Estimating State and Parameters in State Space Models of Spike Trains*

Jakob H. Macke[1,2], Lars Buesing[2,3], and Maneesh Sahani[3]

[1]Max Planck Institute for Biological Cybernetics and Bernstein Center for Computational Neuroscience, Tübingen, Germany;
[2]Gatsby Computational Neuroscience Unit, London, UK
[3]Statistics Department and Center for Theoretical Neuroscience, Grossman Center for the Statistics of Mind, Columbia University, New York, NY

January 16, 2015

1 Introduction

1.1 State space models for neural population spike trains

Neural computations at all scales of evolutionary and behavioural complexity are carried out by recurrently connected networks of neurons that communicate with each other, with neurons elsewhere in the brain, and with muscles through the firing of action potentials or "spikes". To understand how nervous tissue computes, it is therefore necessary to understand how the spiking of neurons is shaped both by inputs to the network and by the recurrent action of existing network activity. Whereas most historical spike data were collected one neuron at a time, new techniques including silicon multi-electrode array recording and scanning 2-photon, light-sheet or light-field fluorescence calcium imaging increasingly make it possible to record spikes from dozens, hundreds and potentially thousands of individual neurons simultaneously. These new data offer unprecedented empirical access to network computation, promising breakthroughs both in our understanding of neural coding and computation (Stevenson & Kording 2011), and our ability to build prosthetic neural interfaces (Santhanam, Ryu, Yu, Afshar & Shenoy 2006). Fulfilment of this promise will require powerful methods for data modelling and analysis, able to capture

*The final version of this draft will appear as a chapter in the book "Advanced State Space Methods for Neural and Clinical Data", Cambridge University Press, Cambridge, UK", edited by Zhe Chen.

the structure of statistical dependence of network activity across neurons and time.

Probabilistic latent state space models are particularly well-suited to this task. Neural activity often appears stochastic, in that repeated trials under the same controlled experimental conditions can evoke quite different patterns of firing. Some part of this variation may reflect differences in the way the computation unfolds on each trial. Another part might reflect noisy creation and transmission of neural signals. Yet more may come from chaotic amplification of small perturbations. As computational signals are thought to be distributed across the population (in a "population code"), variation in the computation may be distinguished by its common impact on different neurons and the systematic evolution of these common effects in time.

A state space model is able to capture such structured variation through the evolution of its latent state trajectory. This latent state provides a summary description of all factors modulating neural activity that are not observed directly. These factors could include processes such as arousal, attention, cortical state (Harris & Thiele 2011) or behavioural states of the animal (Niell & Stryker 2010, Maimon 2011). Alternatively, the latent state might be viewed as a summary description of the 'common inputs' received by the observed neurons from the unobserved parts of the network and other sources (Kulkarni & Paninski 2007, Vidne, Ahmadian, Shlens, Pillow, Kulkarni, Litke, Chichilnisky, Simoncelli & Paninski 2011, Archer, Koester, Pillow & Macke 2015). More practically, it provides a lower dimensional representation of the (possibly single-trial) activity of many neurons, which may be valuable for visualisation or to link to observed behaviour (Churchland, Yu, Sahani & Shenoy 2007, Cunningham & Yu 2014).

In this chapter, we focus on discrete-time state space models of neural spike trains in which the dynamics of the latent state has a very simple structure: The transition process is linear (i.e. the expected state at time $t+1$ is a linear function of the state at time t) and the innovation noise is Gaussian (i.e. the variability of the actual state around this mean follows a normal distribution). Such models have long been studied for tracking, control and other engineering applications. If the observed variables also depend linearly on the state, and any noise corrupting them is also Gaussian, then it gives rise to the statistical model that underlies the Kalman filter (Kalman & Bucy 1961). We will refer to this Gaussian-observation Linear Dynamical System model as *GLDS*. Many algorithms have been developed to identify (or "learn") the parameters of a GLDS model from data, either by using subspace identification (Katayama 2005) or different variants of expectation maximisation (Ghahramani & Hinton 1996).

However, neural spike trains cannot properly be modelled as linear functions of a Gaussian-distributed state with Gaussian noise. Spikes are all-or-nothing events. They are thus correctly described by point processes (Kass, Eden & Brown 2014) or, in discrete time, by binary- or integer-valued stochastic processes. The average spike rate must be non-negative, which is incompatible with a simple linear projection of a Gaussian-distributed state. Furthermore, if data are binned at a timescale likely to characterise neural computation, then

individual bins rarely contain more than a handful of spikes, making the Gaussian model of variability unrealistic. For example, suppose that spikes from neuron firing at 10 spikes/s, were to be binned with 10 ms precision. The expected spike count would then be 0.1 spikes per bin. If the counts were Poisson-distributed, their variance would also be 0.1, resulting in a standard deviation slightly greater than 0.3. This means that a spike count of 1 or more (occuring about every 10th bin) would fall about three standard deviations away from the mean. Under a Gaussian distribution such large-deviation events would occur with probability 0.0013; almost two orders of magnitude more rarely than their Poisson frequency.

Thus, realistic models of neural population activity must take into account the sparse, discrete nature of neural spiking. Although we might retain the linear-Gaussian dynamics of classical state space analysis for now, we depend on algorithms able to handle integer-valued observation processes. In particular, our focus will be on models in which the observations conditioned on the latent state are Poisson distributed. We refer to this class of models as *PLDS* (Macke, Buesing, Cunningham, Yu, Shenoy & Sahani 2012).

1.2 PLDS models in context

The PLDS models fall into the class of hidden Markov models with continuous state variables, and when the expected spike count is an exponential function of the latent state are also special cases of the log-Gaussian Cox Process (Møller, Syversveen & Waagepetersen 1998). Many methods have been developed to estimate state and parameters in state space models (Durbin, Koopman & Atkinson 2001, Doucet & Johansen 2009, Chen & Brown 2013). In the context of neural spike data such models were discussed by Smith & Brown (2003), who proposed an approximate Expectation Maximisation algorithm (Dempster, Laird & Rubin 1977) based on the forward-backwards algorithm, using a Gaussian approximation to the likelihood at each step. Other methods based on the forward-backward algorithm for inference have since been generalised and improved in various ways (Eden, Frank, Barbieri, Solo & Brown 2004, Yu, Afshar, Santhanam, Ryu, Shenoy & Sahani 2006, Kulkarni & Paninski 2007, Lawhern, Wu, Hatsopoulos & Paninski 2010). An algorithm for estimating models of high-dimensional 'common input' and an application to large-scale recordings of retinal ganglion cell activity was given by Vidne et al. (2011). In this approach, the estimation of the parameters of the common-input model is based on a moment transformation approach similar to the one used in the subspace identification method described below (Buesing, Macke & Sahani 2013).

Paninski, Ahmadian, Ferreira, Koyama, Rahnama Rad, Vidne, Vogelstein & Wu (2010) provide an overview of applications of state space models in neuroscience. They point out that direct optimisation of a Gaussian ("Laplace") approximation to the posterior distribution over states (as we will use below) is computationally competitive with or even superior to forward-backward algorithms, and use this approach to construct common input models of neural spike trains (see also Yu, Cunningham, Shenoy & Sahani 2008). Our descrip-

tion of inference and parameter estimation algorithms using a global Laplace approximation is based on that of Macke et al. (2012). More recently, variational methods for state space models (Beal 2003, Emtiyaz Khan, Aravkin, Friedlander & Seeger 2013, Buesing, Machado, Cunningham & Paninski 2014) have also been applied for both state inference and parameter learning in models of neural spike trains (Mangion, Yuan, Kadirkamanathan, Niranjan & Sanguinetti 2011).

All of the above approaches rely on deterministic, Gaussian approximations to the distribution over latent states. While this distribution is uni-modal, it is non-Gaussian which—at least in principle—could lead to bias in the estimation procedure (Turner & Sahani 2011). Sampling-based or Monte Carlo methods (Yuan, Girolami & Niranjan 2012) do not require parametric assumptions on the posterior distribution and are therefore more flexible, but, as their accuracy depends on the number of samples used, can prove to be significantly more computationally expensive. For overview articles on state space models, and application to neural population spike trains see (Chen 2003, Chen & Brown 2013).

2 State space models with linear dynamics and count-process observations

We consider models for spike data recorded simultaneously from q neurons, and discretised in time to yield spike counts $y_{i,t}$ for each neuron $i \in \{1 \dots q\}$ and time bin $t \in \{1 \dots T\}$. We concatenate the observations at time t into the q-dimensional vector $\mathbf{y}_t$ and denote by $\mathbf{y}_{1:T}$ the $q \times T$ matrix of all observations. In the limit of small bin sizes such a model would approach a multivariate point process (Kass et al. 2014). However, here we focus on the discrete-time description and interpretation. It is straightforward to extend the model to capture multiple experimental trials by modelling them as independent, identically distributed (i.i.d.) draws from the same model. This case of multiple trials differs only when estimating model parameters from data and will be discussed below.

