A Supplementary Material

A.1 Automatic Relevance Determination and Kernel

In this work we use the exponentiated quadratic (also known as the “squared exponential”) ARD kernel:

$$K(\mathbf{x}, \mathbf{x}') = \prod_{r=1}^{R} \exp\left(-\frac{(x_r - x'_r)^2}{2\alpha_r}\right). \tag{18}$$

A.2 Derivation of lower bound

The lower bound Equation 6 is derived as follows:

$$\log p(\mathscr{D}_{1:S}, A_{1:S}|\Theta) = \log\left[\iint p(\mathscr{D}_{1:S}, A_{1:S}|f_{1:T})\mathrm{d}p(f_{1:T}|\mathbf{u}_{1:T})p(\mathbf{u}_{1:T})\frac{q(\mathbf{u}_{1:T})}{q(\mathbf{u}_{1:T})}\,\mathrm{d}\mathbf{u}_{1:T}\right] \tag{19}$$

$$\geq \int \prod_t \int \mathrm{d}p(f_t \mid \mathbf{u}_t)\, q(\mathbf{u}_t)\, \mathrm{d}\mathbf{u}_t\ \log\big[p(\mathscr{D}_{1:S}, A_{1:S} \mid f_{1:T})\big] + \iint \mathrm{d}p(f_{1:T} \mid \mathbf{u}_{1:T})\, q(\mathbf{u}_{1:T})\ \log\left[\frac{p(\mathbf{u}_{1:T})}{q(\mathbf{u}_{1:T})}\right]\ \mathrm{d}\mathbf{u}_{1:T} \tag{20}$$

$$= \mathbb{E}_{q(f_{1:T})}\big[\log p(\mathscr{D}_{1:S}, A_{1:S} \mid f_{1:T})\big] - \mathrm{KL}\big(q(\mathbf{u}_{1:T}) \,\big\|\, p(\mathbf{u}_{1:T})\big) \tag{21}$$

$$\triangleq \mathcal{L}(\mathscr{D}_{1:S}, A_{1:S}; \Theta). \tag{22}$$

A.3 Definition of KL

The KL term in Equation 6 is the Kullback–Leibler divergence between T pairs of independent Gaussians distribution and is defined by:

$$\mathrm{KL}\big(q(\mathbf{u}_{1:T}) \,\big\|\, p(\mathbf{u}_{1:T})\big) = \frac{1}{2}\sum_t \left[\mathrm{tr}\left(\mathbf{K}_{zz}^{-1}\mathbf{S}_t\right) + (\vec{1}\bar{u}_t - \mathbf{m})^\top \mathbf{K}_{zz}^{-1}(\vec{1}\bar{u}_t - \mathbf{m}) - M + \log\frac{|\mathbf{K}_{zz}|}{|\mathbf{S}_t|}\right]. \tag{23}$$

A.4 Definition of $\tilde{G}$

The function $\tilde{G}$ that appears in the expectation $\mathbb{E}_{q(f_t)}[\log f_{s,t,n}^2] = \int_{-\infty}^{\infty} \log(f_{s,t,n}^2)\, \mathcal{N}(f_{s,t,n}; \tilde{\mu}_{s,t,n}, \tilde{\sigma}_{s,t,n}^2)\, \mathrm{d}f_{s,t,n}$, Equations 9, is a specialised version of the partial derivative of the confluent hyper-geometric function,

$${}_1F_1(a, b, z) = \sum_{k=0}^{\infty} \frac{(a)_k z^k}{(b)_k\ k!}, \tag{24}$$

with respect to its first argument and is defined by:

$$\tilde{G}(z) = {}_1F_1{}^{(1,0,0)}\left(0, \frac{1}{2}, z\right) = 2z\sum_{j=0}^{\infty} \frac{j!\ z^j}{(2)_j(1\frac{1}{2})_j}, \tag{25}$$

where $(\cdot)_j$ denotes the rising Pochhammer series $(a)_0 = 1,\ (a)_j = a(a+1)(a+2)\ldots(a+j-1)$.

A.5 Definition of Ψ_{zz}

For the ARD Kernel the function $\Psi(\mathbf{z}, \mathbf{z}') = \int_{\mathcal{X}} K(\mathbf{z}, \mathbf{x})K(\mathbf{x}, \mathbf{z}')\mathrm{d}\mathbf{x}$ can be computed in closed form:

$$\Psi(\mathbf{z}, \mathbf{z}') = \prod_{r=1}^{R} \frac{\sqrt{\pi\alpha_r}}{2} \exp\left(-\frac{(z_r - z'_r)^2}{4\alpha_r}\right)\left[\mathrm{erf}\left(\frac{\bar{z}_r - \mathcal{X}_r^{\mathrm{Min}}}{\sqrt{\alpha_r}}\right) - \mathrm{erf}\left(\frac{\bar{z}_r - \mathcal{X}_r^{\mathrm{Max}}}{\sqrt{\alpha_r}}\right)\right], \tag{26}$$

where $\bar{z}_r = \frac{1}{2}(z_r + z'_r)$.

