# Bayesian Decoding of Population Responses 2

Lecture 8

ACN

JV Stone

# Structure

- Tuning curves and Gaussian noise
- Likelihoods by eye
- Interlude: Bayes and MLE
- Evaluating the MLE: Two ‘tricks’
- MLE with Gaussian noise
  - Evaluating the likelihood *L(s)*
  - MLE and least-squares
- MLE with Poisson noise
  - The MLE $\hat{s}$ as a weighted mean
- MLE using recurrent nets

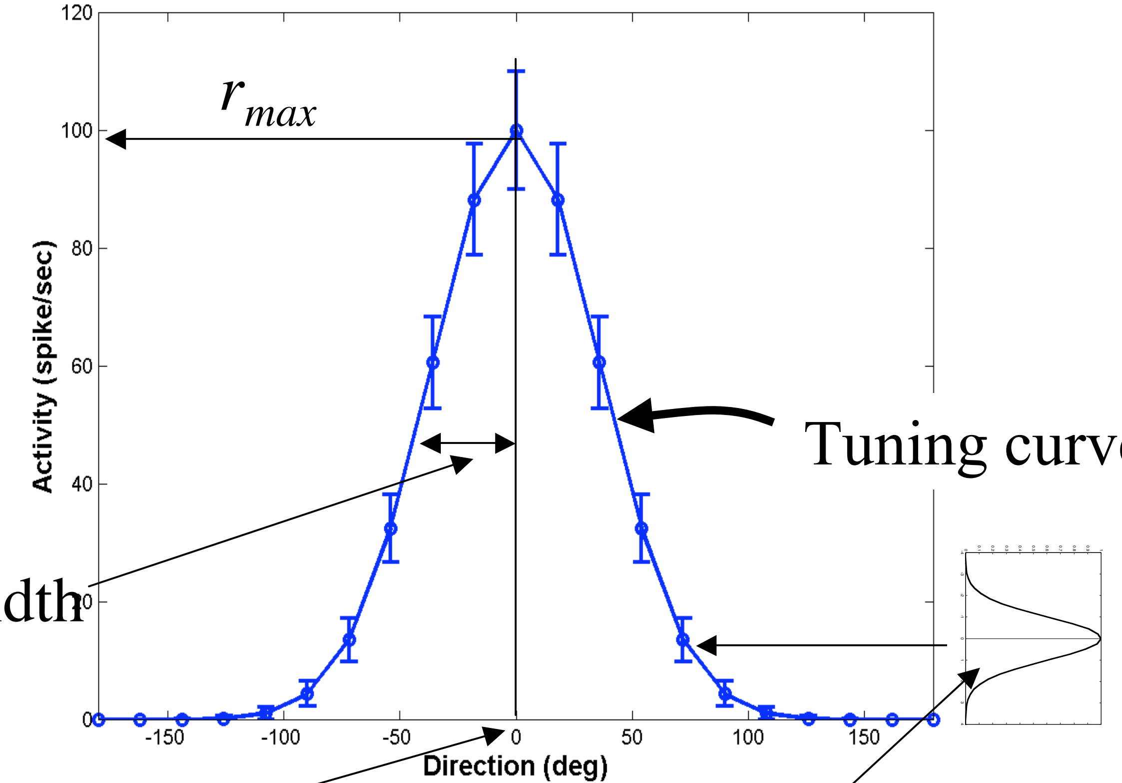
# Notation

| | |
|---|---|
| $f_i(s)$ | tuning curve of *i*th cell |
| $s_i$ | **preferred direction of *i*th cell** |
| $\sigma_i$ | standard deviation of tuning curve of *i*th cell |
| $v_i$ | variance of noise or response distribution |
| $n$ | number of spikes |
| $n_i$ | noise in response of *i*th cell, $n_i = (f_i(s) - r_i)$ |
| $s^*$ | value of stimulus presented to system |
| $\hat{s}$ | estimate of $s^*$ |
| MAP | maximum a posteriori estimate of $s^*$ |
| MLE | maximum likelihood estimate of $s^*$ |
| LSE | least-squares estimate of $s^*$ |

# Tuning Curves and Gaussian noise

$$f_i(s) \;=\; r_{max}\, e^{-(s_i - s)^2/(2\sigma_i^2)}$$

**$s_i$ = preferred direction**




Tuning curve width parameter $\sigma$

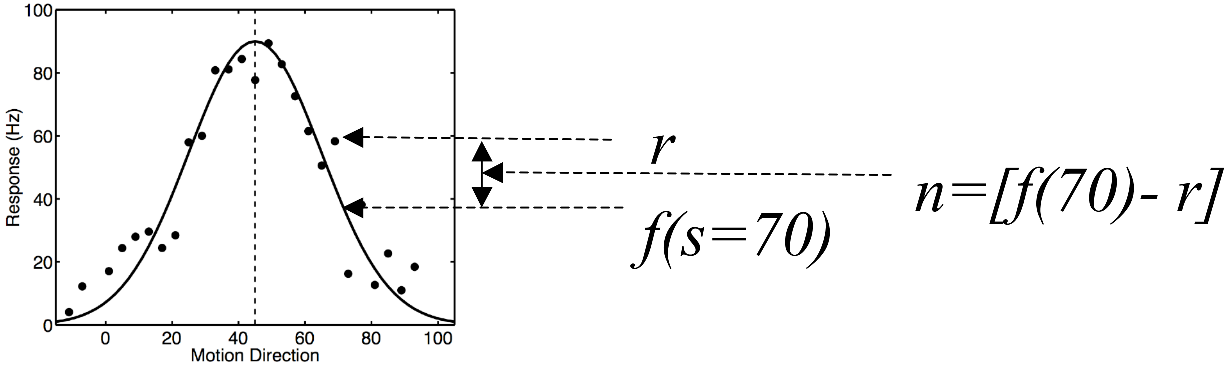
$s_i$ = preferred direction

Noise variance width $v_i$

# Tuning Curves and Gaussian noise

The response $r_i$ to a stimulus value $s$ is the tuning curve height $f_i(s)$ at $s$ plus noise $n_i$

$$r_i = f_i(s) + n_i$$

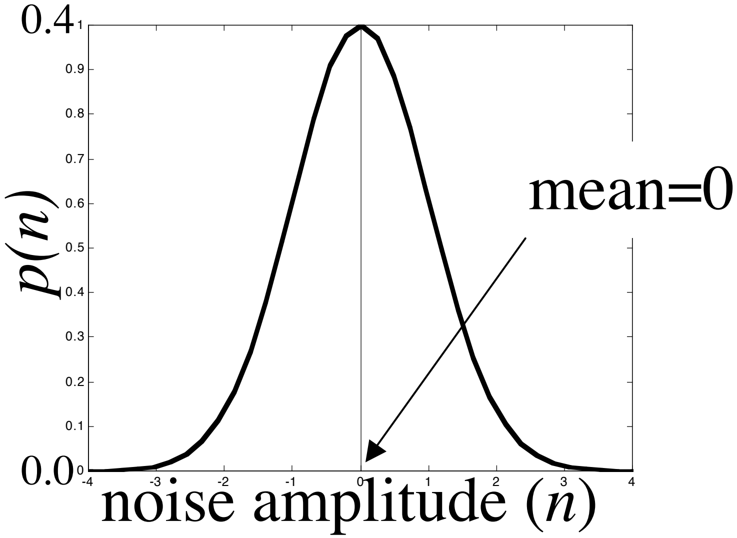



The noise is a random variable with zero mean and variance $v_i$. If noise is Gaussian then the probability (density) of noise value $n_i$ is