It is helpful to introduce intermediate variables $z_{i,t}$ for each neuron i and time bin t. These variables capture the dependence of the spike rate on three factors: i) a parameter d_i that controls the overall mean firing rate of the ith neuron, ii) the current influence of unobserved processes, summarized by a p-dimensional state vector $\mathbf{x}_t$ and iii) any (observed) external covariates $\mathbf{s}_t$. These three factors combine linearly to form a q-dimensional vector $\mathbf{z}_t$ with elements $z_{i,t}$, which we will call the pre-intensity:

$$\mathbf{z}_t = C\mathbf{x}_t + D\mathbf{s}_t + \mathbf{d} \tag{1}$$

The $q \times p$ loading matrix C determines how each neuron is influenced by the latent state $\mathbf{x}_t$, with each row of C containing the couplings of one neuron to the p latent states. The term $\mathbf{s}_t$ can be used to model stimulus drive or the effect of other external variables. It is also often used to describe the influence of spiking history, by using a vector of all relevant recent spiking in the population

(Chornoboy, Schramm & Karr 1988, Eden et al. 2004, Pillow, Shlens, Paninski, Sher, Litke, Chichilnisky & Simoncelli 2008, Truccolo, Hochberg & Donoghue 2010). For example, one choice to model spike refractoriness would be to set $\mathbf{s}_t$ to the counts in the previous time bin $\mathbf{s}_t = \mathbf{y}_{t-1}$, and D to a diagonal matrix of size $q \times q$ with negative entries.

We assume that, given the pre-intensity $z_{i,t}$, the spike-count of neuron i in bin t is a sample from a Poisson distribution with mean $\eta(z_{i,t})$, where $\eta(\cdot)$ is a nonlinear function (which we will call the *spike-rate nonlinearity*) whose range is non-negative. Popular choices for $\eta(\cdot)$ include the exponential function $\eta(\cdot) = \exp(\cdot)$ and the *soft-threshold function* $\eta(\cdot) = \log(1 + \exp(\cdot))$. Thus, the conditional distribution of each count $y_{i,t}$ given the corresponding pre-intensity $z_{i,t}$ is:

$$P(y_{i,t}|z_{i,t}) = \frac{1}{y_{i,t}!}\eta(z_{i,t})^{y_{i,t}} \exp(-\eta(z_{i,t})). \tag{2}$$

The population state $\mathbf{x}_t$ evolves according to linear Gaussian dynamics with an external driving input $\mathbf{u}_t$:

$$\begin{aligned} \mathbf{x}_1 &\sim \mathcal{N}(\mathbf{x}_0, Q_0) \\ \mathbf{x}_t|\mathbf{x}_{t-1} &\sim \mathcal{N}(A\mathbf{x}_{t-1} + B\mathbf{u}_t, Q). \end{aligned} \tag{3}$$

Here, the parameters $\mathbf{x}_0$ and Q_0 denote the expected value and the covariance of the initial state $\mathbf{x}_1$ of each trial. The $p \times p$ matrix A (the *dynamics* matrix) specifies the deterministic component of the evolution from one state to the next, and the matrix Q is the covariance matrix of the innovations that perturb the latent state at each time step. We will assume here that the dynamics of the system are stable (i.e. that the spectral radius of the dynamics matrix A is less than one, meaning that all of its eigenvalues fall within the unit disc), and we denote by Π the asymptotic covariance $\lim_{t\to\infty} \mathrm{Cov}[\mathbf{x}_t]$. This covariance satisfies the relation $\Pi = A\Pi A^\top + Q$, and can thus be obtained from A and Q by solving the discrete-time Lyaponov equation (Buesing, Macke & Sahani 2012).

The vector $\mathbf{u}_t$ and the matrix B allow the model to capture any dependence of the latent state on external covariates. For example, $\mathbf{u}_t$ could model the influence of a stimulus that is believed to modulate neural firing in all neurons in the population. Alternatively, $\mathbf{u}_t$ might be an indicator function that 'measures' the current time in the trial, i.e. $\mathbf{u}_t$ would be a vector of zeros except for the t-th entry, which would be 1. This formulation would allow the model to capture structured variation in the mean firing rates of the population through the evolution of the latent state. Time-varying mean firing rates are typically described by a separate *peri-stimulus time histogram* (PSTH) for each neuron, which would require $q \times T$ parameters to be estimated for each stimulus. In the state-space approach time-varying means are captured by the driving inputs into the latent state, and so only $p \times T$ parameters are needed to describe all the PSTHs (Macke et al. 2012).

As the pre-intensities are also random variables (through both their dependence on $\mathbf{x}$ and possible dependence on previous spike events), the total variance

of each neuron's spiking will be a combination of the Poisson variance and the variance resulting from stochasticity in $\mathbf{z}$. Consequently, spike-counts sampled from the model will be over-dispersed within individual bins, i.e. their variance will be larger than the mean, giving a Fano-Factor (i.e. variance divided by mean) greater than one. However, interactions between bins could, at least in principle, lead to under-dispersion of total spike-counts when summed over multiple bins.

For simplicity, we will assume an exponential spike-rate nonlinearity, i.e. $\eta(.) = \exp(.)$, throughout the remainder of the chapter. This implies that the linear interactions in the latent space result in *multiplicative* interactions in the observed firing rates. Two recent studies have argued that internal states have a primarily multiplicative effect on firing rates (Ecker, Berens, Cotton, Subramaniyan, Denfield, Cadwell, Smirnakis, Bethge & Tolias 2014, Goris, Movshon & Simoncelli 2014), and suggested that state space models which allow for multiplicative interactions would be an direction for future research. Models based on exponential nonlinearities—and, in particular, the PLDS models described in this chapter—lead to such interactions quite naturally. We will also neglect the dependence on external covariates $\mathbf{u}_t$ and $\mathbf{s}_t$ from now on, again to simplify the exposition. [1]

3 Reconstructing the state from neural spike trains

We first consider the problem of *state estimation*. Given the population data $\mathbf{y}_{1:T}$, and the parameters of a state space model, how can we reconstruct the unobserved sequence of states $\mathbf{x}_{1:T}$? State estimation problems are often phrased as the problem of finding the 'most likely' state sequence. Here, we are not only interested in finding the most likely sequence, but also in quantifying uncertainty over estimated states. Hence we seek to characterise as far as possible the full posterior distribution of latent states given the observations $P(\mathbf{x}_{1:T}|\mathbf{y}_{1:T})$.

For state estimation, it is convenient to concatenate the columns of $\mathbf{x}_{1:T}$ to form a $pT \times 1$ vector which we denote by $\mathbf{x}$, i.e. $\mathbf{x} = \mathrm{vec}\,(\mathbf{x}_{1:T}) = (\mathbf{x}_1^\top \ \cdots \ \mathbf{x}_T^\top)^\top$ and similarly $\mathbf{y} = \mathrm{vec}\,(\mathbf{y}_{1:T})$. For the PLDS model, the posterior distribution $P(\mathbf{x}|\mathbf{y})$ does not correspond to a well-studied standard distribution (such as a Gaussian). In particular we do not have closed forms for the expected values under this distribution that will be needed for parameter estimation and model comparison. One way around this difficulty is to approximate $P(\mathbf{x}|\mathbf{y})$ with an appropriate and convenient distribution $q(\mathbf{x})$. As, at least for the exponential spike-rate nonlinearity assumed here, the PLDS posterior $P(\mathbf{x}|\mathbf{y})$ has a single peak (Paninski 2004), we choose to focus on Gaussian approximating forms. That is, we take $q(\mathbf{x}) = q(\mathbf{x}|\mu, \Sigma) = \mathcal{N}(\mu, \Sigma)$ with mean μ and covariance Σ. It will be convenient to refer to the corresponding normal density function by the

[1] However, the accompanying computer implementation is able to handle external covariates.

symbol $\phi(\mathbf{x}|\mu, \Sigma)$:

$$\phi(\mathbf{x}|\mu, \Sigma) = (2\pi)^{-pT/2}|\Sigma|^{-1/2} \exp\left(-\tfrac{1}{2}(\mathbf{x}-\mu)^\top \Sigma^{-1}(\mathbf{x}-\mu)\right). \tag{4}$$

In fact, we will consider two different Gaussian approximations, which select μ and Σ in two different ways. In the first method, μ is set equal to the mode of the distribution, i.e. the latent state sequence with the highest posterior probability, and Σ is determined by the local curvature of the log-posterior at this point. The second method (Emtiyaz Khan et al. 2013) focusses on the Gaussian approximation which has minimal distance to the true posterior, as measured by Kullback-Leibler Divergence (Bishop 2006).

3.1 Inferring the state using the Laplace approximation

Inspection of equation 4 reveals that the log-density of a Gaussian distribution is quadratic in the observation $\mathbf{x}$, from which two properties follow. First, the log-density is highest at the unique point $x = \mu$ and so, conversely, the mean μ can be found by evaluating the mode $\mu = \text{argmax}_{\mathbf{x}} \log \phi(\mathbf{x}|\mu, \Sigma)$. Second, the Hessian matrix of second derivatives of the log-density yields the inverse covariance matrix $\Sigma^{-1} = -\nabla_x^2 \log \phi(\mathbf{x}|\mu, \Sigma)$.

In general, neither property holds for non-Gaussian distributions. However, provided that the distribution is unimodal and smooth, these observations provide a heuristic method for approximating a non-Gaussian distribution by a Gaussian one. [2] This is the approach of the Laplace approximation to the PLDS posterior. The mean of the approximating Gaussian is set to the mode of the log-posterior, and its covariance matrix to the negative inverse Hessian evaluated at that mode (as the value of the Hessian may vary with location for non-Gaussian distributions it is necessary to specify the point at which the derivatives are evaluated). Thus, to find the Laplace approximate Gaussian we need to i) write down the log-posterior over $\mathbf{x}$, ii) find its optimum (typically exploiting gradient information) and iii) find its Hessian. Computation of the Hessian also allows us to use second-order methods during numerical optimisation.

