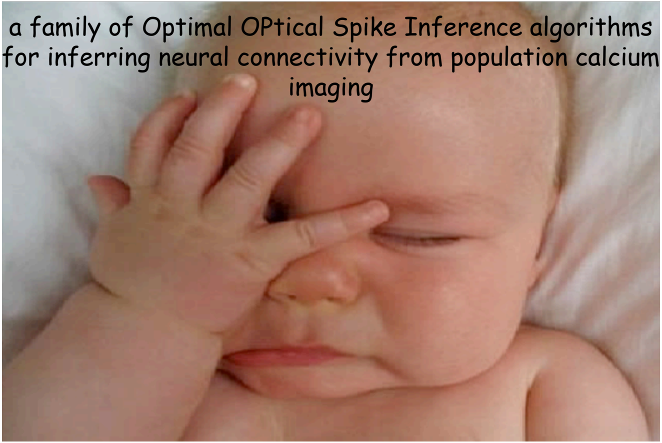
OOPSI

a family of Optimal OPtical Spike Inference algorithms for inferring neural connectivity from population calcium imaging



joshua tzvi cardin vogelstein
johns hopkins university
dec 1, 2009

the most important slide of the talk

acknowledgments

- you + me = us

a little motivation

why are we here?

- animals can do cool stuff
- way cooler than super-computers
- brains are causally related
- brains are well represented as networks of neurons
- the connectivity details are important for this coolness
- no way of determining connectivity (yet) (to our knowledge)
- knowing how leads to . . . love-bombs

outline

1. introduction
2. fast-oopsi: **fast** nonnegative deconvolution (what was spoken)
3. smc-oopsi: **s**equential **M**onte **C**arlo (neuron listening to itself)
4. pop-oopsi: **pop**ulation connectivity (neurons listening to one another)
5. discussion

outline

1. introduction
2. fast-oopsi: **fast** nonnegative deconvolution (what was spoken)
3. smc-oopsi: **s**equential **M**onte **C**arlo (neuron listening to itself)
4. pop-oopsi: **pop**ulation connectivity (neurons listening to one another)
5. discussion

a little neuro background
more specifics

this is your brain (on drugs?)

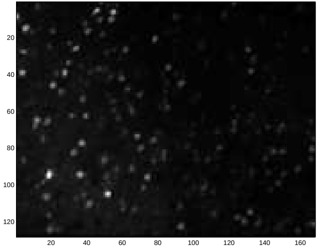
some beliefs some of us might have

- circles are neurons
- brightness (or fluorescence) corresponds with neural activity
- activity is how neurons communicate
- so, watching this movie is like listening in on a cocktail party
- we can use the activity to figure out who is speaking to whom

what's hard about that?

things that make it hard (for us)

- we are not very attentive
- we are hearing impaired
- every neuron is a little different



what did we do?

also known as: “primary aims”

- 1. fast-oopsi what was spoken (fast nonnegative deconvolution)
- 2. smc-oopsi neuron listening to itself (sequential Monte Carlo)
- 3. pop-oopsi neurons listening to one another (population connectivity)

what are we going to do?

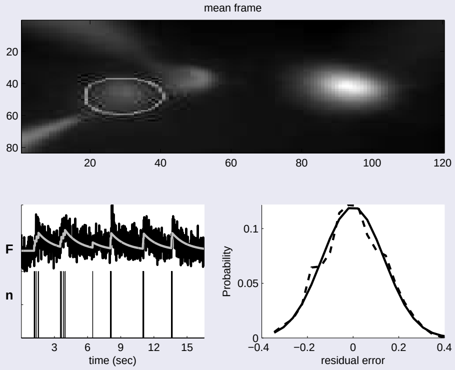
the strategy for each aim

- write down a model, explaining the data
- state our goal
- develop an algorithm to (approximately) achieve that goal
- test the approach on data

outline

model

data

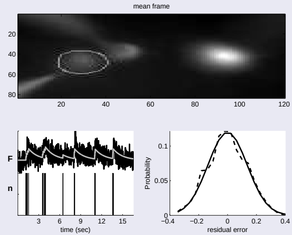


description

- circles = neurons
- black squiggly line = fluorescence
- neuron speaking = spikes
- gray line = calcium
- stuff we don't understand = noise

