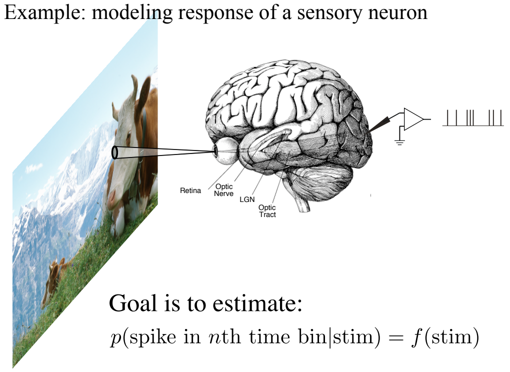
Fitting models to data

- How do we estimate parameters?
  - formulate model + objective function (common choice: ML)
  - optimize
- How good are parameter estimates?
- How well does model fit ?
  - likelihood or posterior comparisons
  - model failures

Example: modeling response of a sensory neuron

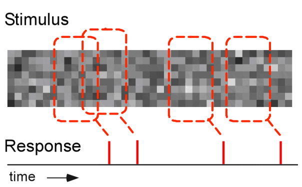


Goal is to estimate:

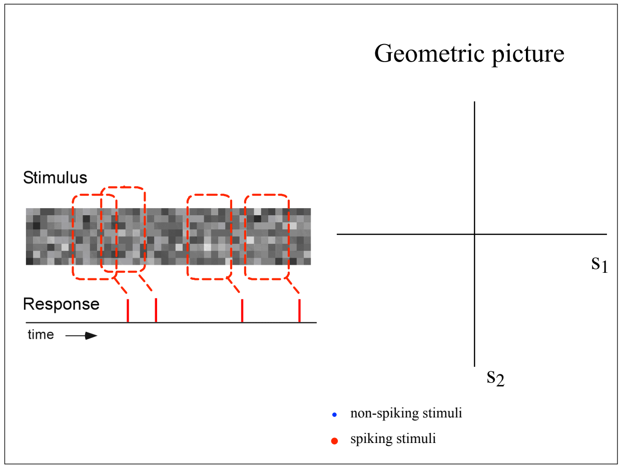
$p(\text{spike in } n\text{th time bin}|\text{stim}) = f(\text{stim})$

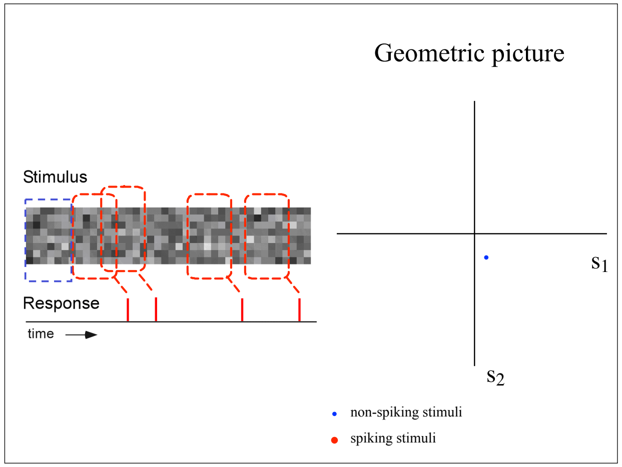
Geometric view

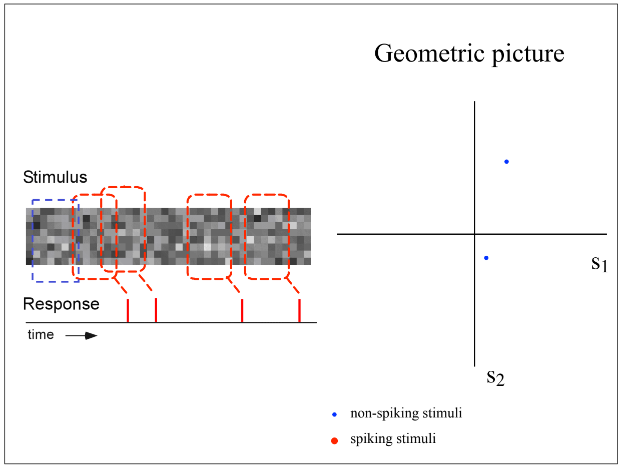
1D stimulus over time
(e.g., flickering bars)