By Bayes' rule, the log-posterior is given by:

$$\log P(\mathbf{x}|\mathbf{y}) = \log P(\mathbf{y}|\mathbf{x}) + \log P(\mathbf{x}) + \text{const}(\mathbf{x}) \tag{5}$$

where $\text{const}(\mathbf{x})$ denotes terms that do not depend on $\mathbf{x}$. The PLDS log-likelihood

[2] The approximation can also be derived by truncating the series expansion of the log-posterior about its mode to second order; and the Bernstein-von Mises theorem implies that, under suitable regularity conditions and in the limit of large data set sizes, the higher-order terms become insignificant and the posterior distribution in fact approaches this Gaussian form.

is:

$$\begin{aligned}
\log P(\mathbf{y}|\mathbf{x}) &= \sum_{t=1}^{T}\sum_{i=1}^{q} \log P(y_{i,t}|\mathbf{x}_t) \\
&= \sum_{t=1}^{T}\sum_{i=1}^{q} \log\left(\tfrac{1}{y_{i,t}!}\exp(z_{i,t})^{y_{i,t}}\exp(-\exp(z_{i,t}))\right) \qquad \text{using (2)} \\
&= \sum_{t=1}^{T}\sum_{i=1}^{q} y_{i,t}z_{i,t} - \exp(z_{i,t}) + \text{const}(\mathbf{x}) \\
&= \sum_{t=1}^{T}\sum_{i=1}^{q} y_{i,t}\left((C\mathbf{x}_t)_i + d_i\right) - \exp\left((C\mathbf{x}_t)_i + d_i\right) + \text{const}(\mathbf{x}) \\
&= \mathbf{y}^\top\left(\tilde{C}\mathbf{x} + \tilde{\mathbf{d}}\right) - \mathbf{e}_{qT}^\top \exp\left(\tilde{C}\mathbf{x} + \tilde{\mathbf{d}}\right) + \text{const}(\mathbf{x})\,. \qquad (6)
\end{aligned}$$

Here, the matrix $\tilde{C}$ is of size $qT \times pT$, and consists of T copies of C along its block-diagonal, and zeros elsewhere, i.e. $\tilde{C} = \mathbb{I}_T \otimes C$ where $\mathbb{I}_T$ denotes the identity matrix of size $T \times T$ and $\otimes$ denotes the Kronecker product. The column vector $\mathbf{e}_{qT}$ is of size $qT \times 1$ with all elements equal to 1, and $\tilde{\mathbf{d}}$ is of size $qT \times 1$ and consists of T copies of $\mathbf{d}$ stacked one atop the other. Finally, the exponential function in the last line is applied element-by-element.

The prior over $\mathbf{x}$ is defined by the dynamics equation (3):

$$\begin{aligned}
\log P(\mathbf{x}) &= \log\phi\left(\mathbf{x}_1|\mathbf{x}_0, Q_0\right) + \sum_{t=2}^{T}\log\phi\left(\mathbf{x}_1|\mathbf{x}_{t-1}, Q\right) \\
&= -\tfrac{1}{2}\left(\mathbf{x}_1 - \mathbf{x}_0\right)^\top Q_0^{-1}\left(\mathbf{x}_1 - \mathbf{x}_0\right) \\
&\quad - \tfrac{1}{2}\sum_{t=2}^{T}\left(\mathbf{x}_t - A\mathbf{x}_{t-1}\right)^\top Q^{-1}\left(\mathbf{x}_t - A\mathbf{x}_{t-1}\right) + \text{const}(\mathbf{x}). \qquad (7)
\end{aligned}$$

Inspection of this equation reveals that $\mathbf{x}_1$ is given by a Gaussian distribution, and each subsequent $\mathbf{x}_t$ depends linearly on its predecessor. Therefore, the joint distribution over all $\mathbf{x}_t$, and equivalently the distribution over $\mathbf{x}$, is a multivariate normal. By collecting the terms which are linear and quadratic in $\mathbf{x}$, one can see that the prior mean μ_π and covariance Σ_π are given by (see Paninski et al. 2010,

for further details and explanation) :

$$\Sigma_\pi^{-1} = \begin{pmatrix} Q_0^{-1} + A^\top Q^{-1} A & -A^\top Q^{-1} & \dots & 0 \\ -Q^{-1}A & A^\top Q^{-1} A + Q^{-1} & -A^\top Q^{-1} & \vdots \\ \vdots & -Q^{-1}A & A^\top Q^{-1} A + Q^{-1} & -A^\top Q^{-1} \\ 0 & \dots & -Q^{-1}A & Q^{-1} \end{pmatrix} \tag{8}$$

$$\mu_\pi = \begin{pmatrix} x_0 \\ Ax_0 \\ A^2 x_0 \\ \vdots \\ A^{T-1} x_0 \end{pmatrix}. \tag{9}$$

Thus, the log-posterior $\log P(\mathbf{x}|\mathbf{y})$ of $\mathbf{x}$ given $\mathbf{y}$ (which we denote $L(\mathbf{x})$) is given by (neglecting the additive constant terms)

$$L(\mathbf{x}) := \mathbf{y}^\top \left(\tilde{C}\mathbf{x} + \tilde{\mathbf{d}}\right) - \mathbf{e}_{qT}^\top \exp\left(\tilde{C}\mathbf{x} + \tilde{\mathbf{d}}\right) - \tfrac{1}{2}(\mathbf{x} - \mu_\pi)^\top \Sigma_\pi^{-1}(\mathbf{x} - \mu_\pi). \tag{10}$$

Taking derivatives with respect to $\mathbf{x}$, we obtain the gradient and Hessian:

$$\nabla_{\mathbf{x}} L = \tilde{C}^\top \mathbf{y} - \tilde{C}^\top \exp\left(\tilde{C}\mathbf{x} + \tilde{\mathbf{d}}\right) - \Sigma_\pi^{-1}(\mathbf{x} - \mu_\pi) \tag{11}$$

$$\nabla_{\mathbf{x}}^2 L = -\tilde{C}^\top \mathrm{diag}\left(\exp(\tilde{C}\mathbf{x} + \tilde{\mathbf{d}})\right) \tilde{C} - \Sigma_\pi^{-1} \tag{12}$$

Using this gradient (and possibly also the Hessian) we are able to numerically maximise the log-posterior with respect to $\mathbf{x}$. This optimisation problem has nice computational properties. In fact, it is equivalent to performing maximum-a-posteriori decoding in generalised linear models with a Gaussian prior (in which case $\mathbf{x}$ would be the stimulus being estimated), and its properties have therefore been well studied. In particular, it is known that the log-posterior is a concave function (as can also be noted by observing that the Hessian is a negative-definite matrix), so it can be optimised without risk of converging to local maxima (Boyd & Vandenberghe 2004).

Finally, because of the sparse and blocked structure of $\tilde{C}$ and Σ_π^{-1}, both the gradient and the Hessian can be calculated efficiently without having to store or invert dense matrices of size $pT \times pT$. As has been pointed out by Paninski et al. (2010), this approach of performing a Laplace approximation on the entire time-series $\mathbf{x}_{1:T}$ ('global Laplace') is therefore as computationally efficient as following the classical approach of using forward-backward message passing to calculate the posterior mode. The full posterior matrix Σ is a dense $pT \times pT$ matrix, so calculating it would be computationally expensive for large pT. However, one rarely requires the full posterior covariance, and in many cases (e.g. for parameter estimation, see below) it is sufficient to calculate just the diagonal blocks Σ_t and immediate off-diagonal blocks $\Sigma_{t,t-1}$, which can be achieved efficiently without explicitly calculating the full Σ.

3.2 Inferring the state distribution using Gaussian variational inference

While the Laplace approximation is often easy to use and efficient to calculate, it has been demonstrated in a variety of settings that alternative approximation schemes provide more accurate approximations to posterior distributions (Nickisch & Rasmussen 2008). One alternative to the Laplace method is known as the variational inference approach. In variational inference, one approximates the posterior $P(\mathbf{x}|\mathbf{y})$ with a distribution $q(\mathbf{x})$ that comes from a specified family of distributions. In the following we will restrict our discussion to Gaussian variational inference, i.e. we restrict $q(\mathbf{x})$ to be a normal distribution, whose mean and covariance are so-called variational parameters and which we denote with μ and Σ. Hence, by construction, the functional form of our posterior approximation will be identical to the one found via Laplace method but, as pointed out by Nickisch & Rasmussen (2008) and Opper & Archambeau (2009), the parameters μ and Σ of the two approximations will generally differ. The variational parameters are found by determining the best Gaussian approximation to the posterior as measured by the Kullback-Leibler divergence between the two distributions:

$$\mu^*, \Sigma^* \quad = \quad \mathrm{argmin}_{\mu,\Sigma} \; \mathrm{D}_{\mathrm{KL}}[q(\mathbf{x}) \| P(\mathbf{x}|\mathbf{y})], \tag{13}$$

where $\mathrm{D}_{\mathrm{KL}}[q(x)\|p(x)] = \int \log(q(x)/p(x))q(x)dx$. The minimisation (13) is equivalent to maximising a lower bound $\mathcal{L}(\mu, \Sigma)$ of the log probability $\log p(\mathbf{y})$ of the data under the model. For the PLDS model considered here, the variational lower bound $\mathcal{L}(\mu, \Sigma) \leq \log p(\mathbf{y})$ reads (up to additive constants)

$$\begin{aligned}\mathcal{L}(\mu, \Sigma) =& \tfrac{1}{2}\left(\log|\Sigma| - \mathrm{tr}[\Sigma_\pi^{-1}\Sigma] - (\mu - \mu_\pi)^\top \Sigma_\pi^{-1}(\mu - \mu_\pi)\right) \\ &+ \mathrm{E}_q[\log P(\mathbf{y}|\mathbf{x})],\end{aligned} \tag{14}$$

where $\mathrm{E}_q[\log P(\mathbf{y}|\mathbf{x})]$ denotes the expectation of the log likelihood under the approximate posterior $q(\mathbf{x})$. In general, this expectation cannot be computed analytically as a function of the variational parameters μ, Σ. However, in our model which has Poisson observations with the canonical link function (i.e. exponential spike-rate nonlinearity) and where the expectation is computed under a Gaussian approximation $q(\mathbf{x})$, it is available in closed form (up to additive constants):

$$\mathrm{E}_q[\log P(\mathbf{y}|\mathbf{x})] = \mathbf{y}^\top \left(\tilde{C}\mu + \tilde{\mathbf{d}}\right) - \mathbf{e}_{qT}^\top \exp(\tilde{C}\mu + \tilde{\mathbf{d}} + \tfrac{1}{2}\,\mathrm{diag}(\tilde{C}\Sigma\tilde{C}^\top)).$$

