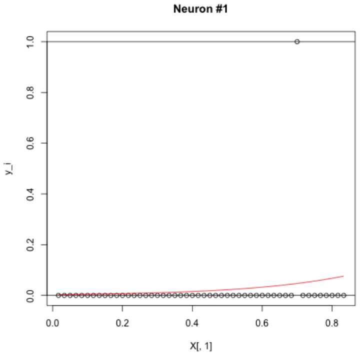
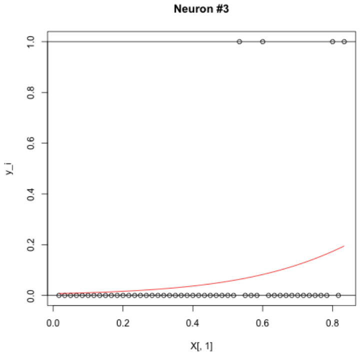
Regularization and clustering of neuron spike data with the Group Lasso

- neuron spikes
- spikes as binary data
- stimulus is continuous
- logistic regression





Logistic Model

- model: neuron i at time t spikes with probability $p_{it} = \sigma(\beta_i X_t)$
- for each neuron we can write down the likelihood and maximize it
- however, much of this data is pretty bad: many neurons had zero spikes!
- IDEA: borrow information from neighboring neurons

Regularization - $\mathcal{L}_2^2$, ridge penalty

- the typical setting: only one coefficient vector β, penalized by its squared norm $\mathcal{L}_2^2$.
- IDEA: borrow information *between* neighboring neurons
- write down the total log-likelihood, and put a penalty on the squared *distance* between estimates for neighboring neurons. Given a graph G:

  $$\text{e.g. } obj(\beta) = logP(y|X,\beta) - \lambda \sum_{(i,j)\in G} \|\beta_i - \beta_j\|_2^2$$

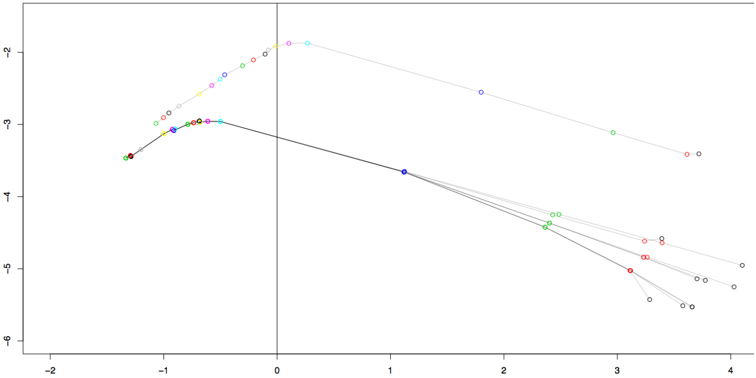
  This pushes estimates towards each other.
- The ridge penalty is often called $\mathcal{L}_2$-regularization. This name is misleading.

Regularization - Lasso and Group Lasso

- IDEA: borrow information between neighboring neurons
- the typical setting: Lasso penalty leads to sparse MAP estimates, i.e. some subset of the parameters will tend to be exactly zero
- here: some of the estimated parameters will be identical
- but it makes more sense to encourage the entire coefficient vectors to be identical
- Group Lasso (Yuan and Lin, 2006): encourages groups of parameters to go to 0 together
- groups of parameters $\equiv$ *vector-valued* parameters

$$obj(\beta) = logP(y|X,\beta) - \lambda \sum_{(i,j)} \|\beta_i - \beta_j\|_2$$

Lassoing the estimates together



- note how for every value of λ, we have a clustering of the neurons: the larger the λ, the fewer clusters.

Interpretation

We can always interpret penalized log-likelihood procedures as MAP inference with a prior that is proportional to the exponential of the penalty.

- e.g. Lasso corresponds to independent double-exponential (a.k.a. Laplace) prior on the coefficient vector.

However, when we have these penalty terms involving differences between β_is, these priors will not be independent.

Optimization

There is no closed-form solution to this optimization problem, so optimization is done by gradient ascent.

- Objective is convex but not automatically efficient
- Simple gradient ascent suffers from zig-zagging
- Interior-point method: start with a smoothed objective and gradually sharpen it

Thanks to Liam Paninski and Tim Teraväinen for discussions.