$$p(n_i) = k_i e^{-n_i^2/(2v_i)}$$

where $v_i$ is the ***variance*** (the square of the ***standard deviation***),

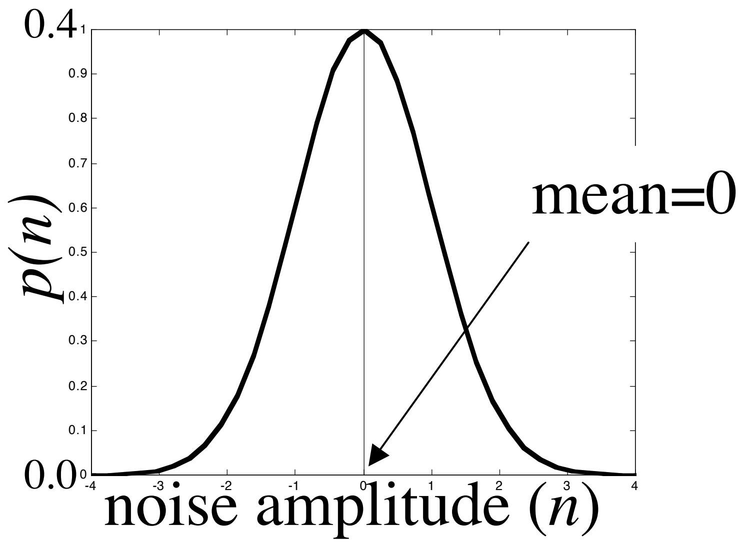
$k_i = 1/sqrt(2\pi v_i)$

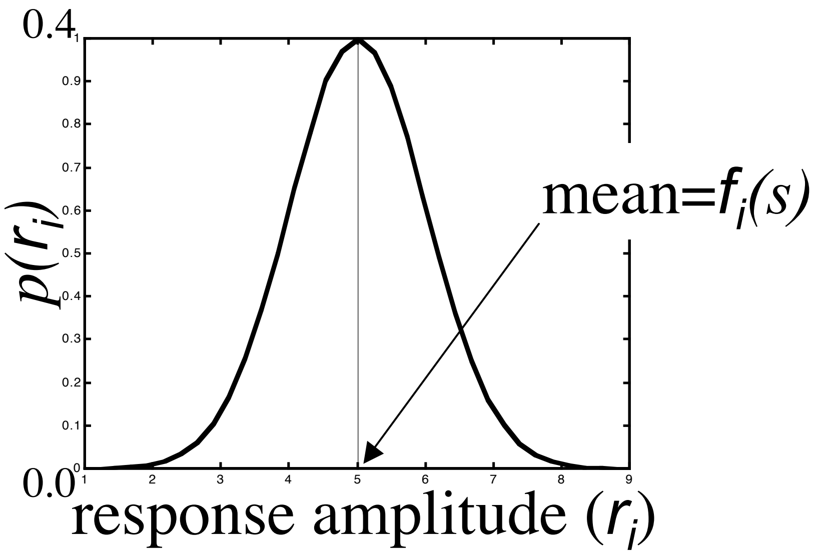


# Tuning Curves and Gaussian noise

If the noise is a random variable with zero mean and variance $v_i$ then the response $r_i$ is also a random variable with variance $v_i$, but with mean $f_i(s)$. The probability of observing a response $r_i$ given a stimulus direction $s$ is

$$p(r_i|s) \quad = \quad k_i e^{-(r_i - f_i(s))^2/(2v_i)}$$



# Interim summary

Now have equations for Gaussian tuning function

$$f_i(s) \;=\; r_{max}\, e^{-(s_i-s)^2/(2\sigma_i^2)}$$

… and for pdf of firing rate $r_i$ with Gaussian noise

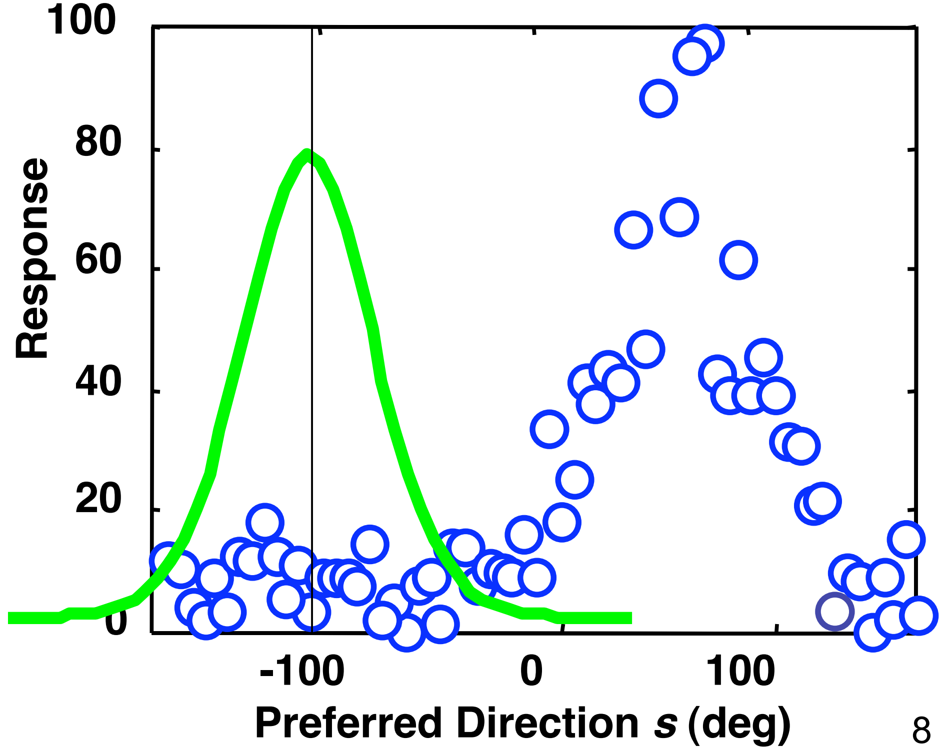
$$p(r_i|s) \;=\; k_i e^{-(r_i-f_i(s))^2/(2v_i)}$$