Given the above expression, one can in principle directly optimise $\mathcal{L}(\mu, \Sigma)$ to obtain the approximation $q(\mathbf{x}) = \phi(\mu^*, \Sigma^*)$ to the posterior $P(\mathbf{x}|\mathbf{y})$. In practice however, this optimisation can be computationally very expensive for two reasons. First, the posterior covariance Σ has $\mathcal{O}(p^2T^2)$ entries, and hence for large T it is an optimisation over a large number of variables. Second, as Σ is a covariance matrix, it needs to be constrained to be a symmetric, semi-definite matrix. There are well-established tools for solving such so-called

semi-definite programs (see Boyd & Vandenberghe 2004), but this constraint further increases the computational cost. Fortunately however, it has been shown (Opper & Archambeau 2009) that this optimisation problem has further structure that can be exploited to speed up the computation. The inverse of the optimal posterior covariance Σ^{-1} (the posterior *precision*) is given by sum of the prior precision Σ_π^{-1} and a simple correction term,

$$\Sigma^{-1}(\lambda) = \Sigma_\pi^{-1} + \tilde{C}\,\mathrm{diag}(\lambda)\tilde{C}^\top, \tag{15}$$

where λ is a vector of dimension qT (i.e. one element per observed spike count) with non-negative entries. Furthermore, the posterior mean can be expressed using the same λ as

$$\mu(\lambda) = \mu_\pi - \Sigma_\pi \tilde{C}^\top(\lambda - \mathbf{y}). \tag{16}$$

One straight-forward way to exploit this special structure would be to plug (16) and (15) into (14) to express $\mathcal{L}$ as a function of λ and optimise $\mathcal{L}(\lambda)$ over the positive orthant (as all elements of λ are non-negative). Unfortunately, this optimisation is non-convex and converges very slowly, as pointed out by Emtiyaz Khan et al. (2013). However, these authors also showed that the special form of the optimal variational parameters (16) and (15) can be used more efficiently by converting the original (primal) optimisation of $\mathcal{L}(\mu, \Sigma)$ over μ, Σ into a dual optimisation problem (Boyd & Vandenberghe 2004). In this dual problem, we have to minimise the objective function $D(\lambda)$ over $\lambda \geq 0$:

$$\lambda^* = \mathrm{argmin}_{\lambda \geq 0}\, D(\lambda) \tag{17}$$

$$\begin{aligned} D(\lambda) :=& \tfrac{1}{2}(\lambda - \mathbf{y})^\top \tilde{C}\Sigma_\pi \tilde{C}^\top(\lambda - \mathbf{y}) - (\tilde{C}\mu + \overline{\mathbf{d}})^\top(\lambda - \mathbf{y}) \\ &+ \tfrac{1}{2}\log|\Sigma(\lambda)| + \sum_{n=1}^{qT} \lambda_n(\log\lambda_n - 1), \end{aligned} \tag{18}$$

where $\Sigma(\lambda)$ is defined in (15). By contrast to the optimisation of (14) over $\mathcal{O}(qT)^2$ variables, the optimisation (17) is over only qT variables, it is strictly convex and converges quickly if solved by standard gradient-based optimisation techniques. Having solved (17) for the unique optimum λ^*, we can then compute the optimal posterior mean $\mu^* = \mu(\lambda^*)$ and covariance $\Sigma^* = \Sigma(\lambda^*)$ from (16) and (15).

To find the optimum of (17) we use a gradient-based pseudo-Newton method (lBFGS) (Boyd & Vandenberghe 2004). The gradient of $D(\lambda)$ is given by

$$\nabla_\lambda D(\lambda) = \quad \tilde{C}\Sigma_\pi\tilde{C}^\top(\lambda - \mathbf{y}) - \tilde{C}\mu_\pi - \overline{\mathbf{d}} + \log\lambda - \tfrac{1}{2}\,\mathrm{diag}(\tilde{C}\Sigma(\lambda)\tilde{C}^\top). \tag{19}$$

Naive evaluation of $D(\lambda)$ and $\nabla_\lambda D(\lambda)$ requires the computation of $\Sigma(\lambda)$, which according to (15) is an inversion of a $pT \times pT$ matrix, yielding a computational cost of $\mathcal{O}(p^3T^3)$ per optimisation iteration, which would be prohibitive for data sets with large T. However, using the Markovian structure of the LDS prior on $\mathbf{x}$ and properties of $\tilde{C}$, the cost $D(\lambda)$ and its gradient $\nabla_\lambda D(\lambda)$

(and hence each iteration of the optimisation) can be evaluated in $\mathcal{O}(p^3T)$ in the following way. As $\tilde{C}$ is block-diagonal, the term $\operatorname{diag}(\tilde{C}\Sigma(\lambda)\tilde{C}^\top)$ requires only the computation of the block-diagonal entries of $\Sigma(\lambda)$. Furthermore, we leverage the special structure of $\Sigma(\lambda)$: Equation (15) shows that the inverse of $\Sigma(\lambda)$ is the sum of the inverse LDS prior covariance matrix Σ_π^{-1} and the block-diagonal term $\tilde{C}\operatorname{diag}(\lambda)\tilde{C}^\top$. Hence, $\Sigma^{-1}(\lambda)$ is exactly equal to the posterior precision matrix of an "effective" LDS model with an observation model given by $P(\mathbf{y}_t|\mathbf{x}_t) = \phi(\mathbf{y}_t|C\mathbf{x}_t + \mathbf{d}, \operatorname{diag}(\lambda_t^{-1}))$. Therefore, numerical evaluation of the block-diagonal of $\Sigma(\lambda)$ is equivalent to determining the block-diagonal of the posterior covariance of the "effective" LDS model, for which we can use standard Kalman smoothing which requires $\mathcal{O}(p^3T)$ (Paninski et al. 2010, Buesing et al. 2014). Kalman smoothing also yields the term $\log|\Sigma(\lambda)|$ without additional cost. Finally, multiplications of a vector of dimension qT with $\tilde{C}$ and Σ_π can also be carried out in $\mathcal{O}(p^3T)$ as the former is block-diagonal and the latter has a banded inverse (given in (8)). This shows that each iteration for optimising the dual cost (18) requires only $\mathcal{O}(p^3T)$ time complexity. In general, we cannot hope to do better as this is the same complexity as a Kalman smoothing computation as well as the cost of computing the gradient and Hessian for the Laplace approximation.

4 Estimating model parameters

We have described above how to find an approximation to the distribution of the state trajectory given the observed data *and known parameters*. However, in many cases we have available only observed spike data from which we would like to characterise both the dynamical structure of the network activity—i.e. the parameters of the state space model—and the trajectories on all trials. Here, we describe an iterative method (Expectation Maximisation) for estimating the parameters from data, and a closed-form "spectral" algorithm which provides a good initialisation for Expectation Maximisation. In the following we denote the model parameters by $\theta = (A, Q, Q_0, \mathbf{x}_0, C, \mathbf{d})$. We now also consider multiple experimental recordings, modelled as independent, identically distributed trials indexed by $k = 1, \ldots, K$. Let the data and the state variables from the kth trial be denoted by $\mathbf{y}^k$ and $\mathbf{x}^k$ and, slightly overloading the notation from above, we now denote the data across all trials by $\mathbf{y}$.

4.1 Expectation Maximisation: Estimating parameters via iteratively optimising a cost function

Expectation Maximisation (EM) (Dempster et al. 1977) is a general method for estimating parameters of latent variable models, such as the state space model described above, from data. EM can be motivated from a standard maximum likelihood (ML) perspective. In ML estimation, we seek those model parameters θ that maximise the likelihood, and hence also the log-likelihood, of the data under the model $\log P(\mathbf{y}|\theta) = \log \int P(\mathbf{y}, \mathbf{x}|\theta)\, d\mathbf{x}$. For many latent

variable models however, integration of the likelihood over the latent variables does not have a closed form solution, so it is difficult to evaluate $P(\mathbf{y}|\theta)$ as a function of the model parameters. EM circumvents this problem by replacing direct maximisation of the likelihood, with maximisation of a lower bound, which we denote by $\mathcal{Q}$:

$$\mathcal{Q}_q(\theta) := E_q[\log P(\mathbf{x}, \mathbf{y}|\theta) - \log q(\mathbf{x})] \leq \log P(\mathbf{y}|\theta). \tag{20}$$

This bound is valid for any distribution $q(\mathbf{x})$, and is tight if and only if $q(\mathbf{x})$ is equal to the posterior over the state variables $P(\mathbf{x}|\mathbf{y}, \theta)$. "Vanilla" EM is a coordinate-ascent-style algorithm for maximising $\mathcal{Q}_q(\theta)$. Given initial parameters θ', one computes $q(\mathbf{x}) = P(\mathbf{x}|\mathbf{y}, \theta')$. Given $q(\mathbf{x})$, ones maximizes $\mathcal{Q}_q(\theta)$ with respect to the parameters θ. Iterating over these two steps is guaranteed to increase the lower bound $Q_q(\theta)$ and it converge to a (local) maximum of the likelihood. As we have seen, however, in the PLDS model it is not possible to evaluate the exact posterior distribution $P(\mathbf{x}|\mathbf{y}, \theta)$. Hence, we are compelled to approximate $q(\mathbf{x})$. Either the Laplace or the variational approximation can be used, and we refer to the resulting algorithms as lEM or vEM respectively. Note that, if q is constrained to be Gaussian, the bounds of (20) and (14) become identical. Thus, in vEM both the inference of q and the optimisation with respect to θ increase the same lower bound, and convergence is guaranteed (albeit not necessarily to a local maximum of the likelihood). By contrast, no similar convergence guarantee is available for lEM and in practice we regularly observed that the lower bound (20) actually decreases for lEM after a number of iterations.

