}_1, \dots, \mathbf{u}_T]$ (where \mathbf{u}_t is of size $p \times 1$); let $\text{vec}(\mathbf{U})$ be the concatenation of the columns of \mathbf{U} . Let the $p(T+1) \times p(T+1)$ matrix $\boldsymbol{\Sigma}$ denote the covariance matrix of $(\text{vec}(\mathbf{U}))$. This covariance can be computed as

$$\begin{aligned} & \boldsymbol{\Sigma}[tp+1 : (t+1)p, (t+h)p+1 : (t+h+1)p] \\ &= \text{cov}(\mathbf{u}_t, \mathbf{u}_{t+h}) \\ &= \text{cov}(\mathbf{u}_t, \mathbf{A}_{t+h} \mathbf{A}_{t+h-1} \dots \mathbf{A}_{t+1} \mathbf{u}_t) \\ &= \text{cov}(\mathbf{u}_t) \left(\prod_{\tau=t+1}^{t+h} \mathbf{A}_\tau \right)' \\ &= \text{cov}(\mathbf{u}_t) \tilde{\mathbf{A}}_{t+1,t+h}' \end{aligned}$$

where by $\boldsymbol{\Sigma}[tp+1 : (t+1)p, (t+h)p+1 : (t+h+1)p]$, we mean the sub-matrix composed of the consecutive rows from $tp+1$ to $(t+1)p$ and the consecutive columns from $(t+h)p+1$ to $(t+h+1)p$ (indices are 1-based). If needed, we can also compute the marginal correlation between the mean of i_1 th ROI at time t_1 and the mean of i_2 th ROI at time t_2 :

$$\text{correlation}(\mathbf{u}_{t_1}[i_1], \mathbf{u}_{t_2}[i_2]) = \frac{\boldsymbol{\Sigma}[t_1p+i_1, t_2p+i_2]}{\sqrt{\boldsymbol{\Sigma}[t_1p+i_1, t_1p+i_1] \boldsymbol{\Sigma}[t_2p+i_2, t_2p+i_2]}} \quad (9)$$

Next, we discuss how to evaluate the objective function that contains the marginal likelihood of the observed multi-trial sensor data and a prior term for the parameters (i.e., $\log f(\{\mathbf{y}_t^{(r)}\}_{t=0, r=1}^{T,q}) + \log f(\boldsymbol{\theta})$). Since $\log f(\boldsymbol{\theta})$ is easy to compute, we mainly focus on evaluating $-\log f(\{\mathbf{y}_t^{(r)}\}_{t=0, r=1}^{T,q})$. Let the $n \times (T+1)$ matrix $\mathbf{Y} = (\mathbf{y}_0, \mathbf{y}_1, \dots, \mathbf{y}_T)$ denote the sensor time series in an arbitrary trial. Let $\text{vec}(\mathbf{Y})$ be a vector obtained by concatenating the columns in \mathbf{Y} .

Let $\tilde{\mathbf{C}}$ be an $nT \times pT$ block-diagonal matrix, where $\tilde{\mathbf{C}}[tn+1 : (t+1)n, tp+1 : (t+1)p] = \mathbf{C}$, for $t = 0, 1, \dots, T$; $\tilde{\mathbf{C}}[tn+1 : (t+1)n, tp+1 : (t+1)p]$ denotes the intersection of the sub-rows (rows $tn+1$ to $(t+1)n$) and the sub-columns (columns $tp+1$ to $(t+1)p$). Similarly, let $\tilde{\mathbf{R}}$ be an $nT \times nT$ block diagonal matrix, where $\tilde{\mathbf{R}}[tn+1 : (t+1)n, tn+1 : (t+1)n] = \mathbf{R}$, for $t = 0, 1, \dots, T$. Then we have

$$\text{cov}(\text{vec}(\mathbf{Y})) = \tilde{\mathbf{C}} \boldsymbol{\Sigma} \tilde{\mathbf{C}}' + \tilde{\mathbf{R}}$$

and

$$-\log f(\{\mathbf{y}_t^{(r)}\}_{t=0, r=1}^{T, q}) \propto q \log \det(\text{cov}(\text{vec}(\mathbf{Y}))) + \sum_{r=1}^q \text{tr}(\text{cov}(\text{vec}(\mathbf{Y}))^{-1} \text{vec}(\mathbf{Y}^{(r)}) \text{vec}(\mathbf{Y}^{(r)})')$$

+ some constant irrelevant to the parameters

We use the following equations to facilitate the computation

$$\begin{aligned} \text{cov}(\text{vec}(\mathbf{Y}))^{-1} &= \tilde{\mathbf{R}}^{-1} - \tilde{\mathbf{R}}^{-1} \tilde{\mathbf{C}} (\Sigma^{-1} + \tilde{\mathbf{C}}' \tilde{\mathbf{R}}^{-1} \tilde{\mathbf{C}})^{-1} \tilde{\mathbf{C}}' \tilde{\mathbf{R}}^{-1} \\ \log \det(\text{cov}(\text{vec}(\mathbf{Y}))) &= \log \det(\Sigma^{-1} + \tilde{\mathbf{C}}' \tilde{\mathbf{R}}^{-1} \tilde{\mathbf{C}}) + \log \det(\Sigma) + \log \det(\tilde{\mathbf{R}}) \end{aligned}$$

