Introduction to temporal structure in neural data

Il Memming Park*

July 1, 2024

In this worksheet*, we will deal with discrete time series x_t defined on discrete time indices $t = 0, 1, \ldots$, assuming a periodic sampling scheme. For example, for voltage measured at 1 kHz sampling rate, x_1 would denote the voltage measured $\Delta = 1/1000\text{ s} = 1\text{ ms}$ after x_0. For spike trains, sampling interval of 0.1 ms to 1 ms range are typically used.

1. (Warm up: a linear neuron model) Let's consider a simple neuron with continuous neural activity (e.g. membrane potential) over time y_t, which is linearly related to the stimulus over time x_t:

$$y_t = a x_{t-1} \tag{1}$$

where a is a constant.

(a) Let $x_{-1:4} = [0, 1, 0, 2, -1, 0]$, write out the values of $y_{0:5}$, i.e., what are the values of $y_0, y_1, \ldots, y_5$?

(b) Let's say the experimentalist has drawn independent samples† from the standard normal distribution at each time point to use as the stimulus x_t, i.e.,

$$x_t \overset{iid}{\sim} \mathcal{N}(0, 1) \tag{2}$$

for all t. This is known as the *white noise excitation* which is very useful for studying the response of an unknown system. Now that input to the system is random, the response is also random. What is the mean of y_t, that is, what is the value of $\mathbb{E}[y_t]$?

> **Fun Fact:** expectation is linear, specifically, $\mathbb{E}[aX] = a\,\mathbb{E}[X]$ for a constant a.

(c) Evaluate $\mathbb{E}[x_t y_t]$

> **Fun Fact:** if X and Y are independent, $\mathbb{E}[XY] = \mathbb{E}[X]\,\mathbb{E}[Y]$

(d) Evaluate $\mathbb{E}[x_{t-1} y_t]$

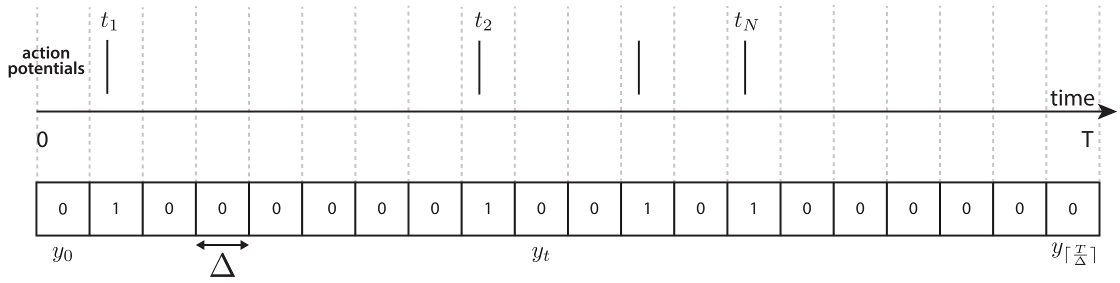
(e) Evaluate $\mathbb{E}[x_{t-2} y_t]$

---

*Special thanks to Ábel Ságodi for the python code.

*Available online: `https://github.com/memming/temporal-neural-data-lectures`

†we abuse notation and use the same notation for observation and the corresponding random variables.



For a spiking neuron, the neural response is a spike train for which we have two notations to represent them. Let's consider a time window of $[0, T]$. First, as an ordered sequence of time points of action potentials, $(0 < t_1 < t_2 < \ldots < t_N < T)$ where there are N action potentials. Assuming a fixed sampling period Δ, the second representation is a vector of length $\lceil \frac{T}{\Delta} \rceil$. Each element of the vector represents a time period, that is y_t corresponds to the time window $[t\Delta, (t+1)\Delta)$. This vector will have a value of 1 for the time windows that contains an action potential, and 0 for the absence (see illustration above). This second representation is useful in analysis and called the *binned spike train* in general. Note that if Δ is large, there may be more than 1 action potential in the time window, in which case, we store the number of action potentials in the vector.

2. To characterize the relationship between the stimulus (input) and the spiking response of a neuron (output), we can use the **spike-triggered-average (STA)**:

$$\mathrm{STA}(\tau) = \frac{1}{N} \sum_t y_t x_{t-\tau} \tag{3}$$

where x_t is the stimulus, y_t is the binned spike train, and N is the total number of spikes. STA is the average stimulus conditioned on each of the spiking events (non-zero y_t). Larger τ represents times farther past relative to the triggering spike. Evaluate the *expected* STA (discussed in Dayan and Abbott[2]), $\mathbb{E}[\sum_t y_t x_{t-\tau}] / \mathbb{E}[N]$ and sketch them for the following neuron models with independently random stimulus x_t where $x_t = -1$ with probability $1/2$ and $x_t = 1$ with probability $1/2$.