Given a posterior approximation $q(\mathbf{x}) = \prod_{k=1}^{K} q(\mathbf{x}^k)$ with mean μ^k and covariance Σ^k on trial k from either Laplace or variational inference, the optimisation of $\mathcal{Q}_q(\theta)$ with respect to the parameters θ is reasonably straightforward. $\mathcal{Q}_q(\theta)$ can be written as (neglecting terms constant in θ):

$$\begin{aligned}
\mathcal{Q}_q(\theta) &= \sum_{k=1}^{K} \int q(\mathbf{x}^k) \log P(\mathbf{x}^k, \mathbf{y}^k|\theta) d\mathbf{x}^k && (21)\\
&= \sum_{k=1}^{K} \int q(\mathbf{x}^k) \log P(\mathbf{y}^k|\mathbf{x}^k, \theta) d\mathbf{x}^k + \int q(\mathbf{x}^k) \log P(\mathbf{x}^k|\theta) d\mathbf{x}^k \\
&=: \mathcal{Q}_q^{obs}(C, d) + \mathcal{Q}_q^{dyn}(A, Q, x_0, Q_0). && (22)
\end{aligned}$$

Thus this cost function decomposes into a sum of two terms, $\mathcal{Q}_q^{obs}$ and $\mathcal{Q}_q^{dyn}$ which only depend on the parameters of the observation model (C, d) or on the parameters of the latent dynamics (A, Q, x_0, Q_0) respectively, and which can therefore be optimised separately. As we have assumed linear dynamics, the cost function $\mathcal{Q}_q^{dyn}$ is the familiar term that arises with linear Gaussian state space models (Ghahramani & Roweis 1999), and all parameters can be updated

in closed form:

$$x_0 = \frac{1}{K}\sum_{k=1}^{K}\mu_1^k \tag{23}$$

$$Q_0 = \frac{1}{K}\sum_{k=1}^{K}\Sigma_{11}^k + \frac{1}{K}\sum_k (x_0-\mu_1^k)(x_0-\mu_1^k)^\top$$

$$A = \frac{1}{K(T-1)}\left(\sum_{k=1}^{K}\sum_{t=2}^{T} M_{t,t-1}^k\right)\left(\sum_{k=1}^{K}\sum_{t=2}^{T} M_{t-1,t-1}^k\right)^{-1}$$

$$Q = \frac{1}{K(T-1)}\sum_{k=1}^{K}\sum_{t=2}^{T}\left(M_{t,t}^k + AM_{t-1,t-1}^k A^\top - AM_{t-1,t}^k - M_{t,t-1}^k A^\top\right),$$

where we have used the shorthand $M_{t,s}^k = \Sigma_{t,s}^k + \mu_t^k \mu_s^{k\top}$, $\Sigma_{t,s}^k$ for the posterior covariance of $\mathbf{x}_t^k$ and $\mathbf{x}_s^k$, and μ_t^k for the posterior mean of $\mathbf{x}_t^k$.

The cost function of the observation model, $\mathcal{Q}_q^{obs}$, has to be optimised numerically. As the log-posterior $\log P(\mathbf{y}^k|\mathbf{x}^k,\theta)$ is concave in our model and integration over a Gaussian density preserves concavity (Boyd & Vandenberghe 2004), $\mathcal{Q}_q^{obs}$ is also a concave function, so its maximum can be found efficiently and without risk of getting stuck in a local maximum. $\mathcal{Q}_q^{obs}$ and its gradient with respect to C and d are given by:

$$\mathcal{Q}_q^{obs} = \sum_{k=1}^{K}\sum_{t=1}^{T}\mathbf{y}_t^{k\top}\left(C\mu_t^k + \mathbf{d}\right) - \mathbf{e}_q^\top \hat{\mathbf{y}}_t^k \tag{24}$$

$$\nabla_d \mathcal{Q}_q^{obs} = \sum_{k=1}^{K}\sum_{t=1}^{T}\mathbf{y}_t^k - \hat{\mathbf{y}}_t^k \tag{25}$$

$$\nabla_C \mathcal{Q}_q^{obs} = \sum_{k=1}^{K}\sum_{t=1}^{T}\mathbf{y}_t^k \mu_t^{k\top} - \operatorname{diag}(\hat{\mathbf{y}}_t^k)\left(\mathbf{e}_q \mu_t^{k\top} + C\Sigma_t^k\right) \tag{26}$$

$$\hat{\mathbf{y}}_t^k := \exp(C\mu_t^k + \mathbf{d} + \tfrac{1}{2}\operatorname{diag}(C\Sigma_t^k C^\top)). \tag{27}$$

In practice, updating $\mathbf{d}$ before C in each iteration leads to much better results than updating C first [3]. In fact, in the case of an exponential nonlinearity, the update in $\mathbf{d}$ also has a closed-form solution and is given by

$$\mathbf{d} = \log\left(\sum_{k,t}\mathbf{y}_t^k\right) - \log\left(\sum_{k,t}\exp\left(C\mu_t^k + \tfrac{1}{2}\operatorname{diag}(C\Sigma_t^k C^\top)\right)\right). \tag{28}$$

4.2 Learning parameters of the model through spectral learning

Although each step of both lEM and vEM has a single unique solution, the model likelihood and the lower bound (20) both have multiple local optima.

[3] C and d can also be updated jointly, as implemented in the accompanying computer code.

Thus both forms of EM (and, indeed, any other iterative maximum-likelihood method) will converge to a nearby local maximum that is selected by the particular initial values from which the iterations are begun. Indeed, with lEM there is no guarantee that even the lower bound increases at each step, implying that the algorithm may not converge, and convergence may be difficult to identify. Both algorithms are also computationally expensive, typically requiring multiple iterations, and each iteration of the E and M-step requires numerical optimisation as a subroutine.

In this section, we describe an alternative approach to estimating the parameters of the PLDS model. The approach is based on a technique known as *subspace identification* (SSID), also called *spectral learning* (Katayama 2005), which was originally developed for models with *linear* dynamics and *Gaussian linear* observations. It is based on finding an eigendecomposition of a matrix constructed from the observed data. This decomposition is typically fast to compute and there is no risk of getting stuck in local minima, which renders such methods both computationally efficient and robust. As we will show, these advantages can be carried over to the more general setting of dynamical system models with nonlinear and non-Gaussian observation processes. Here we provide a sketch of the algorithm as applied to the PLDS model; a full description can be found in (Buesing et al. 2012).

The basic intuition behind the approach is the following. To use standard, Gaussian subspace identification, we need the mean and time-lagged covariances of the observations from a linear Gaussian Dynamical System. As can be seen from the definition (2), in the PLDS model, any 'nonlinearity' and 'non-Gaussianity' is only present at the level of the observations $y^{i,t}$. These are generated by pushing the underlying pre-intensities $z_{i,t}$ (which are generated by a linear Gaussian dynamical system) element-wise through the nonlinear Poisson observation model. Although we cannot observe $\mathbf{z}$ directly (it is a latent variable), it turns out that we can nevertheless estimate its means and covariance matrix from the means and covariances of the observed spike-counts $\mathbf{y}$ by applying a fixed and deterministic nonlinear transformation. Having thus recovered the means and covariance of $\mathbf{z}$, the standard subspace identification methods apply and provide estimates of the unknown parameters θ of the PLDS model.

This nonlinear spectral learning algorithm provides a fast and robust algorithm for parameter estimation, but one which often provides less accurate parameter estimates than EM. Therefore, it is often useful to combine the two approaches, and to use spectral learning as a means of generating an initial setting of parameters and then to further improve on this initial value using EM.