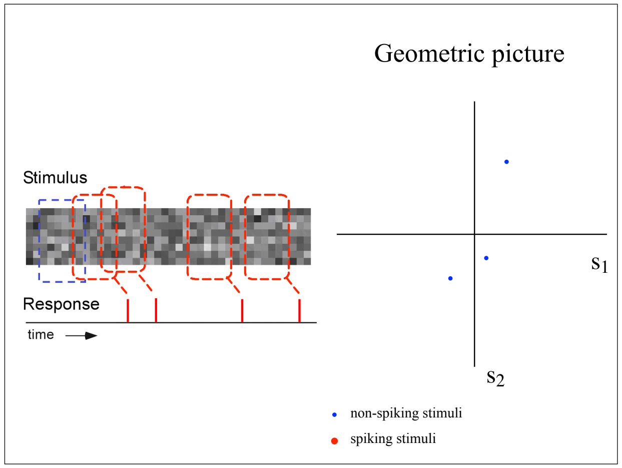


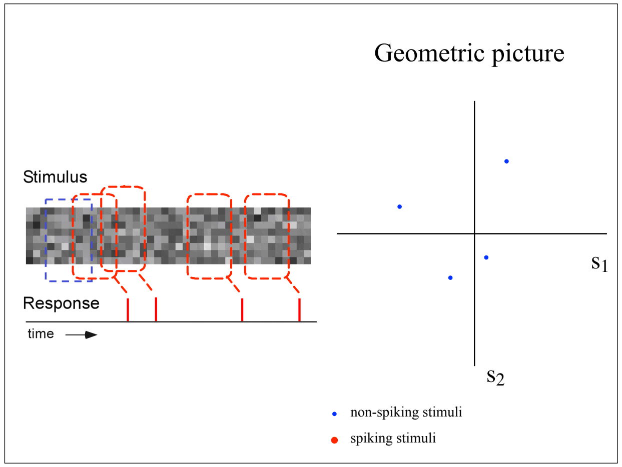
- 8 x 6 stimulus block
  = 48-dimensional vector

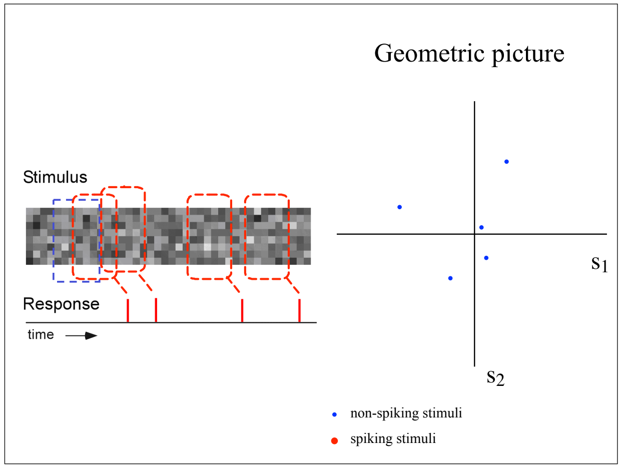

Geometric picture
Stimulus
Response
time
s1
s2
non-spiking stimuli
spiking stimuli

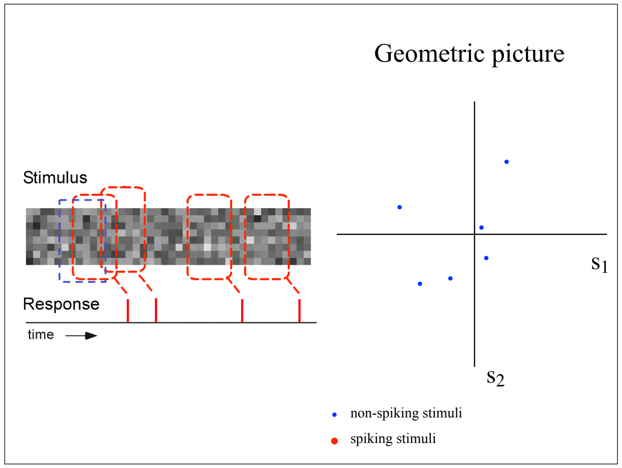

Geometric picture
Stimulus
Response
time
s1
s2
non-spiking stimuli
spiking stimuli

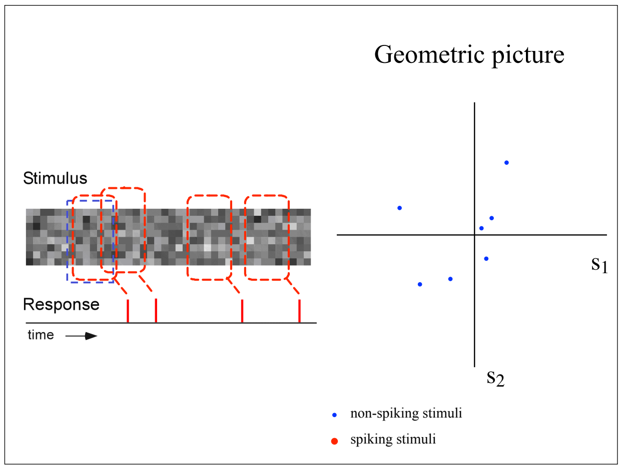

Geometric picture
Stimulus
Response
time
s1
s2
non-spiking stimuli
spiking stimuli