Estimating the source space activity

Estimating the source space activity \mathbf{J}_t is not our primary goal in this model. However, we can still estimate \mathbf{J}_t once the ROI means \mathbf{u}_t is estimated in the end of the E-M algorithm. Given the hierarchical model below

$$\begin{aligned} \mathbf{J}_t | \mathbf{u}_t &\sim \mathcal{N}(\mathbf{L} \mathbf{u}_t, \mathbf{Q}_J) \\ \mathbf{y}_t | \mathbf{J}_t &\sim \mathcal{N}(\mathbf{G} \mathbf{J}_t, \mathbf{Q}_e) \end{aligned}$$

we have

$$\mathbf{J}_t | (\mathbf{u}_t, \mathbf{y}_t) \sim \mathcal{N}((\mathbf{G}' \mathbf{Q}_e^{-1} \mathbf{G} + \mathbf{Q}_J^{-1})^{-1} (\mathbf{G}' \mathbf{Q}_e^{-1} \mathbf{y}_t + \mathbf{Q}_J^{-1} \mathbf{L} \mathbf{u}_t), (\mathbf{G}' \mathbf{Q}_e^{-1} \mathbf{G} + \mathbf{Q}_J^{-1})^{-1}) \quad (10)$$

Given an estimate of \mathbf{u}_t , which we denote by $\hat{\mathbf{u}}_t$, we can estimate \mathbf{J}_t by the mean term in Equation 10,

$$\hat{\mathbf{J}}_t = (\mathbf{G}' \mathbf{Q}_e^{-1} \mathbf{G} + \hat{\mathbf{Q}}_J^{-1})^{-1} (\mathbf{G}' \mathbf{Q}_e^{-1} \mathbf{y}_t + \hat{\mathbf{Q}}_J^{-1} \mathbf{L} \hat{\mathbf{u}}_t)$$

where $\hat{\mathbf{Q}}_J$ is based on the estimations of σ_i^2 s. Note that to make the notation succinct, we have omitted the trial index $^{(r)}$. But such an estimate step can easily be done for each time point t and each trial.

Additional details about the simulations

In the simulations, the bilateral pericalcarine sulci and the parahippocampal gyri were automatically labeled by the Freesurfer software (3) according to the anatomy. With $p = 2$ ROIs, $T = 20$ time points and $q = 200$ trials, we had a relatively large number of observations. Therefore, in the prior of $\{\mathbf{A}_t\}_{t=1}^T$, we preselected $\lambda_0 = 0$ and $\lambda_1 = 0.1$, expecting that these values only added minimal regularization. In practice, these values can be selected as the ones that maximize the cross-validated log-likelihood of the sensor data. When using the posterior estimates of \mathbf{u}_t to predict \mathbf{y}_t and quantify how much variance of the sensor data was explained by the state-variables, we observed about 5.95% in our simulations when $a = 2$ and 10.84% when $a = 5$.

It is worth noting that the in the two-step MNE method, although the estimated \mathbf{u}_t could be temporally correlated with the true \mathbf{u}_t in the simulations, the scale was much smaller. For example, Figure 1a shows the true \mathbf{u}_t (blue) in one trial, one ROI (ROI 2), and one simulation ($\alpha = 5$), as well as the estimated \mathbf{u}_t in the two-step MNE method (red), and the posterior estimate of \mathbf{u}_t by the state-space model (green). The one from the two-step MNE method was close to zero. This could be due to the shrinkage effect of the L_2 penalty with a relatively large penalization parameter λ . However, using a smaller λ resulted in extremely noisy estimations of \mathbf{u}_t , which were not correlated with the true \mathbf{u}_t .

Although the auto-regressive (AR) coefficients (\mathbf{A}_t s) should not be very sensitive to the scales of \mathbf{u}_t , when the scaling is different for each ROI, the off-diagonal entries might be affected. In contrast, the diagonal entries, which describe the dependence of the ROI activity on the one-step-back history of itself, are unlikely to be affected by the scales. To give the two-step MNE method a chance to correct for scaling for the off-diagonal entries, we also computed a different measurement of the relative error. Let \mathbf{A}_t denote the true AR coefficients. For each pair of ROIs, (i, j) , we multiply the estimates ($\hat{\mathbf{A}}_t$) with a scalar α , such that $\sum_{t=1}^T (\mathbf{A}_t[i, j] - \alpha \hat{\mathbf{A}}_t[i, j])^2$ was minimized, and then we defined the relative error as

$$\sqrt{\frac{\sum_{t=1}^T (\mathbf{A}_t[i, j] - \alpha \hat{\mathbf{A}}_t[i, j])^2}{\sum_{t=1}^T (\mathbf{A}_t[i, j])^2}}$$

We took the average of the relative error after scaling for the off-diagonal (i, j) pairs (i.e. $i \neq j$) (Figure 1b); in the results, we still observed that the state-space model yielded smaller errors than the two-step MNE method.