where $k_i=1/sqrt(2\pi v_i)$

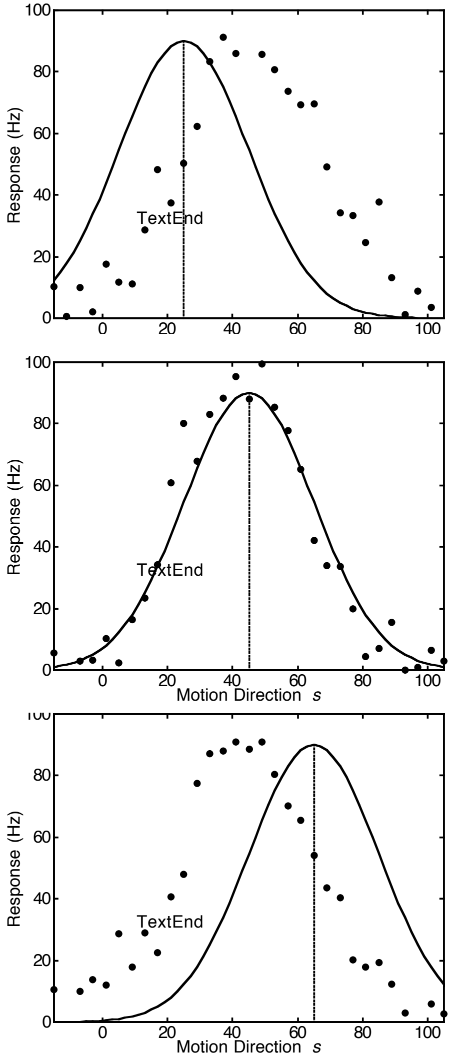
# Likelihoods by eye

For a stimulus with direction *s*, the *response vector* $\mathbf{r}=(r_1, r_2, \ldots, r_{60})$ (circles) of $m=60$ neuronal responses has a ***likelihood*** defined by the conditional pdf $p(\mathbf{r}|s)$.

For example, the probability $p(\mathbf{r}|s)$ of observing the response vector (circles)
given a stimulus with direction $s=-100$ is very low.
(If $s=-100$ then the circles would lie close to the solid tuning curve).



# Likelihoods by eye

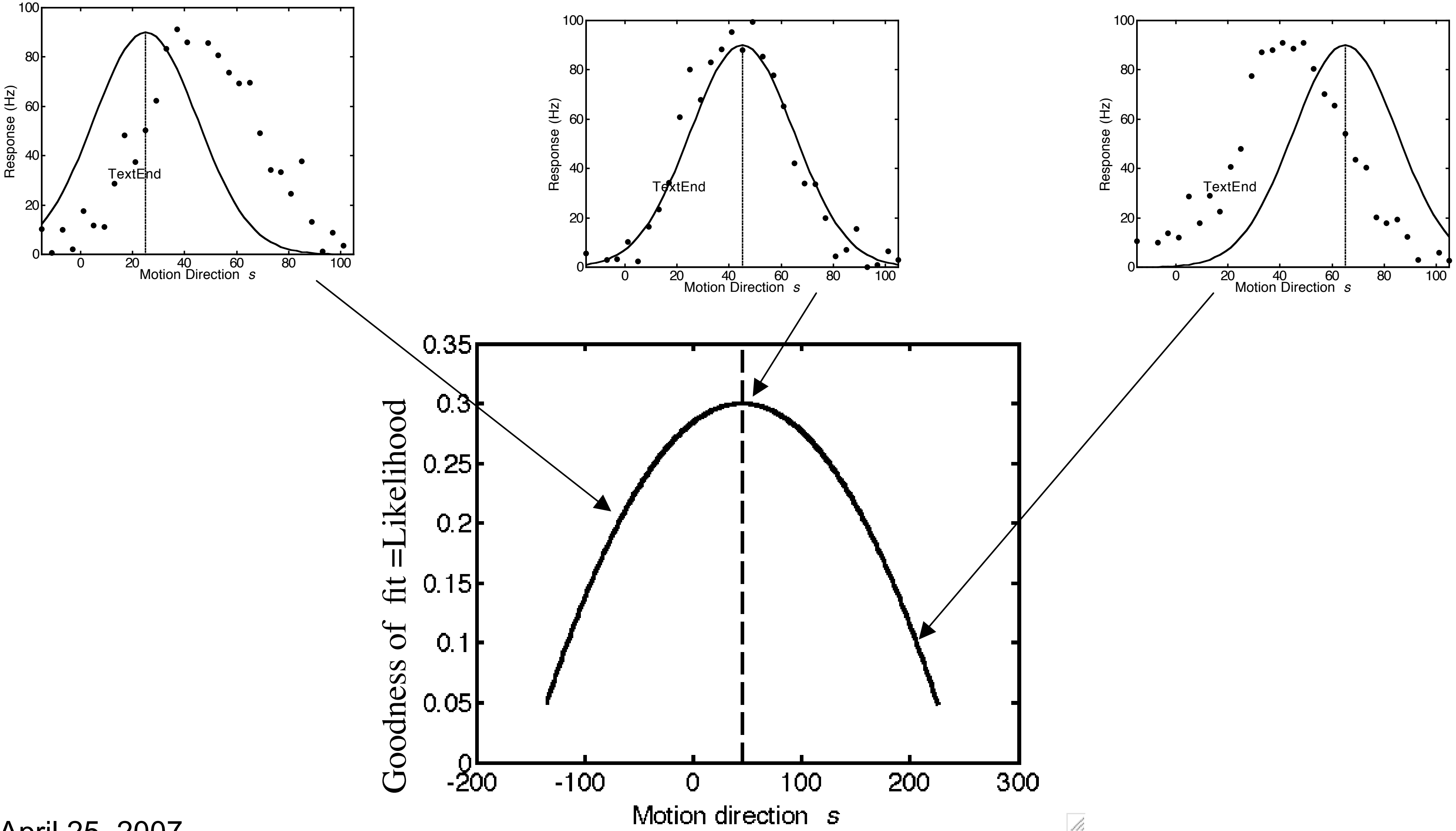



The likelihood $p(\mathbf{r}|s)$ of *observing* the responses $\mathbf{r}=(r_1, r_2, \ldots, r_{60})$ (dots) given $s=25$ is low, ***because if s=25 then responses would be close to solid curve***.

The likelihood $p(\mathbf{r}|s)$ of observing the responses $\mathbf{r}=(r_1, r_2, \ldots, r_{60})$ (dots) given $s=45$ is high, ***because if s=45 then responses would be close to solid curve***.

The likelihood $p(\mathbf{r}|s)$ of observing the responses $\mathbf{r}=(r_1, r_2, \ldots, r_{60})$ (dots) given $s=65$ is low, ***because if s=65 then responses would be close to solid curve***.