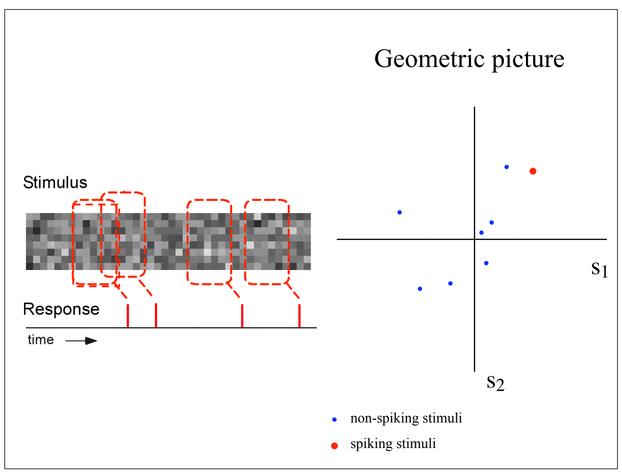

Geometric picture
Stimulus
Response
time
s_1
s_2
non-spiking stimuli
spiking stimuli


Geometric picture
Stimulus
Response
time
s_1
s_2
non-spiking stimuli
spiking stimuli

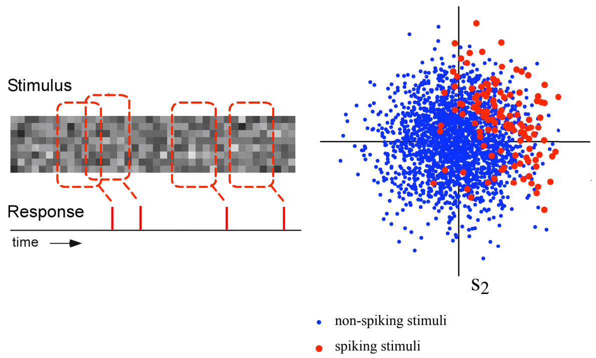

Geometric picture
Stimulus
Response
time
s_1
s_2
non-spiking stimuli
spiking stimuli


Geometric picture
Stimulus
Response
time
s1
s2
non-spiking stimuli
spiking stimuli

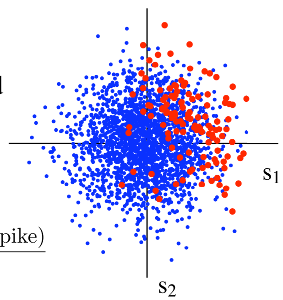

Geometric picture
Stimulus
Response
time
s1
s2
non-spiking stimuli
spiking stimuli


Geometric picture
Stimulus
Response
time
s1
s2
non-spiking stimuli
spiking stimuli

Geometric picture



Response is captured by relationship between the distribution of red points (spiking stim) and blue+red points (all stim), expressed in terms of Bayes' rule:



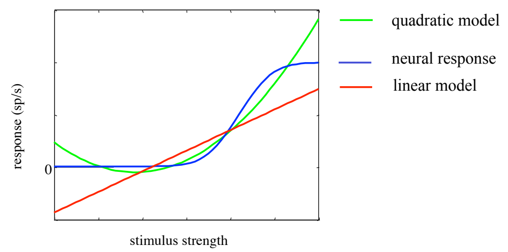
$$P(\text{spike}|\vec{s}) = \frac{P(\vec{s}\ |\text{spike})P(\text{spike})}{P(\vec{s})}$$

Cannot estimate directly ("curse of dimensionality"). We need a **model**

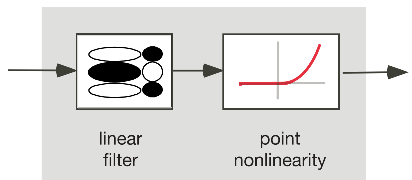
Some tractable model options

- Low-order polynomial [Volterra '13; Wiener '58; DeBoer and Kuyper '68; ...]
- Low-dimensional subspace [Bialek '88; Brenner etal '00; Schwartz etal '01; Touryan and Dan '02; ...]
- Recursive linear with exponential nonlinearity [Truccolo etal '05; Pillow etal '05]

Low-order polynomial model

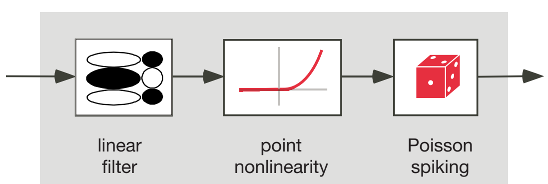


Example: LN cascade model



- Threshold-like nonlinearity => linear classifier
- Classic model for Artificial Neural Networks
  - McCullough & Pitts (1943), Rosenblatt (1957), etc
- No spikes (output is firing rate)

