Probabilistic models for neural data
Session 4: Generalized linear models #2

To do before this session:

- **Read Pillow (2006) and prepare assigned presentation**
- **Read up on this session's statistical concepts**
- **Complete pre-session quiz**

In this session we will first discuss a book chapter by Pillow (2006), where he introduces GLMs as one variant of likelihood-based approaches to modeling neural data. Please note that the models called GLMs in this chapter are slightly different from the ones we have called GLMs in the last session. I will explicitly compare and contrast these uses in the second half of the session, in which we will continue our discussion of generalized linear models, and will focus on modeling population activity, and their use for decoding. Furthermore, as the theory underlying decoding doesn't require a long discussion, we will additionally have a look at graphical models, and the expectation maximization (EM) algorithm. The latter is a standard algorithm for inference in models with latent variables, and its use will show up in some of the papers we will be discussing.

Paper: Pillow (2006). Likelihood-based approaches to modeling the neural code

When reading the paper, please focus on the following:

- Why did they pick a particular form for p(y|x) rather than characterize it fully (e.g., by tabulating the frequency of a particular spike count y for each stimulus x)? What are the benefits of the p(y|x) chosen in the paper?
- What are the benefits of the LNP model over the STA? When do they become equivalent?
- What is the benefit of including the spike history in GLMs?
- What are the conditions under which GLM likelihood has no (non-global) local maxima? Why is this a desirable property?
- Why do they focus on ML estimation rather than computing the full Bayesian posterior?
- Why is cross-validation required, and why does it work?
- (Advanced) What independence assumptions are made if we set the non-linearity to f(z)=exp(z+b)?
- (Advanced) If Δ in Eq. (3.3) is small, y_i will be either zero or one (not more than one spike per time bin), and we can use the approximation exp(x)≈1+x for small x to turn Eq. (3.3) into a Bernoulli distribution. What does this tell us about the relationship between the inhomogeneous Poisson distribution and the Bernoulli distribution?

Note that, unlike stated in the paper, Eq. (3.9) does not follow from Bayes' rule, but instead uses p(A,B|C) = p(A|B,C) p(B|C), which is a standard joint probability decomposition (i.e., the product rule).

Don’t worry too much about the detailed relationship between LNP models and the STC. Furthermore, applying GLMs to fitting integrate-and-fire neurons is an advanced topic, as it is generally hard to compute the likelihood function for these models. You should skim the corresponding section 3.2.3, but feel free to skip the details. Time-rescaling (section 3.3.2) is an advanced model validation step and you should have heard about it, but won’t be expected to be able to implement it.

To get a better idea of the underlying graphical model, consider the stimulus sequence x_1, x_2, …, where x_n denotes the stimulus in the *n*th time bin (note that this is a different notation from the one used in the paper, Eq. (2,3)), and y_1, y_2, … is the corresponding sequence of spike counts, how would the graphical model for the LNP model look like? How would this model change for GLMs (as discussed in the paper)?

Presentations:

1. The neural coding problem (Fig. 3.1 and Sec. 3.1)
    a. Describe the general neural coding problem (Fig. 3.1).
    b. What is the classical approach to the problem, and what is its motivation?
    c. What is the probabilistic approach to neural coding, and how does it differ from the classical approach?
2. The LNP model (Fig. 3.2 and Secs. 3.2 & 3.2.1)
    a. Describe the maximum likelihood fitting problem in general (Eq. (3.1)).
    b. Describe the components of the LNP model (Fig. 3.2), and their potential interpretation. What is the implicit assumption underlying the inhomogeneous Poisson process in this model? Why might this assumption be violated in neural data?
    c. Qualitatively outline the derivation of Eq. (3.5) without going through the steps in detail. Can you give an interpretation of the different terms and how they contribute to the maximum likelihood estimate? In particular, when should the activation function $f(\cdot)$ return a high value, and when a low one, in order to maximize the probability of observed spikes?
    d. Try to link the LNP model to the statistical concepts introduced in session 3.
3. Relationship to spike-triggered average (STA) (Fig. 3.3)
    a. Introduce the spike-triggered average (STA); for this you will need to consult other sources (e.g., Wikipedia).
    b. Describe Fig. 3.3. How is the estimation error computed, and why is it shown as a function of time?
    c. (Advanced) Try to explain the argument below Eq. (3.6). You can assume that the second term on the right-hand side becomes proportional to k without explaining the details or consulting [16].
    d. (Advanced) What assumptions does an exponential activation function make about how different parts of the stimulus vector x impact the neuron’s spike rate? Hint: think of the vector product as a weighted sum of the elements of x, and then turn the sum in the exponential into a product over exponentials.