# Likelihoods by eye



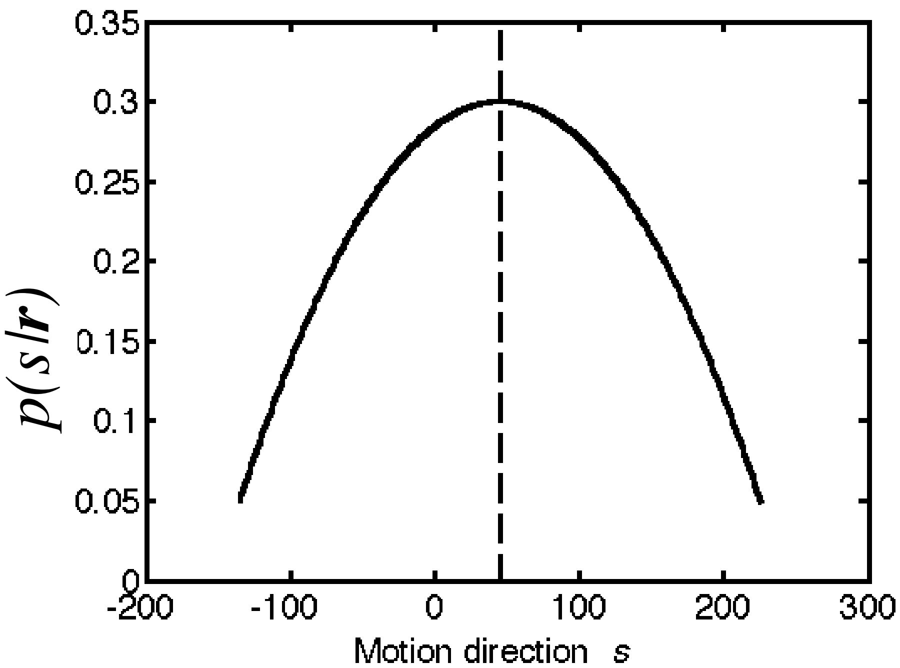
# Interlude: Bayes and MLE

But the quantity we *want* is the ***posterior distribution*** $p(s|\mathbf{r})$.

The quantity we *have* (or will, in a minute) is the ***likelihood*** $p(\mathbf{r}|s)$.

These are related via Bayes' rule:

$$p(s|\mathbf{r}) = p(\mathbf{r}|s)\, p(s) / p(\mathbf{r}).$$

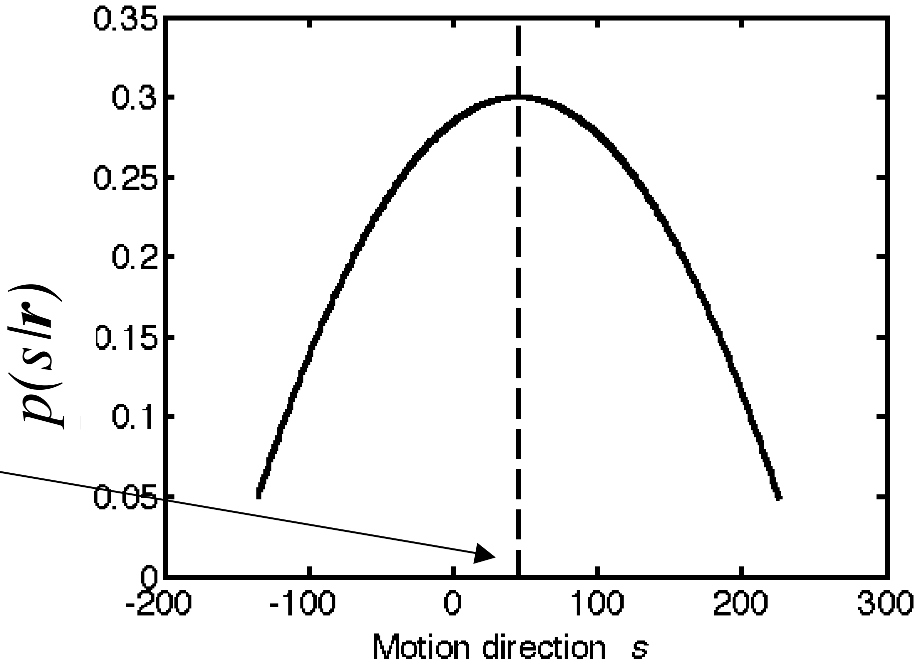


# Why do we want the posterior pdf?

Because the posterior $p(s|\boldsymbol{r})$ tells us the probability that each value of $s$ gave rise to the observed response vector $\boldsymbol{r}$.

The value of $s$ that maximises the posterior $p(s|\boldsymbol{r})$ is the ***maximum a posteriori*** (MAP) estimate $\hat{s}$ of $s^*$.

MAP



## Uniform priors

Bayes' rule:

$$p(s|\boldsymbol{r}) = p(\boldsymbol{r}|s)\, p(s) / p(\boldsymbol{r}).$$

If we assume that all directions $s$ are equally probable then we have a ***uniform prior*** $p(s)$. This means that $p(s)$ is the same for all values of $s$, so we can treat $p(s)$ as a constant (i.e. we can ignore it).

As $\boldsymbol{r}$ is the data we have observed we can treat $p(\boldsymbol{r})$ as a constant (and ignore it too).

## Uniform priors

Given the above, Bayes' rule

$$p(s|\boldsymbol{r}) = p(\boldsymbol{r}|s)\, p(s) / p(\boldsymbol{r})$$

reduces to

$$p(s|\boldsymbol{r}) = \text{constant} \times p(\boldsymbol{r}|s).$$

In other words, the most probable value of $s$ given $\boldsymbol{r}$ is the same as the most probable value of $\boldsymbol{r}$ given $s$ (the likelihood of $s$).

# MAP and MLE

Given that $p(s|\mathbf{r}) = p(\mathbf{r}|s)$, we can find the best estimate $\hat{s}$ of $s^*$ by finding that value of $s$ which maximises the ***likelihood*** $p(\mathbf{r}|s)$.

This value of $s$ is the ***maximum likelihood estimate*** (MLE) of $s^*$.

**Thus, if we have a uniform prior then the MLE and the MAP are the same.**

**End of 'Bayes and MLE' interlude.**

Next, we will find the $s$ value $\hat{s}$ that maximises $p(\mathbf{r}|s)$…

# Evaluating the MLE: The 'product trick'

Crucially, if the noise in each neuron's response is *independent* of the noise in the other cells' responses then we can use a 'product trick' to evaluate the probability of observing $\boldsymbol{r}$

$$p(\boldsymbol{r}|s) = p(r_1|s) \times p(r_2|s) \times \ldots \times p(r_{60}|s)$$

This can be written more succinctly as

$$p(\boldsymbol{r}|s) = \Pi_i \, p(r_i|s),$$

where the capital Greek letter $\Pi$ (pi) indicates that all terms indexed $i$ are multiplied together (e.g. the probabilities of the responses of different neurons).