LNP cascade model



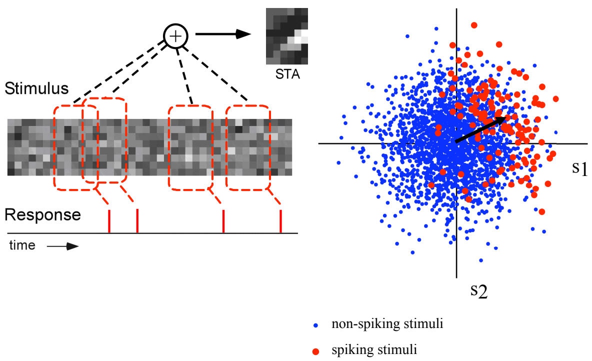
- Simplest descriptive spiking model
- Easily fit to (extracellular) data
- Descriptive, and interpretable (although *not* mechanistic)

Simple LNP fitting

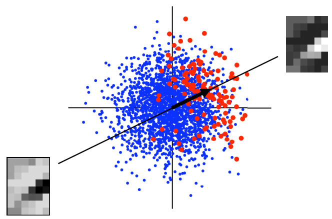
- Assuming:
  - stochastic stimuli, spherically distributed
  - average of spike-triggered ensemble (STA) is shifted from that of raw ensemble
- The STA (i.e., linear regression!) gives an **unbiased** estimate of w (for any f). *[on board]*
- For exponential f, this is the ML estimate! *[on board]*

[Bussgang 52; de Boer & Kuyper 68]

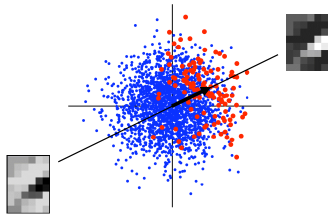
Computing the STA



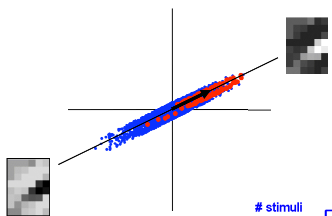
STA corresponds to a “direction” in stimulus space



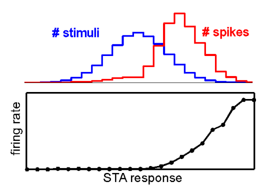
Projecting onto the STA



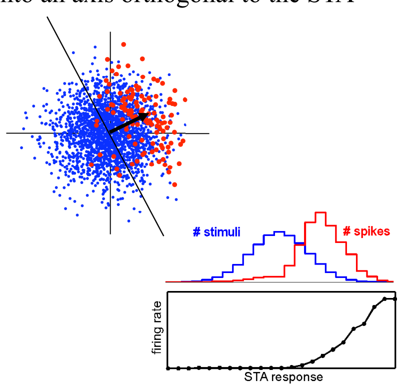
Solving for nonparametric nonlinearity



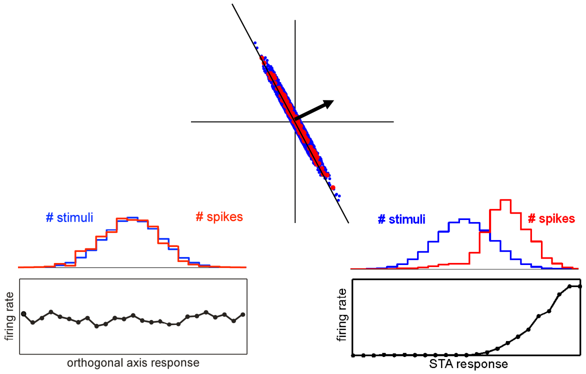
$$P(\text{spike}|\vec{s}) = \frac{P(\vec{s}\ |\text{spike})P(\text{spike})}{P(\vec{s})}$$



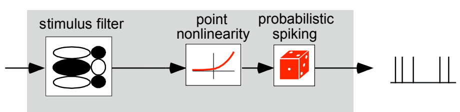
Projecting onto an axis orthogonal to the STA