4. Relationship to the spike-triggered covariance (STC) (Fig. 3.4)
    a. Introduce the spike-triggered covariance (STC); for this you will need to consult other sources (e.g., Wikipedia).
    b. Describe Fig. 3.4.
    c. What is the benefit of the LNP model over STA & STC? When do they become equivalent?
5. Generalized linear model (Fig. 3.5).
    a. Describe how, qualitatively, GLMs differ from the LMP model? (See note about Eq. (3.9) further above) How does Eq. (3.12) differ from Eq. (3.5), and what is the benefit of this difference?
    b. Technically, why is the GLM no longer an inhomogeneous Poisson process? What assumption of such processes might be violated? Why is this a useful property of GLMs?
6. The issue of local maxima (Sec. 3.2.2)
    a. Describe how gradient ascent for optimization works. For this you will need to consult other sources (e.g., Wikipedia on ‘gradient descent’). What are local maxima and why are they an issue for gradient ascent?
    b. Describe how local maxima can be avoided in GLMs.
    c. Can you think of activation functions that violate the required conditions to ensure the absence of local maxima? Hint: think of the response of some neurons to increasingly larger inputs.
7. Model validation (Sec. 3.3)
    a. Describe the proposed cross-validation procedure. Why can’t we simply assess goodness-of-fit on the training set?
    b. Describe how fitted GLMs can be used for stimulus decoding. Under which conditions is MAP estimation not plagued by local maxima? (Advanced) Why could a bad encoding model still lead to a good decoding model?
    c. (Advanced) Describe, qualitatively, the idea underlying time rescaling.

Statistical concepts: generalized linear models (GLMs), graphical models, and expectation maximization

Using generalized linear models for decoding
Please revisit section 3.3.3 of Pillow (2007) paper, which discusses stimulus decoding using GLMs by a straight-forward application of Bayes’ rule. Consider the following:

- Under which conditions can local maxima of the MAP estimate be avoided?
- Under which circumstances can a bad encoding model still be a good decoding model?

Graphical models (PRML 8.0-8.2)
Graphical models are an important tool in Bayesian inference that illustrates the model’s structure and highlights the dependence and independence of different variables involved in these models. As sequential data models have more complex variable dependencies, we will

discuss graphical models before moving to models for sequential data. In particular, you should focus on:

- The graphical representation of conditional (in)dependencies in Directed Acyclic Graphs (DAGs)
- The meaning of plates, and the associated i.i.d. assumption
- The difference between observed and latent variables
- The components of generative models
- Achieving tractability through conditional independence
- D-separation induced by observed variables

The most important concepts in this chapter are those of generative models, and d-separation. In particular the latter will become essentials to achieve tractability in models of sequential data (Session 6).

Expectation Maximization (EM) algorithm (PRML 9.0, 9.2-9.3.1, 9.4)

While we won't discuss mixture models in detail in this course, the EM algorithm is easiest understood in the context of such models. Therefore, we will look at Gaussian mixture models (PRML 9.2.0) before diving into the application of the EM algorithm to such models (PRML 9.2.2). In PRML 9.2.0-9.2.2 make sure to understand

- The benefit of introducing the binary random vector $\boldsymbol{z}$ as a trick to model the mixing coefficients $\boldsymbol{\pi}$, leading to the useful form Eq. (9.11) that turns the sum in Eq. (9.7) into a product
- The difference between the responsibilities (Eq. (9.13)) and class labels (e.g., Fig. 9.5))
- The rationale behind the different steps leading to the EM-algorithm in PRML 9.2.2, in particular the use of soft mixture assignments through the responsibilities

Do not worry too much about the issues of maximum likelihood inference when applied to Gaussian mixtures, as those are specific to such types of models, which are not central to this course.

PRML 9.2.2 introduces the EM algorithm in an intuitive, but non-rigorous way. Nonetheless, it paves the way for a more general introduction of the EM algorithm in PRML 9.3 that supports maximum likelihood inference for parametric models that contain a set of unobserved latent variables that depend on these parameters. In PRLM 9.3.0-9.3.1 make sure to understand

- The difference between observed and latent variables, and between the complete and incomplete dataset
- The benefit of having to only deal with the complete dataset likelihood $p(\boldsymbol{X},\boldsymbol{Z}|\theta)$ rather than the incomplete dataset likelihood $p(\boldsymbol{X}|\theta)$ (think about their difference in structure in Gaussian mixtures)
- The general application of the EM algorithm to the Gaussian mixture model (PRML 9.3.1)

You can skip PRML 9.3.2 and the sections after that, which discuss relations and application of the EM algorithm to other models. (Advanced) PRML 9.4 provides a more generic explanation for the assumptions underlying the EM algorithm and why it works. It might be an interesting read for those who are more mathematically inclined, as it also opens the door to generalizations of the EM algorithm, but can otherwise be skipped.

Exercise

(to be completed *after* Session 4)

This week's exercise focuses on fitting GLMs to neural data. You will load pre-processed calcium traces of a set of neurons of a mouse performing a virtual reality decision-making task, will fit a set of GLMs of increasing complexity, and will interpret the results. The aim is to put into practice what we have discussed in theory in this and the previous session.

To get started with the exercise, open the Colab notebook at
https://colab.research.google.com/drive/1ByO6a-wUfo61QjTYMOrKzN0xWHKtEIP0?usp=sharing

This notebook contains all the required instructions for completing the exercise, as well as some questions that you should address in your writeup.

Once you have completed the exercise, please describe the results of your exercise, and your interpretation of the results (suggested length: 450-650 words). Use figures to help describe your results; please embed the figures in your text description. The exercise contains a set of discussion questions that you should address in the write-up. In particular, please describe what caused the fits to change the way they did in response to your manipulations. This point is often the most difficult, and might require lots of thought. Please be as specific as possible, and submit the write-up before noon on the day of the next session.