Point-Process Generalized Linear Models (PPGLMs) for spike train analysis

M. Rule

November 8, 2011

In neuroscience, we often want to understand how neuronal spiking relates to other variables. For this, tools like linear regression and correlation often work well enough. However, linear approaches presume that the underlying relationships are linear and the noise is Gaussian, neither of which are true for neural datasets. The Generalized Linear Model (GLM) extends linear methods to better account for nonlinearity and non-Gaussian noise.

0.0.1 GLMs for spike train analysis

We're interested in understanding how a neural spike train $y(t)$ relates to some other covariates $\mathbf{x}(t)$. In continuous time, a spike train is modeled as a train of impulses at each spike time t_i, $y(t) = \sum_i \delta(t - t_i)$. In practice, we always work in discrete time. It is common to choose a time step Δt small enough so that at most one spike occurs in each time bin.

Two types of GLM that are common in spike train analysis:

1. The Poisson GLM assumes that spikes arise from an inhomogeneous Poisson process with time-varying rate $\lambda(t)$ that depends log-linearly on the covariates $\mathbf{x}(t)$. In practice, it is common to treat the Poisson GLM in discrete time by assuming that the rate $\lambda(t)$ is constant within each time-bin:

$$y_t \sim \text{Poisson}(\lambda_t \cdot \Delta t)$$
$$\lambda_t = \exp\left(\mathbf{w}^\top \mathbf{x}_t\right)$$

2. The Bernoulli GLM models each time-bin of a binary, discrete-time spike train $y_t \in \{0, 1\}$ as a Bernoulli "coin flip" with $\Pr(y{=}1) = p$.

$$y_t \sim \text{Bernoulli}(p_t)$$
$$p_t = \frac{1}{1 + \exp[-\mathbf{w}^\top \mathbf{x}(t)]}$$

0.0.2 Maximum likelihood

GLMs are typically estimated using maximum likelihood when testing how covariates $\mathbf{x}(t)$ influence spiking. (Other fitting procedures may be more appropriate when GLMs are used to model spiking dynamical systems.) The maximum likelihood procedure finds the weights $\mathbf{w}$ that maximize the likelihood of observing spike train $\mathbf{y} = \{y_1, .., y_T\}$, given covariates $\mathbf{X} = \{\mathbf{x}_1, .., \mathbf{x}_T\}$, over all T recorded time-bins.

$$\mathbf{w} = \underset{\mathbf{w}}{\operatorname{argmax}} \Pr(\mathbf{y}|\mathbf{w}, \mathbf{X})$$

In practice, likelihood maximization is typically phrased in terms of minimizing the negative log-likelihood. Working with log-likelihood is more numerically stable, and the problem is phrased as minimization so that it is more straightforward to plug-in to off-the-shelf minimization code.

$$\mathbf{w} = \underset{\mathbf{w}}{\operatorname{argmin}} - \ln \Pr(\mathbf{y}|\mathbf{w}, \mathbf{X})$$

Assuming observations are independent, the negative log-likelihood factors into a sum over time-points. Equivalently, one can minimize the average negative log-likelihood, over samples, $-\langle \ln \Pr(y_t|\mathbf{w}, \mathbf{x}_t) \rangle$. The maximum likelihood parameters are typically solved for via gradient descent or the Newton-Raphson method. This requires computing the gradient and hessian of the negative log-likelihood.

0.0.3 Gradient and Hessian for the Poisson GLM

We can calculate the gradient and Hessian for the Poisson GLM by substituting the Poisson probability density function, $\Pr(y) = \lambda^y e^{-\lambda}/y!$, in to our expression for the negative log-likelihood:

$$-\langle \ln \Pr(y_t|\mathbf{w}, \mathbf{x}_t) \rangle = \langle \lambda_t - y_t \ln(\lambda_t) - \ln(y!) \rangle$$

Because the term $\ln(y!)$ does not depend on $\mathbf{w}$, we can ignore it without affecting the location of our optimum. Substituting in $\lambda = \exp(\mathbf{w}^\top \mathbf{x})$, we get:

$$\ell = \left\langle \exp(\mathbf{w}^\top \mathbf{x}) - y_t \mathbf{w}^\top \mathbf{x}_t \right\rangle$$

Finding weight $\mathbf{w}$ that minimize ℓ is equivalent to solving for the maximum-likelihood weights. The gradient and Hessian of of ℓ in $\mathbf{w}$ are:

$$\begin{aligned} \frac{\partial \ell}{\partial w_i} &= \left\langle [\exp(\mathbf{w}^\top \mathbf{x}_t) - y_t] x_i \right\rangle = \langle (\lambda_t - y_t) x_i \rangle \\ \frac{\partial \ell}{\partial w_i \partial w_j} &= \left\langle [\exp(\mathbf{w}^\top \mathbf{x}_t)] x_i x_j \right\rangle = \langle \lambda_t x_i x_j \rangle \end{aligned}$$

In matrix notation, these derivatives can then be written as:

$$\begin{aligned} \nabla \ell &= \langle \mathbf{x}(\lambda - y) \rangle \\ \mathbf{H}\ell &= \langle \lambda \mathbf{x}\mathbf{x}^\top \rangle \end{aligned}$$

0.0.4 Gradient and Hessian for the Bernoulli GLM

The Bernoulli GLM is similar, with the observation probability given by the Bernoulli distribution $\Pr(y) = p^y(1-p)^{1-y}$

$$\begin{aligned}\langle -\ln \Pr(y_t|\mathbf{w}, \mathbf{x}_t)\rangle &= -\langle y_t \ln(p_t) + (1-y_t)\ln(1-p_t)\rangle \\ &= -\left\langle y_t \ln\left(\tfrac{p_t}{1-p_t}\right) + \ln(1-p_t)\right\rangle\end{aligned}$$