(a)

$$y_t = \begin{cases} 1 & \text{if } x_{t-1} = 1 \\ 0 & \text{otherwise} \end{cases} \tag{4}$$

Fun Fact: Expectation is linear. $\mathbb{E}[X + Y] = \mathbb{E}[X] + \mathbb{E}[Y]$

(b)

$$y_t = \begin{cases} 1 & \text{if } x_{t-1} = -1 \textbf{ and } x_{t-2} = 1 \\ 0 & \text{otherwise} \end{cases} \tag{5}$$

(c)

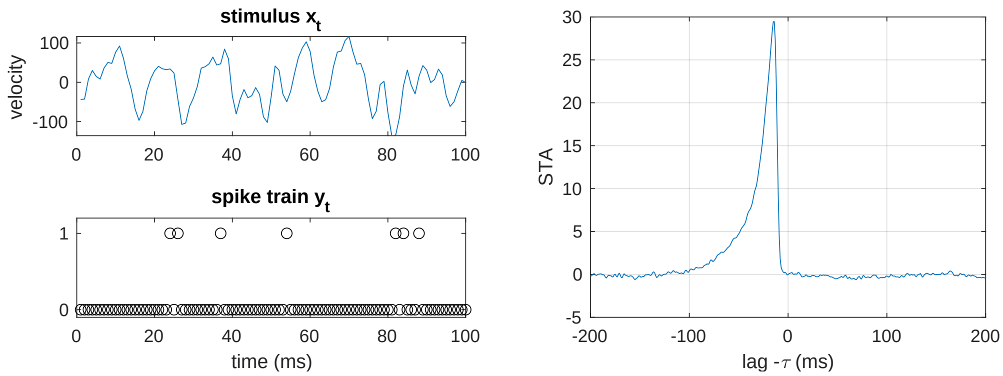
$$y_t = \begin{cases} 1 & \text{if } x_{t-1} = -1 \textbf{ or } x_{t-2} = 1 \\ 0 & \text{otherwise} \end{cases} \tag{6}$$

Fun Fact: STA does not work for an arbitrary stimulus distribution, but it works for white gaussian noise and other elliptically symmetric distributions[6,8].

We will be analyzing fly visual neuron data from de Ruyter van Steveninck et al.[3]. In particular, spikes from the H1 neuron in response (y_t) to visual motion velocity input (x_t) is provided to you.

3. **Homework 1:** Load the H1 dataset, compute the STA, and plot it.

   The result should look something like the plot below. Note that I have plotted it in reverse time, so that negative lag corresponds to the past.



4. Explain what you see in the STA figure above.
   (a) Observe that the STA is positive. What does this mean?

   (b) Observe that the peak and deviation from zero of the STA are on the left of 0 lag. What does this mean?

   (c) Observe that the rise of the STA peak is fast but decays over 100 ms. What does this mean?

   (d) What kind of neuron model would produce an STA with such a shape?

Before we can discuss nonlinear neuron models, let's consider some linear neurons with internal dynamics, that is the internal state of the neuron influences the time evolution of the neural response. (Unlike our previous linear neuron model which forgot its past instantaneously.)

5. Consider a **system** (or a filter or a neuron model) that takes an input time series u_t, and produces x_t. We shall denote such system as: $x_t = \mathcal{H}[u_t]$. In particular, we shall consider a first order recursive model (or filter),

$$x_t = \mathcal{H}[u_t] \triangleq a x_{t-1} + u_t \tag{7}$$

where a is a constant. If you are familiar with conductance based models or leaky-integrate-and-fire neuron model, note their similarity.

(a) Let $x_0 = 0$ and $u_t = 0$ for $t \leq 0$. Given u_t for $t > 0$, write out the 5 next terms of x_t, i.e., what are the values of $x_1, x_2, \ldots, x_5$? What's the general expression for x_t?

(b) Let $x_0 = 0$ and $u_t = 0$ for all time except at $t = 0$, $u_0 = 1$. What's the general expression for x_t? Sketch them out for $a = 0.5$ and $a = -0.9$.

