Probabilistic models for neural data
Session 7: State space models #2

To do before this session:

- **Read Macke et al. (2015) and prepare assigned presentation**
- **Read up on this session's statistical concepts**
- **Complete pre-session quiz**

In this session we will make our first venture into state space models for neural data. These combine all the concepts of the previous sessions. First, they assume that the observed high-dimensional neural population activity is generated from some low-dimensional latent state, thus invoking dimensionality reduction. Second, they assume the latent dynamical state to evolve according to some stochastic process, linking them to time series models. Third, particular model variants (including the one we discuss in this session) assume that the relation between the observed neural activity and the latent state is described by a GLM.

This session's paper introduces the simplest (useful) state space model variant, in which the latent state evolves according to some linear-Gaussian process, and the neural population activity is modeled by an exponential-Poisson GLM with the latent state as input. The discussed paper is mostly theoretical and only evaluates the model on simulated data. Nonetheless, it provides the foundation for the following two weeks, in which we will discuss more complex state space models and their application to (this time, real) neural data.

After that, we will prepare next week's paper discussion by introducing Gaussian processes. Next week's paper assumes that the time-evolution of lower-dimensional latent states is modeled by such processes. These processes provide priors over continuous functions and are able to mimic a wide range of stochastic processes by relatively minor modifications of their covariance kernels. Thus, they form the basis for state space models that can capture a potentially wide range of latent state dynamics.

Note that the major purpose of neural state space models is data visualization and, sometimes, model comparison. They are particularly helpful for making sense of complex datasets in order to develop novel hypotheses. These hypotheses are then usually tested through other means.

Paper: Macke et al. (2015). Estimating state and parameters in state space models of spike trains

The paper uses various approximation schemes to make posterior inference and parameter estimation tractable. We only discuss a subset of these schemes as statistical concepts, and you can skip the paper sections that describe the non-discussed schemes (i.e., variational inference, Sec. 3.2; spectral learning, Sec. 4.2). When reading the paper, please focus on the following:

- What is the benefit of choosing a Gaussian-linear model to describe the latent state dynamics?
- What is their motivation for choosing a Poisson rather than a Gaussian model for the (binned) spiking neural population data?
- Why can the posterior not be computed in closed-form?
- What is the difference between the population state $\boldsymbol{x}_t$ and the pre-intensity $\boldsymbol{z}_t$?
- What is the difference between $\boldsymbol{s}_t$ and $\boldsymbol{u}_t$ in their model?
- Why does their model predict Fano factors larger than one?
- What makes their model parameters unidentifiable (think about the similarities to PCA)?
- (Advanced) under which conditions does their model become equivalent to a standard GLM?

Section 3.1 of the paper describes finding the posterior by the Laplace approximation. You should understand the general idea of the procedure and the assumptions made, but can skip the exact details of how the mathematical expressions are efficiently computed (e.g., most of page 9). The same applies to the application of the EM algorithm to find the maximum likelihood estimates of the parameters, where you again should understand the general idea, but can skip the detailed mathematical expressions to implement the EM algorithm (e.g., most of page 14). As mentioned further above, feel free to skip Sections 3.2 and 4.2, as they use statistical concepts that we haven't previously discussed.

Once you have read the paper, try to draw the graphical model that represents the model's structure. You can start with that of a HMM as a starting point, and then identify points of divergence (particularly when including the $\boldsymbol{s}_t$'s and $\boldsymbol{u}_t$'s). Try to relate this graphical model to that of Markov chains and of GLMs.

Presentations:

1. The model (single time step *t*; Secs. 1&2)
    a. Describe the data $\boldsymbol{y}_{1:T}$ that is being modeled.
    b. Describe how the data $\boldsymbol{y}_t$ is assumed to be generated by the corresponding pre-intensities $\boldsymbol{z}_t$ (Eq. (2)). How does this assumption relate to GLMs?
    c. Describe how the pre-intensities $\boldsymbol{z}_t$ relate the latent state $\boldsymbol{x}_t$ and the stimulus $\boldsymbol{s}_t$ (Eq. (1)). What are the dimensionalities about the involved vectors, and what assumptions does this imply?
    d. What are possible interpretations of the latent state $\boldsymbol{x}_t$ and the stimulus $\boldsymbol{s}_t$ (Secs. 1&2)?
2. The model (linking time steps *1:T*; Secs. 1&2)
    a. Describe how the model assumes the latent state $\boldsymbol{x}_t$ to evolve over time (Eq. (3)). How does this assumption differ from the one underlying the latent state transitions in the Kalman filter?
    b. What is the role of $\boldsymbol{u}_t$ in these dynamics?
    c. Try to draw the graphical model that describes this model (Hint: start with the one for an HMM, and then add the parts that aren't yet present).
    d. Why does the model feature Fano-Factors larger than one?

