A tutorial overview of the Kalman filter and latent linear dynamical systems

Jonathan W. Pillow
Princeton Neuroscience Institute
Princeton University
http://pillowlab.princeton.edu

November 14, 2023

Abstract

This paper provides a tutorial introduction to the Kalman filter and the model underlying it.

1 Introduction

In this tutorial, we will introduce Gaussian latent linear dynamical system (LLDS) models and derive the classic Kalman filtering and smoothing algorithms. We will then derive the expectation-maximization algorithm for fitting the parameters of a Gausian LLDS model to data.

In the appendix, we provide several additional derivations, such as the treatment of intpus and an efficient implementation of LLDS model inference involving block-sparse matrices.

The Kalman filter, also known as the Kalman-Bucy filter [1, 2], was first derived from the perspective of least squares, with no explicit connection to Bayesian inference. For more on the origins of the Kalman filter see [3].

1.1 Notation

We will use $\mathcal{N}(\mathbf{x}; \mu, \Sigma)$ to denote a multivariate normal probability density function over $\mathbf{x}$ with mean μ and variance Σ. When it is obvious what the variable in question is, we will abbreviate this more simply as $\mathcal{N}(\mu, \Sigma)$.

To do: flesh this out.

2 The Latent Linear Dynamical System (LLDS) model

Consider the latent linear dynamical system (LLDS) defined by:

$$\mathbf{x}_t = A\mathbf{x}_{t-1} + \mathbf{w}_t, \qquad \mathbf{w}_t \sim \mathcal{N}(0, Q) \qquad \textit{(latent dynamics)} \tag{1}$$

$$\mathbf{y}_t = C\mathbf{x}_t \quad + \mathbf{v}_t, \qquad \mathbf{v}_t \sim \mathcal{N}(0, R) \qquad \textit{(observations)} \tag{2}$$

Here $\mathbf{x}_t \in \mathbb{R}^m$ is a “latent” or “unobserved” or “state” vector, $\mathbf{y}_t \in \mathbb{R}^n$ is the “observed” or “measured” or “output” vector, also known as the “emissions”, and $\mathbf{w}_t \in \mathbb{R}^m$ and $\mathbf{v}_t \in \mathbb{R}^n$ denote vectors of zero-mean Gaussian noise

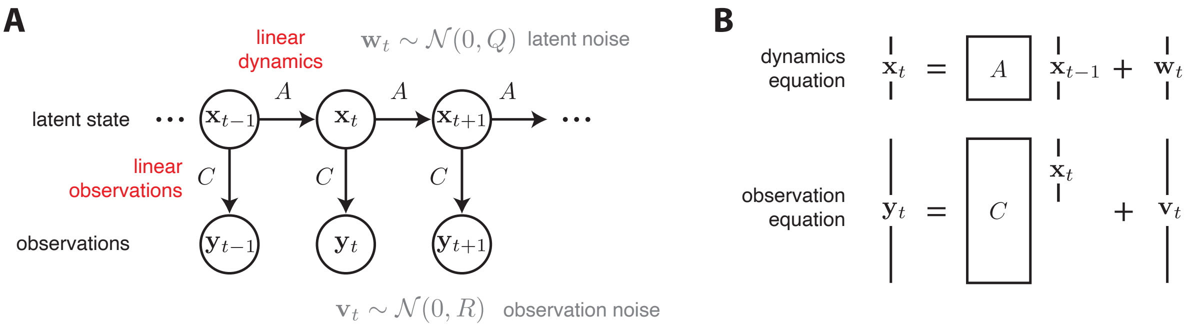
A
linear dynamics
$\mathbf{w}_t \sim \mathcal{N}(0, Q)$ latent noise
latent state
A
$\mathbf{x}_{t-1}$
$\mathbf{x}_t$
$\mathbf{x}_{t+1}$
linear observations
C
observations
$\mathbf{y}_{t-1}$
$\mathbf{y}_t$
$\mathbf{y}_{t+1}$
$\mathbf{v}_t \sim \mathcal{N}(0, R)$ observation noise
B
dynamics equation
$\mathbf{x}_t = A \; \mathbf{x}_{t-1} + \mathbf{w}_t$
observation equation
$\mathbf{y}_t = C \; \mathbf{x}_t + \mathbf{v}_t$

Figure 1: Graphical model and equations for a Gaussian latent linear dynamical system (LLDS) model, also known as the Kalman filtering model.

that corrupts the latent state and the observations, respectively. In the control literature, this model is known as a *discrete-time state-space linear system* [4].

In this model, the latent vector $\mathbf{x}_t$ evolves linearly according to an $m \times m$ dynamics matrix A, and observations $\mathbf{y}_t$ result from a linear projection of the $\mathbf{x}_t$ onto the observation matrix C. To give the model a well-defined initial state, we add a prior over the latent at time step one:

$$\mathbf{x}_1 \sim \mathcal{N}(0, Q_0). \qquad \textit{(initial latent)} \tag{3}$$

Typically we have $m < n$, so that the high-dimensional observations $\mathbf{y}_t$ are explained by the lower-dimensional $\mathbf{x}_t$. The LLDS model parameters are thus given by

$$\theta = \{A, C, Q, R, Q_0\}, \tag{4}$$

consisting of linear dynamics matrix A, observation matrix C, the latent variable noise covariance Q, observation noise covariance R, and initial latent covariance Q_0. Fig. 1 shows an illustration of the LLDS model, depicted either as a graphical model (left) or a set of matrix equations (right).

2.1 Identifiability

Because the latent variables are not observed, a Gaussian LLDS model is only identifiable up to a linear transformation of its latent space. To see this, consider a LLDS model with parameters A, C, Q, R, Q_0. Now let us define a new latent variable $\mathbf{x}'_t$ that is related to the latent variables of the original model by an invertible linear transformation:

$$\mathbf{x}'_t = H\mathbf{x}_t, \tag{5}$$

for some invertible $m \times m$ matrix H. Then we can create a new LLDS model with parameters

$$A' = HAH^{-1}, \quad C' = CH^{-1}, \tag{6}$$

$$Q' = HQH^\top, \quad Q'_0 = HQH^\top, \tag{7}$$

and noise covariance R unchanged. This new model assigns the same probability to the observed data as the original model. This is easy to see by substituting the $H\mathbf{x}_t$ into the system equations (eq. 1-3) for the new model.

One implication of this non-identiability is that we can transform an LLDS model in order to simplify its parameters or increase the interpretability of its latents without changing the model. For example, we can transform by $H = Q^{-\frac{1}{2}}$, so that the dynamics noise covariance Q' is the identity. If we wish to make C semi-orthogonal, so its columns are orthogonal unit vectors, we could define the transform by $H^{-1} = V\Sigma^{-1}$, where $C = U\Sigma V^\top = C$ is the singular value decomposition of the original C matrix. Note that we are restricted to transforming the dynamics matrix A to a similarity transformation of itself, since $A = HAH^{-1}$. Thus we cannot alter the eigenvalues of A, and we cannot diagonalize A if it has complex eigenvalues. Note also that such transformations do *not* allow us to alter the column space of C, or to modify the noise covariance R in any way.

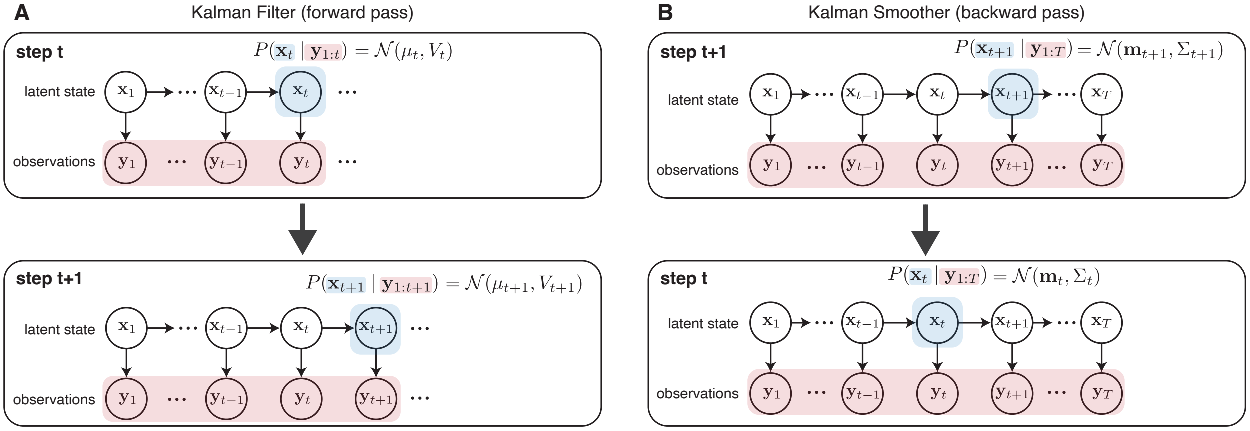

Figure 2: Comparison between Kalman filtering (A) and Kalman smoothing (B).

2.2 Stability and asymptotic behavior

The stability of an LLDS model is determined by the eigenvalues of the dynamix matrix A, which are in general complex valued. The imaginary component of each eigenvalue gives rise to rotations or oscillations along an associated pair of eigenmodes, while the absolute value of the eigenvalues determines stability.

If the absolute value of any eigenvalue exceeds one, $\max(\text{abs}(\text{eig}(A))) > 1$, then the LLDS is unstable. This means that if we simulate the model, the latent vector will diverge exponentially to infinity.

By contrast, if all eigenvalues have absolute value less than one, $\max(\text{abs}(\text{eig}(A))) < 1$, then the LLDS model is stable. In this regime, the noiseless LLDS model, in which $Q = 0$, will have a latent state that converges to a stable fixed point at 0. If the latent noise is non-zero, however, then the latent state has a stable asymptotic Gaussian distribution with mean 0 and covariance V_∞, where this asymptotic covariance can be computed as the solution to the discrete Lyapunov equation:

$$V_\infty = AV_\infty A^\top + Q. \tag{8}$$

In this case, the observations will also have a marginally Gaussian distribution with mean zero and covariance $CV_\infty C^\top + R$. (Thus, a simple sanity check when simulating an LDS model or fitting an LDS model to data is to examine whether the sample covariance of the data, $\text{cov}(\mathbf{y}_t \mathbf{y}_t^\top)$ is close to $CV_\infty C^\top + R$, where a discrete Lyapunov equation solver is used to obtain V_∞ from A and Q.)