(c) Note that this system is (1) *linear*, i.e., $\mathcal{H}[\beta u_t] = \beta \mathcal{H}[u_t]$, and (2) *time-invariant*, i.e., $x_t = \mathcal{H}[u_t] \implies x_{t+\tau} = \mathcal{H}[u_{t+\tau}]$ for any time shift τ.

Let's add some noise to the linear time-invariant system. This provides foundational intuition for temporal data (time series) analysis.

6. Assume a white gaussian noise $\eta_t \sim \mathcal{N}(0, \sigma^2)$ with variance σ^2. The *whiteness* comes from being independently drawn at each time step t. Consider the following **autoregressive process (or model) of order 1**, or **AR(1)**,

$$X_t = a X_{t-1} + \eta_t. \tag{8}$$

When $|a| < 1$, AR(1) is *stationary*, that is, for any n and time indices $t_1, \ldots, t_n$, an arbitrary joint expectation shifted in time by τ, $\mathbb{E}[f(X_{t_1+\tau}, X_{t_2+\tau}, \ldots, X_{t_n+\tau})]$ for any function f, does not depend on τ. In other words, you cannot tell t by looking at the random process, since it's always the same probability law.

(a) What is the mean, $\mathbb{E}[X_t]$?

(b) What is the variance, $\mathrm{var}[X_t]$?

> **Fun Fact:** $\mathrm{var}(aX) = a^2 \, \mathrm{var}(X)$.

> **Fun Fact:** if X and Y are independent, $\mathbb{E}[f(X)g(Y)] = \mathbb{E}[f(X)] \, \mathbb{E}[g(Y)]$, also, $\mathrm{var}(X + Y) = \mathrm{var}(X) + \mathrm{var}(Y)$.

(c) **Homework 2:** What is the autocovariance at lag 1, $\mathrm{cov}(X_{t+1}, X_t)$?

(d) What is the autocovariance function $\gamma(t, s) = \mathrm{cov}(X_t, X_s)$? (*Hint: consider* $t \geq s$ *case first*)?

(e) If η_t is actually the external stimulus (measured or generated by the experimentalist), what is the expected STA without the $1/N$ term? (I know, this model doesn't spike.) Sketch them out for $a = 0.9$ and $a = -0.5$.

(f) **Homework 3:** Simulate AR(1) models with $a = 0.9$ and $a = -0.9$ and $\sigma = 0.01$ for $T = 10,000$ time steps. Compute the sample mean and sample variance over time and plot the sample autocorrelation function. Compare these with the theoretical values you derived above. Is the process consistent with being *ergodic*, that is, do the time averages agree with the expectations?

(g) **Homework 4:** Let $x \triangleq x_{t-1}$ and $y \triangleq x_t$ and observe that (8) is just a straight line with no bias. Simulate an AR(1) model with $a = -0.7$ and $\sigma = 0.1$ for $T = 10,000$ time steps. Given the time series generated by this AR(1) model, estimate the AR coefficient a using linear regression (least squares).

STA is almost the same as a more widely used quantity, the *cross-correlation function*, which is not restricted to spike trains.

Cross-correlation function:

$$R_{xy}(t, s) = \mathbb{E}[x_t y_s] \tag{9}$$

If the cross-correlation function only depends on how far the two time points $\tau = t - s$ are, that is, when it is time-invariant (a.k.a. wide-sense stationary), we can simply write:

$$R_{xy}(\tau) = \mathbb{E}[x_{t+\tau} y_t] = \mathbb{E}[x_t y_{t-\tau}] \tag{10}$$

Similarly, **autocorrelation** is defined as the cross-correlation with itself:

$$R_{xx}(\tau) = \mathbb{E}[x_{t+\tau} x_t] = \mathbb{E}[x_t x_{t-\tau}] \tag{11}$$

7. How would you estimate auto- and cross-correlation functions from time series data?

Fun Fact: By the law of large numbers, expectations can be approximated by sample averages.

(a) Given m data points over time, we can write an estimator (called the *sample cross-correlation*) as,

$$\hat{R}_{xy}(\tau) = \begin{cases} \frac{1}{m-\tau-1} \sum_{t=0}^{m-\tau-1} x_{t+\tau} y_t & \tau \geq 0 \\ \hat{R}_{xy}(-\tau) & \tau < 0 \end{cases} \tag{12}$$

Is this estimator unbiased?

(b) How is the (raw) cross-correlation function $R_{x,y}(t, s)$ related to the $\operatorname{cov}(x_t, y_s)$?