model

data



description

- circles = neurons
- black squiggly line = fluorescence
- neuron speaking = spikes
- gray line = calcium
- stuff we don't understand = noise

equations

$$F_t = \alpha C_t + \beta + \sigma\varepsilon_t, \qquad \varepsilon_t \overset{iid}{\sim} \mathcal{N}(0,1)$$

$$C_t = -(1 - \Delta/\tau)C_{t-1} + n_t$$

$$n_t \sim \text{Poisson}(\lambda\Delta)$$

goal

finding the most likely spike train given the data

$$\widehat{\mathbf{n}} = \underset{\mathbf{n}}{\text{argmax}}\, P(\mathbf{n}|\mathbf{F}) = \underset{\mathbf{n}}{\text{argmax}}\, P(\mathbf{F}|\mathbf{n})P(\mathbf{n})$$

some fancy terms

- posterior: $P(\mathbf{n}|\mathbf{F})$ is the prob. of a spike train, given the observations
- likelihood: $P(\mathbf{F}|\mathbf{n})$ is the likelihood of the data, given the spikes
- prior: $P(\mathbf{n})$ is the probability of any particular sequence of spikes

goal

finding the most likely spike train given the data

$$\widehat{\mathbf{n}} = \operatorname*{argmax}_{\mathbf{n}} P(\mathbf{n}|\mathbf{F}) = \operatorname*{argmax}_{\mathbf{n}} P(\mathbf{F}|\mathbf{n}) P(\mathbf{n})$$

some fancy terms

- posterior: $P(\mathbf{n}|\mathbf{F})$ is the prob. of a spike train, given the observations
- likelihood: $P(\mathbf{F}|\mathbf{n})$ is the likelihood of the data, given the spikes
- prior: $P(\mathbf{n})$ is the probability of any particular sequence of spikes

model equations

$$F_t = \alpha C_t + \beta + \sigma \varepsilon_t, \qquad \varepsilon_t \overset{iid}{\sim} \mathcal{N}(0,1) \tag{1}$$

$$C_t = -(1-\Delta/\tau) C_{t-1} + n_t, \qquad n_t \overset{iid}{\sim} \text{Poisson}(\lambda\Delta) \tag{2}$$

goal

finding the most likely spike train given the data

$$\widehat{\mathbf{n}} = \operatorname*{argmax}_{\mathbf{n}} P(\mathbf{n}|\mathbf{F}) = \operatorname*{argmax}_{\mathbf{n}} P(\mathbf{F}|\mathbf{n})P(\mathbf{n})$$

model equations

$$F_t = \alpha C_t + \beta + \sigma \varepsilon_t, \qquad \varepsilon_t \overset{iid}{\sim} \mathcal{N}(0,1) \tag{1}$$

$$C_t = -(1 - \Delta/\tau) C_{t-1} + n_t, \qquad n_t \overset{iid}{\sim} \text{Poisson}(\lambda\Delta) \tag{2}$$

plugging in

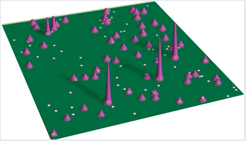
- likelihood defined by eq. (1)
- prior defined by eq. (2)

algorithm = blind navigator

finding most likely sequence of spikes

- we can't actually search for the most likely sequence of spikes, because their are too many, and the search space is too mountainous

nonlinear optimization



algorithm = blind navigator

finding most likely sequence of spikes

- search space is too mountainous
- we can approximate the landscape to just be one big mountain

log-concave maximization

algorithm = blind navigator

finding most likely sequence of spikes

- search space is too mountainous
- one big mountain
- we use an approximation that is the closest big mountain