# Evaluating the MLE: The 'log trick'

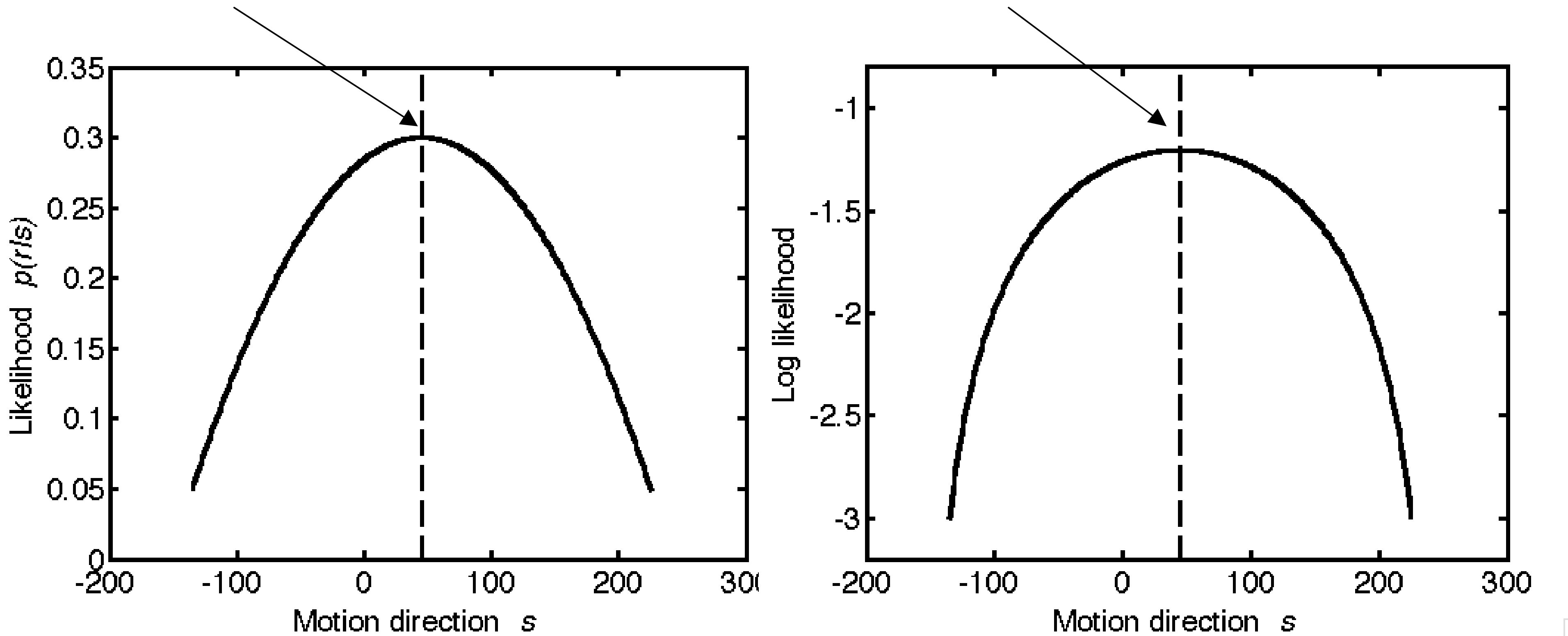
In general, the *log of the product* of a set of numbers is the same as the *sum of the logs* of those numbers. Thus the ***log likelihood*** is

$$\begin{aligned} L(s) &= \ln p(\mathbf{r}|s) \\ &= \ln \prod_i p(r_i|s) \\ &= \sum_i \ln p(r_i|s) \end{aligned}$$

The symbol $\Sigma$ (the greek capital letter sigma) indicates summation of terms indexed by the subscript *i*.

# Evaluating the MLE: The ‘log trick’

We can do this because any change in $s$ that increases $p(\mathbf{r}|s)$ also increases $\ln p(\mathbf{r}|s)$.
Therefore the value $\hat{s}$ of $s$ (-100) that maximises the likelihood $p(\mathbf{r}|s)$ also maximises the log of the likelihood $L(s) = \ln p(\mathbf{r}|s)$.



# MLE and Gaussian noise

- Using Gaussian noise allows us to see that:
- Finding that value of $s$ which maximises the log likelihood function reduces to weighted least-squares.
- Thus MLE = weighted least-squares for Gaussian noise.
- If the noise variances of all response are the same then *weighted* least-squares reduces to least-squares. In this case, MLE = least-squares (LSE).
- Finally, if we also assume a uniform prior then the MAP = least-squares.

## MLE: likelihood function

If we assume Gaussian noise then the likelihood

$$p(\boldsymbol{r}|s) = \Pi_i\, p(r_i|s),$$

can be re-written

$$p(\boldsymbol{r}|s) = \Pi_i\ k_i\, exp(-(f_i(s) - r_i)^2/2v_i).$$

where the subscript identifies each neuron.

# MLE: Log likelihood function

By taking the logs we have the ***log likelihood function***

$$L(s) \quad = ln\ p(\mathbf{r}|s)$$
$$= k_{logn} - \sum_i \ (f_i(s) - r_i)^2/v_i,$$

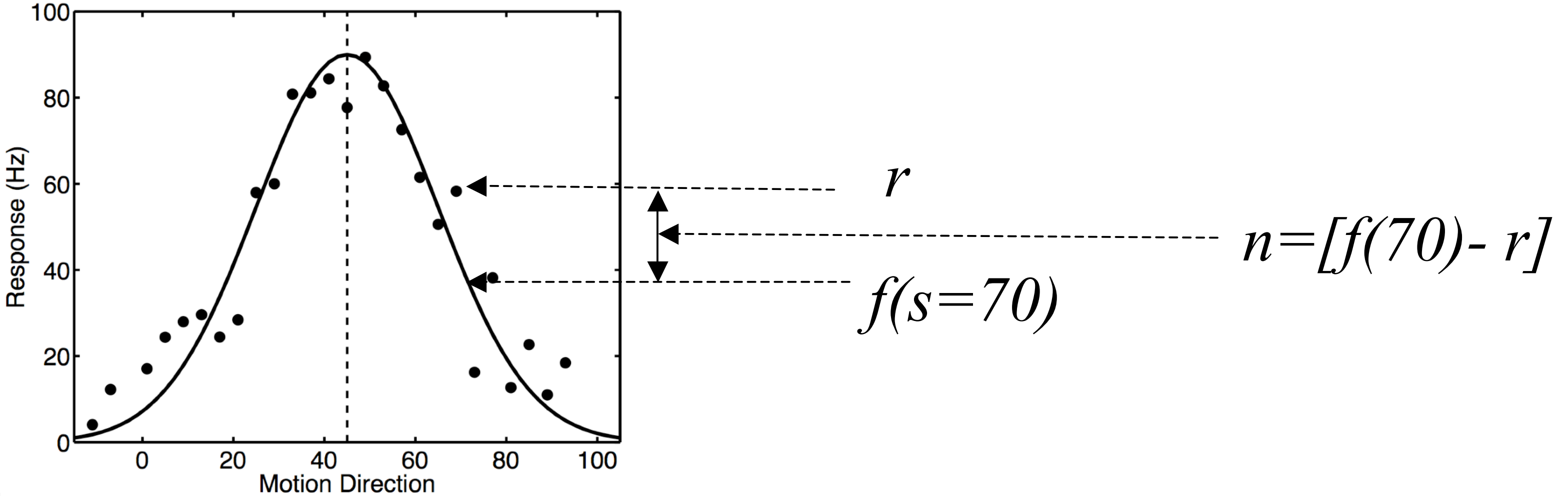
Notice that we have 'wrapped up' all the irrelevant constants into a single constant $k_{logn} = -60\ ln\ k_n$ .

# MLE: Log likelihood function

$$L(s) = k_{logn} - \sum_i \; ( f_i(s) - r_i )^2 / v_i,$$

If we ignore the constants then this equation states that,

“*the log probability that the ith neuron has a (noisy) response* $r_i$ *is equal to … [**minus** the squared distance between* $r_i$ *and* $f_i(s)$ *(the height of the neuron’s tuning curve at s)], where this distance is ‘discounted’ by the variance* $v_i$ *in each neuron’s output*”.