(c) Propose a “good” estimator for correlation coefficient:

Fun Fact: Correlation coefficient is

$$\rho(X, Y) = \frac{\operatorname{cov}(X, Y)}{\sqrt{\operatorname{var}(X) \operatorname{var}(Y)}} \in [-1, 1] \tag{13}$$

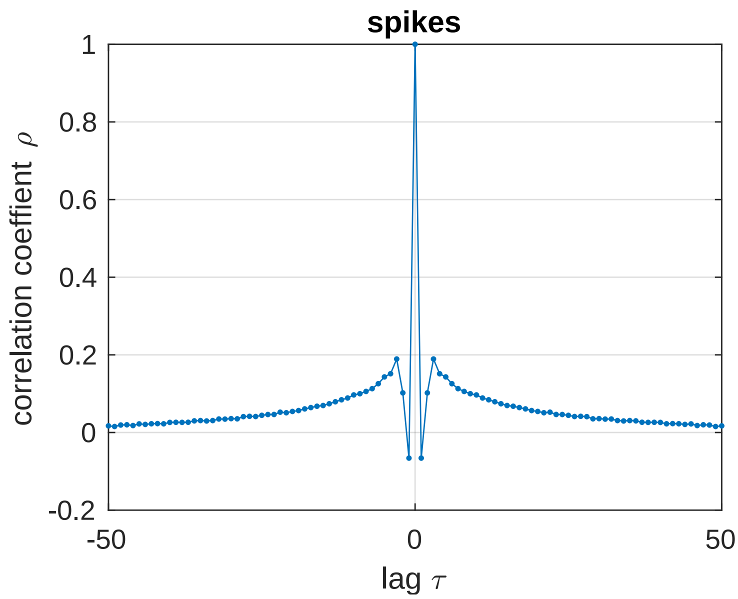
(d) In Python, the function `numpy.correlate` or `scipy.signal.correlate` can compute auto- and cross-correlation functions. However, it only computes the "raw" correlations without the normalization factor. Try the following Python code:

```
T = 2000; maxLags = 100; a1 = 0.83;
x = np.random.normal(0,1,(T))
y = np.random.normal(0,1,(T))
for t in range(3, len(y)-1):
    y[t+1] += a1 * y[t] + x[t-3]  # AR(1)

fig, ax = plt.subplots(figsize=(15,5), nrows=1, ncols=3)

xc =  scipy.signal.correlate(x, y, "full")[len(x)-maxLags:len(x)+
                                            maxLags+1];
```

Try to implement the normalizations for the cross-correlation, both for the unbiased normalization and the normalization to the correlation coefficients.

8. From the same H1 dataset, compute the sample autocorrelation function (normalized to correlation coefficients) of the spike train using `scipy.signal.correlate` for various lags.

   We get the following typical shape.



(a) Why is it symmetric?

(b) Why is the peak at lag $\tau = 0$, and why is it 1?

(c) Why is the autocorrelation negative at $\tau = 1$?

(d) What does it mean that the autocorrelation approaches 0 at $\tau = 50$?

9. A stationary, finite variance time series can be thought of as the random superposition of sines and cosines oscillating at various frequencies. Specifically, for such processes, the Fourier transform of the autocovariance function is called the **power spectral density**, and it captures how much power is captured by each frequency.

> **Fun Fact:** $e^{2\pi i\omega} = \cos(2\pi\omega) + i\sin(2\pi\omega)$ (Euler's formula)

> Denote the autocovariance function as $\gamma(\tau) = \text{cov}(X_t, X_{t+\tau})$. The power spectral density is defined as:
>
> $$f(\omega) \triangleq \sum_{\tau=-\infty}^{\infty} \gamma(\tau) e^{-2\pi i\omega\tau} \tag{14}$$
>
> The inverse transform is given by,
>
> $$\gamma(\tau) = \int_{-1/2}^{1/2} f(\omega) e^{2\pi i\omega\tau} \mathrm{d}\omega \tag{15}$$
>
> **Fun Fact:** These are two sides of the same coin!