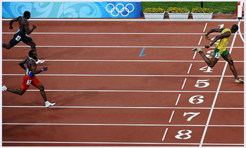
nonnegative constraint with interior point method

algorithm = blind navigator

finding most likely sequence of spikes

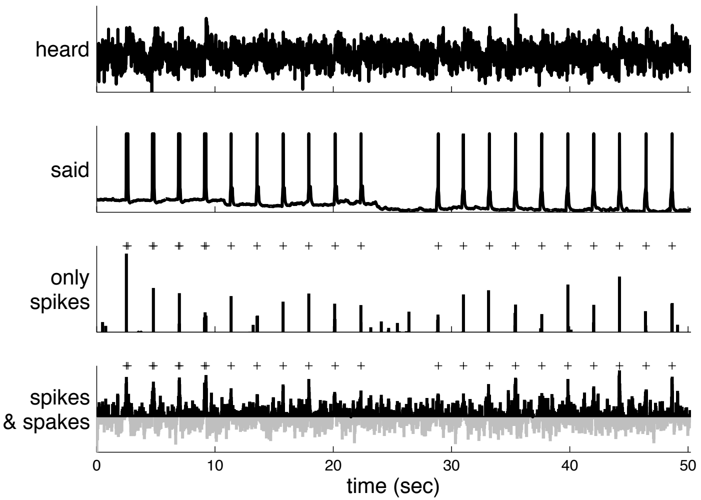
- search space is too mountainous
- one big mountain
- closest big mountain
- we can run up the mountain

Gaussian elimination on tridiagonal Hessian



main results

our best guess is pretty good for real data



demo

who wants to see a demo?