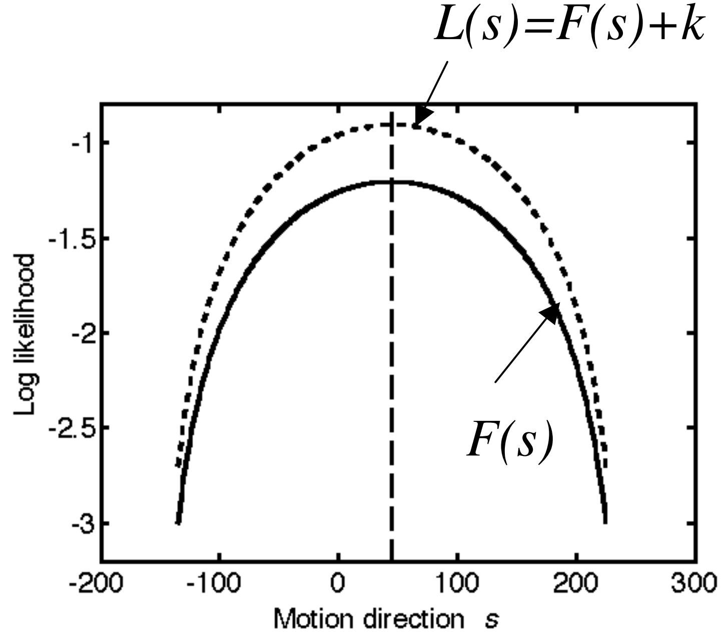
## MLE: Constant disposal

The value $\hat{s}$ of $s$ that maximises

$$L(s) = k_{logn} - \sum_i \left(f_i(s) - r_i\right)^2 / v_i,$$

is unaffected by the value of the constant $k_{logn}$, so $\hat{s}$ also maximises

$$F(s) = -\sum_i \left(f_i(s) - r_i\right)^2 / v_i$$



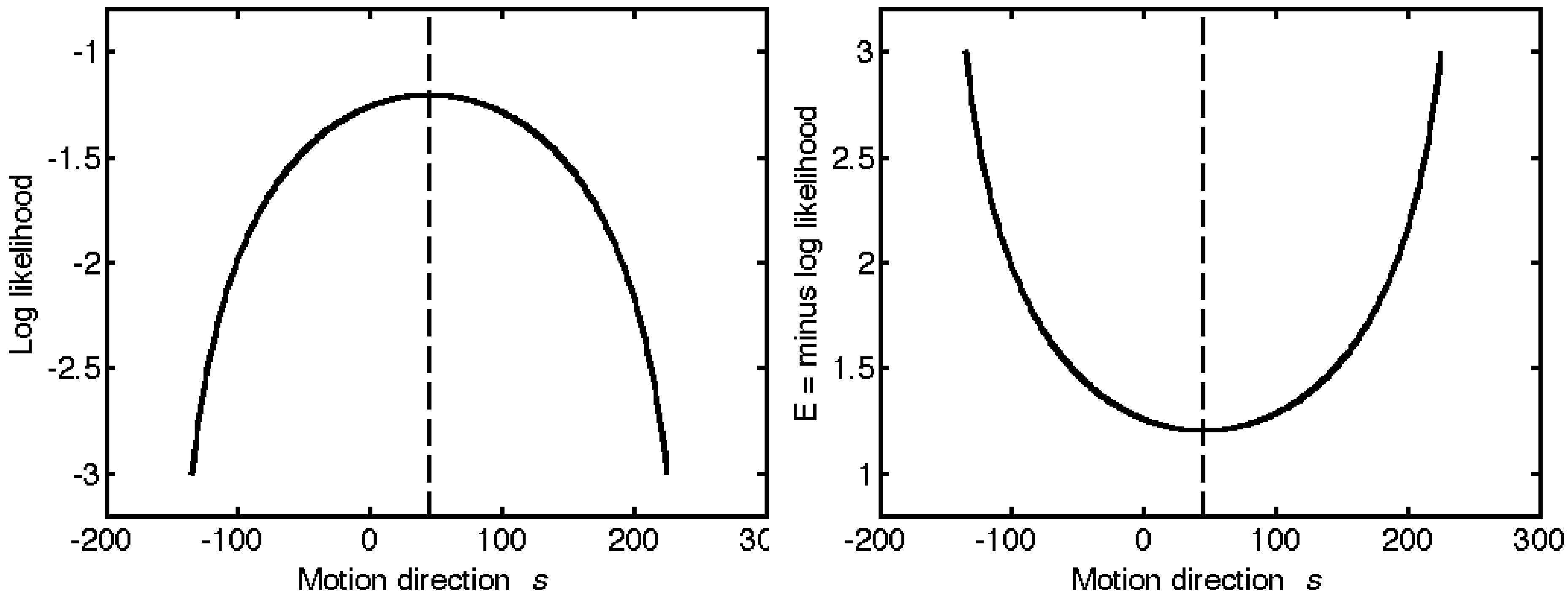
# MLE and least-squares

Rather than finding that value $\hat{s}$ of $s$ that *maximises* (the fit)

$$F(s) = -\sum_i \left( f_i(s) - r_i \right)^2 / v_i,$$

we can find that value of $s$ that *minimises* $-F(s)$ (the *mis-fit*)

$$E(s) = \sum_i \left( f_i(s) - r_i \right)^2 / v_i.$$

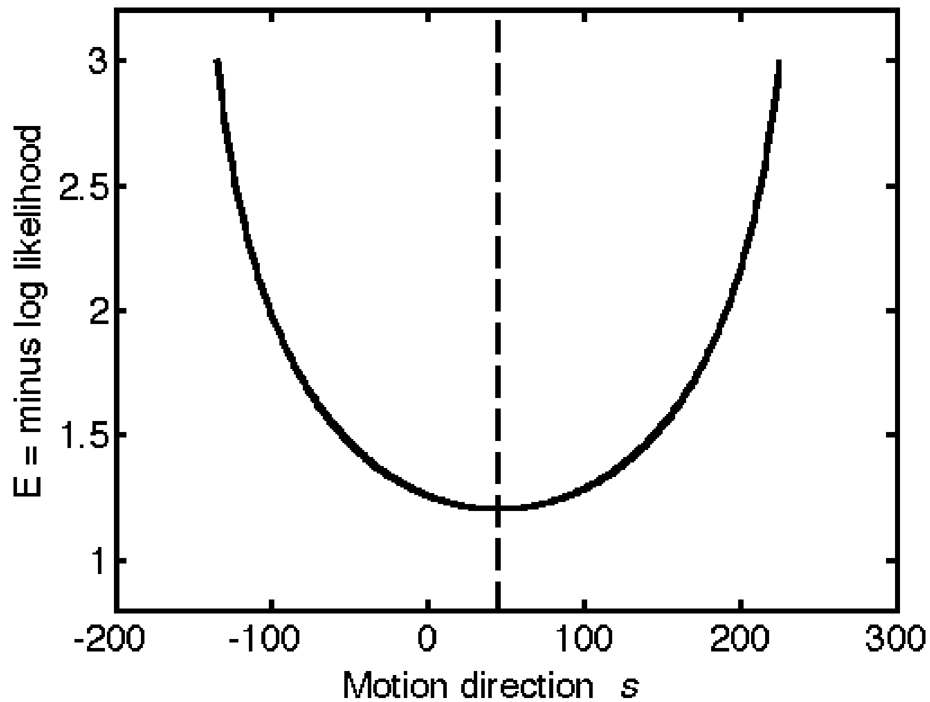


# MLE and least-squares

If all noise variances are equal then $v_i$ is a constant which can be ignored (for the same reason we ignored $k_{logn}$ above)

$$E(s) = \sum_i (f_i(s) - r_i)^2.$$

The value $\hat{s}$ of $s$ that minimises $E(s)$ is the thus ***least-squares estimate*** (LSE) of $s^*$, and is *equivalent* to the MLE if the variance of Gaussian response noise is the same for all neurons.