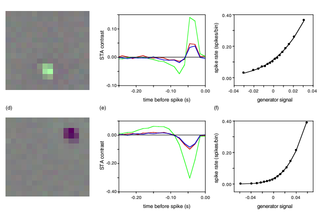


Projecting onto an axis orthogonal to the STA

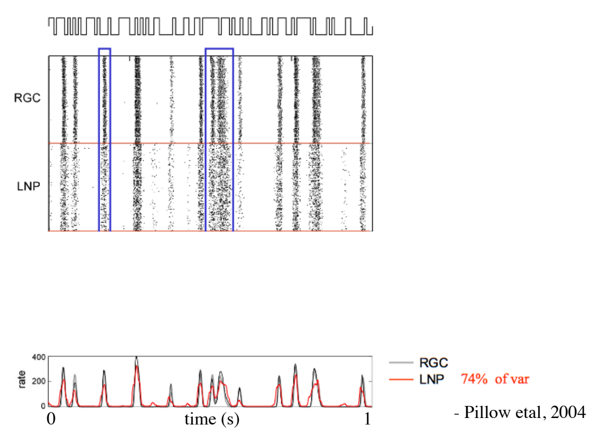


LNP model, fit to retinal ganglion cells



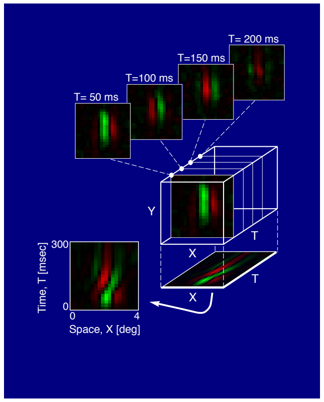


[Chichilnisky & Kalmer, 2002]

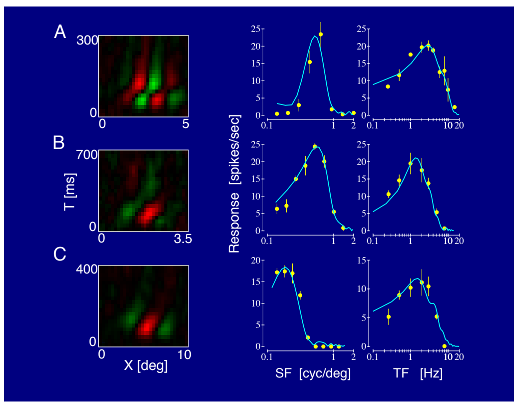


- Pillow etal, 2004

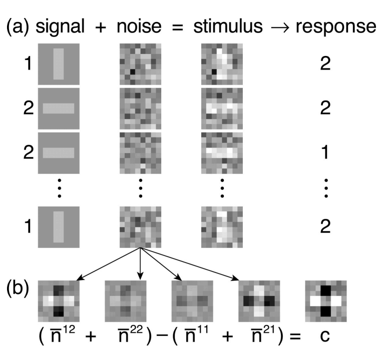
V1
simple
cell



- Ozhawa, etal

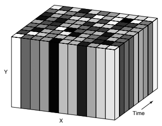


Psychophysical “Classification Image”

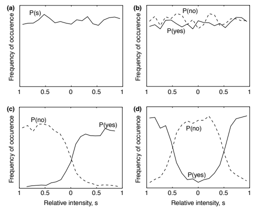


Murray R F J Vis 2011;11:2

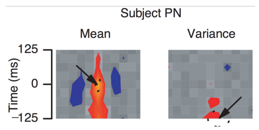
Stimuli: 11x9 movie of bars, uniform random intensities



Task:
Is center bar of middle frame brighter or darker than the mean?



Simulation:
a) raw stimulus distribution
b) cond. dist. for irrelevant bar
c) cond. dist. for linear response model
d) cond. dist. for quadratic (contrast) response



[Neri & Heeger, 2002]

ML estimation of LNP

If $f_\theta(\vec{k} \cdot \vec{x})$ is convex (in argument and theta),
and $log f_\theta(\vec{k} \cdot \vec{x})$ is concave,
the likelihood of the LNP model is convex
(for all data, $\{n(t), \vec{x}(t)\}$)

Examples: $e^{(\vec{k} \cdot \vec{x}(t))}$

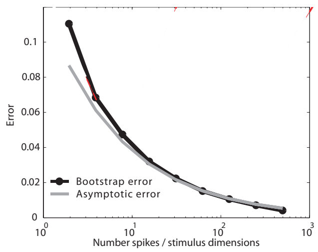
$(\vec{k} \cdot \vec{x}(t))^\alpha, \quad 1 < \alpha < 2$

[Paninski, '04]

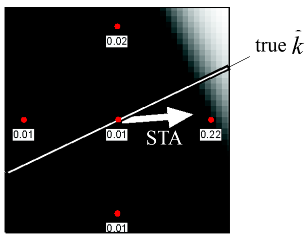
Sources of STA estimation error

- Finite data (convergence goes as 1/N)
- Non-spherical stimuli (estimator can be biased)
- Model failures. Examples:
  - symmetric nonlinearity (causes no change in STE mean)
  - response not captured by 1D linear projection
  - spike history dependence (non-Poisson)