(a) What is the power spectral density of white gaussian noise? (*Hint: white gaussian noise is independent gaussian noise with zero mean.*)

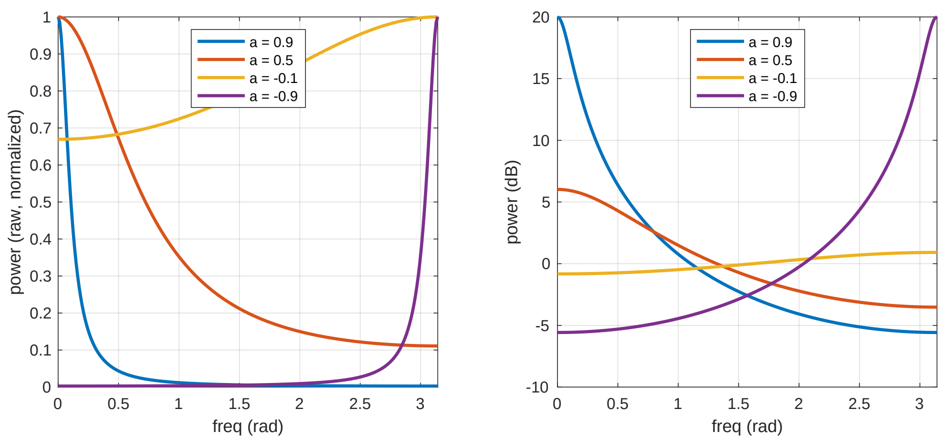
(b) **Homework 5:** Generate $T = 10,000$ long white gaussian noise with variance 2 (i.e., standard deviation $\sqrt{2}$). Estimate and plot the power spectral density.

> The Python package numpy has the command `numpy.random.normal`(μ, σ) which generates "independent" gaussian noise with mean μ and variance σ. The Python package scipy has the commands `scipy.signal.welch` and `scipy.signal.periodogram` can be used to estimate power spectrum given a time series. Try plotting the output!
>
> ---
>
> **Fun Fact:** absolute square of the `scipy.fft.fft` of the raw signal is <u>not</u> a (statistically) consistent estimator for the power spectral density. (Don't ever use this method!)

(c) What is the power spectral density of AR(1) process?

$$
\begin{aligned}
f(\omega) &= \sum_{\tau=-\infty}^{\infty} \gamma(\tau) e^{-2\pi i \omega \tau} && (16)\\
&= \frac{\sigma^2}{1-a^2} \sum_{\tau=-\infty}^{\infty} a^{|\tau|} e^{-2\pi i \omega \tau} && (17)\\
&= \frac{\sigma^2}{1-a^2} \left(1 + \sum_{\tau=1}^{\infty} a^{\tau} \left(e^{2\pi i \omega \tau} + e^{-2\pi i \omega \tau}\right)\right) && (18)\\
&= \frac{\sigma^2}{1-a^2} \left(1 + \frac{a e^{2\pi i \omega \tau}}{1-e^{2\pi i \omega \tau}} + \frac{a e^{-2\pi i \omega \tau}}{1-e^{-2\pi i \omega \tau}}\right) && (19)\\
&= \frac{\sigma^2}{1-a^2} \frac{1 - a e^{2\pi i \omega \tau} a e^{-2\pi i \omega \tau}}{(1-a e^{2\pi i \omega \tau})(1-a e^{-2\pi i \omega \tau})} && (20)\\
&= \frac{\sigma^2}{1 - 2a\cos(2\pi\omega) + a^2} && (21)
\end{aligned}
$$

(d) For what range of a values does the AR(1) act as a *low pass filter*, that is, its power spectrum peaks in low frequencies? See below for the plot of (21).



(e) **Homework 6:** Estimate and plot the power spectral density of time series generated from two AR(1) models with parameters of your choice.

A general order p autoregressive model is,

$$X_t \sim \mathcal{N}\left(\sum_{\tau=1}^{p} a_\tau X_{t-\tau}, \sigma^2\right) \tag{22}$$

10. Autoregressive model with order 2, AR(2), can be written as,

$$X_t = a_1 X_{t-1} + a_2 X_{t-2} + \eta_t. \tag{23}$$

(a) Define $\vec{X}_t = \begin{bmatrix} X_t \\ X_{t-1} \end{bmatrix}$. Rewrite (23) in matrix-vector form,

$$\vec{X}_t = \mathbf{A}\vec{X}_{t-1} + \vec{\eta}_t \tag{24}$$

$$\mathbf{A} = \begin{bmatrix} & , & \\ & , & \end{bmatrix}, \qquad \vec{\eta}_t = \begin{bmatrix} \quad \end{bmatrix} \tag{25}$$

> Fun matrix-vector product rule:
>
> $$\begin{bmatrix} A_{11} & A_{12} \\ A_{21} & A_{22} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \begin{bmatrix} A_{11}x_1 + A_{12}x_2 \\ A_{21}x_1 + A_{22}x_2 \end{bmatrix} \tag{26}$$

(b) In general, when the *eigenvalues* of $\mathbf{A}$ are of magnitude strictly less than 1 (inside the unit circle), the corresponding AR(p) model is stable. What are the eigenvalues for the AR(2)?