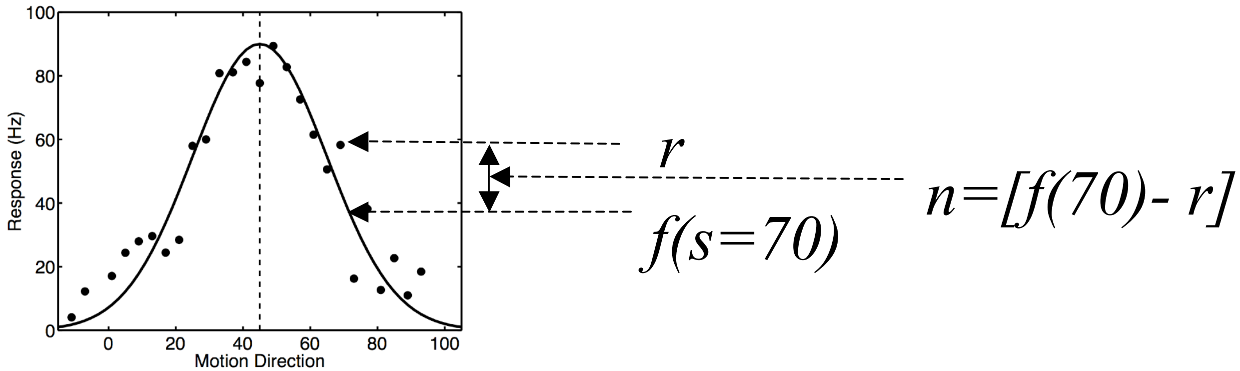
# Least-squares as 'best fit'

In order to find that model tuning function position $s=\hat{s}$ which minimises $E$, we need to evaluate $E$ for all values of $s$.

For each putative value $s$, we must evaluate the square difference for all $m$ responses (one response per cell).

In short, want to find direction $s$ such that the sum of squares is minimised:

$$E(s) = \sum_i^m [\, f_i(s) - r_i \,]^2$$



## Interim summary

- If we assume a uniform prior for $s$ then the MAP reduces to MLE.
- If noise is Gaussian and we ignore all constants, then MLE reduces to weighted least-squares.
- If noise variances of responses are the same then weighted least-squares reduces to least-squares.
- Thus, the least-squares methods used in statistics can be equivalent to MAP under certain conditions.

# MLE with Poisson noise

- Using Poisson noise, the MLE is given by a weighted average of cells' preferred directions, where this weighting is proportional to firing rate of each cell.

- Note that $r$ in previous lecture meant mean (expected) firing rate, and that $r_i$ here means observed (non-average) firing rate. Similarly subtle notation used in D&A, so beware!
- Note, *expected value* is a technical term for *mean*.

# MLE with Poisson noise

If noise is *Poisson* then the likelihood is

$$p(\mathbf{r}|s) \;=\; \prod_i \frac{(f_i(s)T)^{r_iT}}{(r_iT)!} e^{-f_i(s)T}$$

$r_iT=n_i$, is the *observed* number of spikes in time $T$

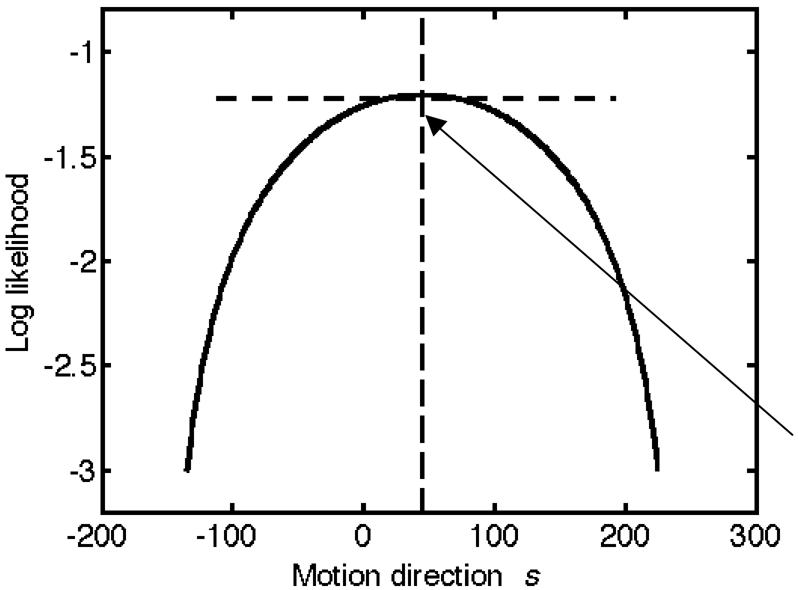
$f_i(s)T = E[n]=\lambda$, the *mean* number of spikes in time $T$

# MLE with Poisson noise

Define $L(s) = \ln p(\mathbf{r}|s)$. At the peak in $L(s)$, where $s = \hat{s}$, the derivative $\partial L(s)/\partial s = 0$.

Ignoring additive constants, in $L(s)$, we obtain

$$\begin{aligned}\partial L(s)/\partial s &\approx \sum_i r_i \frac{\partial \ln f_i(s)}{\partial s} \\ &= \sum_i r_i \frac{f'(s)}{f(s)} \\ &= 0\end{aligned}$$