Finally, if any eigenvalue has absolute value equal to 1, the noiseless LLDS model has neutral stability, meaning that it neither diverges nor converges to zero. If the covariance Q is non-singular, moreover, the variance along the associated subspace grows linearly with time. Thus, the model does not converge to a stationary distribution.

3 Inferring the latents: Kalman filter and smoother

Kalman filtering is an algorithm for computing the posterior distribution over latent $\mathbf{x}_t$ given $\mathbf{y}_{1:t}$, the observations from time bin 1 to time t. Kalman filtering is a "foward" filtering equation that operates by iteratively updating the posterior over $\mathbf{x}_t$ using the data from each time point (Fig. 2A).

By contrast, *Kalman smoothing* involves propagating information backwards from the final time step T in order to update the posterior over $\mathbf{x}_t$ at all previous times. It corresponds to a "backwards" filtering operation that starts at time T and propagates information iteratively to the latent variable at each previous time stpes (Fig. 2B).

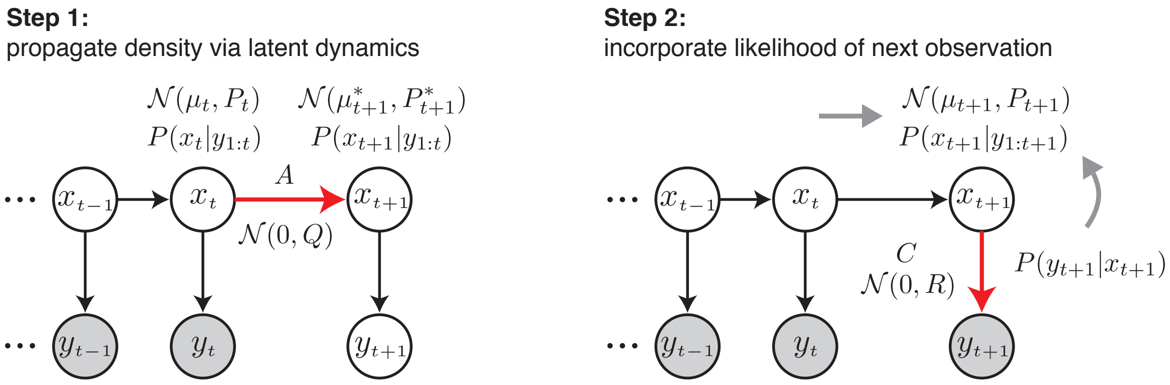

Figure 3: Two steps in the kalman filter.

3.1 Kalman filter

The Kalman Filter is a recursive algorithm for computing $P(\mathbf{x}_t \mid \mathbf{y}_{1:t})$, the posterior over the latent state at time t given all the data up to and including time t, from $P(x_{t-1} \mid \mathbf{y}_{1:t-1})$ the posterior over the latent at the previous time step, given the model parameters θ. The resulting distribution is necessarily Gaussian, allowing us to write:

$$P(\mathbf{x}_t \mid \mathbf{y}_{1:t}) = \mathcal{N}(\mu_t, V_t), \tag{9}$$

parametrized by a mean μ_t and covariance V_t. Note that μ_t is also the least-squares estimate of the signal $\mathbf{x}_t$ given the data up to time t.

The recursive update rule for the Kalman filter can then be broken down into two steps (see Fig. 1):

Step 1 (Prediction). The "prediction" step of the Kalman filter takes $P(\mathbf{x}_{t-1} \mid \mathbf{y}_{1:t-1})$, the posterior over the latent at time $t-1$ given observations from 1 to $t-1$, and propagates it one time step forward using the linear dynamics A and the latent "innovations" noise covariance Q. This produces the *prior* over the latent state at time t given the observations from time step 1 up to $t-1$:

$$P(\mathbf{x}_t \mid \mathbf{y}_{1:t-1}) = \mathcal{N}(\mu_{t*}, V_{t*}), \qquad \textit{(Kalman filter prior)} \tag{10}$$

where

$$\begin{aligned} \mu_{t*} &= A\mu_{t-1}, && \textit{(prior mean)} && (11)\\ V_{t*} &= AV_{t-1}A^\top + Q, && \textit{(prior covariance)} && (12) \end{aligned}$$

which follow from Gaussian fun facts for linear and additive transformations of a Gaussian distribution (eqs. 118-119). The only exception is that on the very first time step we have no previous observations, thus the prior is simply:

$$P(\mathbf{x}_1) = \mathcal{N}(0, Q_0), \qquad \textit{(Kalman filter prior, first time bin)} \tag{13}$$

thus we have that $\mu_{1*} = 0$ and $V_{1*} = Q_0$.

Step 2 (Update). The second step of the Kalman filter is the "update" step, which incorporates information from the observation $\mathbf{y}_t$ about the latent $\mathbf{x}_t$. This involves using Bayes' rule to combine the likelihood term $P(\mathbf{y}_t|\mathbf{x}_t)$ with the prior $P(\mathbf{x}_t|\mathbf{y}_{1:t-1})$ computed in the prediction step. The product of these two terms is proportional to $P(\mathbf{x}_t \mid \mathbf{y}_{1:t})$, the posterior distribution for time step t. From Bayes' rule, we have:

$$P(\mathbf{x}_t \mid \mathbf{y}_{1:t}) \propto P(\mathbf{y}_t \mid \mathbf{x}_t)P(\mathbf{x}_t \mid \mathbf{y}_{1:t-1}) = \mathcal{N}(\mathbf{y}_t; C\mathbf{x}_t, R)\mathcal{N}(\mathbf{x}_t; \mu_{t*}, V_{t*}). \tag{14}$$

Because both prior and likelihood are exponentiated quadratic forms in $\mathbf{x}_t$, we can use Gaussian identities (eq. 126) to obtain the Gaussian form of the posterior distribution:

$$P(\mathbf{x}_t \mid \mathbf{y}_{1:t}) = \mathcal{N}(\mu_t, V_t) \qquad \textit{(Kalman filter output)} \tag{15}$$

where

$$\mu_t = V_t(C^\top R^{-1}\mathbf{y}_t + V_{t*}^{-1}\mu_{t*}), \qquad \textit{(Kalman filter mean)} \tag{16}$$

$$V_t = (V_{t*}^{-1} + C^\top R^{-1} C)^{-1}. \qquad \textit{(Kalman filter covariance)} \tag{17}$$

Marginal likelihood

Note that the normalizing constant in the application of Bayes' rule above (eq. 14) corresponds to $P(\mathbf{y}_t \mid \mathbf{y}_{1:t-1})$, the marginal probability of the observation $\mathbf{y}_t$ given all previous observations. This quantity can also be derived by noting that $\mathbf{y}_t$ is a linear transformation of $\mathbf{x}_t$ plus Gaussian noise with covariance R. From the prior over $\mathbf{x}_t$ (eq. 10) and Gaussian fun facts (eq. 118-119), we obtain the following expression for the conditional marginal likelihood:

$$P(\mathbf{y}_t \mid \mathbf{y}_{1:t-1}) = \mathcal{N}(\mathbf{y}_t; C\mu_{t*}, CV_{t*}C^\top + R), \tag{18}$$

where μ_{t*} and V_{t*} are the prior mean and variance of the latent at time step t (eqs. 11-12).

The marginal probability of the entire dataset is given by the product of these terms:

$$P(\mathbf{y}_{1:T}) = P(\mathbf{y}_1)\prod_{t=2}^{T} P(\mathbf{y}_t \mid \mathbf{y}_{1:t-1}), \tag{19}$$

where $P(\mathbf{y}_1) = \mathcal{N}(\mathbf{y}_1; 0, CQ_0C^\top + R) = \mathcal{N}(\mathbf{y}_1; C\mu_{1*}, CV_{1*}C^\top + R)$ is the distribution of the first observation.

The log marginal likelihood can therefore be fully computed during the forward pass of the Kalman filter, without needing the backward pass of the Kalman smoother (which we discuss below). We obtain:

$$\log P(\mathbf{y}_{1:T}) = -\tfrac{1}{2}\sum_{t=1}^{T}\Big(\log|2\pi(CV_{t*}C^\top + R)| + (\mathbf{y}_t - C\mu_{t*})^\top (CV_{t*}C^\top + R)^{-1}(\mathbf{y}_t - C\mu_{t*})\Big). \tag{20}$$

3.2 Kalman Smoother

The Kalman Smoother represents the "backwards" complement to the Kalman Filter. The Kalman filter marches forward in time to update the posterior over the latent variable given *past* observations; the Kalman smoother, by contrast, marches backward in time to update the posterior over the latents with information about *future* observations.

Because these distributions are Gaussian, the Kalman smoother amounts to an algorithm for updating the mean and covariance of the latent given future observations. Thus the Kalman smoother produces:

$$P(\mathbf{x}_t \mid \mathbf{y}_{1:T}) = \mathcal{N}(\mathbf{m}_t, \Sigma_t), \tag{21}$$

where $\mathbf{m}_t$ and Σ_t are the mean and covariance of the posterior distribution over the latent vector at time bin t.

The Kalman smoother begins at time step T, where the Kalman filter terminates. The posterior over the latents at T is obtained direclty from the last step of the Kalman filter, thus we have:

$$P(\mathbf{x}_T \mid \mathbf{y}_{1:T}) = \mathcal{N}(\mathbf{m}_T, \Sigma_T) = \mathcal{N}(\mu_T, V_T). \tag{22}$$

The Kalman then proceeds recursively, taking $P(\mathbf{x}_{t+1} \mid \mathbf{y}_{1:T})$, the posterior over $\mathbf{x}_{t+1}$ given all observations from past and future, and propagating it one step backwards in time to obtain $P(\mathbf{x}_t \mid \mathbf{y}_{1:T})$. This requires combining future information from $\mathbf{y}_{t+1:T}$ with past information from $\mathbf{y}_{1:t}$ computed via the Kalman filter.