(c) Generalize the matrix form representation for AR(p):

(d) Show that thanks to the special structure of $\mathbf{A}$, roots of the characteristic polynomial for AR(p), is

$$\det(\mathbf{A} - \lambda\mathbb{I}) = \lambda^p - a_1\lambda^{p-1} - \cdots - a_p = 0 \tag{27}$$

(e) For the general AR(p), the power spectral density has the form:

$$f(\omega) = \frac{\sigma^2}{|1 - \sum_k a_k e^{-i2\pi\omega k}|^2} \tag{28}$$

When the eigenvalues are complex-valued, they come in conjugate pairs. Argue that the power spectral density has peaks at the corresponding angles by viewing (28) as evaluating a function in the complex plane restricted to the unit circle (parameterized as $z = e^{i2\pi\omega}$). The eigenvalues of $\mathbf{A}$ are poles of the said function.

(f) **Homework 7:** Use Python function `numpy.linalg.eig` to obtain eigenvalues of a matrix. Construct a stable AR(2) model by choosing parameters and generate time series of length $T = 100,000$. Plot the sample autocorrelation function and estimated power spectral density.

(g) Another way of writing the AR(2) as matrix equations is,

$$X_t = \begin{bmatrix} a_1 & a_2 \end{bmatrix} \begin{bmatrix} X_{t-1} \\ X_{t-2} \end{bmatrix} + \eta_t. \tag{29}$$

Find a way to stack the vectors to write this for all observed data, $x_1, \ldots, x_m$, as one bigger matrix equation. (The new covariate matrix is called the *design matrix*.)

$$\begin{bmatrix} x_3 \\ x_4 \\ x_5 \\ \vdots \\ x_m \end{bmatrix} = \begin{bmatrix} \quad , \quad \\ \quad , \quad \\ \quad , \quad \\ \vdots \\ \quad , \quad \end{bmatrix} \begin{bmatrix} \quad \end{bmatrix} + \begin{bmatrix} \quad \\ \quad \\ \quad \\ \vdots \\ \quad \end{bmatrix} \tag{30}$$

(h) **Homework 8:** Note that above form (30) is a least squares regression. Using the data from question (f), estimate the coefficients for the autoregressive model using least squares regression (use `sklearn.linear_model.LinearRegression`) Compare the result with the true model parameters.

(i) **Homework 9:** Read Ozaki[5] chapters 1-3. Learn about Wiener-Khinchin theorem, aliasing effect, deterministic vs. stochastic modeling, and the ARMA.

Note that AR(p) is also a **generalized linear model (GLM)**[‡] with Gaussian observation noise. However, with Gaussian observation noise, we won't get spikes, hence not great for action potentials over time. To model noisy spiking patterns (a.k.a. point process observations), we can still use GLM but with a Poisson observation noise.

Autoregressive Point Process (a.k.a. Poisson-GLM)

$$y_t \sim \text{Poisson}\left(\exp\left(\sum_{\tau=1}^{p} h_\tau y_{t-\tau} + b\right)\right) \tag{31}$$

Note the similarity between this GLM formulation (autoregressive point process) and the AR(p) model.

11. Consider the GLM in (31). The autoregressive weights h_τ are called the *history filter* since they dictate how the spiking history impacts the probability of firing now.

(a) Recall the absolute refractory period of neurons: Neurons can't fire another action potential within this period. When the time bin size is smaller than the absolute refractory period, what would be the value of h_1 that best explains the neural activity?

(b) **Homework 10:** What is the probability of firing an action potential in the current time bin if there were no spikes for the past p time bins?

[‡]not to be confused with *general* linear model!

(c) What is the log-likelihood of this model?

> **Fun Fact:** For a Poisson random variable $\mathbf{Y} \sim \text{Poisson}(\Lambda)$, its likelihood for $y \in \{0, 1, \ldots\}$ is,
>
> $$P(\mathbf{Y} = y \mid \Lambda) = \frac{\Lambda^y \exp(-\Lambda)}{y!} \tag{32}$$
>
> We omit $\mathbf{Y}$ in our notation when it's obvious from context.