4.2.1 Subspace-ID for LDS models with linear-Gaussian observations

We briefly review here the linear SSID algorithm that we will use: the Ho-Kalman SSID algorithm (Ho & Kalman 1966, Katayama 2005) for linear-Gaussian

LDS models. In this setting, the model for the dynamics on $\mathbf{x}$ is as before (Gaussian distribution over initial state, linear dynamics, Gaussian innovation noise), but the observation model is now linear with Gaussian noise with covariance R:

$$\mathbf{y}_t \mid \mathbf{z}_t \sim \mathcal{N}(\mathbf{z}_t, R). \tag{29}$$

Provided the generative model is stationary (i.e., $\mathbf{x}_0 = 0$ and $Q_0 = \Pi$), SSID algorithms yield consistent estimates of the parameters $A, C, Q, R, \mathbf{d}$ in a non-iterative way. Ho-Kalman SSID takes as input the empirical estimate of the (so-called) "future-past Hankel matrix" H which is defined as the cross-covariance between time-lagged vectors $\mathbf{y}_t^+$ (the "future") and $\mathbf{y}_t^-$ (the "past") of the observed data:

$$H := \mathrm{Cov}[\mathbf{y}_t^+, \mathbf{y}_t^-] \qquad \mathbf{y}_t^+ := \begin{pmatrix} \mathbf{y}_t \\ \vdots \\ \mathbf{y}_{t+k-1} \end{pmatrix} \qquad \mathbf{y}_t^- := \begin{pmatrix} \mathbf{y}_{t-1} \\ \vdots \\ \mathbf{y}_{t-k} \end{pmatrix}.$$

The parameter κ is called the Hankel size and has to be chosen so that $\kappa \geq p$. The key to SSID is that H has rank equal to the dimensionality p. As any dependence between past and future data is mediated by the latent state x, the covariance has to have the same rank as $\mathbf{x}_t$, i.e. rank p. Furthermore, the Hankel matrix can be decomposed in terms of the model parameters A, C, Π. For example, in the simplest case of $k = 1$,

$$H = \mathrm{Cov}[\mathbf{y}_t, \mathbf{y}_{t-1}] = C\mathrm{Cov}\left[\mathbf{x}_t, \mathbf{x}_{t-1}\right] C^\top = CA\mathrm{Cov}\left[\mathbf{x}_t, \mathbf{x}_{t-1}\right] C^\top \approx CA\Pi C^\top.$$

The SSID algorithm first takes the singular value decomposition (SVD) of the empirical estimate $\widehat{H}$ of H. The singular value spectrum of $\widehat{H}$ suggests a suitable value of p by inspection (e.g. take p to be the number of eigenvalues that are significantly greater than 0). From the corresponding low-rank approximation to $\widehat{H}$ the model parameters A, C as well as the covariances Q and R can be found by linear regression and by solving an algebraic Riccati equation; $\mathbf{d}$ is given simply by the empirical mean of the data.

4.2.2 Subspace-ID for the PLDS model by moment conversion

Consider now the PLDS in which the Gaussian observation process (29) is replaced by the PLDS observation model (2). In the PLDS model $y_{t,i} \perp y_{s,j} \mid \mathbf{z}_t$, i.e. all observations are independent given $\mathbf{z}_t$. Further, $y_{t,i} \mid \mathbf{z}_t$ is Poisson distributed with mean and variance $\mathrm{E}[\mathbf{y}_t|\mathbf{z}_t] = \mathrm{Var}[\mathbf{y}_t|\mathbf{z}_t] = \exp(\mathbf{z}_t)$.

We consider the covariance matrix $\mathrm{Cov}[\mathbf{y}^\pm]$ of the combined $2\kappa q$-dimensional future-past vector $\mathbf{y}^\pm = ({\mathbf{y}^+}^\top, {\mathbf{y}^-}^\top)^\top$ which is defined by stacking $\mathbf{y}^+$ and $\mathbf{y}^-$ (here and henceforth we drop the subscripts t as it is superfluous given the assumed stationarity of the process). We denote the mean and covariance matrix of the normal distribution of $\mathbf{z}^\pm$ (defined analogously to $\mathbf{y}^\pm$) by ρ and Λ. We then have (Buesing et al. 2013)

$$\mathrm{E}[y_i^\pm] =: m_i = \mathrm{E}_z[\exp(z_i^\pm)] = \exp\left(\tfrac{1}{2}\Lambda_{ii} + \rho_i\right) \tag{30}$$

$$\mathrm{E}[(y_i^\pm)^2] =: S_{ii} = m_i + \exp(\Lambda_{ii}) m_i^2. \tag{31}$$

For the off-diagonal second moments we have ($i \neq j$)

$$\mathrm{E}[y_i^{\pm} y_j^{\pm}] := S_{ij} = m_i m_j \exp(\Lambda_{ij}). \tag{32}$$

As we can estimate the first and second moments m_i and S_{ij} of $\mathbf{y}$ directly from the data, equations (30)-(32) form a system of $4kq + kq(2kq-1)$ nonlinear equations in $4kq + kq(2kq-1)$ unknowns ρ, Λ (with symmetric $\Lambda = \Lambda^\top$). The equations above can be solved efficiently by separately solving one 2-dimensional system (equations 30-31) for each pair of unknowns ρ_i, Λ_{ii}, $\forall i \in \{1, \ldots, \kappa q\}$. They have a closed-form solution for the PLDS model (see also (Krumin & Shoham 2009)):

$$\rho_i = 2\log(m_i) - \tfrac{1}{2}\log(S_{ii} - m_i) \tag{33}$$

$$\Lambda_{ii} = \log(S_{ii} - m_i) - 2\log(m_i). \tag{34}$$

Once the ρ_i and Λ_{ii} are known, equation (32) reduces to a 1-dimensional nonlinear equation for Λ_{ij} for each pair of indices ($i < j$):

$$\Lambda_{ij} = \log(S_{ij}) - \log(m_i m_j). \tag{35}$$

One can see that the above equations do not have solutions if any one of the terms in the logarithms is non-positive, which may happen with moments m_i, S_{ij} computed from a finite number of samples or a mis-specified model. It is therefore useful to scale the matrix S (by left and right multiplication with the same diagonal matrix) such that all Fano factors that are initially smaller than 1 are set to a given threshold (in simulations we used $1 + 10^{-2}$). This procedure ensures that there exists a unique solution (ρ, Λ) to the moment conversion (33)-(35). It is still the case that the resulting matrix Λ might not be positive semidefinite (Macke, Berens, Ecker, Tolias & Bethge 2009), but this can be rectified by finding its eigendecomposition, thresholding the eigenvalues (EVs) and then reconstructing Λ.

For sufficiently large data sets generated from a "true" PLDS model, observed Fano factors will be greater than one with high probability. Assuming stationarity, the moment conversion asymptotically yields the unique, correct moments ρ and Λ of the Gaussian log-rates $\mathbf{z}$. The Ho-Kalman method yields consistent estimates of $A, C, Q, \mathbf{d}$ given the true μ and Λ. Hence, the proposed two-stage method yields consistent estimates of the parameters $A, C, Q, \mathbf{d}$ of a stationary PLDS. We call this algorithm PLDSID (Buesing et al. 2013).

5 Results

In this section, we illustrate the behaviour of the algorithms described above on synthetic data. We generated the data by sampling neural population spike trains from ground truth PLDS models with randomly generated parameters θ. The latent dynamics of all PLDS models were chosen to be ten-dimensional with time-constants τ_i ranging from 30 to 120 time bins, where the τ_i derive from

the eigenvalues λ_i of the dynamics matrix A according to $\tau_i = -(\log|\lambda_i|)^{-1}$. The elements of the loading matrix C were drawn independently from $\mathcal{N}(0,1)$, and the remaining parameters Q and $\mathbf{d}$ where scaled such that the sampled spike trains were sparse, with about 20% of bins non-empty and about 2% of bins containing more than one spike. By setting $\mathbf{x}_0 = 0$ and Q_0 equal to the stationary covariance matrix Π, we ensured that the state-space process was statistically stationary throughout.

We first compared the accuracy of the Laplace and the variational methods at inferring the latent state trajectories, when the true parameters of the generating models are known. Initially, we simulated spike trains for $q = 10$ neurons lasting $T = 250$ time steps and compared the resulting estimates of the latent trajectories (Figure 1a). The two methods produced estimates of the mean trajectory which were very similar. Furthermore, the posterior uncertainty estimates (derived from the marginal posterior variances) of the two algorithms proved to be almost indistinguishable by eye.

However, a closer investigation revealed that the relative quality of the two posterior approximations varied systematically with the number of observed neurons q. To illustrate this, we generated 6 different ground truth PLDS models (as described above) and from each model we sampled $K = 10$ latent trajectories $\mathbf{x}_{1:T}$ with $T = 250$ time steps each, resulting in 60 trials in total. From these latent trajectories, we generated data sets with different observation dimensions ranging from $q = 1$ to $q = 2000$ and compared the quality of the variational and Laplace estimates of the latent states as a function of q. To measure the approximation quality, we evaluated the log-probability $\log q(\mathbf{x}_{1:T})$ of the true (unobserved) latent trajectory $\mathbf{x}_{1:T}$ under the two different approximate posteriors. For a good posterior approximation, the true trajectories should have higher probability than under a poor one. In Figure 1b we report the difference of the log-probabilities under the variational and the Laplace posteriors collected over the 60 trials for each value of q. The results show that when the dimensionality of the observation is small, the true trajectories have roughly the same posterior probability under both the variational and Laplace approximations; this is to be expected as the posterior in this low-data regime is very close to the (Gaussian) prior which both approximations capture faithfully. For intermediate values of q, the latent trajectories were roughly $10 \approx \exp(2)$ times more likely under the variational than under the Laplace approximation, showing that the variational approach consistently outperforms the Laplace approximation in this intermediate regime. Thus, the local curvature at the mode of the posterior, as used by the Laplace approximation, provides a poorer guide to appropriate distribution than does direct minimisation of a KL-divergence.[4] The probability ratio goes down again for large q, as the posterior concentrates around the maximum likelihood value $\mathbf{x}^* = \mathrm{argmax}_{\mathbf{x}} P(\mathbf{y}|\mathbf{x}, \theta)$.