To see how this can be accomplished, we start by writing the posterior over $\mathbf{x}_t$ as the marginalization of joint posterior over $\mathbf{x}_t$ and $\mathbf{x}_{t+1}$:

$$\begin{aligned} P(\mathbf{x}_t \mid \mathbf{y}_{1:T}) &= \int P(\mathbf{x}_t, \mathbf{x}_{t+1} \mid \mathbf{y}_{1:T})\, d\mathbf{x}_{t+1} && (23)\\ &= \int P(\mathbf{x}_t \mid \mathbf{x}_{t+1}, \mathbf{y}_{1:T})\, P(\mathbf{x}_{t+1}|\mathbf{y}_{1:T})\, d\mathbf{x}_{t+1} && (24)\\ &= \int P(\mathbf{x}_t \mid \mathbf{x}_{t+1}, \mathbf{y}_{1:t})\, P(\mathbf{x}_{t+1}|\mathbf{y}_{1:T})\, d\mathbf{x}_{t+1}, && (25)\\ &= \int P(\mathbf{x}_t \mid \mathbf{x}_{t+1}, \mathbf{y}_{1:t})\, \mathcal{N}(\mathbf{x}_{t+1}; \mathbf{m}_{t+1}, \Sigma_{t+1})\, d\mathbf{x}_{t+1}, && (26) \end{aligned}$$

where (eq. 24) is simply an application of the definition of conditional probability, and (eq. 25) follows from the fact that $\mathbf{x}_t$ is conditionally independent of future observations $\mathbf{y}_{t+1:T}$ given $\mathbf{x}_{t+1}$, meaning that $P(\mathbf{x}_t \mid \mathbf{x}_{t+1}, \mathbf{y}_{1:T}) = P(\mathbf{x}_t \mid \mathbf{x}_{t+1}, \mathbf{y}_{1:t})$. In (eq. 26), we have simply substituted the Gaussian from the previous step of the Kalman smoother for $P(\mathbf{x}_{t+1} \mid \mathbf{y}_{1:T})$.

The first term in (eq. 26), $P(\mathbf{x}_t \mid \mathbf{x}_{t+1}, \mathbf{y}_{1:t})$, can be obtained from terms computed during the forward pass of the Kalman filter, since it involves only past observations $\mathbf{y}_{1:t}$. To compute it, we can start by deriving the joint distribution $P(\mathbf{x}_t, \mathbf{x}_{t+1} \mid \mathbf{y}_{1:t})$. From the Kalman filtering equations (eqs. 10-12), the joint distribution over a pair of adjacent latent states given previous observations is:

$$P\left(\begin{bmatrix}\mathbf{x}_t \\ \mathbf{x}_{t+1}\end{bmatrix} \mid \mathbf{y}_{1:t}\right) = \mathcal{N}\left(\begin{bmatrix}\mu_t \\ A\mu_t\end{bmatrix}, \begin{bmatrix} V_t & V_t A^\top \\ AV_t & AV_tA^\top\end{bmatrix}\right) = \mathcal{N}\left(\begin{bmatrix}\mu_t \\ \mu_{t+1*}\end{bmatrix}, \begin{bmatrix} V_t & V_t A^\top \\ AV_t & V_{t+1*}\end{bmatrix}\right), \tag{27}$$

where μ_{t+1*} and V_{t+1*} in the right-most expression denote the prior mean and covariance over $\mathbf{x}_{t+1}$ given the data up to time bin t. We can then apply the Gaussian fun fact for conditionalization (eq. 121) to obtain:

$$P(\mathbf{x}_t \mid \mathbf{x}_{t+1}, \mathbf{y}_{1:t}) = \mathcal{N}(\mu_t + V_tA^\top V_{t+1*}^{-1}(\mathbf{x}_{t+1} - \mu_{t+1*}), V_t - V_tA^\top V_{t+1*}^{-1}AV_t). \tag{28}$$

Note that this is equivalent to writing:

$$\mathbf{x}_t = \mu_t + J_t(\mathbf{x}_{t+1} - \mu_{t+1*}) + n, \quad n \sim \mathcal{N}(0, V_t - J_tV_{t+1*}J_t^\top), \tag{29}$$

where $J_t = V_tA^\top V_{t+1*}^{-1}$, and the additive noise n has covariance $V_t - V_tA^\top V_{t+1*}^{-1}AV_t = V_t - J_tV_{t+1*}J_t^\top$. This expression allows us to see how to compute the marginalization in (eq. 25), since $\mathbf{x}_t$ is written as an affine transformation of $\mathbf{x}_{t+1}$, which we already know to have distribution $\mathcal{N}(\mathbf{m}_{t+1}, \Sigma_{t+1})$, plus Gaussian noise.

The recursive updates for the Kalman smoother mean and covariance can then be derived from the Gaussian fun facts for linear transformations (eq. 119) and sums (eq. 118),

$$\begin{aligned} P(\mathbf{x}_t \mid \mathbf{y}_{1:T}) &= \mathcal{N}(\mathbf{m}_t, \Sigma_t), \quad \text{where} && (30)\\ \mathbf{m}_t &= \mu_t + J_t(\mathbf{m}_{t+1} - \mu_{t+1*}). \quad && \textit{(Kalman smoother mean)} \quad (31)\\ \Sigma_t &= V_t + J_t(\Sigma_{t+1} - V_{t+1*})J_t^\top. \quad && \textit{(Kalman smoother covariance)} \quad (32) \end{aligned}$$

Note the appealing symmetry in these equations, which is that they both involve taking the mean and covariance from the Kalman filter, μ_t and V_t, and modify them by the difference between the posterior moments from the Kalman smoother, $\mathbf{m}_{t+1}$ and Σ_{t+1}, and the prior moments μ_{t+1*} and V_{t+1*}, computed using Kalman filter, scaled by J_t.

Note that this same derivation allows us to see that the covariance between $\mathbf{x}_t$ and $\mathbf{x}_{t+1}$, which we will need later for the M-step updates, is given by:

$$\Sigma_{t,t+1} \triangleq \operatorname{cov}(\mathbf{x}_t, \mathbf{x}_{t+1}) = J_t\Sigma_{t+1}. \tag{33}$$

4 Fitting via Expectation-Maximization (EM)

Although it is possible to fit the parameters of the Gaussian LLDS model using direct numerical optimization of the log marginal likelihood (eq. 20), it is often more computationally efficient to use expectation-maximization (EM), an iterative procedure for maximum likelihood estimation in latent variable models [5].

The EM algorithm consists of the alternation of expectation ("E") and maximization ("M") steps. Each of these steps is guaranteed to increase a lower bound on the marginal likelihood or "evidence". In classical machine learning literature, this lower bound was often referred to as the "negative free energy", while in recent literature it is commonly known as the "evidence lower bound" or ELBO. Note that in the current setting of Gaussian LLDS models, the ELBO provides a tight lower bound on the log-marginal likelihood, thus it is equal to the log marginal likelihood after each E step.

It is beyond the scope of this tutorial to provide a general treatment of EM (see [6] or [7]). However, we will provide a detailed derivation of the relevant E and M steps, both of which have closed-form expressions for Gaussian LLDS models.

4.1 E step

The E step involves computing a quantity known as the complete-data log-likelihood given the model parameters. Let θ denote the parameters at the beginning the k'th step of EM. The complete data log-likelihood is given by:

$$L_{CD}(\theta) \triangleq \mathbb{E}\big[\, \log P(\mathbf{y}_{1:T}, \mathbf{x}_{1:T} \mid \theta)\,\big], \qquad \textit{(complete-data log-likelihood)} \tag{34}$$

where expectation is taken with respect to $P(\mathbf{x}_{1:T} \mid \mathbf{y}_{1:T}, \theta)$, the posterior over the latents given θ, which is itself Gaussian (eq. 107).

For the Gaussian LLDS model, the complete data log-likelihood can be written as the sum of two terms, one coming from the latent dynamics and another coming from the observations:

$$\begin{aligned}
L_{CD}(\theta) &= \mathbb{E}\Big[\log P(\mathbf{x}_{1:T} \mid \theta)\Big] + \mathbb{E}\Big[\log P(\mathbf{y}_{1:T} \mid \mathbf{x}_{1:T}, \theta)\Big] && (35)\\
&= \mathbb{E}\left[\log\left(P(\mathbf{x}_1|\theta)\prod_{t=2}^{T} P(\mathbf{x}_t \mid \mathbf{x}_{t-1}, \theta)\right)\right] + \mathbb{E}\left[\log\left(\prod_{t=1}^{T} P(\mathbf{y}_t \mid \mathbf{x}_t, \theta)\right)\right] && (36)\\
&= \mathbb{E}\Big[\log \mathcal{N}(\mathbf{x}_1; 0, Q_0)\Big] + \mathbb{E}\left[\sum_{t=2}^{T} \log \mathcal{N}(\mathbf{x}_t; A\mathbf{x}_{t-1}, Q)\right] + \mathbb{E}\left[\sum_{t=1}^{T} \log \mathcal{N}(\mathbf{y}_t; C\mathbf{x}_t, R)\right]. && (37)
\end{aligned}$$

The complete data log-likelihood can thus be written as the sum of three expectations. The first contains only the initial latent x_1, the second consists of a sum over latent dynamics terms, and the third a sum over likelihood terms. Note that these expectations involve only pairs of adjacent latent vectors, thus evaluating them requires only the joint marginals, $P(\mathbf{x}_t, \mathbf{x}_{t+1} \mid \mathbf{y}_{1:T}, \theta)$, which are provided by the Kalman smoother (Sec. 3.2).

We will now derive and then simplify each of these terms explicitly. The first expectation term, involving x_1, is given by:

$$\begin{aligned}
\mathbb{E}\Big[\log \mathcal{N}(\mathbf{x}_1; 0, Q_0)\Big] &= -\tfrac{1}{2}\mathbb{E}[\mathbf{x}_1^\top Q_0^{-1}\mathbf{x}_1] - \tfrac{1}{2}\log|2\pi Q_0| \\
&= -\tfrac{1}{2}\mathrm{Tr}\left[Q_0^{-1}\mathbb{E}[\mathbf{x}_1\mathbf{x}_1^\top]\right] - \tfrac{1}{2}\log|2\pi Q_0| \\
&= -\tfrac{1}{2}\mathrm{Tr}[Q_0^{-1}(\mathbf{m}_1\mathbf{m}_1^\top + \Sigma_1)] - \tfrac{1}{2}\log|2\pi Q_0|, && (38)
\end{aligned}$$

where we have relied on the "circular shift" trace identity, $\mathrm{Tr}[ABC] = \mathrm{Tr}[BCA]$, and the identity for expectations of quadratic forms under a Gaussian distribution (eq. 122).