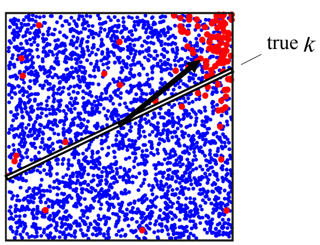
Variance behavior of STA



Example 1:
"sparse" noise



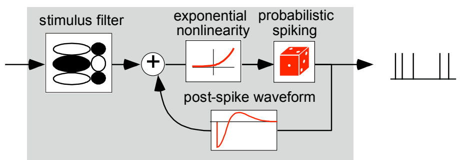
Example 2:
uniform noise



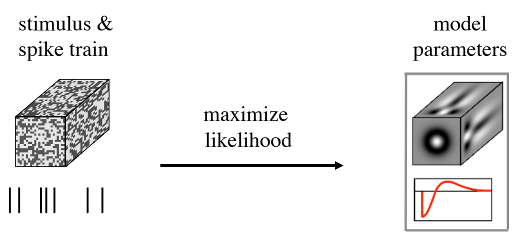
LNP model limitations

- Neural response depends on spike history
  => introduce spike history feedback
- Symmetric nonlinearities and/or multi-dimensional front-end (e.g., V1 complex cells)
  => spike-triggered covariance, subspace analyses
- White noise doesn't drive mid- to late-stage neurons well
  => cascade LNP on top of an "afferent" model

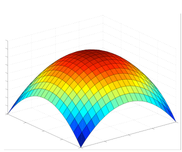
Recursive LNP



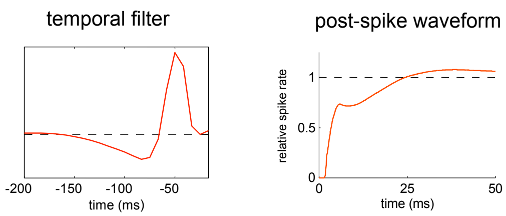
[Truccolo et al '05; Pillow et al '05]

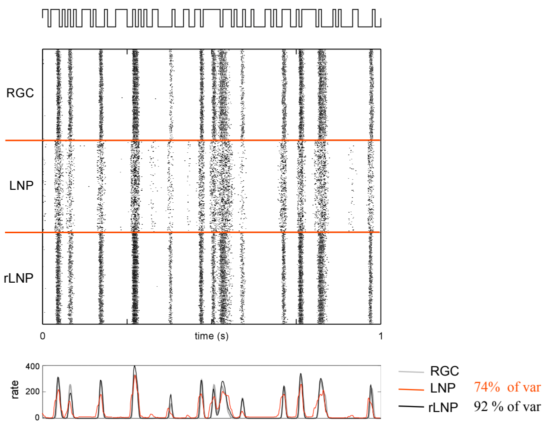


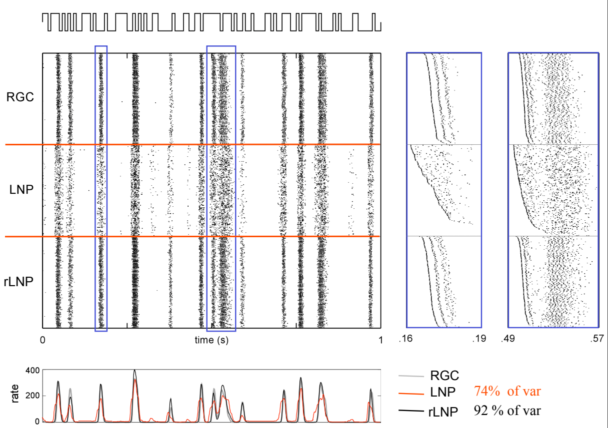
Critical: Likelihood function has no local maxima [Paninski 04]

