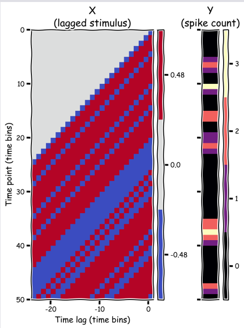
Neuromatch Academy: Generalised Linear Models - Summary Sheet[1]

Generalized Linear Models

Create design matrix

To create the **design matrix** which organizes the stimulus intensities in matrix form such that the ith row has the stimulus frames preceding timepoint i.

In this example, we will create the design matrix $\mathbf{X}$ using $d = 25$ time lags. That is, $\mathbf{X}$ should be a $T \times d$ matrix. $d = 25$ is a choice we're making based on our prior knowledge of the temporal window that influences RGC responses. Here, spike count is $\mathbf{Y}$ is predicted from out stimulus $\mathbf{X}$,

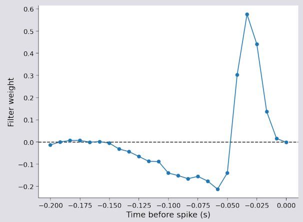


Fit Linear-Gaussian regression model

The maximum likelihood estimate of θ in this model can be solved analytically using the equation:

$$\hat{\boldsymbol{\theta}} = (\mathbf{X}^\top \mathbf{X})^{-1}\mathbf{X}^\top \mathbf{y}. \tag{1}$$

The resulting maximum likelihood filter estimates are:



Generalized Linear Models

Poisson regression

Poisson regression is a generalized linear model form of regression analysis used to model count data, like spikes. In the Poisson GLM,

$$\log P(\mathbf{y} \mid \mathbf{X}, \theta) = \sum_t \log P(y_t \mid \mathbf{x_t}, \theta), \tag{2}$$

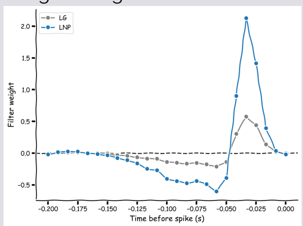
where

$$P(y_t \mid \mathbf{x_t}, \theta) = \frac{\lambda_t^{y_t} \exp(-\lambda_t)}{y_t!}, \text{ with rate } \lambda_t = \exp(\mathbf{x_t}^\top \theta). \tag{3}$$

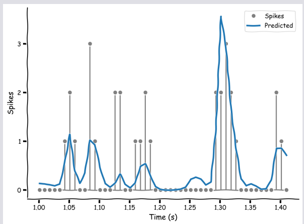
Now, taking the log likelihood for all the data we obtain: $\log P(\mathbf{y} \mid X, \theta) = \sum_t (y_t \log(\lambda_t) - \lambda_t - \log(y_t!))$. Because we are going to minimize the negative log likelihood with respect to the parameters θ, we can ignore the last term that does not depend on θ. For faster implementation, let us rewrite this in matrix notation:

$$\mathbf{y}^\top \log(\lambda) - \mathbf{1}^\top \lambda, \text{ with rate } \lambda = \exp(\mathbf{X}\theta) \tag{4}$$

Finally, don't forget to add the minus sign for your function to return the negative log likelihood.



Spike Prediction



Generalized Linear Models

Logistic regression

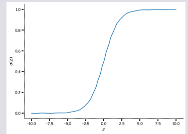
Logistic Regression is a binary classification model. It is a GLM with a logistic link function and a Bernoulli (i.e. coin-flip) noise model. The fundamental input/output equation of logistic regression is:

$$\hat{y} \equiv p(y = 1|x, \theta) = \sigma(\theta^T x) \tag{5}$$

Note that we interpret the output of logistic regression, $\hat{y}$, as the probability that $y = 1$ given inputs x and parameters θ.
Here $\sigma()$ is a "squashing" function called the sigmoid function or logistic function. Its output is in the range $0 \leq y \leq 1$. It looks like this:

$$\sigma(z) = \frac{1}{1 + \exp(-z)} \tag{6}$$

Recall that $z = \theta^T x$. The parameters decide whether $\theta^T x$ will be very negative, in which case $\sigma(\theta^T x) \approx 0$, or very positive, meaning $\sigma(\theta^T x) \approx 1$.



Regularisation

Regularization forces a model to learn a set solutions you a priori believe to be more correct, which reduces overfitting because it doesn't have as much flexibility to fit idiosyncrasies in the training data. This adds model bias, but it's a good bias because you know (maybe) that parameters should be small or mostly 0.
L_2 **regularization**
Regularization comes in different flavors. A very common one uses an L_2 or "ridge" penalty. This changes the objective function to

$$-\log \mathcal{L}'(\theta|X, y) = -\log \mathcal{L}(\theta|X, y) + \frac{\beta}{2}\sum_i \theta_i^2, \tag{7}$$

where β is a *hyperparameter* that sets the *strength* of the regularization.

---

[1] 't Hart et al., (2022). Neuromatch Academy: a 3-week, online summer school in computational neuroscience. Journal of Open Source Education, 5(49), 118. https://doi.org/10.21105/jose.00118