The second expectation, containing latent dynamics terms, can be obtained similarly:

$$\mathbb{E}\left[\sum_{t=2}^{T} \log \mathcal{N}(\mathbf{x}_t; A\mathbf{x}_{t-1}, Q)\right] = -\tfrac{1}{2}\mathbb{E}\left[\sum_{t=2}^{T}(\mathbf{x}_t - A\mathbf{x}_{t-1})^\top Q^{-1}(\mathbf{x}_t - A\mathbf{x}_{t-1})\right] - \tfrac{T-1}{2}\log|2\pi Q|$$

$$= -\tfrac{1}{2}\mathrm{Tr}\left[Q^{-1}\sum_{t=2}^{T}\mathbb{E}[\mathbf{x}_t\mathbf{x}_t^\top] - 2Q^{-1}A\sum_{t=2}^{T}\mathbb{E}[\mathbf{x}_{t-1}\mathbf{x}_t^\top] + A^\top Q^{-1}A\sum_{t=1}^{T-1}\mathbb{E}[\mathbf{x}_t\mathbf{x}_t^\top]\right] - \tfrac{T-1}{2}\log|2\pi Q|$$

$$= -\tfrac{1}{2}\mathrm{Tr}\left[Q^{-1}\left(\sum_{t=2}^{T}\mathbf{m}_t\mathbf{m}_t^\top + \Sigma_t\right) - 2Q^{-1}A\left(\sum_{t=2}^{T}\mathbf{m}_{t-1}\mathbf{m}_t^\top + \Sigma_{t-1,t}\right) + A^\top Q^{-1}A\left(\sum_{t=1}^{T-1}\mathbf{m}_t\mathbf{m}_t^\top + \Sigma_t\right)\right] - \tfrac{T-1}{2}\log|2\pi Q|. \quad (39)$$

The third expectation, containing likelihood terms, is given by:

$$\mathbb{E}\left[\sum_{t=1}^{T}\log\mathcal{N}(\mathbf{y}_t; C\mathbf{x}_t, R)\right] = -\tfrac{1}{2}\mathbb{E}\left[\sum_{t=1}^{T}(\mathbf{y}_t - C\mathbf{x}_t)^\top R^{-1}(\mathbf{y}_t - C\mathbf{x}_t)\right] - \tfrac{T}{2}\log|2\pi R|$$

$$= -\tfrac{1}{2}\mathrm{Tr}\left[R^{-1}\sum_{t=1}^{T}\mathbf{y}_t\mathbf{y}_t^\top - 2R^{-1}C\sum_{t=1}^{T}\mathbb{E}[\mathbf{x}_t\mathbf{y}_t^\top] + C^\top R^{-1}C\sum_{t=1}^{T}\mathbb{E}[\mathbf{x}_t\mathbf{x}_t^\top]\right] - \tfrac{T}{2}\log|2\pi R|$$

$$= -\tfrac{1}{2}\mathrm{Tr}\left[R^{-1}\left(\sum_{t=1}^{T}\mathbf{y}_t\mathbf{y}_t^\top\right) - 2R^{-1}C\left(\sum_{t=1}^{T}\mathbf{m}_t\mathbf{y}_t^\top\right) + C^\top R^{-1}C\left(\sum_{t=1}^{T}\mathbf{m}_t\mathbf{m}_t^\top + \Sigma_t\right)\right] - \tfrac{T}{2}\log|2\pi R|. \quad (40)$$

We can combine all three expectations into one simplified expression for $L_{CD}(\theta)$ in terms of the model parameters and sufficient statistics extracted from the data (using the Kalman smoother):

$$\begin{aligned} L_{CD}(\theta) = &-\tfrac{1}{2}\mathrm{Tr}\left[Q_0^{-1}M_{(1,1)}\right] - \tfrac{1}{2}\mathrm{Tr}\left[Q^{-1}M_{(2,T)}\right] + \mathrm{Tr}\left[Q^{-1}AM_\Delta\right] - \tfrac{1}{2}\mathrm{Tr}\left[A^\top Q^{-1}AM_{(1,T-1)}\right] \\ &- \tfrac{1}{2}\mathrm{Tr}\left[R^{-1}Y\right] + \mathrm{Tr}\left[R^{-1}C\tilde{Y}\right] - \tfrac{1}{2}\mathrm{Tr}\left[C^\top R^{-1}CM_{(1,T)}\right] \\ &- \tfrac{1}{2}\log|2\pi Q_0| - \tfrac{T-1}{2}\log|2\pi Q| - \tfrac{T}{2}\log|2\pi R|, \end{aligned} \quad (41)$$

with sufficient statistics given by:

$$M_{(t_1,t_2)} = \sum_{t=t_1}^{t_2}\mathbf{m}_t\mathbf{m}_t^\top + \Sigma_t, \qquad Y = \sum_{t=1}^{T}\mathbf{y}_t\mathbf{y}_t^\top, \quad (42)$$

$$M_\Delta = \sum_{t=2}^{T}\mathbf{m}_{t-1}\mathbf{m}_t^\top + \Sigma_{t-1,t}, \qquad \tilde{Y} = \sum_{t=1}^{T}\mathbf{m}_t\mathbf{y}_t^\top. \quad (43)$$

Thus, the first step for each E-step is to run the Kalman Filter-Smoother to obtain the posterior means $\mathbf{m}_t$, covariances Σ_t and cross-covariances $\Sigma_{t,t+1}$ (eqs. 31-33), which can then be combined into the sufficient statistics above.

4.2 M-step

The M-step seeks to find the maximum of the complete-data-loglikelihood $L_{CD}(\theta)$ as a function of the model parameters θ. For the Gaussian LLDS model, this maximum can be found in closed form by differentiating L_{CD} with respect to θ, setting it to zero, then solving for θ.

Here we provide the partial derivatives of L_{CD} (eq. 35) with respect to each of the model parameters:

$$\frac{\partial}{\partial A}L_{CD} = Q^{-1}M_\Delta^\top - Q^{-1}AM_{(1,T-1)} \quad (44)$$

$$\frac{\partial}{\partial C}L_{CD} = R^{-1}\tilde{Y}^\top - R^{-1}CM_{(1,T)} \quad (45)$$

$$\frac{\partial}{\partial Q^{-1}}L_{CD} = \tfrac{T-1}{2}Q - \tfrac{1}{2}M_{(2,T)} + \tfrac{1}{2}AM_\Delta + \tfrac{1}{2}M_\Delta^\top A^\top - \tfrac{1}{2}AM_{(1,T-1)}A^\top \quad (46)$$

$$\frac{\partial}{\partial R^{-1}} L_{CD} = \tfrac{T}{2} R - \tfrac{1}{2} Y + \tfrac{1}{2} C\tilde{Y} + \tfrac{1}{2}\tilde{Y}^\top C^\top - \tfrac{1}{2} C M_{(1,T)} C^\top \tag{47}$$

$$\frac{\partial}{\partial Q_0^{-1}} L_{CD} = \tfrac{1}{2} Q_0 - \tfrac{1}{2} M_{(1,1)}, \tag{48}$$

which can be obtained using matrix identities found in [8] (Sec. 2.1 and 2.5), and we have simplified the last three equations by differentiating with respect to Q^{-1}, R^{-1}, and Q_0^{-1}. By setting these partial derivatives and solving, we obtain the following M-step updates to the model parameters:

$$\hat{A} = M_\Delta^\top M_{(1,T-1)}^{-1} = \left(\sum_{t=2}^{T} \mathbf{m}_{t-1}\mathbf{m}_t^\top + \Sigma_{t-1,t}\right)^\top \left(\sum_{t=1}^{T-1} \mathbf{m}_t \mathbf{m}_t^\top + \Sigma_t\right)^{-1} \tag{49}$$

$$\hat{C} = \tilde{Y}^\top M_{(1,T)}^{-1} = \left(\sum_{t=1}^{T} \mathbf{y}_t \mathbf{m}_t^\top\right)\left(\sum_{t=1}^{T} \mathbf{m}_t \mathbf{m}_t^\top + \Sigma_t\right)^{-1} \tag{50}$$

$$\hat{Q} = \frac{1}{T-1}\left(M_{(2,T)} - A M_\Delta - M_\Delta^\top A^\top + A M_{(1,T-1)} A^\top\right) \tag{51}$$

$$\hat{R} = \frac{1}{T}\left(Y - C\tilde{Y} - \tilde{Y}^\top C^\top + C M_{(1,T)} C^\top\right) \tag{52}$$

$$\hat{Q}_0 = M_{(1,1)}. \tag{53}$$

Fitting the model involves alternating E and M states until convergence, which is guaranteed to attain a local optimum of the marginal likelihood. Note however that there may be multiple local optima. To obtain the global optimum, it is therefore common to run EM from multiple random initializations, or to use heuristic methods such as subspace identification methods

5 Notes

5.1 Other topics to include

- subspace fitting methods
- identifiability?
- reduced versions: diagonal R or noiseless setting: $R = \epsilon I$ (which is DMD). $C = 0 \implies$ no dynamics (Gaussian mixture model).

5.2 Things to cite:

- Liam's "new look" paper [9]
- Lars' NIPS paper on spectral methods [10]
- Byron's paper on GPFA [11].
- Byron Yu's KF notes [12].
- Max Welling's notes [13].
- KF tutorials: [14, 15].
- Book on classical non-Bayesian treatment: [16].
- Book on Bayesian treatment: [3].

Appendix A: Standard Kalman Filtering Equations

Here we provide the "standard" Kalman filtering equations, which differ from those provided in the main text (eqs. 10-17) and rely on a quantity known as the *Kalman gain*. These formulas can be derived from the Kalman filtering equations in the main text using the matrix inversion lemma (eq. 129), and thus result in identical values for the mean and covariance of the latent state.

However, it should be noted that these "standard" updates require the inversion of an $n \times n$ matrix on each time step (where n is the dimensionality of the observations $\mathbf{y}$), whereas the updates in the main text require inverting a matrix of size $m \times m$ (where m is the dimensionality of the latent state $\mathbf{x}$). It is thus unclear to this author why standard texts seem to favor the version of the Kalman filtering equations provided here.

