Let

, where .

Use AR(1) for

**Constraints:**

Easy one: Just let and be diagonal. (Now is free, yay!)

Maybe just let diagonal is enough? Why?

A picture containing shape

Description automatically generated

Because the number of constraint parameters is , which is equivalent to the number of constraints in traditional constraint for **normal factor model**. **Try later.**

Using less constraints will lead to a faster convergence.

**Key derivation for clustering: Marginal distribution of .**

I no longer need to use auxiliary parameters.

Denote

Maybe use the Laplace approximation?

, where is the Poisson density with mean .

If I use the Laplace approximation, I need to compute . This is quite painful. If there are neurons and clusters at current step, I need to run Newton-Raphson times!

Here, I just use the least square estimates, i.e., .

**Better approximation? Think…**

**Simulation 1: known label**

Here I use , but since now I include , the effective . So, there’s one more factor than the real



Trace for X



**Simulation 2: same setting as in simulation 1, but remove the label.**



OK, perfect. Let’s see what happens to a more nasty simulation.

**Simulation 3: as in simulation 3 in the slides**

Spikes



The trace of clusters:



Wow… Looks quite nasty, but when we check the clustering results for each neuron (at iteration 50 & 100), things are more meaningful.

|  |  |
| --- | --- |
| Iteration 50 | Iteration 100 (end of the chain) |
|  |  |

How about just use Poisson regression to estimate ? Better…

|  |  |
| --- | --- |
| trace | Iteration 100 (end of the chain) |
|  |  |

I also tried to run by using the Laplace approximation for . The results are similar.

**Although, the results seem (just seem) not as good as previous, but I strongly prefer the current model:**

1. The new model is cleaner & more elegant. Moreover, it has a better interpretation with less constraints, i.e., mean part () + variance part (). The value of just controls how detailed we want to model the variance.
2. The convergence & fitting in terms of mean firing rate is (**so**) much faster, especially when doing the clustering.
3. The seemly perfectness (, **which is fake**) in the previous method is because the convergence of is slow, and this gives a huge space for to tune itself… After convergence, the auxiliary will never beat , since is observed for . So, after a few iterations, the algorithm will just stop searching. It will even stop at some totally irrational results sometimes.
4. The previous method is not robust… When change the variance of or change , the results will change a lot.
5. The new method just does a more detailed clustering, not only consider the waveform but also take amplitude into consideration.
6. Maybe current imperfect performance comes from bad approximation of marginal likelihood.

**TODO**

The number of latent factor should be optimized. Choose by using the shrinkage prior and adaptive Gibbs sampler. See if I can easily do that.