Next, we compared the quality of parameter estimates obtained using each of the two approximations during the E-step of the EM algorithm (i.e. lEM

[4] In fact, our performance measure provides a crude numerical correlate to a KL-divergence between the true posterior and the approximation: but this is the opposite direction of divergence to that minimised by the variational method.

using the Laplace approximation and vEM using the variational approach) and also investigated the efficacy of the spectral learning method as an initialiser for EM. To this end, we used a data set of $K = 100$ trials with $T = 250$ time bins each and $q = 100$ observed dimensions. We found that, for this data set, both approximation methods increased the variational bound at each iteration, a property that was guaranteed to be true for vEM, but not for lEM (Figure 5a). Initialisation using the parameters returned by spectral learning led to faster convergence but—for this data set—not to better final values of the lower bound (Figure 5a). As we know the true generating parameters for these simulated data it was possible in principle to also quantify performance by comparing the estimated parameters to these true values. Howewver, state-space models of the form we have considered here are not fully identifiable: there are many different parameter values (related by a linear transformation) that all result in the same effective distribution of observations. Thus, it is possible for learning to converge on parameter estimates that are different from, but equivalent to, the generating values and so comparing estimated parameters element-by-element could yield misleading results. Fortunately, each equivalent set of parameters shares certain invariant properties, and it was these that we compared. Specifically, we quantified the accuracy of the estimated loading matrix C by measuring the subspace angle between the true and estimated matrices. We found that the the spectral algorithm provided very good estimates of the loading-matrix C and led to fast convergence, and that there was little difference in the performance of the two approximation methods. Similarly, we compared estimates of the dynamics matrix A by comparing the eigenvalues of the learnt matrices to those of the generating ones. As with C, we found that both approximations worked well, and that initialisation with the spectral method led to faster convergence. Finally, we calculated the stationary covariance of the spike-counts predicted for each set of learnt parameters using the methods described by Buesing et al. (2012) and Buesing et al. (2013). For this performance metric, we found that vEM consistently outperformed lEM. Thus, while the more accurate posterior estimation of vEM did yield some benefits in learning, lEM was broadly competitive on the simulated data used here. It is worth noting that lEM is computationally much cheaper than the variational method, and so the former might be preferable when computational considerations are important. Finally, we found that spectral initialisation led to considerably faster convergence than from random initial values.

6 Discussion

We have presented two methods for inferring low-dimensional state descriptions from population spike train data. We found that (global) Laplace and variational inference methods both yielded similar estimates for the mean state trajectory, and also comparable estimates for the marginal variances and covariances of the state trajectory, although in numerical evaluations the variational approach proved more accurate for a range of observed dimensionalities. We

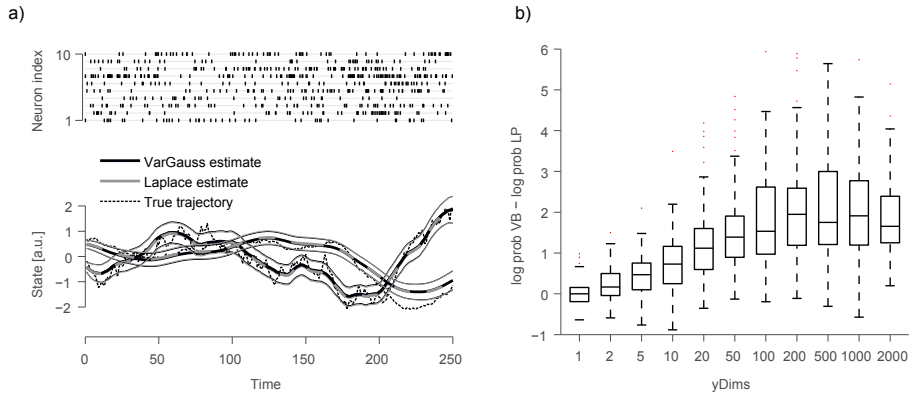

Figure 1: **Estimating latent states from population spike trains a) Simulated raster and state reconstructions:** Top: Simulated spike raster from 10 neurons. Bottom: Two dimensions of the true state trajectory (black dotted) and posterior mean estimates of two inference methods. Both variational (thick black, VB) and Laplace approximations (thick gray, LP) have very similar reconstructions of the mean trajectory. Thin lines indicate estimated posterior uncertainty around the mean, and they are also highly similar in this case. **b) Performance of the two inference methods as a function of observed dimension q:** Despite the apparent similarities in (a), there are systematic differences between the two methods which become apparent in the average across multiple data sets. We generated 600 data sets of each observation dimension (ranging from 1 to 2000), and calculated the difference of log-probability of the true trajectory under the two posterior approximations.

conclude that both methods are useful for state inference, and—given the lower computational cost—the Laplace approximation may be preferable when qualitatively accurate estimates are sought. However, when the two approximations were used within the Expectation Maximisation algorithm for parameter identification, only the variational method is guaranteed to increase the lower bound on the marginal likelihood at each iteration. Thus, parameter identification is often improved by the use of variational inference, although the benefit was slim in the experiments reported here. Finally, we found that using a spectral identification method adapted to non-linear outputs also provided reliable estimates of some parameters, and significantly sped up convergence times for the EM algorithm.

7 Summary

- State space-models with linear dynamics provide a powerful and flexible class of models for describing the statistical structure and temporal dynamics of neural population activity.

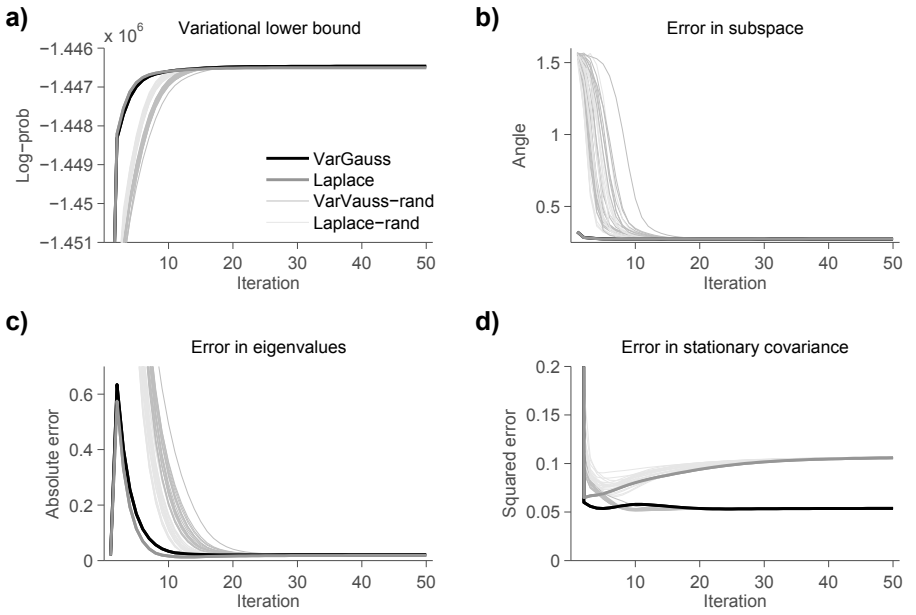

Figure 2: **Parameter estimation.** The performance of different parameter estimation algorithms on a synthetic data set with known ground truth (see text for details). **a) Variational lower bound:** For each algorithm, we calculated the variational lower bound (equation 20) after each EM iteration. Thick lines indicate performance of vEM and lEM initialised with PLSID, thin lines indicate performance on EM runs initialised randomly. **b) Error in reconstruction of subspace:** We also compared the performance of the different methods in reconstructing the parameters of the ground-truth model which was used to generate data. We quantified reconstruction performance as the subspace angle between C_{true} and $C_{estimated}$ for each algorithm and iteration. We found that both Laplace and variational approaches achieved good reconstruction performance and initialisation with PLSID led to faster convergence than use of random initialisations. **c) Error in eigenvalues of dynamics matrix:** We quantified the ability of algorithms to reconstruct the eigenvalues of the dynamics matrix (which determine the temporal correlations of the system). **d) Error in stationary correlation matrix:** We calculated the stationary covariance of the latent state $\mathbf{x}_t$ and from this calculated the stationary covariance of the $\mathbf{y}_t$. As PLSID directly optimises the stationary covariance, EM did not lead to a performance increase in this measure. In fact, we found the Laplace method for this data set to lead to worse estimates of the stationary covariance. In summary, usage of vEM initialised by PLSID led to best results and convergences in the fewest number of iterations.

- Neural spike-trains constitute multivariate point-processes (or, in discrete time, count-processes) which need to be modelled with non-Gaussian observation models.
- Learning the parameters of such models is a challenging problem, but multiple algorithms have been proposed both for parameter estimation as well as for reconstructing state trajectories from observed spike-trains.
- When using the Expectation Maximisation (EM), approximation methods need to be used in order to calculate the posterior distribution over states given observations. We found Gaussian variational inference (Emtiyaz Khan et al. 2013) to narrowly outperform the Laplace approximation for this task.
- Spectral learning algorithms which are modified to account for the non-Gaussianity of the observation model are faster than EM and are not sensitive to the choice of initial conditions.
- We found that best results were achieved by initialising parameters using the spectral learning algorithm, and then refining this initial value using the the EM algorithm based on Gaussian variational inference.

Acknowledgements

We acknowledge support from the Gatsby Charitable Foundation; an EU Marie Curie Fellowship to JHM (hosted by MS); DARPA REPAIR N66001-10-C-2010 and NIH CRCNS R01- NS054283 to MS; as well as the Bernstein Center Tübingen funded by the German Ministry of Education and Research (BMBF; FKZ: 01GQ1002). Portions of the research described here have been presented previously (Macke et al. 2012, Buesing et al. 2013, Buesing et al. 2014). We would like to thank our co-authors on these papers, E. Archer for contributions to the accompanying code, and Florian Sandhäger for comments on the manuscript.

A software-implementation of the methods described in this chapter can be found at `http://www.gatsby.ucl.ac.uk/resources/plds/`.