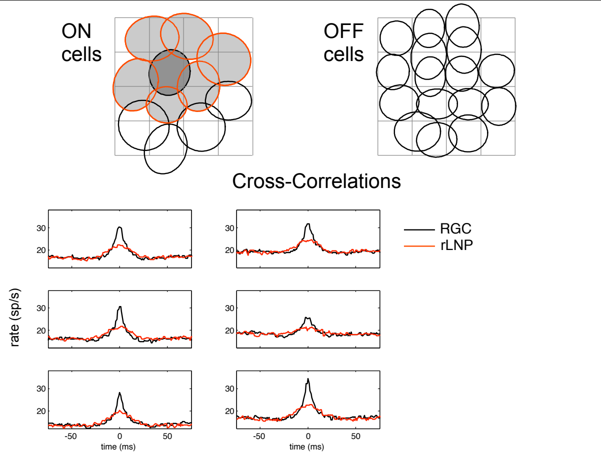


Example ON cell

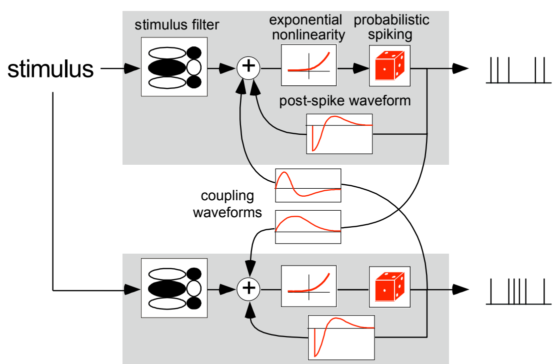


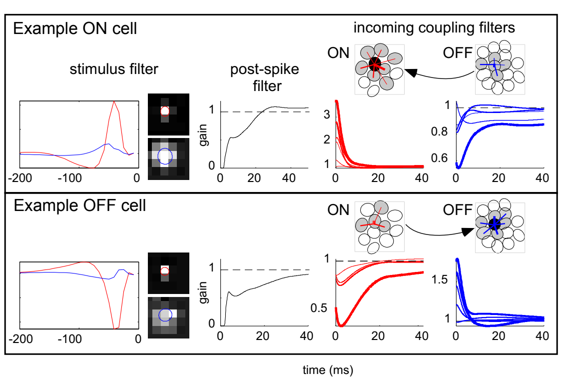




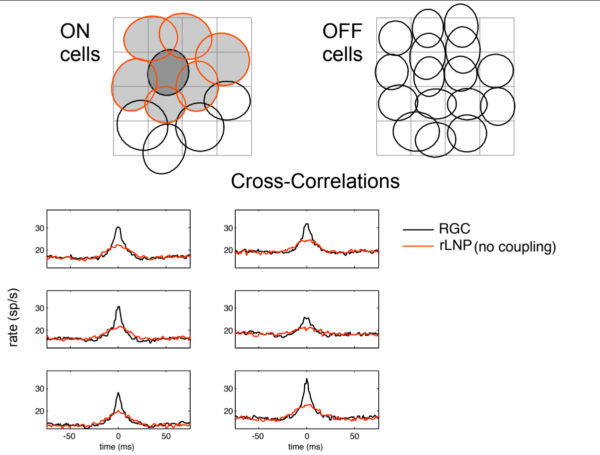


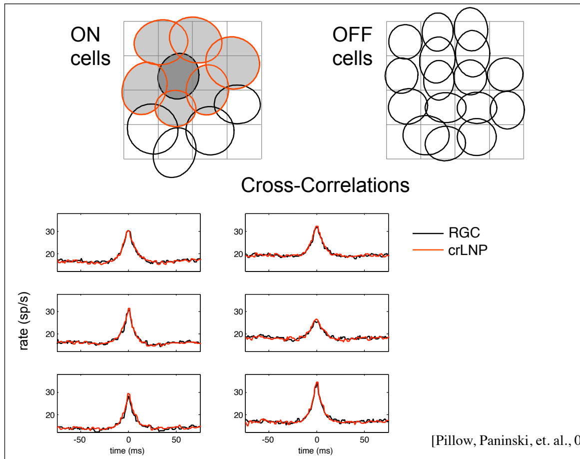
Coupled recursive LNP (crLNP)





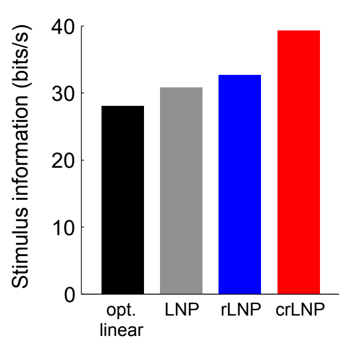
[Pillow, Paninski, et. al., 08]





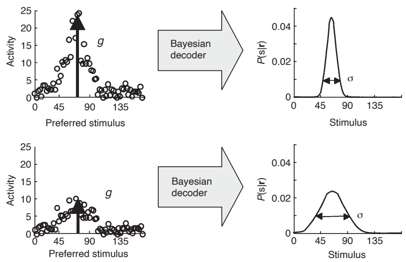
[Pillow, Paninski, et. al., 08]

Decoding



[Pillow, Paninski, et. al., 08]

Probabilistic Population Codes



$$p(\theta\,|\,\vec{r}) \propto p(\vec{r}\,|\,\theta)p(\theta)$$

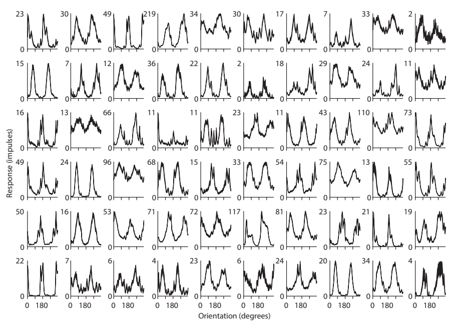
For independent Poisson firing rates: $p(\theta\,|\,\vec{r}) \propto \prod_i \frac{e^{-f_i(\theta)}f_i(\theta)^{r_i}}{r_i!}p(\theta)$

Integration by summing spikes!