The "standard" Kalman filtering equations are given by: **Step 1 (Prediction):**

$$\begin{aligned} \mu_{t*} &= A\mu_{t-1} && \text{(KF prior mean)} \qquad (54)\\ V_{t*} &= AV_{t-1}A^\top + Q && \text{(KF prior covariance)} \qquad (55)\end{aligned}$$

Step 2 (Update):

$$\begin{aligned} K_t &= V_{t*}C^\top(CV_{t*}C^\top + R)^{-1} && \text{(Kalman gain)} \qquad (56)\\ \mu_t &= \mu_{t*} + K_t(\mathbf{y}_t - C\mu_{t*}) && \text{(KF posterior mean)} \qquad (57)\\ V_t &= V_{t*} - K_t(CV_{t*}C^\top + R)K_t^\top && \text{(KF posterior covariance).} \qquad (58)\end{aligned}$$

To derive the relationship, we can apply the matrix inversion lemma (eq. 129) to the posterior covariance formula from the main text (eq. 17):

$$\begin{aligned} V_t = (V_{t*}^{-1} + C^\top R^{-1}C)^{-1} &= V_{t*} - V_{t*}C^\top(CV_{t*}C^\top + R)^{-1}CV_{t*} \qquad (59)\\ &= V_{t*} - K_t(CV_{t*}C^\top + R)K_t^\top, \qquad (60)\end{aligned}$$

where K_t is the Kalman gain defined above (eq. 56).

To update the formula for the mean (eq. 57), we can plug this expression for V_t into (eq. 16), and then perform some algebra:

$$\begin{aligned} \mu_t &= V_t(C^\top R^{-1}\mathbf{y}_t + V_{t*}^{-1}\mu_{t*}) \qquad (61)\\ &= (V_{t*} - K_tCV_{t*})(C^\top R^{-1}\mathbf{y}_t + V_{t*}^{-1}\mu_{t*}) \qquad (62)\\ &= \left[(V_{t*} - K_tCV_{t*})C^\top R^{-1}\mathbf{y}_t\right] + \left[(V_{t*} - K_tCV_{t*})V_{t*}^{-1}\mu_{t*}\right] \qquad (63)\\ &= \left[(V_{t*} - V_{t*}C^\top(CV_{t*}C^\top + R)^{-1}CV_{t*})C^\top R^{-1}\mathbf{y}_t\right] + \left[\mu_{t*} - K_tC\mu_{t*}\right] \qquad (64)\\ &= \left[V_{t*}C^\top(I - (CV_{t*}C^\top + R)^{-1}CV_{t*}C^\top)R^{-1}\mathbf{y}_t\right] + \left[\mu_{t*} - K_tC\mu_{t*}\right] \qquad (65)\\ &= \left[V_{t*}C^\top(CV_{t*}C^\top + R)^{-1}\mathbf{y}_t\right] + \left[\mu_{t*} - K_tC\mu_{t*}\right] \qquad (66)\\ &= \mu_{t*} + K_t(\mathbf{y}_t - C\mu_{t*}), \qquad (67)\end{aligned}$$

where on line 66 we used the identity $(I - (A+B)A)^{-1}B^{-1} = (A+B)^{-1}$, which follows from the fact that $I = (A+B)^{-1}(A+B) \implies I - (A+B)^{-1}A = (A+B)^{-1}B \implies$ the identity.

Appendix B: Incorporating inputs

One common extension to the LLDS model defined in (eq. 1-2) is to incorporate linear dependence of both the latents and observations on a set of external inputs signal $\mathbf{u}_t \in \mathbb{R}^d$. The extended model becomes:

$$\mathbf{x}_t = A\mathbf{x}_{t-1} + B\mathbf{u}_t + \mathbf{w}_t, \qquad \mathbf{w}_t \sim \mathcal{N}(0, Q) \qquad \textit{(latents with inputs)} \qquad (68)$$

$$\mathbf{y}_t = \; C\mathbf{x}_t \quad + D\mathbf{u}_t + \mathbf{v}_t, \qquad \mathbf{v}_t \sim \mathcal{N}(0, R) \qquad \textit{(observations with inputs)} \tag{69}$$

where $B \in \mathbb{R}^{m \times d}$ and $D \in \mathbb{R}^{n \times d}$ are matrices capturing the influence of the inputs on the latents and observations, respectively.

B.1 Kalman filter

The addition of inputs affects the Kalman filter only via its mean. First, the prior over the latent $\mathbf{x}_t$ at time step t given previous observations $\mathbf{y}_{1:t-1}$, denoted $\mathcal{N}(\mu_{t*}, V_{t*})$ (eq. 10), has an updated mean (eq. 11) given by:

$$\mu_{t*} = A\mu_{t-1} + B\mathbf{u}_t, \quad \textit{(prior mean w/ inputs)}, \tag{70}$$

while the prior covariance V_{t*} (eq. 12) is unchanged. For the first time bin, we now have a non-zero mean: $\mu_{1*} = B\mathbf{u}_1$.

Second, the inputs affect the posterior mean (the Kalman filter output) via their influence on observations $\mathbf{y}_t$. The posterior over $\mathbf{x}_t$ given observations $\mathbf{y}_{1:t}$, denoted $\mathcal{N}(\mu_t, V_t)$ (eq. 15), has updated mean given by:

$$\mu_t = V_t(C^\top R^{-1}(\mathbf{y}_t - D\mathbf{u}_t) + V_{t*}^{-1}\mu_{t*}), \qquad \textit{(Kalman filter mean w/ inputs)} \tag{71}$$

where again the covariance V_t (eq. 17) remains unchanged.

The marginal likelihood $P(\mathbf{y}_{1:T})$, which is also computed during Kalman filtering, is also changed so that the mean of $\mathbf{y}_t \mid \mathbf{y}_{1:t-1}$ incorporates the effect of the inputs. The terms in the marginal likelihood (eq. 19) become:

$$P(\mathbf{y}_t \mid \mathbf{y}_{1:t-1}) = \mathcal{N}(C\mu_{t*} + D\mathbf{u}_t, CV_{t*}C^\top + R), \tag{72}$$

which is also valid for the first time bin, since $\mu_{1*} = 0$ and $V_{1*} = Q_0$, and thus $P(\mathbf{y}_1) = \mathcal{N}(D\mathbf{u}_1, CQ_0C^\top + R)$.

The inclusion of inputs has no affect on the Kalman smoothing equations.

B.2 Learning parameters via EM

The inclusion of inputs enlarges the Gaussian LLDS model parameters to $\theta = \{A, B, C, D, Q, Q_0, R\}$, and the presence of B and D affect the EM update equations for all other parameters.

With inputs, the complete data log-likelihood (eq. 35) to be computed in the E-step becomes:

$$L_{CD}(\theta) = \; \mathbb{E}\left[\log\mathcal{N}(\mathbf{x}_1; B\mathbf{u}_1, Q_0) + \sum_{t=2}^{T}\log\mathcal{N}(\mathbf{x}_t; A\mathbf{x}_{t-1} + B\mathbf{u}_t, Q) + \sum_{t=1}^{T}\log\mathcal{N}(\mathbf{y}_t; C\mathbf{x}_t + D\mathbf{u}, R)\right] \tag{73}$$

$$\begin{aligned}
&= -\tfrac{1}{2}\mathbb{E}[(\mathbf{x}_1 - B\mathbf{u}_1)^\top Q_0^{-1}(\mathbf{x}_1 - B\mathbf{u}_1)] - \tfrac{1}{2}\log|2\pi Q_0| \\
&\quad - \tfrac{1}{2}\mathbb{E}\left[\sum_{t=2}^{T}(\mathbf{x}_t - A\mathbf{x}_{t-1} - B\mathbf{u}_t)^\top Q^{-1}(\mathbf{x}_t - A\mathbf{x}_{t-1} - B\mathbf{u}_t)\right] - \tfrac{T-1}{2}\log|2\pi Q| \\
&\quad - \tfrac{1}{2}\mathbb{E}\left[\sum_{t=1}^{T}(\mathbf{y}_t - C\mathbf{x}_t - D\mathbf{u}_t)^\top R^{-1}(\mathbf{y}_t - C\mathbf{x}_t - D\mathbf{u}_t)\right] - \tfrac{T}{2}\log|2\pi R| \qquad (74)\\
&= -\tfrac{1}{2}\mathrm{Tr}\left[Q_0^{-1}\left(M_{(1,1)} - 2B\tilde{U}_{(1,1)} + BU_{(1,1)}B^\top\right)\right] - \tfrac{1}{2}\log|2\pi Q_0| - \tfrac{T-1}{2}\log|2\pi Q| - \tfrac{T}{2}\log|2\pi R| \\
&\quad - \tfrac{1}{2}\mathrm{Tr}\left[Q^{-1}\left(M_{(2,T)} + AM_{(1,T-1)}A^\top + BU_{(2,T)}B^\top - 2AM_\Delta - 2B\tilde{U}_{(2,T)} + 2B\tilde{U}_\Delta A^\top\right)\right] \\
&\quad - \tfrac{1}{2}\mathrm{Tr}\left[R^{-1}\left(Y + CM_{(1,T)}C^\top + DU_{(1,T)}D^\top - 2C\tilde{Y} - 2DU_{\mathbf{y}} + 2D\tilde{U}_{(1,T)}C^\top\right)\right], \qquad (75)
\end{aligned}$$

where the sufficient statistics now correspond to:

$$\begin{aligned}
M_{(t_1,t_2)} &= \sum_{t=t_1}^{t_2} \mathbf{m}_t \mathbf{m}_t^\top + \Sigma_t & U_{(t_1,t_2)} &= \sum_{t=t_1}^{t_2} \mathbf{u}_t \mathbf{u}_t^\top & \tilde{U}_{(t_1,t_2)} &= \sum_{t=t_1}^{t_2} \mathbf{u}_t \mathbf{m}_t^\top \\
M_\Delta &= \sum_{t=1}^{T-1} \mathbf{m}_t \mathbf{m}_{t+1}^\top + \Sigma_{t,t+1} & \tilde{U}_\Delta &= \sum_{t=1}^{T-1} \mathbf{u}_{t+1} \mathbf{m}_t^\top & U_{\mathbf{y}} &= \sum_{t=1}^{T} \mathbf{u}_t \mathbf{y}_t^\top \\
Y &= \sum_{t=1}^{T} \mathbf{y}_t \mathbf{y}_t^\top & \tilde{Y} &= \sum_{t=1}^{T} \mathbf{m}_t \mathbf{y}_t^\top .
\end{aligned} \tag{76}$$

The M-step updates for the dynamics model weights A and B, and for the observation model weights C and D can now be computed jointly as:

$$\begin{bmatrix} \hat{A} & \hat{B} \end{bmatrix} = \begin{bmatrix} M_\Delta^\top & \tilde{U}_{(2,T)}^\top \end{bmatrix} \begin{bmatrix} M_{(1,T-1)} & \tilde{U}_\Delta^\top \\ \tilde{U}_\Delta & U_{(2,T)} \end{bmatrix}^{-1} \tag{77}$$

$$\begin{bmatrix} \hat{C} & \hat{D} \end{bmatrix} = \begin{bmatrix} \tilde{Y}^\top & U_{\mathbf{y}}^\top \end{bmatrix} \begin{bmatrix} M_{(1,T)} & \tilde{U}_\Delta(1,T)^\top \\ \tilde{U}_\Delta(1,T) & U_{(1,T)} \end{bmatrix}^{-1} , \tag{78}$$

where we have ignored the contribution of the very first term to B, since this allows us to eliminate the dependence on Q and Q_0 when computing updates for A and B, greatly simplifying the expression in (eq. 77). This is a small price to pay, however, since it concerns only the input contribution to the latent vector in very first time bin.