$\partial L(s)/\partial s = 0$

see p105-6 of D&A

## MLE as a weighted mean

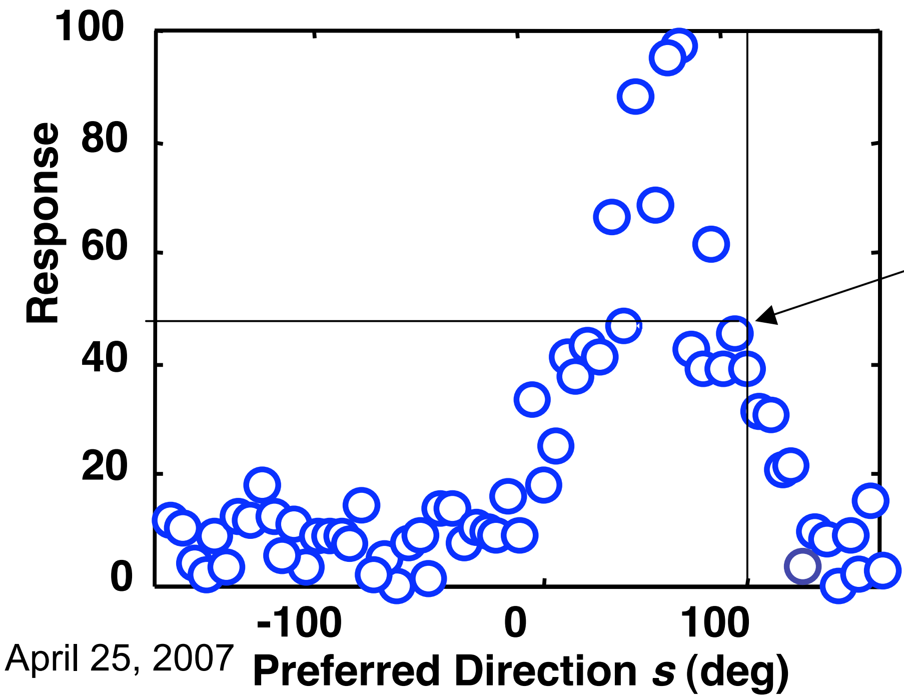
Solving for $s$ in $\partial L(s)/\partial s = 0$ yields $\hat{s}$. If we assume the widths $\sigma_i$ of cell tuning curves are the same then

$$\hat{s} = \frac{\sum r_i s_i}{\sum r_i}$$

Thus the MLE $\hat{s}$ is a ***weighted average*** of the preferred values $s_i$ of a population of cells, where the contribution of each $s_i$ to $\hat{s}$ is proportional to $r_i$.

# MLE as a weighted mean

- It is as if each cell 'votes' for its own preferred value $s_i$. The magnitude of that vote is the cell's relative firing rate $r_i$ (ie normalised by $\sum r_i$).
- Note that the vote $r_i$ for $s_i$ tends to be large if the input value $s^*$ is close to $s_i$.




'Vote' for $s_i$=100 is 50Hz.

$$\hat{s} = \frac{\sum r_i s_i}{\sum r_i}$$

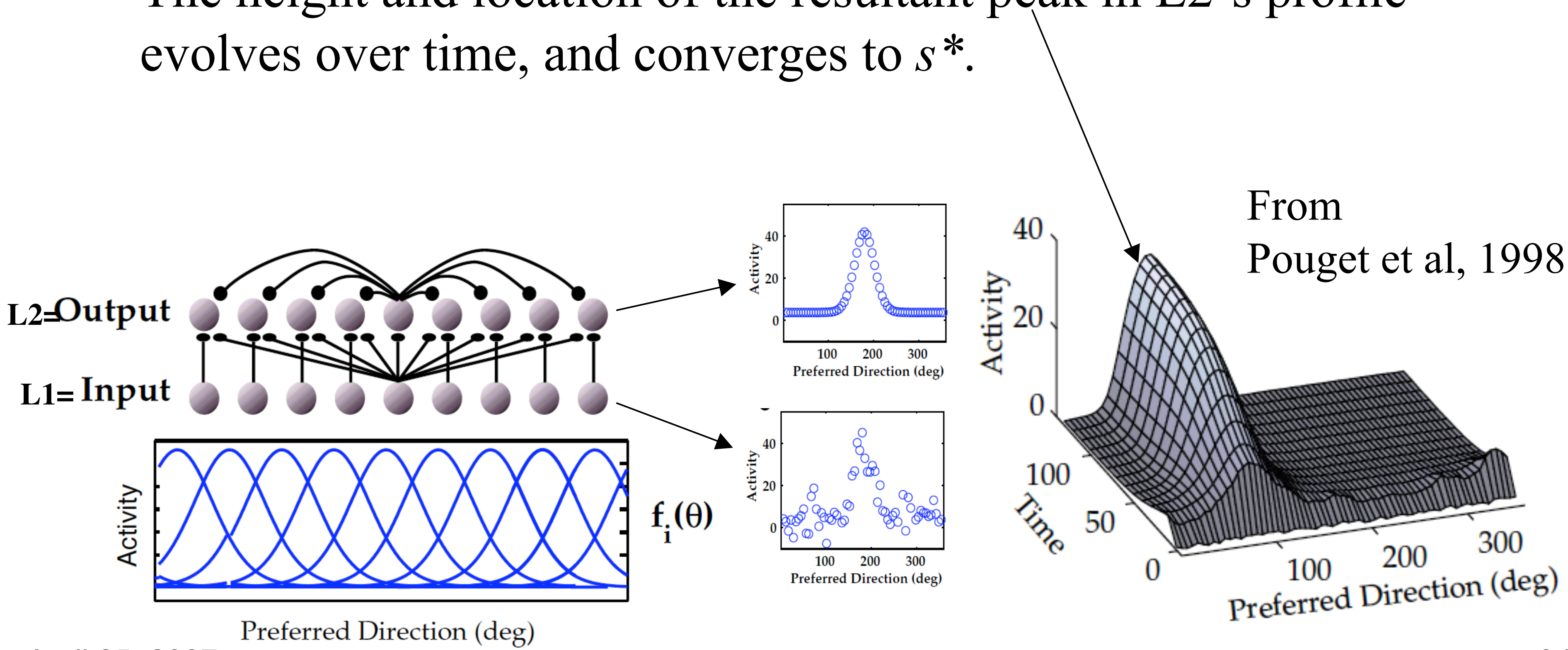
## Interim summary

- A standard method for finding the maximum of a function $L(s)$ is to find an expression for the function's derivative $\partial L(s)/\partial s$, and then estimate that value of $s=\hat{s}$ which makes $\partial L(s)/\partial s=0$.
- Using Poisson noise, the MLE is found to be a weighted mean of preferred values $s_i$, where each cell gets to 'vote' for its own preferred value $s_i$ only. The magnitude of this vote is proportional to the cell's response $r_i$ to the input stimulus $s^*$.
- Cells with preferred values $s_i$ close to the input value $s^*$ have large firing rates, so the preferred values of these cells will tend to dominate the value of $\hat{s}$.

# MLE using Recurrent Networks

- Input cell layer L1 receive same transient stimulus value $s^*$.
- Noisy responses in L1 are 'cleaned up' by output cells in L2, with local excitatory and long-range inhibitory lateral connections.
- The height and location of the resultant peak in L2's profile evolves over time, and converges to $s^*$.

From
Pouget et al, 1998



# Reference

**Essential**

- Frisby and Stone, 2007, chapter 'Seeing Motion', p240-248.
- D&A, p104-6 gives an account of decoding using Poisson noise.
- D&A p258-9 gives a brief account of MLE via recurrent nets, but a more detailed account is in Pouget (below) …

**Background**


- Pouget, A., Zhang, K., Deneve, S. and Latham, P. E. Statistically efficient estimation using population coding, Neural Computation 10, 373–401, 1998.
- Sivia, DS, Data Analysis: A Bayesian Tutorial, 1996. Chapters 2 and 3 give a good account of Bayes and MLE.