Bibliography

Archer, E., Koester, U., Pillow, J. W. & Macke, J. H. (2015), Low-dimensional models of neural population activity in sensory cortical circuits, *in* 'Advances in Neural Information Processing Systems', Vol. 27, Curran Associates, Inc.

Beal, M. J. (2003), Variational Algorithms for Approximate Bayesian Inference, PhD thesis, Gatsby Unit, University College London.

Bishop, C. (2006), *Pattern recognition and machine learning*, Springer New York.

Boyd, S. P. & Vandenberghe, L. (2004), *Convex Optimization*, Cambridge Univ Press.

Buesing, L., Machado, T., Cunningham, J. P. & Paninski, L. (2014), Clustered factor analysis of multineuronal spike data, *in* 'Advances in Neural Information Processing Systems', Vol. 27, Curran Associates, Inc.

Buesing, L., Macke, J. H. & Sahani, M. (2012), 'Learning stable, learning regularised latent models of neural population dynamics', *Network: Computation in Neural Systems* **23**(1-2), 24–47.

Buesing, L., Macke, J. H. & Sahani, M. (2013), Spectral learning of linear dynamics from generalised-linear observations with application to neural population data, *in* 'Advances in Neural Information Processing Systems', Vol. 25, pp. 1691–1699.

Chen, Z. (2003), 'Bayesian filtering: From kalman filters to particle filters, and beyond', *Statistics* **182**(1), 1–69.

Chen, Z. & Brown, E. N. (2013), 'State space model', *Scholarpedia* **8**(3), 30868. revision 132960.

Chornoboy, E., Schramm, L. & Karr, A. (1988), 'Maximum likelihood identification of neural point process systems', *Biological Cybernetics* **59**(4), 265–275.

Churchland, M. M., Yu, B. M., Sahani, M. & Shenoy, K. V. (2007), 'Techniques for extracting single-trial activity patterns from large-scale neural recordings', *Current Opinion in Neurobioly* **17**(5), 609–618.

Cunningham, J. P. & Yu, B. M. (2014), 'Dimensionality reduction for large-scale neural recordings', *Nat Neurosci* .

Dempster, A. P., Laird, N. M. & Rubin, D. B. (1977), 'Maximum likelihood from incomplete data via the em algorithm', *Journal of the Royal Statistical Society. Series B (Methodological)* **39**(1), 1–38.

Doucet, A. & Johansen, A. M. (2009), 'A tutorial on particle filtering and smoothing: Fifteen years later', *Handbook of Nonlinear Filtering* **12**, 656–704.

Durbin, J., Koopman, S. J. & Atkinson, A. C. (2001), *Time series analysis by state space methods*, Vol. 15, Oxford University Press Oxford.

Ecker, A. S., Berens, P., Cotton, R. J., Subramaniyan, M., Denfield, G. H., Cadwell, C. R., Smirnakis, S. M., Bethge, M. & Tolias, A. S. (2014), 'State dependence of noise correlations in macaque primary visual cortex', *Neuron* **82**(1), 235–48.

Eden, U. T., Frank, L. M., Barbieri, R., Solo, V. & Brown, E. N. (2004), 'Dynamic analysis of neural encoding by point process adaptive filtering', *Neural Computation* **16**(5), 971–998.

Emtiyaz Khan, M., Aravkin, A., Friedlander, M. & Seeger, M. (2013), Fast dual variational inference for non-conjugate latent gaussian models, *in* 'Proceedings of the 30th International Conference on Machine Learning', pp. 951–959.

Ghahramani, Z. & Hinton, G. E. (1996), 'Parameter estimation for linear dynamical systems', *University of Toronto Technical Report* **6**(CRG-TR-96-2).

Ghahramani, Z. & Roweis, S. T. (1999), Learning nonlinear dynamical systems using an em algorithm, *in* 'Proceedings of the 1998 conference on Advances in neural information processing systems II', MIT Press, Cambridge, MA, USA, pp. 431–437.

Goris, R. L. T., Movshon, J. A. & Simoncelli, E. P. (2014), 'Partitioning neuronal variability', *Nature Neuroscience* **17**(6), 858–65.

Harris, K. D. & Thiele, A. (2011), 'Cortical state and attention.', *Nature Reviews Neuroscience* **12**(9), 509–523.

Ho, B. L. & Kalman, R. E. (1966), 'Effective construction of linear state-variable models from input/output functions.', *Regelungstechnik* **14**(12), 545–548.

Kalman, R. E. & Bucy, R. S. (1961), 'New results in linear filtering and prediction theory', *Trans. Am. Soc. Mech. Eng., Series D, Journal of Basic Engineering* **83**, 95–108.

Kass, R. E., Eden, U. & Brown, E. (2014), *Analysis of Neural Data*, Springer.

Katayama, T. (2005), *Subspace methods for system identification*, Springer.

Krumin, M. & Shoham, S. (2009), 'Generation of Spike Trains with Controlled Auto-and Cross-Correlation Functions', *Neural Computation* pp. 1–23.

Kulkarni, J. E. & Paninski, L. (2007), 'Common-input models for multiple neural spike-train data', *Network: Computation in Neural Systems* **18**(4), 375–407.

Lawhern, V., Wu, W., Hatsopoulos, N. & Paninski, L. (2010), 'Population decoding of motor cortical activity using a generalized linear model with hidden states.', *Journal of Neuroscience Methods* **189**(2), 267–280.

Macke, J., Berens, P., Ecker, A., Tolias, A. & Bethge, M. (2009), 'Generating spike trains with specified correlation coefficients.', *Neural Computation* **21**(2), 397–423.

Macke, J. H., Buesing, L., Cunningham, J. P., Yu, B. M., Shenoy, K. V. & Sahani, M. (2012), Empirical models of spiking in neural populations, *in* 'Advances in Neural Information Processing Systems', Vol. 24, Curran Associates, Inc.

Maimon, G. (2011), 'Modulation of visual physiology by behavioral state in monkeys, mice, and flies', *Current Opinion in Neurobiology* **21**(4), 559–64.

Mangion, A. Z., Yuan, K., Kadirkamanathan, V., Niranjan, M. & Sanguinetti, G. (2011), 'Online variational inference for state-space models with point-process observations.', *Neural Computation* **23**(8), 1967–1999.

Møller, J., Syversveen, A. & Waagepetersen, R. (1998), 'Log gaussian cox processes', *Scandinavian journal of statistics* **25**(3), 451–482.

Nickisch, H. & Rasmussen, C. E. (2008), 'Approximations for binary gaussian process classification.', *Journal of Machine Learning Research* **9**(10).

Niell, C. M. & Stryker, M. P. (2010), 'Modulation of visual responses by behavioral state in mouse visual cortex.', *Neuron* **65**(4), 472–479.

Opper, M. & Archambeau, C. (2009), 'The variational gaussian approximation revisited', *Neural Computation* **21**(3), 786–792.

Paninski, L. (2004), 'Maximum likelihood estimation of cascade point-process neural encoding models.', *Network: Computation in Neural Systems* **15**(4), 243–262.

Paninski, L., Ahmadian, Y., Ferreira, D., Koyama, S., Rahnama Rad, K., Vidne, M., Vogelstein, J. & Wu, W. (2010), 'A new look at state-space models for neural data', *Journal of Computational Neuroscience* **29**, 107–126.

Pillow, J. W., Shlens, J., Paninski, L., Sher, A., Litke, A. M., Chichilnisky, E. J. & Simoncelli, E. P. (2008), 'Spatio-temporal correlations and visual signalling in a complete neuronal population.', *Nature* **454**(7207), 995–999.

Santhanam, G., Ryu, S. I., Yu, B. M., Afshar, A. & Shenoy, K. V. (2006), 'A high-performance brain-computer interface.', *Nature* **442**(7099), 195–198.

Smith, A. C. & Brown, E. N. (2003), 'Estimating a state-space model from point process observations', *Neural Computation* **15**(5), 965–991.

Stevenson, I. H. & Kording, K. P. (2011), 'How advances in neural recording affect data analysis', *Nature Neuroscience* **14**(2), 139–42.

Truccolo, W., Hochberg, L. R. & Donoghue, J. P. (2010), 'Collective dynamics in human and monkey sensorimotor cortex: predicting single neuron spikes.', *Nature Neuroscience* **13**(1), 105–111.

Turner, R. E. & Sahani, M. (2011), Two problems with variational expectation maximisation for time-series models, *in* D. Barber, A. T. Cemgil & S. Chiappa, eds, 'Inference and Learning in Dynamic Models', Cambridge University Press.

Vidne, M., Ahmadian, Y., Shlens, J., Pillow, J., Kulkarni, J., Litke, A., Chichilnisky, E., Simoncelli, E. & Paninski, L. (2011), 'Modeling the impact of common noise inputs on the network activity of retinal ganglion cells.', *Journal of Computational Neuroscience* .

Yu, B. M., Afshar, A., Santhanam, G., Ryu, S. I., Shenoy, K. & Sahani, M. (2006), Extracting dynamical structure embedded in neural activity, *in* Y. Weiss, B. Schölkopf & J. Platt, eds, 'Advances in Neural Information Processing Systems 18', MIT Press, Cambridge, MA, pp. 1545–1552.

Yu, Byron, M., Cunningham, J. P., Shenoy, K. V. & Sahani, M. (2008), Neural decoding of movements: From linear to nonlinear trajectory models, *in* 'Neural Information Processing', Springer, pp. 586–595.

Yuan, K., Girolami, M. & Niranjan, M. (2012), 'Markov chain monte carlo methods for state-space models with point process observations.', *Neural Computation* **24**(6), 1462–1486.