A.6 Detailed Derivation of the Collapsed Bound

The set of all possible assignments is:

$$\big\{\{A_1^{(1)}=1,\ldots,A_S^{(N_S)}=1\},\ldots,\{A_1^{(1)}=T,\ldots,A_S^{(N_S)}=T\}\big\},$$

In the collapsed bound we sum over all the possible assignments to each of the allocation variables:

$$\begin{aligned}
\log p(\mathscr{D}_{1:S}|\Theta) &= \log \sum\nolimits_{A_{1:S}} p(\mathscr{D}_{1:S}, A_{1:S}|\Theta) && (27)\\
&\geq \log \sum\nolimits_{A_{1:S}} \exp\big(\mathcal{L}(\mathscr{D}_{1:S}, A_{1:S};\Theta)\big) && (28)\\
&= \log \sum\nolimits_{A} \exp\Big(\mathfrak{B} + \sum\nolimits_s \sum\nolimits_n \sum\nolimits_t \mathbb{1}\{A_s^{(n)}=t\}\mathfrak{A}_{s,t,n}\Big) && (29)\\
&= \log\Big[\exp(\mathfrak{B}) \times \sum\nolimits_{A_1^{(1)}=1}^{T} \cdots \sum\nolimits_{A_S^{(N_S)}=1}^{T} \prod\nolimits_s \prod\nolimits_n \exp\Big(\sum\nolimits_t \mathbb{1}\{A_s^{(n)}=t\}\mathfrak{A}_{s,t,n}\Big)\Big] && (30)\\
&= \log\Big[\exp(\mathfrak{B}) \times \prod\nolimits_s \prod\nolimits_n \sum\nolimits_{A_s^{(n)}=1}^{T} \exp\Big(\sum\nolimits_t \mathbb{1}\{A_s^{(n)}=t\}\mathfrak{A}_{s,t,n}\Big)\Big] && (31)\\
&= \log\Big[\exp(\mathfrak{B}) \times \prod\nolimits_s \prod\nolimits_n \sum\nolimits_t \exp\left(\mathfrak{A}_{s,t,n}\right)\Big] && (32)\\
&= \mathfrak{B} + \sum\nolimits_s \sum\nolimits_n \log \sum\nolimits_t \exp \mathfrak{A}_{s,t,n} && (33)\\
&\triangleq \mathcal{L}(\mathscr{D}_{1:S};\Theta) && (34)
\end{aligned}$$

A.7 Benchmark

The benchmark kernel smoother optimises the leave-one-out training objective:

$$\Sigma_s^* = \underset{\Sigma}{\operatorname{argmax}} \sum\nolimits_{i=1}^{N_s} \log \sum\nolimits_{j\neq i=1}^{N_s} \mathcal{N}_{\mathcal{X}}(\mathbf{x}^{(s,i)};\mathbf{x}^{(s,j)},\Sigma). \quad (35)$$

We can construct the test log-likelihood for the held-out datasets as:

$$\log p(\mathscr{H}_{1:S}|\mathscr{D}_{1:S},\Sigma_{1:S}^*) = \sum_{s=1}^{S}\sum_{n=1}^{\tilde{N}_h} \log \sum_{t=1}^{T} a_{s,t}\; b_{t,m(h,n)} - |\Delta\mathbf{x}| \sum_{s=1}^{S}\sum_{t=1}^{T}\sum_{b=1}^{B} a_{s,t} b_{t,b}$$

where $m(h,n)$ is a function that maps a test data point $\tilde{\mathbf{x}}^{(h,n)}$ into the d^{th} grid-cell. For the CT case the weight matrix $\mathbf{A}$ is optimised for the test data.

A.8 Mixed Continuous Discrete Co-ordinate Spaces

This Ψ-function in the mixed co-ordinate space case is $\Psi(z_r, z_r') = \sum_{x_r} K(z_r, x_r)K(x_r, z_r)$. When using Kronecker structure $\Psi_{z_2 z_2}$ is simply $\mathbf{K}_{z_2 z_2}\mathbf{K}_{z_2 z_2}$ if $\mathcal{Z}$ contains all feeding station locations and the discrete dimension is $r = 2$.

A.9 Adapting LPPA to Model Dynamic Interaction Networks

LPPA can be used to model dynamic pair-wise interactions between V nodes, where is each sender i and receiver j is associated with a set of observations $\{\mathcal{D}_{i,j}\}_{i,j=1}^{V}$ and a rate functions $\lambda_{i,j}$. A straight forward approach is a triple factorisation typical of network models (Schmidt and Morup, 2013). Each rate function is constructed as $\lambda_{i,j} = \sum_{v=1}^{C}\sum_{w=1}^{C} \Omega_{i,v} f_{v,w}^2 \Omega_{j,w}$, where C is the number of "communities".

To modify LPPA we simply need to map $\mathcal{D}_{i,j}$ and $\lambda_{i,j}$ to $\mathcal{D}_s$ and λ_s, to map $f_{v,w}^2$ to f_t^2 and compute $\gamma_{s,t}$ from $\Omega_{i,v}$ and $\Omega_{j,w}$. These mappings will be different depending on whether we wish to model a symmetric network with $\mathcal{D}_{i,j} = \mathcal{D}_{j,i}$, and/or a network in which reflexive interaction is by definition empty, i.e. $\mathcal{D}_{i,i} = \emptyset$, thus making no contribution to the likelihood.

Since the cost of this algorithm increases quadratically as $P = C^2$, we might also consider a simpler model in which only intra-community interaction is allowed. In this case we may model the rate function as $\lambda_{i,j} = \sum_t \Omega_{i,t} f_t^2 \Omega_{j,t}$ for symmetric networks, or $\lambda_{i,j} = \sum_t \Omega_{i,t} f_t^2 \Upsilon_{j,t}$ where asymmetry is introduced via a third factor Υ.