(d) What is the difference in log-likelihood for the homogeneous Poisson model (i.e. GLM with no history filter), and the GLM with $p = 1$ if the history filter captured the absolute refractory period per spike? Part of the superior performance of GLMs[10] can be contributed to this.

Do you have other task variables and external stimuli that may drive the neuron you are analyzing[9]?

GLM with covariates

$$y_t \sim \text{Poisson}\left(\exp\left(\sum_{\tau=1}^{p} h_\tau y_{t-\tau} + \sum_{k=1}^{K}\sum_{\tau=0}^{q_k-1} w_\tau^k x_{t-\tau}^k + b\right)\right) \tag{33}$$

where K covariate time series x_t^k are introduced with maximum $q_k - 1$ time lags.

How do we fit the parameters of a GLM given spike trains and covariates? Among many potential estimators, one of the most popular and principled estimator is the maximum likelihood estimator (MLE). In most cases, MLE is *statistically efficient*, that is, the variance of the estimator decays at the fastest possible rate as a function of sample size. Also, this particular form of MLE has a unique optimal solution, but it may not be the best solution for simulating from the GLM[1].

In Python, you can use `statsmodels.api.GLM` to perform MLE-based Poisson regression that can be used to fit our GLM formulations.

```
import statsmodels.api as sm

log_link = sm.families.links.Log() #defines as the link function the
                                          logarithm

#initialize Poisson regression:
poisson_log = sm.GLM(Y, X, family=sm.families.Poisson(log_link))

poisson_log_results = poisson_log.fit() #fit model

poisson_log_results.summary() #get summary of fit
```

Note natural logarithm is chosen as the "link function". This corresponds to the exponential inverse link function in (31) and (33).

12. **Homework 11:** Following code simulates from an autoregressive Poisson-GLM. Use `statsmodels.api.GLM` to estimate the parameters.

```
# simulate from a GLM
T = 1000; #just 1000 time points, try increasing this if you'd like
x = np.random.normal(0,1,(T)) #white noise stimulus
y = np.zeros((T)) #pre-allocate some space to put the spikes

for t in range(3,T): # start from time 3, because we need y[t-3]
    y[t] = np.random.poisson(np.exp(-5*y[t-1] - 1.5*y[t-2] - 0.1*y[t-3]
                                            + 0.3*x[t]-3))

# form a design matrix for glm
Y = y[3:T]
```

13. Let $\mathcal{I} = 1, 2, \ldots, n$ be time indices. The **covariance matrix** Σ is an $n \times n$ real-valued matrix, where each entry is given as $\Sigma_{i,j} = \operatorname{cov}(X_i, X_j)$ for $i, j \in \mathcal{I}$.

   (a) What are the diagonal entries of Σ? What would they be if the process is stationary?

   (b) If the system is stationary, the covariance matrix of the time series has a repeating structure called the *Toeplitz* structure. How many unique values can the covariance matrix have? Are there entries of the second row that are the same as the first row?

   (c) Let $\mathbf{X}$ be the data matrix of size $n \times (T - n)$,

$$\mathbf{X} = \begin{bmatrix} X_T & X_{T-1} & \cdots & X_n \\ X_{T-1} & X_{T-2} & \cdots & X_{n-1} \\ \vdots & \vdots & \ddots & \vdots \\ X_{T-n+1} & X_{T-n+1} & \cdots & X_1 \end{bmatrix} \tag{34}$$

   Additionally assume that the observed time series $\mathbf{X}$ has zero mean. Show that the following estimates the covariance matrix:

$$\hat{\Sigma} = \frac{1}{T-n}\mathbf{X}\mathbf{X}^\top \qquad \text{(sample covariance matrix)} \tag{35}$$

   where $(\cdot)^\top$ denotes the *transpose* operation.

   Recall a popular (biased but consistent) sample covariance estimator for $\{(X_i, Y_i)\}_{i=1}^N$:

$$\operatorname{cov}(X, Y) \approx \frac{1}{N}\sum_i \left(X_i - \frac{1}{N}\sum_j X_j\right)\left(Y_i - \frac{1}{N}\sum_k Y_k\right) \tag{36}$$

14. What if we have two simultaneously observed time series? Bivariate autoregressive model is a simple extension of AR(p) that can handle this:

$$X_t = \sum_{\tau=1}^{p} a_\tau X_{t-\tau} + \sum_{\tau=1}^{p} b_\tau Y_{t-\tau} + \eta_t \tag{37}$$

$$Y_t = \sum_{\tau=1}^{p} c_\tau X_{t-\tau} + \sum_{\tau=1}^{p} d_\tau Y_{t-\tau} + \epsilon_t \tag{38}$$

where (X_t, Y_t) are zero mean and simultaneously observed, and η_t and ϵ_s are independent for all t, s.