Finally, the M-step updates for covariances Q, R, and Q_0 are given by:

$$\begin{aligned}
\hat{Q} \;=\; \frac{1}{T-1} \Big(& M_{(2,T)} \;+\; A M_{(1,T-1)} A^\top \;+\; B U_{(2,T)} B^\top \\
& -\; A M_\Delta \;-\; M_\Delta^\top A^\top \;-\; B \tilde{U}_{(2,T)} \;-\; \tilde{U}_{(2,T)}^\top B^\top \;+\; B \tilde{U}_\Delta A^\top \;+\; A \tilde{U}_\Delta^\top B^\top \Big)
\end{aligned} \tag{79}$$

$$\begin{aligned}
\hat{R} \;=\; \frac{1}{T} \Big(& Y \;+\; C M_{(1,T)} C^\top \;+\; D U_{(1,T)} D^\top \\
& -\; C \tilde{Y} \;-\; \tilde{Y}^\top C^\top \;-\; D U_{\mathbf{y}} \;-\; U_{\mathbf{y}}^\top D^\top \;+\; D \tilde{U}_{(1,T)} C^\top \;+\; C \tilde{U}_{(1,T)}^\top D^\top \Big)
\end{aligned} \tag{80}$$

$$\hat{Q}_0 = M_{(1,1)} + B U_{(1,1)} B^\top - B \tilde{U}_{(1,1)} - \tilde{U}_{(1,1)}^\top B^\top . \tag{81}$$

B.3 Identifiability

As discussed in Sec. 2.1, Gaussian LLDS models are only identifiable up to an (invertible) linear transformation of the latent space. When transforming the latent variables by an arbitary invertible linear transformation H, resulting in new latents $\mathbf{x}'_t = H\mathbf{x}_t$, we can obtain an equivalent model by transforming the input matrix B according to $B' = HB$. There is no need to transform D.

Appendix C: Combining data from multiple experiments

In the main text we have assumed that the data consisted of a single time series of length T. However, it is common to fit a single LLDS model to data from multiple experiments, consisting of independent time series $\{\mathbf{y}_{1:T_1}^{(1)}, \ldots, \mathbf{y}_{1:T_k}^{(k)}\}$ of lengths $T_1, \ldots, T_k$.

As in the original derivation, fitting the model can proceed via EM. The E-step involves running the Kalman Filter-Smoother on each independent time series to obtain the posterior mean and covariance of the latents, which can be performed in parallel across datasets.

For the M-step, we simply sum the complete data log-likelihood (eq. 41 or eq. 75) across datasets before differentiating and solving for the M-step updates.

For the model without inputs, the M-step updates (originally given in eq. 49-53) become:

$$\hat{A} = \left(\sum_{i=1}^{k} M_\Delta^{(i)}\right)^\top \left(\sum_{i=1}^{k} M_{(1,T_i-1)}^{(i)}\right)^{-1} \tag{82}$$

$$\hat{C} = \left(\sum_{i=1}^{k} \tilde{Y}^{(i)}\right)^\top \left(\sum_{i=1}^{k} M_{(1,T_i)}^{(i)}\right)^{-1} \tag{83}$$

$$\hat{Q} = \frac{1}{\sum_{i=1}^{k}(T_i - 1)} \left(\sum_{i=1}^{k} M_{(2,T_i)}^{(i)} - AM_\Delta^{(i)} - {M_\Delta^{(i)}}^\top A^\top + AM_{(1,T_i-1)}^{(i)} A^\top\right) \tag{84}$$

$$\hat{R} = \frac{1}{\sum_{i=1}^{k} T_i} \left(Y^{(i)} - C\tilde{Y}^{(i)} - (\tilde{Y}^{(i)})^\top C^\top + CM_{(1,T_i)}^{(i)} C^\top\right) \tag{85}$$

$$\hat{Q}_0 = \frac{1}{k}\left(\sum_{i=1}^{k} M_{(1,1)}^{(i)}\right), \tag{86}$$

using sufficient statistics $M^{(i)}$, $M_\Delta^{(i)}$, $Y^{(i)}$, $\tilde{Y}^{(i)}$ computed from each dataset (eqs. 42-43).

Note that the formulas provided here are the nearly same as using the original M-step updates (eq. 49-53), but with sufficient statistics replaced by their sums across datasets; the only exception is the updates for the covariance matrices Q, R, and Q_0, which must take account of the number of datasets k and number of time points T_i in each.

Thus, given multiple time series in a model with inputs and parameters C and D, the the M-step updates are identical to those in Appendix B (eqs. 77-81), but with denominators of $\sum_{i=1}^{k}(T_i - 1)$, $\sum_{i=1}^{k} T_i$ and k for Q, R, and Q_0, respectively.

Appendix D: Incorporating priors

The EM algorithm converges to a maximum of the log-likelihood function. However, in cases where data is limited relative to the number of parameters, it may be desirable to regularize our estimates by incorporating a prior over the model parameters. Mathematically, this is equivalent to adding a penalty (equal to the log-prior) to the complete-data log-likelihood. This penalty will alter the M-step updates, and the EM algorithm will now converge to a maximum of the posterior, thereby producing a *maximum a posteriori* (MAP) estimate of the model parameters.

For the dynamics matrix A and observation matrix C, it is straightforward to incorporate a Gaussian prior, which is equivalent to adding a quadratic penalty to the complete data log-likelihood. Suppose we have zero-mean Gaussian priors:

$$A_{[j,:]} \sim \mathcal{N}(0, \Lambda_A) \tag{87}$$

$$C_{[j,:]} \sim \mathcal{N}(0, \Lambda_C), \tag{88}$$

where $A_{[j,:]}$ is the jth row of A, $C_{[j,:]}$ is the jth row of C, and Λ_A and Λ_C represent $m \times m$ and $n \times n$ covariance matrices, respectively.

If we assume we have the same prior over all rows of A and over all rows of C, this is equivalent to adding the terms $-\frac{1}{2}\mathrm{Tr}[A\Lambda_A^{-1}A^\top]$ and $-\frac{1}{2}\mathrm{Tr}[C\Lambda_C^{-1}C^\top]$ to the complete data log-likelihood (eq. 41). The gradients of the penalized complete-data log-likelihood with respect to A and C are now:

$$\frac{\partial}{\partial A} L_{CD} = Q^{-1} M_{\Delta}^{\top} - Q^{-1} A M_{(1,T-1)} - A \Lambda_A^{-1} \tag{89}$$

$$\frac{\partial}{\partial C} L_{CD} = R^{-1} \tilde{Y}^{\top} - R^{-1} C M_{(1,T)} - C \Lambda_C^{-1}. \tag{90}$$

Note that if the dataset consists of k independent time series (as discussed in Appendix C), then in the above equations we set the sufficient statistics equal to their sum across datasets, that is: $M_{\Delta} = \sum_{i=1}^{k} M_{\Delta}^{(i)}$, $M_{(1:\tau)} = \sum_{i=1}^{k} M_{(1,\tau)}^{(i)}$, and $\tilde{Y} = \sum_{i=1}^{k} \tilde{Y}^{(i)}$.

If Q or R is diagonal, then we have the following closed-form M-step updates for each row of A or C:

$$\hat{A}_{[j,:]} = \left(M_{\Delta}^{\top}\right)_{[j,:]} \left(M_{(1,T-1)} + Q_{jj} \Lambda_A^{-1}\right)^{-1} \tag{91}$$

$$\hat{C}_{[j,:]} = \left(\tilde{Y}^{\top}\right)_{[j,:]} \left(M_{(1,T)} + R_{jj} \Lambda_C^{-1}\right)^{-1}, \tag{92}$$

where Q_{ii} and R_{ii} indicate the diagonal elements of the (diagonal) covariances Q and R, respectively, and $X_{[j,:]}$ indicates the j'th row of a matrix X.

Note that if either prior covariance is proportional to the identity, $\Lambda_A = \sigma_A^2 I$ or $\Lambda_C = \sigma_C^2 I$, then the penalty temrs above are equal to $\frac{Q_{jj}}{\sigma_A^2} I$ or $\frac{R_{jj}}{\sigma_C^2} I$, respectively. This is equivalent to ridge regression with ridge parameter $\lambda = \frac{Q_{jj}}{\sigma_A^2}$ or $\lambda = \frac{R_{jj}}{\sigma_C^2}$, so the updates for each row are given by:

$$\hat{A}_{[j,:]} = \left(M_{\Delta}^{\top}\right)_{[j,:]} \left(M_{(1,T-1)} + \left(\frac{Q_{jj}}{\sigma_A^2}\right) I\right)^{-1} \tag{93}$$

$$\hat{C}_{[j,:]} = \left(\tilde{Y}^{\top}\right)_{[j,:]} \left(M_{(1,T)} + \left(\frac{R_{jj}}{\sigma_C^2}\right) I\right)^{-1}, \tag{94}$$

although note that the ridge parameter for each row of A or C differs depending on the noise variance Q_{jj} or R_{jj} for that latent or observed dimension, respectively.