    e. What do they assume about $\boldsymbol{u}_t$ and $\boldsymbol{s}_t$ for the remainder of the chapter (end of Sec. 2)?
3. Inferring states by the Laplace approximation (Secs. 3 & 3.1)
    a. Describe the need to approximate $p(\boldsymbol{x}_{1:T}|\boldsymbol{y}_{1:T})$, and the form of approximation $q(\boldsymbol{x})=\Phi(\boldsymbol{x}|\boldsymbol{\mu},\boldsymbol{\Sigma})$ they choose (Eq. (4)).
    b. Describe how $\boldsymbol{\mu}$ and $\boldsymbol{\Sigma}$ are determined by the Laplace approximation in the context of this work.
    c. Without going through the details of the math, what properties of the log-posterior make it easy to find its maximum to determine $\boldsymbol{\mu}$? What properties allow the Hessian to be computed efficiently?
    d. (Advanced) If they approximate the observation likelihood $p(\boldsymbol{y}_t|\boldsymbol{x}_t)$ by a Laplace approximation (Gaussian in $\boldsymbol{x}_t$) for each time bin t separately rather than globally across all time bins, which algorithm could they have used instead of the Laplace approximation to find $p(\boldsymbol{x}_{1:T}|\boldsymbol{y}_{1:T})$ (Hint: think of the relationship to the Kalman filter)?
4. Inferring parameters by the EM algorithm (Secs. 4 & 4.1; focus only on the IEM variant)
    a. The EM algorithm relies on three components: the observed data, the latent states, and model parameters. Identify each of these components in the discussed model.
    b. Having identified those, what is the “incomplete-data” likelihood that we aim to maximize, and the “complete-data” likelihood that is used in the EM algorithm for the maximization (revisit PRML & session 4 slides)?
    c. Describe (conceptually) the E and M steps. What is the posterior that is computed in the E step for this model? How is this posterior used in the M step to computed updated parameters for this model?
5. Simulation results (latent state estimation; Sec. 5 & Fig. 1)
    a. Describe how the data was generated to test the performance of PLDS.
    b. Why did the authors first check the quality of the latent state estimates while setting the parameters to their known ground-truth values? What does Fig. 1a show?
    c. Describe Fig. 1b. How do the authors compare the quality of the Laplace approximation to the Variational Bayes approximation? Why, after an initial divergence of the performance of the two approximations, do they become more similar again for large q?
6. Simulation results (parameter estimation; Sec. 5 & Fig. 2)
    a. (Advanced) Describe why the model parameters of state-space models are not fully identifiable (Hint: it’s related to the non-identifiability of the latent space rotation in probabilistic PCA).
    b. Describe the invariant parameter properties that they use to assess parameter recovery.
    c. Describe Fig. 2b-d. Note that Fig. 2d is based on the asymptotic covariance for $\boldsymbol{x}_t$ that (see Sec. 2), through Eq. (1), determines the asymptotic covariance of $\boldsymbol{z}_t$, and, in turn, of $\boldsymbol{y}_t$.

Statistical concepts: Kernel methods and Gaussian processes

The next session's paper describes the time-evolution of the latent states by Gaussian processes, which describe priors over functions by specifying the correlation between function values $f(x)$ and $f(y)$ as a function of the distance in the function arguments x and y. These correlations are specified through kernel functions which we thus need to go through before moving to Gaussian processes. If you have skipped reading PRML 3.3.3 in Session 2, you should start with this section before reading the below.

Kernel methods (PRML 6.0-6.2)
Kernel methods provide another perspective on regression problems by characterizing the different training samples by their pairwise relationships (e.g., their similarity) through the use of kernel functions. This allows regression problems to be directly solved in kernel space, with the benefit of a flexible choice of kernel functions, but with the downside of increased computational complexity. When reading through PRML 6.0-6.2, make sure to understand the following:

- How regression problems can be solved in their dual form as function of the Gram matrix
- The constraints imposed on valid kernel functions, and how valid kernel functions can be constructed

You don't need to worry too much about kernel functions applied to probabilistic models, as they won't be required to understand the next session's paper.

Gausssian processes (PRML 6.4-6.4.3)
Gaussian processes define priors over functions, and use kernels to specify the covariance (i.e., similarity) of function values for close-by data points. The higher the covariance, the more similar the function values are assumed to be. In this session's paper, that function that is specified by Gaussian processes is the time evolution of latent states. Thus, the Gaussian process specifies how similar these latent states are expected to be for close-by times. When reading about Gaussian processes, make sure to understand the following:

- Why it is possible to tractably define priors over theoretically infinite function spaces, and what unique property of multivariate Gaussians enable this tractability
- The "exponential of quadratic form" kernel and its properties
- How the posterior predictive density is derived
- The computational complexity of Gaussian processes as a function of the size of the training set
- How the hyperparameters of the covariance kernel can be tuned

The remaining sections of this chapter deal with the use of Gaussian processes to classification, and the use of the Laplace approximation to regain tractability. As these concepts won't be discussed in next week's paper, you can skip them.