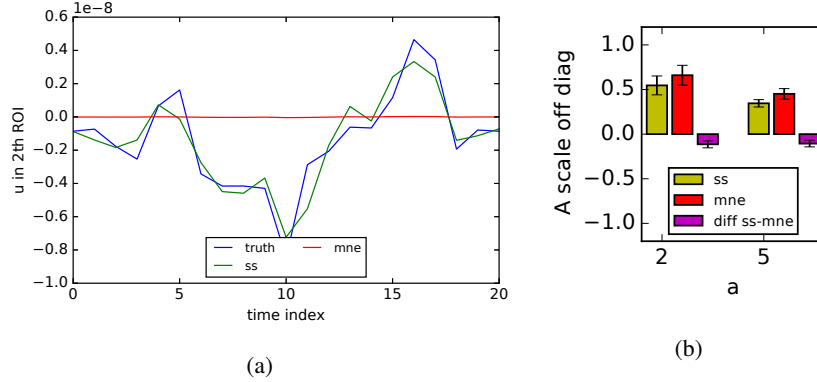


Figure 1: (a), Visualization of u_t in ROI 2 in one trial from a simulation. (The legends: truth (blue), true values; ss (green), estimates by the state-space model; mne (red), estimates by the two-step MNE method.) (b) Relative error of AR coefficients after scaling, averaged within off-diagonal entries by the state-space models and the two-step MNE method, as well as the paired difference. The error bars show standard errors across individual simulations.

Additional details about the empirical MEG data

Data in the MEG experiment was collected using a 306-channel whole-head MEG system (Elekta Neuromag, Helsinki, Finland) at the Brain Mapping Center at the University of Pittsburgh. During the experiment, a human participant processed naturalistic photographs of scenes. In each trial, while the participant fixated their eyes on a “+” symbol in the center of a gray screen, one image centered in the screen appeared and lasted for 200 ms. Afterwards, the screen switched back to the “+” symbol on the gray background. The participant was instructed to press a button if the image was identical to the previous one. In the entire session, 362 images of scenes were used and each image was repeated 5 times (excluding the consecutive repetitions) in a pseudo-random order. The participant gave written informed consent and was financially compensated. All procedures followed the principles in the Declaration of Helsinki and were approved by the Institutional Review Boards of Carnegie Mellon University and the University of Pittsburgh.

The MEG recordings were acquired at 1 kHz, high-pass filtered at 0.1 Hz and low-pass filtered at 330 Hz. In the preprocessing, the raw data was filtered with a 1-110 Hz bandpass filter, and then with a notch filter at 60 Hz to reduce the power-line interference; independent component analysis (ICA) was used to decompose the MEG data into multiple components, and the components that were highly correlated with eye blinks and heartbeats were removed. After the preprocessing, the MEG data was cropped from -140 ms to 960 ms for each trial, where 0 is the stimulus onset. A signal space projection (SSP) was applied to the data; the SSP constructed a low-dimensional linear subspace characterizing the empty room noise (via principal component analysis), and removed from the experimental MEG recordings the projection onto this subspace, so that neural signals orthogonal to the principal components of empty room noise remained. For each sensor, a baseline temporal mean within -140 ms to -40 ms was subtracted for all time points, separately for each trial. The data was further down-sampled at 100 Hz (i.e., one time point for every 10 ms). For each image, data across 5 repetitions were averaged, and further cropped from -100 to 700 ms. Finally, the sample mean of the $n \times (T + 1)$ sensor data across the $q = 362$ observations were subtracted off. The preprocessing, the source-space definition, the forward modeling, and the minimum-norm estimates were done in the MNE (MNE-python) software (4; 5).

References

- [1] R. H. Shumway and D. S. Stoffer. An approach to time series smoothing and forecasting using the EM algorithm. *Journal of time series analysis*, 3(4):253–264, 1982.

- [2] S. Boyd and L. Vandenberghe. *Convex Optimization*. Cambridge University Press, New York, NY, USA, 2004.
- [3] B. Fischl, D. H. Salat, E. Busa, M. Albert, M. Dieterich, C. Haselgrove, A. Van Der Kouwe, R. Killiany, D. Kennedy, S. Klaveness, et al. Whole brain segmentation: automated labeling of neuroanatomical structures in the human brain. *Neuron*, 33(3):341–355, 2002.
- [4] A. Gramfort, M. Luessi, E. Larson, D. A. Engemann, D. Strohmeier, C. Brodbeck, R. Goj, M. Jas, T. Brooks, L. Parkkonen, et al. Meg and eeg data analysis with mne-python. *Frontiers in neuroscience*, 7:267, 2013.
- [5] A. Gramfort, M. Luessi, E. Larson, D. A. Engemann, D. Strohmeier, C. Brodbeck, L. Parkkonen, and M. S. Hämäläinen. MNE software for processing MEG and EEG data. *Neuroimage*, 86:446–460, 2014.