(a) If $b_\tau = 0$ for all τ, X would not depend on Y. Does this mean that for all lags τ, the cross-correlation between X_t and $Y_{t+\tau}$ are zero?

(b) Consider the following two models of time series X_t:

$$X_t = \sum_{\tau=1}^{p} a_\tau X_{t-\tau} + \sum_{\tau=1}^{p} b_\tau Y_{t-\tau} + \eta_t \tag{39}$$

$$X_t = \sum_{\tau=1}^{p} a'_\tau X_{t-\tau} + \eta'_t \tag{40}$$

Which model has more free parameters? Which model is more likely to explain more variance of X_t?

(c) What would your null hypothesis be if you wanted to claim that past of Y has influence on X?

> If (39) is significantly better at explaining X_t than (40), we say that Y **Granger causes** X.

(d) What is the effect of sampling rate in the interpretation of Granger causality?

(e) Discuss the advantages and disadvantages of Granger causality. When does it fail to reveal the physical causal structure between X and Y?

(f) Can you extend Granger causality to point processes using GLM? Outline a procedure.

References

[1] Diego M. Arribas, Yuan Zhao, and Il Memming Park. Rescuing neural spike train models from bad MLE. In *Advances in Neural Information Processing Systems (NeurIPS)*, October 2020.

[2] Peter Dayan and L F Abbott. *Theoretical Neuroscience: Computational and Mathematical Modeling of Neural Systems*. MIT Press, Cambridge, MA, USA, 2001.

[3] Rob R de Ruyter van Steveninck, Geoffrey D Lewen, Steven P Strong, Roland Koberle, and William Bialek. Reproducibility and variability in neural spike trains. *Science*, 275:1805–1808, 1997. ISSN 0036-8075.

[4] Matthew Dowling, Yuan Zhao, and Il Memming Park. Non-parametric generalized linear model. September 2020.

[5] Tohru Ozaki. *Time Series Modeling of Neuroscience Data*. CRC Press, January 2012. ISBN 9781420094602. doi: 10.1201/b11527. URL `http://dx.doi.org/10.1201/b11527`.

[6] Liam Paninski. Convergence properties of three spike-triggered analysis techniques. *Network: Computation in Neural Systems*, 14:437–464, August 2003.

[7] Il Memming Park and Jonathan W. Pillow. Bayesian spike triggered covariance analysis. In *Advances in Neural Information Processing Systems (NIPS)*, 2011.

[8] Il Memming Park, Evan Archer, Nicholas Priebe, and Jonathan W. Pillow. Spectral methods for neural characterization using generalized quadratic models. In *Advances in Neural Information Processing Systems (NIPS)*, 2013.

[9] Il Memming Park, Miriam L. R. Meister, Alexander C. Huk, and Jonathan W. Pillow. Encoding and decoding in parietal cortex during sensorimotor decision-making. *Nature Neuroscience*, 17(10):1395–1403, October 2014. ISSN 1097-6256. doi: 10.1038/nn.3800.

[10] Jonathan W Pillow, Jonathon Shlens, Liam Paninski, Alexander Sher, Alan M Litke, E J Chichilnisky, and Eero P Simoncelli. Spatio-temporal correlations and visual signalling in a complete neuronal population. *Nature*, 454(7207):995–999, August 2008. ISSN 0028-0836. doi: 10.1038/nature07140.

[11] E P Simoncelli, Liam Paninski, Jonathan Pillow, and Odelia Schwartz. Characterization of neural responses with stochastic stimuli. In Michael S Gazzaniga, editor, *The Cognitive Neurosciences III*, chapter 23, page 327. The MIT Press, 2004. ISBN 00262072548.

[12] Patrick A Stokes and Patrick L Purdon. A study of problems encountered in granger causality analysis from a neuroscience perspective. *Proceedings of the National Academy of Sciences of the United States of America*, 114(34):E7063–E7072, August 2017. ISSN 0027-8424, 1091-6490. doi: 10.1073/pnas.1704663114.