discussion

fast nonnegative deconvolution

- accurate because only spikes are allowed
- quick
- not introspective

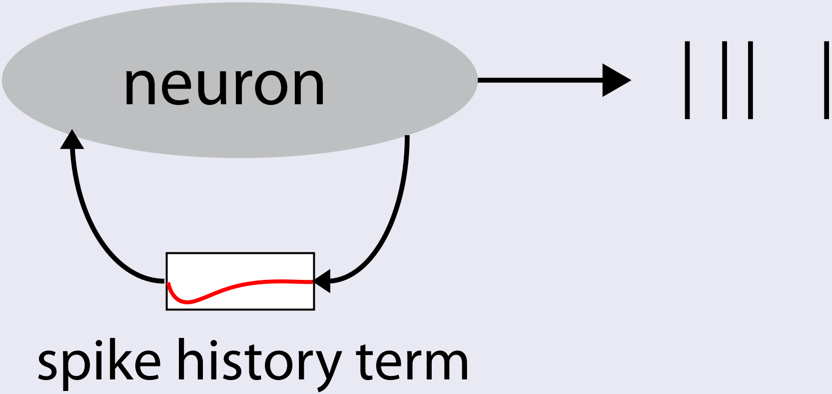
dash

outline

model

neuron listening to itself

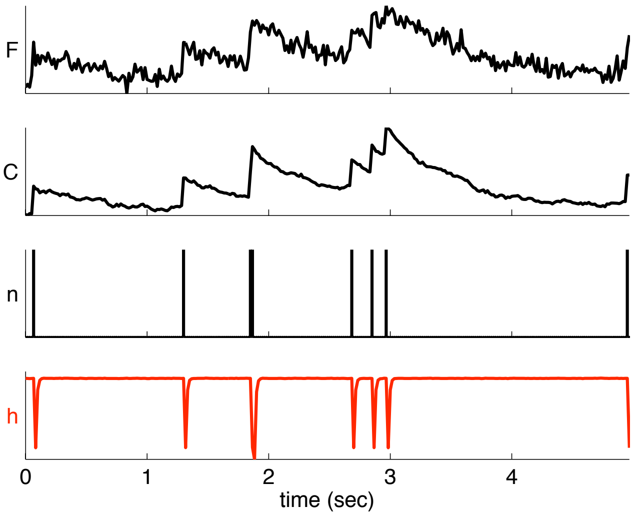
model



model

neuron listening to itself

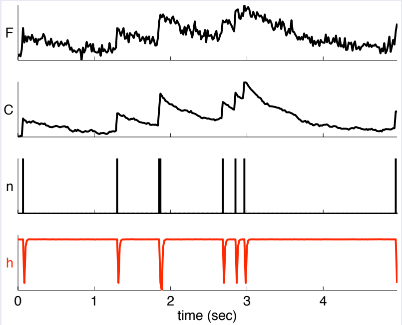
schematic



model

neuron listening to itself

schematic



generative model

$$F_t = \alpha \frac{C_t}{C_t + k_d} + \beta + \sigma_F \varepsilon_F$$

$$C_t = \gamma_c C_{t-1} + C_b + A n_t + \sigma_c \varepsilon_c$$

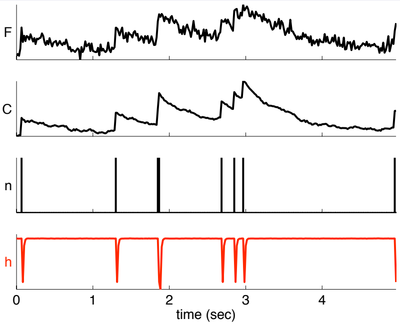
$$n_t \sim \text{Bernoulli}(\text{f}(\mathbf{w}^{\mathsf{T}}\mathbf{h}_t)\Delta)$$

$$h_t = \gamma_h h_{t-1} - n_t + \sigma_h \varepsilon_h$$

model

neuron listening to itself

schematic



generative model

$$F_t = \alpha \frac{C_t}{C_t + k_d} + \beta + \sigma_F \varepsilon_F$$

$$C_t = \gamma_c C_{t-1} + C_b + A n_t + \sigma_c \varepsilon_c$$

$$n_t \sim \text{Bernoulli}(\text{f}(\mathbf{w}^{\mathsf{T}} \mathbf{h}_t)\Delta)$$

$$h_t = \gamma_h h_{t-1} - n_t + \sigma_h \varepsilon_h$$

some thoughts

- we can now have each neuron listen to itself
- previous methods won't work here

new goal, new method

new goal

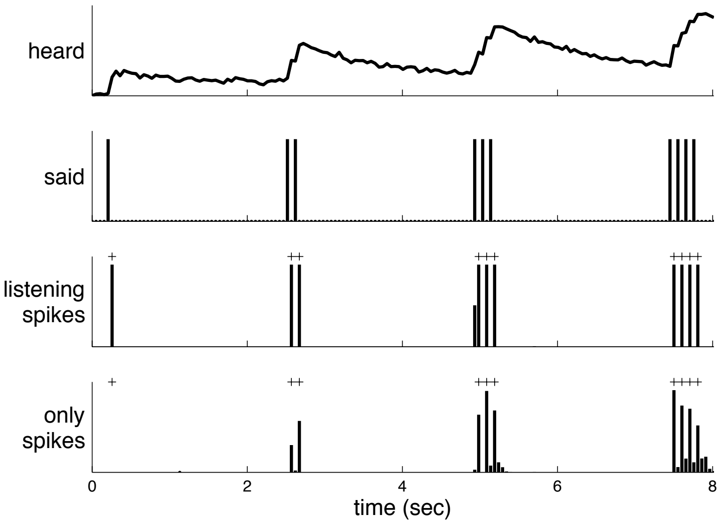
- find the probability of a spike happening in each frame, incorporating the neuron listening to itself

new method: sequential Monte Carlo methods, which is an approximate forward-backward technique

- step forward, guess at each time how likely is it that a spike happened
- get all your friends to do the same
- repeat for each frame
- when at the end, turn around to go backward and count the votes for each frame

main result

listening helps in real neurons



demo

who wants to see a demo?

discussion

smc-oopsi

- more accurate than fast nonnegative deconvolution
- allows each neuron to listen to itself
- can be extended

elastic girl

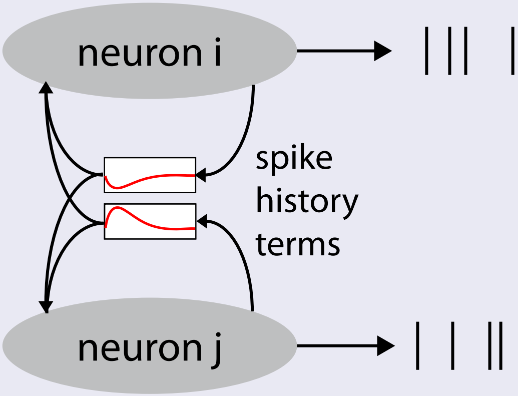
outline

pop-oopsi: **pop**ulation connectivity

model

neurons listening to one another

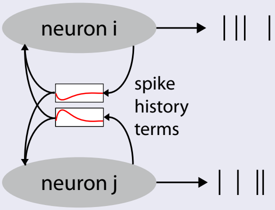
schematic



model

neurons listening to one another

schematic



model

$$F_i(t) = \alpha_i C_i(t)/(C_i(t)+k_d) + \beta_i + \sigma_i^F \varepsilon_i^F$$

$$C_i(t) = \gamma_i^c C_i(t-1) + C_i^b + A_i n_i(t) + \sigma_i^c \varepsilon_i^c$$