In the case that we do not wish to restrict Q or R to diagonal, we can use Kronecker identities to derive the vectorized M-step update for A and C. Specifically, we use the identity $\mathrm{vec}(ABC) = (C^{\top} \otimes A)\mathrm{vec}(B)$, where $\mathrm{vec}(\cdot)$ is the "vectorize" operator that transforms a matrix into a vector by stacking its columns. With this identity, we can set the partial derivatives (eqs. 89-90) to zero and obtain the following solutions:

$$\mathrm{vec}(\hat{A}) = \left(M_{(1,T-1)}^{\top} \otimes I + \Lambda_A^{-1} \otimes Q\right)^{-1} \mathrm{vec}\left(M_{\Delta}^{\top}\right) \tag{95}$$

$$\mathrm{vec}(\hat{C}) = \left(M_{(1,T)}^{\top} \otimes I + \Lambda_C^{-1} \otimes R\right)^{-1} \mathrm{vec}\left(\tilde{Y}^{\top}\right). \tag{96}$$

Note that these solutions require inverting an $m^2 \times m^2$ matrix for $\hat{A}$ and and an $mn \times mn$ matrix for $\hat{C}$, which have computational costs of $O(m^6)$ and $(Om^3 n^3)$, respectively. This may be prohibitive when the latent dimensionality m or observation dimensionality n is large. In such cases, it may be more efficient to use conjugate gradient methods to numerically minimize (or approximately minimize) the complete-data log-likelihood for A and C using the gradients in (eqs. 89-90).

D.1 Priors for the LLDS model with inputs

To add: priors on A, B, C, D

Appendix E: Speeding up inference

Appendix F: Structured assumptions

F.1 Diagonal covariances

F.2 Diagonal input weights

F.3 Block structure in A

Appendix G: Missing data

Appendix H: Implementation using block-sparse matrices

Here we describe the LLDS model using block-sparse matrices and vectors of concatenated observations and latents, which facilitates efficient computations of marginal likelihood and posterior distribution.

H.1 Expressing the model with block-sparse matrices

Let us begin by stacking the latents $\{\mathbf{x}_t\}$, observations $\{\mathbf{y}_t\}$, latent noise $\{\mathbf{w}_t\}$, and observed noise $\{v_t\}$ for all time steps $t \in \{1, \ldots, T\}$ into four gigantic column vectors $\mathbf{x}$, $\mathbf{y}$, $\mathbf{w}$, $\mathbf{x}$, respectively. Thus, for example, we have

$$\mathbf{x} = \begin{bmatrix} \mathbf{x}_1 \\ \vdots \\ \mathbf{x}_T \end{bmatrix}. \tag{97}$$

Then we can rewrite the LLDS model in matrix form:

$$\mathbf{H}\mathbf{x} = \mathbf{w} \tag{98}$$

$$\mathbf{y} = \mathbf{C}\mathbf{x} + \mathbf{v}, \tag{99}$$

where $\mathbf{H}$ is a block bi-diagonal matrix given by

$$\mathbf{H} = \begin{bmatrix} I & & & & \\ -A & I & & & \\ & -A & I & & \\ & & \ddots & \ddots & \\ & & & -A & I \end{bmatrix}, \tag{100}$$

constructed so that the t'th row-block expresses the condition that $\mathbf{x}_t - A\mathbf{x}_{t-1} = \mathbf{w}_t$, and $\mathbf{C}$ is a block diagonal matrix with copies of the observation matrix C along the main diagonal:

$$\mathbf{C} = I_T \otimes C \triangleq \begin{bmatrix} C & & \\ & \ddots & \\ & & C \end{bmatrix}, \tag{101}$$

where $I_T \otimes C$ denotes Kronecker product between the $T \times T$ identity matrix and C.

We can also compactly write the distributions over the noise vectors:

$$\mathbf{w} \sim \mathcal{N}(0, \mathbf{Q}), \quad \mathbf{Q} = \begin{bmatrix} Q_0 & & & \\ & Q & & \\ & & \ddots & \\ & & & Q \end{bmatrix} \tag{102}$$

$$\mathbf{v} \sim \mathcal{N}(0, \mathbf{R}), \quad \mathbf{R} = I_T \otimes R, \tag{103}$$

where the block structure of mQ accounts for the fact that Q_0 is the prior over the initial latent.

H.2 Prior, Conditional, Posterior, & Marginal

Now, we can derive a simple expressions for the prior, conditional, marginal and posterior distributions.

First, the dynamics equation (eq. 98) and the distribution for latent noise $\mathbf{w}$ (eq. 102) implies that $\mathbf{Hx} \sim \mathcal{N}(0, \mathbf{Q})$. This implies that we can write the prior over the latent as a linear transformation of the latent noise:

$$\mathbf{x} \sim \mathcal{N}(0, \tilde{\mathbf{Q}}), \qquad \textit{(prior)} \tag{104}$$

$$\tilde{\mathbf{Q}} = \mathbf{H}^{-1}\mathbf{Q}\mathbf{H}^{-\top} = (\mathbf{H}^\top \mathbf{Q}^{-1}\mathbf{H})^{-1}. \tag{105}$$

While the prior covariance $\tilde{\mathbf{Q}}$ is a full $mT \times mT$ matrix, the right-most expression on the second line shows that its inverse $\tilde{\mathbf{Q}}^{-1}$ is the block-tridiagonal matrix $\mathbf{H}^\top \mathbf{Q}^{-1}\mathbf{H}$, which is requires only linear memory in T.

Second, the observation equation (eq. 99) implies that the conditional distribution of $\mathbf{y}$ given $\mathbf{x}$ can be written

$$\mathbf{y} \mid \mathbf{x} \sim \mathcal{N}(\mathbf{Cx}, \mathbf{R}). \qquad \textit{(conditional)} \tag{106}$$

This allows us to apply the generic formula for linear-Gaussian systems (Gaussian Fun Fact #4, eq. 126), which implies that the posterior distribution (which is also the output of the Kalman smoother) is:

$$\mathbf{x} \mid \mathbf{y} = \mathcal{N}\Big((\mathbf{C}^\top \mathbf{R}^{-1}\mathbf{C} + \tilde{\mathbf{Q}}^{-1})^{-1}\mathbf{C}^\top \mathbf{R}^{-1}\mathbf{y}, (\mathbf{C}^\top \mathbf{R}^{-1}\mathbf{C} + \tilde{\mathbf{Q}}^{-1})^{-1}\Big). \qquad \textit{(posterior)} \tag{107}$$

Note that the posterior covariance is a full matrix, but its inverse is once-again tri-diagonal.

Finally, the marginal likelihood can be derived by observing that $\mathbf{y}$ is a linear transformation of $\mathbf{x}$ plus Gaussian noise (Gaussian fun fact #1, eq. 119):

$$\mathbf{y} \sim \mathcal{N}(0, \mathbf{C}\tilde{\mathbf{Q}}\mathbf{C}^\top + \mathbf{R}). \qquad \textit{(marginal)} \tag{108}$$

H.3 Efficient evaluation

Log marginal likelihood

The log of the marginal-likelihood (eq. 107), which is used for fitting θ or evaluating test performance, can be written explicitly as:

$$\log P(\mathbf{y}) = -\tfrac{1}{2}\log|\boldsymbol{\Lambda}| - \tfrac{1}{2}\mathbf{y}^\top \boldsymbol{\Lambda}^{-1}\mathbf{y} - \tfrac{T}{2}\log(2\pi), \qquad \textit{(log marginal likelihood)} \tag{109}$$

where $\boldsymbol{\Lambda} = \mathbf{C}\tilde{\mathbf{Q}}\mathbf{C}^\top + \mathbf{R}$ is the marginal covariance of $\mathbf{y}$ given θ. However, $\boldsymbol{\Lambda}$ is a $mT \times mT$ matrix, which requires $O(m^2T^2)$ storage and $O(m^3T^3)$ computational cost to invert, which is infeasible for even medium-length datasets.

We can achieve dramatically higher efficiency by exploiting the special block-diagonal structure of the matrices $\mathbf{C}$, $\mathbf{R}$, and $\tilde{\mathbf{Q}}^{-1}$. First, we can use the matrix inversion lemma (eq. 129) to write the inverse covariance:

$$\boldsymbol{\Lambda}^{-1} = (\mathbf{C}\tilde{\mathbf{Q}}\mathbf{C}^\top + \mathbf{R})^{-1} = \mathbf{R}^{-1} - \mathbf{R}^{-1}\mathbf{C}(\tilde{\mathbf{Q}}^{-1} + \mathbf{C}^\top \mathbf{R}^{-1}\mathbf{C})^{-1}\mathbf{C}^\top \mathbf{R}^{-1}. \tag{110}$$

This allows us to evaluate the quadratic term in (eq. 109) as:

$$\mathbf{y}^\top \mathbf{\Lambda}^{-1}\mathbf{y} = \mathbf{y}^\top \mathbf{R}^{-1}\mathbf{y} - \mathbf{y}^\top \mathbf{R}^{-1}\mathbf{C}(\tilde{\mathbf{Q}}^{-1} + \mathbf{C}^\top \mathbf{R}^{-1}\mathbf{C}) \setminus (\mathbf{C}^\top \mathbf{R}^{-1}\mathbf{y}), \qquad \textit{(quadratic term)} \qquad (111)$$

where '$B \backslash A$' denotes "matrix left divide" of A by B, which can be implemented via `B \ A` in Matlab, and by `numpy.linalg.solve(A,B)` in python. This operation is equivalent to $B^{-1}A$, but far more efficient as it does not require explicitly forming the inverse of B, and is only $O(T)$ complexity for the block-tridiagonal matrix $(\tilde{\mathbf{Q}}^{-1} + \mathbf{C}^\top \mathbf{R}^{-1}\mathbf{C})$.

Secondly, we can use the matrix-determinant lemma (eq. 130) to efficiently evaluate the log-determinant term in (eq. 109):

$$\begin{aligned} \log|\Lambda| &= \log|\mathbf{C}\tilde{\mathbf{Q}}\mathbf{C}^\top + \mathbf{R}| = \log\Big(|\tilde{\mathbf{Q}}||\mathbf{R}||\tilde{\mathbf{Q}}^{-1} + \mathbf{C}^\top \mathbf{R}^{-1}\mathbf{C}|\Big) \\ &= -\log|\tilde{\mathbf{Q}}^{-1}| - \log|\mathbf{R}| + \log|\tilde{\mathbf{Q}}^{-1} + \mathbf{C}^\top \mathbf{R}^{-1}\mathbf{C}|, \qquad \textit{(log-determinant term)} \qquad (112) \end{aligned}$$

where $\mathbf{R}$ is block-diagonal and $\tilde{\mathbf{Q}}^{-1}$ and $(\tilde{\mathbf{Q}}^{-1} + \mathbf{C}^\top \mathbf{R}^{-1}\mathbf{C})$ are both block-tridiagonal matrices whose determinants can be obtained in $O(T)$ time.