[Ma, Beck, Latham & Pouget, 06]

Population Decoding

The data: tuning curves f_i



[Graf, Kohn, Jazayeri & Movshon, 11]

Probabilistic Population Codes

Poisson independent decoder:

$$\log L(\theta) = \log\left(\prod_{i=1}^{N} p(r_i\,|\,\theta)\right) = \sum_{i=1}^{N} \log\left(\frac{f_i(\theta)^{r_i}}{r_i!}\exp(-f_i(\theta))\right)$$

$$= \sum_{i=1}^{N} \log(f_i(\theta))r_i - \sum_{i=1}^{N} f_i(\theta) - \sum_{i=1}^{N} \log(r_i!) = \sum_{i=1}^{N} W_i(\theta)r_i + B(\theta)$$

For discrimination, compute a linear discriminant:

$$\log LR(\theta_1,\theta_2) = \log\left(\frac{L(\theta_1)}{L(\theta_2)}\right) = \log L(\theta_1) - \log L(\theta_2)$$

$$= \sum_{i=1}^{N} [W_i(\theta_1) - W_i(\theta_2)]r_i + [B(\theta_1) - B(\theta_2)]$$

$$= \sum_{i=1}^{N} w_i(\theta_1,\theta_2)r_i + b(\theta_1,\theta_2)$$

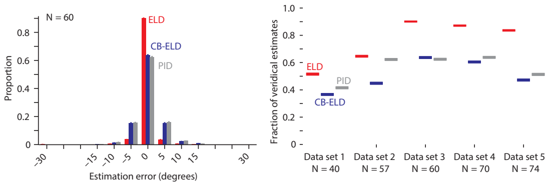
[Graf, Kohn, Jazayeri & Movshon, 11]

Probabilistic Population Codes

Alternatively, take the set of response vectors to each orientation and compute an SVM of identical form, the empirical linear decoder:

$$y(\theta_1, \theta_2) = \sum_{i=1}^{N} w_i(\theta_1, \theta_2) r_i + b(\theta_1, \theta_2) \equiv \log LR(\theta_1, \theta_2)$$

Finally, shuffle the firing rates independently across trials for each neuron and orientation and retrain the SVM, yielding a correlation-blind empirical linear decoder.



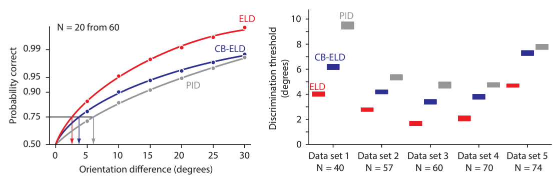
[Graf, Kohn, Jazayeri & Movshon, 11]

Probabilistic Population Codes

Alternatively, take the set of response vectors to each orientation and compute an SVM of identical form, the empirical linear decoder:

$$y(\theta_1, \theta_2) = \sum_{i=1}^{N} w_i(\theta_1, \theta_2) r_i + b(\theta_1, \theta_2) \equiv \log LR(\theta_1, \theta_2)$$

Finally, shuffle the firing rates independently across trials for each neuron and orientation and retrain the SVM, yielding a correlation-blind empirical linear decoder.



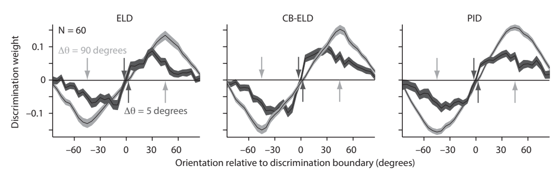
[Graf, Kohn, Jazayeri & Movshon, 11]

Probabilistic Population Codes

Alternatively, take the set of response vectors to each orientation and compute an SVM of identical form, the empirical linear decoder:

$$y(\theta_1, \theta_2) = \sum_{i=1}^{N} w_i(\theta_1, \theta_2) r_i + b(\theta_1, \theta_2) \equiv \log LR(\theta_1, \theta_2)$$

Finally, shuffle the firing rates independently across trials for each neuron and orientation and retrain the SVM, yielding a correlation-blind empirical linear decoder.



[Graf, Kohn, Jazayeri & Movshon, 11]