Then, using $p = [1+\exp(-\mathbf{w}^\top \mathbf{x})]^{-1}$, i.e. $\mathbf{w}^\top \mathbf{x} = \ln[p/(1-p)]$, we get:

$$\ell = \left\langle \ln[1+\exp(\mathbf{w}^\top \mathbf{x}_t)] - y_t \mathbf{w}^\top \mathbf{x}_t \right\rangle$$

The gradient and Hessian of of ℓ in $\mathbf{w}$ are:

$$\begin{aligned}\frac{\partial \ell}{\partial w_i} &= \left\langle \left[\frac{\exp(\mathbf{w}^\top \mathbf{x}_t)}{1+\exp(\mathbf{w}^\top \mathbf{x}_t)} - y_t\right] x_i \right\rangle \\ &= \langle (p_t - y_t) x_i \rangle \\ \frac{\partial \ell}{\partial w_i \partial w_j} &= \langle p_t(1-p_t) x_i x_j \rangle\end{aligned}$$

In matrix notation:

$$\begin{aligned}\nabla \ell &= \langle \mathbf{x}(p-y)\rangle \\ \mathbf{H}\ell &= \langle p(1-p)\cdot \mathbf{x}\mathbf{x}^\top \rangle\end{aligned}$$

0.0.5 Iteratively reweighted least squares

PPGLMs can also be fit to spike train data using Iteratively Reweighted Least Squares (IRLS). Recall that for a linear model $\mathbf{y} = \mathbf{w}^\top \mathbf{x}$, the Ordinary Least Squares (OLS) solution is:

$$\mathbf{w} = \langle \mathbf{x}\mathbf{x}^\top \rangle^{-1} \langle \mathbf{x}\mathbf{y}^\top \rangle.$$

The IRLS approach phrases optimizing the parameters of the GLM in terms of repeated iterations of a reweighted least-squares problem. To derive this, first recall the definition of the Newton-Raphson update:

$$\mathbf{w}_{n+1} = \mathbf{w}_n - \mathbf{H}\ell(\mathbf{w}_n)^{-1} \nabla \ell(\mathbf{w}_n)$$

For the Poisson GLM, this is

$$\mathbf{w}_{n+1} = \mathbf{w}_n + \langle \lambda \mathbf{x}\mathbf{x}^\top \rangle^{-1} \langle \mathbf{x}(y-\lambda)\rangle$$

For e.g. the Poisson GLM, IRLS rewrites this as a least squares problem by defining weights λ and pseudo-variables $z = \mathbf{w}^\top \mathbf{x} + \frac{1}{\lambda}(y - \lambda)$. We can confirm that the IRLS update is equivalent to the Newton-Raphson update:

$$\begin{aligned}
\mathbf{w}_{n+1} &= \langle \lambda \mathbf{x}\mathbf{x}^\top \rangle^{-1} \langle \lambda \mathbf{x} z^\top \rangle \\
&= \langle \lambda \mathbf{x}\mathbf{x}^\top \rangle^{-1} \left[\langle \lambda \mathbf{x} [\mathbf{x}^\top \mathbf{w}_n + \tfrac{1}{\lambda}(y - \lambda)] \rangle \right] \\
&= \langle \lambda \mathbf{x}\mathbf{x}^\top \rangle^{-1} \left[\langle \lambda \mathbf{x}\mathbf{x}^\top \rangle \mathbf{w}_n + \langle \mathbf{x}(y - \lambda) \rangle \right] \\
&= \mathbf{w}_n + \langle \lambda \mathbf{x}\mathbf{x}^\top \rangle^{-1} \langle \mathbf{x}(y - \lambda) \rangle
\end{aligned}$$

0.0.6 Expected log-likelihoods

It's also possible to approximately fit $\mathbf{w}$ knowing only the mean and covariance of $\mathbf{x}$. This reduces computational complexity, since it avoids having to process the whole data matrix when optimizing $\mathbf{w}$. Previously, we calculated negative log-likelihood as an expectation over the data time-points. Here, we instead calculate these expectations based on a Gaussian model of the covariates $\mathbf{x} \sim \mathcal{N}(\mu_\mathbf{x}, \Sigma_\mathbf{x})$. For example in the Poisson case, the gradient of the log-likelihood is :

$$\nabla \ell = \langle \mathbf{x}(\lambda - y) \rangle = \langle \mathbf{x} \exp(\mathbf{w}^\top \mathbf{x}) \rangle - \langle \mathbf{x} y \rangle$$

The term $\langle \mathbf{x} y \rangle$ does not depend on $\mathbf{w}$, and can be computed in advance. The term $\langle \mathbf{x} \exp(\mathbf{w}^\top \mathbf{x}) \rangle$ has a closed-form solutions based on the log-normal distribution:

$$\begin{aligned}
\langle \mathbf{x}\lambda \rangle &= [\langle \mathbf{x} \rangle + \mathbf{w}^\top \Sigma_\mathbf{x}] \langle \lambda \rangle \\
\langle \lambda \rangle &= \langle \exp(\mathbf{w}^\top \mathbf{x}) \rangle \\
&= \exp(\mathbf{w}^\top \mu_\mathbf{x} + \tfrac{1}{2} \mathbf{w}^\top \Sigma_\mathbf{x} \mathbf{w})
\end{aligned}$$

The Hessian also has a closed form. This avoids having to recompute the re-weighted mean/covariances on every iteration. However, one still must calculate a mean and covariance initially. This approximation will only remain valid in the case that $\mathbf{x}$ is truly Gaussian. However, it can be used to pick an initial $\mathbf{w}_0$ before continuing with Newton-Raphson.