Posterior mean and covariance

We can efficiently compute the posterior mean over the latents given the data (eq. 107 by once again using the 'backslash' operator to avoid explicitly inverting an $mT \times mT$ matrix:

$$\mathbb{E}[\mathbf{x} \mid \mathbf{y}] = (\mathbf{C}^\top \mathbf{R}^{-1}\mathbf{C} + \tilde{\mathbf{Q}}^{-1}) \setminus \mathbf{C}^\top \mathbf{R}^{-1}\mathbf{y}. \qquad \textit{(posterior mean)} \qquad (113)$$

As noted above, the posterior covariance over the latents given the data, $(\mathbf{C}^\top \mathbf{R}^{-1}\mathbf{C} + \tilde{\mathbf{Q}}^{-1})^{-1}$, is a full $mT \times mT$ matrix, but its inverse, $\mathbf{C}^\top \mathbf{R}^{-1}\mathbf{C} + \tilde{\mathbf{Q}}^{-1}$, is symmetric and block-tridiagonal. We can exploit this structure to efficiently compute the main and above-diagonal blocks of the posterior covariance, which are useful for quantifying uncertainty in the latents or for the EM algorithm discussed below.

Specifically, the diagonal blocks Σ_t and immediate above-diagonal blocks $\Sigma_{t,t+1}$ of the posterior covariance can be computed in linear time via an efficient recursion [17]. A Matlab implementation of this algorithm is available at https://github.com/pillowlab/invBlockTriDiag.

H.4 Incorporating inputs

Let $\mathbf{u}_x$ denote the vector obtained by vectorizing and stacking the linearly projected inputs $B\mathbf{u}_t$ from all time bins, and $\mathbf{u}_y$ denote the vector obtained by vectorizing and stacking the linearly projected inputs $D\mathbf{u}_t$ from all time bins.

With inputs, the prior on the vectorized latents (eq. 104) becomes

$$\mathbf{x} \sim \mathcal{N}(\mathbf{H}^{-1}\mathbf{u}_x, \tilde{\mathbf{Q}}), \qquad (114)$$

the conditional distribution of the observations (eq. 106) becomes

$$\mathbf{y} \mid \mathbf{x} \sim \mathcal{N}(\mathbf{C}\mathbf{H}^{-1}\mathbf{x} + \mathbf{u}_y, \mathbf{R}), \qquad (115)$$

and the posterior over the latents (eq. 107) becomes

$$\mathbf{x} \mid \mathbf{y} = \mathcal{N}\Big(\mathbf{\Sigma}(\mathbf{C}^\top \mathbf{R}^{-1}(\mathbf{y} - \mathbf{u}_y) + \tilde{\mathbf{Q}}^{-1}\mathbf{u}_x)\,,\, \mathbf{\Sigma}\Big), \qquad (116)$$

where the posterior covariance $\mathbf{\Sigma} = (\mathbf{C}^\top \mathbf{R}^{-1}\mathbf{C} + \tilde{\mathbf{Q}}^{-1})^{-1}$ is unchanged from the version of the model without inputs.

Finally, the marginal likelihood (eq. 108) becomes:

$$\mathbf{y} \sim \mathcal{N}(\mathbf{C}\mathbf{H}^{-1}\mathbf{u}_x + \mathbf{u}_y, \mathbf{C}\tilde{\mathbf{Q}}\mathbf{C}^\top + \mathbf{R}). \qquad (117)$$

Appendix I: Useful Identities

I.1 Gaussian Fun Facts

Here we provide a list of Gaussian identities or “fun facts”, which are useful for the derivations in the paper.

Fact 1: Sums.

$$\begin{aligned} & x \sim \mathcal{N}(a, A), \quad y \sim \mathcal{N}(b, B) \\ \implies \quad & x + y \sim \mathcal{N}(a + b, A + B). \end{aligned} \tag{118}$$

Fact 2: Linear Transformations.

$$\begin{aligned} & x \sim \mathcal{N}(a, A), \quad y = Cx \\ \implies \quad & y \sim \mathcal{N}(a, CAC^\top). \end{aligned} \tag{119}$$

Fact 3: Products of Gaussian densities.

$$\mathcal{N}(a, A) \cdot \mathcal{N}(b, B) \propto \mathcal{N}(c, C) \tag{120}$$

$$\text{where} \quad C = (A^{-1} + B^{-1})^{-1}, \quad c = C(A^{-1}a + B^{-1}b).$$

Fact 4: Conditionals.

$$\begin{aligned} \text{If} \quad & \begin{bmatrix} x \\ y \end{bmatrix} \sim \mathcal{N}\left(\begin{bmatrix} a \\ b \end{bmatrix}, \begin{bmatrix} A & C \\ C^T & B \end{bmatrix} \right) \\ \text{then} \quad & x \mid y \sim \mathcal{N}\left(a + CB^{-1}(y - b), A - CB^{-1}C^\top\right) \\ & y \mid x \sim \mathcal{N}\left(b + C^\top A^{-1}(x - a), B - C^\top A^{-1}C\right). \end{aligned} \tag{121}$$

Fact 5: Expectations of quadratic forms

$$\begin{aligned} & x \sim \mathcal{N}(a, A) \\ \text{then} \quad & \mathbb{E}[(x-b)^\top B(x-b)] = (a-b)^\top B(a-b) + \mathrm{Tr}[BA] \\ & = \mathrm{Tr}[B(a-b)(a-b)^\top + BA]. \end{aligned} \tag{122}$$

Fact 6: The Linear-Gaussian model.

Here we consider a linear-Gaussian system defined by a Gaussian prior over a latent variable x, which is transformed linearly and corrupted by additive Gaussian noise to obtain an observation y:

$$x \sim \mathcal{N}(a, A) \tag{123}$$

$$y = Cx + n, \quad n \sim \mathcal{N}(0, R), \tag{124}$$

where the second equation is equivalent to writing $y \mid x \sim \mathcal{N}(Cx, R)$. First, we can use the Gaussian fun facts above (eq. 118-119) to obtain the marginal distribution over y:

$$y \sim \mathcal{N}(Ca, CAC^\top + R) \tag{125}$$

We can also use “completing the square” to derive the posterior distribution over x given y:

$$x \mid y \sim \mathcal{N}(d, D), \tag{126}$$

where covariance and mean are given by

$$D = (C^\top R^{-1} C + A^{-1})^{-1}, \quad d = D(C^\top R^{-1} y + A^{-1} a). \tag{127}$$

Lastly, the joint distribution over x and y is given by:

$$\begin{bmatrix} x \\ y \end{bmatrix} = \mathcal{N}\left(\begin{bmatrix} a \\ Ca \end{bmatrix}, \begin{bmatrix} A & AC^\top \\ CA & CAC^\top + R \end{bmatrix} \right). \tag{128}$$

I.2 Matrix Identities

Matrix Inversion Lemma

If A and B are invertible matrices, then:

$$(A^{-1} + XB^{-1}X^\top)^{-1} = A - AX(B + X^\top AX)^{-1} X^\top A \tag{129}$$

Matrix Determinant Lemma

If A and B are invertible matrices, then:

$$\det|A + XBX^\top| = \det|A| \det|B| \det|X^\top A^{-1} X + B^{-1}| \tag{130}$$

References

[1] R. E. Kalman. A new approach to linear filtering and prediction problems. *Journal of Basic Engineering*, 82(1):35–45, 03 1960.

[2] R. E. Kalman and R. S. Bucy. New results in linear filtering and prediction theory. *Transactions of the ASME, Journal of Basic Engineering*, 83(3):95–108, 1961.

[3] Simo Särkkä. *Bayesian filtering and smoothing*, volume 3. Cambridge University Press, 2013.

[4] Joao P Hespanha. *Linear systems theory*. Princeton university press, 2018.

[5] A. Dempster, N. Laird, and R. Rubin. Maximum likelihood from incomplete data via the EM algorithm. *J. Royal Statistical Society, B*, 39(1):1–38, 1977.

[6] C. M. Bishop. *Pattern recognition and machine learning*. Springer New York:, 2006.

[7] Kevin P Murphy. *Machine learning: a probabilistic perspective*. MIT press, 2012.

[8] Kaare Brandt Petersen and Michael Syskind Pedersen. The matrix cookbook. *Technical University of Denmark*, 7(15):510, 2008.

[9] Liam Paninski, Yashar Ahmadian, Daniel Gil Ferreira, Shinsuke Koyama, Kamiar Rahnama Rad, Michael Vidne, Joshua Vogelstein, and Wei Wu. A new look at state-space models for neural data. *J Comput Neurosci*, 1:107–126, 2010.

[10] Lars Buesing, Jakob Macke, and Maneesh Sahani. Spectral learning of linear dynamics from generalised-linear observations with application to neural population data. In *Advances in Neural Information Processing Systems 25*, pages 1691–1699, 2012.

[11] B. M. Yu, J. P. Cunningham, G. Santhanam, S. I. Ryu, K. V. Shenoy, and M. Sahani. Gaussian-process factor analysis for low-dimensional single-trial analysis of neural population activity. *Journal of Neurophysiology*, 102(1):614, 2009.

[12] Byron M. Yu, K. V. Shenoy, and M. Sahani. Derivation of Kalman filtering and smoothing equations. Technical report, Stanford University, 2004.

[13] Max Welling. The kalman filter. Technical report, California Institute of Technology, 2010.

[14] Ramsey Faragher et al. Understanding the basis of the kalman filter via a simple and intuitive derivation. *IEEE Signal processing magazine*, 29(5):128–132, 2012.

[15] A.L. Barker, D.E. Brown, and W.N. Martin. Bayesian estimation and the kalman filter. *Computers & Mathematics with Applications*, 30(10):55 – 77, 1995.

[16] Ali H Sayed. *Adaptive filters*. John Wiley & Sons, 2011.

[17] G. B. Rybicki and D. G. Hummer. An accelerated lambda iteration method for multilevel radiative transfer. i-non-overlapping lines with background continuum. *Astronomy and Astrophysics*, 245:171–181, 1991. Appendix B.