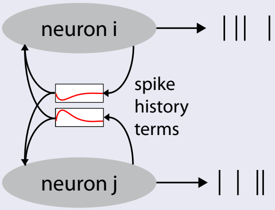
$$n_i(t) \sim \text{Bernoulli}\left(\text{f}\left(\sum_{j=1}^{N} w_{ij} h_j(t)\right)\Delta\right)$$

$$h_j(t) = \gamma_j^h h_j(t-1) + n_j(t) + \sigma_i^h \varepsilon_j^h$$

model

neurons listening to one another

schematic



model

$$F_i(t) = \alpha_i C_i(t)/(C_i(t)+k_d) + \beta_i + \sigma_i^F \varepsilon_i^F$$

$$C_i(t) = \gamma_i^c C_i(t-1) + C_i^b + A_i n_i(t) + \sigma_i^c \varepsilon_i^c$$

$$n_i(t) \sim \text{Bernoulli}\Big(\text{f}\big(\sum_{j=1}^{N} w_{ij} h_j(t)\big)\Delta\Big)$$

$$h_j(t) = \gamma_j^h h_j(t-1) + n_j(t) + \sigma_i^h \varepsilon_j^h$$

some thoughts

- listening to each other
- how carefully is each listening to the others: w_{ij} is the synaptic weight
- description of the whole party: $\mathbf{w} = \{w_{ij}\}_{i,j \le N}$ is the connectivity matrix

goal and algorithm

find the most likely connection matrix, given the fluorescence

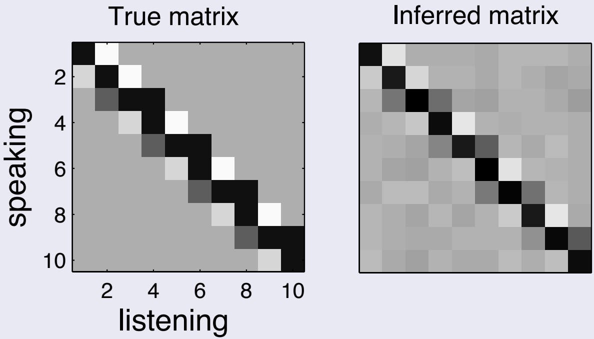
$$\widehat{\mathbf{w}} = \operatorname*{argmax}_{\mathbf{w}} P(\mathbf{n}|\mathbf{F};\mathbf{w})$$

obtaining $\widehat{\mathbf{w}}$

- initialize spike trains using smc-oopsi
- for each neuron
  1. assume it is listening to everybody
  2. estimate how much it cares about each other neuron, w_{ij}
- put it all together

main result

we can determine who is speaking to whom



discussion

population connectivity

- can accurately identify who is speaking to whom
- not yet vetted on real data

outline

discuss

what did we do?

we've built some useful tools, wethinks. . .

- fast-oopsi (fast nonnegative deconvolution) is fast and accurate
- smc-oopsi improves inference results, by allowing each neuron to listen to itself
- pop-oopsi seems to correctly identify who is speaking to whom by allowing the neurons to listen to one another

what's next?

woopsi?

what's next?

woopsi?

applying to real data

- use data where neurons are labeled either excitatory or inhibitory
- can't confirm how attentive each neuron is, but at least whether each is attentive
- multiple stabbings confirms how attentive any pair of neurons are to one another

the most important slide of the talk

acknowledgments...

- you
- moral and financial support: eric young
- theory support: liam paninski's group (baktash and yuriy), bruno
- data support: rafa yuste's group (brendon, adam, tanya, tim)
- emotional support: "me", family, friends, the earth, the universe, etc.

this talk has been brought to you by...

- the letters: y, e, s
- NIDCD DC00109
- and the number: 1