Welcome!
#pod-031
Week2, Day1

(Reviewed by Deepak)

neuromatch
academy

facebook
Reality Labs

Penn
University of Pennsylvania

mindCORE
Center for Outreach, Research, and Education

UC Irvine

Queen's University

PennState

University of Minnesota

CIFAR

IEEE brain

SIMONS FOUNDATION

TEMPLETON WORLD
CHARITY FOUNDATION

THE KAVLI FOUNDATION

think : theory
Center for Theoretical Neuroscience

CHEN INSTITUTE
TIANQIAO & CHRISSY

wellcome

GATSBY

Bernstein Network
Computational Neuroscience

NB
DT

hhmi | janelia
Research Campus

Agenda

Developing a Bayesian model for localizing sounds based on audio and visual cues.

This model will combine **prior** information about where sounds generally originate with sensory information about the **likelihood** that a specific sound came from a particular location.

Resulting **posterior distribution** not only allows us to make optimal decision about the sound's origin, but also lets us quantify how uncertain that decision is.

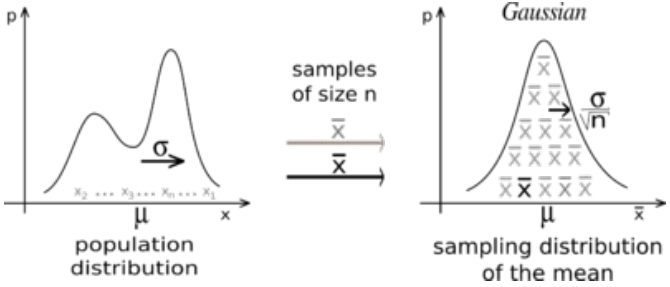
Bayesian techniques are therefore useful **normative models**: the behavior of human or animal subjects can be compared against these models to determine how efficiently they make use of information.

Note: fundamental building blocks for Bayesian statistics: the Gaussian distribution and the Bayes Theorem.

W2D1_pod 031

Central limit theorem

The **central limit theorem** states that if you have a population with mean μ and standard deviation σ and take sufficiently large random samples from the population with replacement , then the distribution of the sample means will be approximately normally distributed.


p
σ
x_2 ... x_3 ... x_n ... x_1
μ
x
population distribution
samples of size n
$\bar{x}$
$\bar{x}$
p
Gaussian
$\frac{\sigma}{\sqrt{n}}$
$\bar{x}$
μ
$\bar{x}$
sampling distribution of the mean

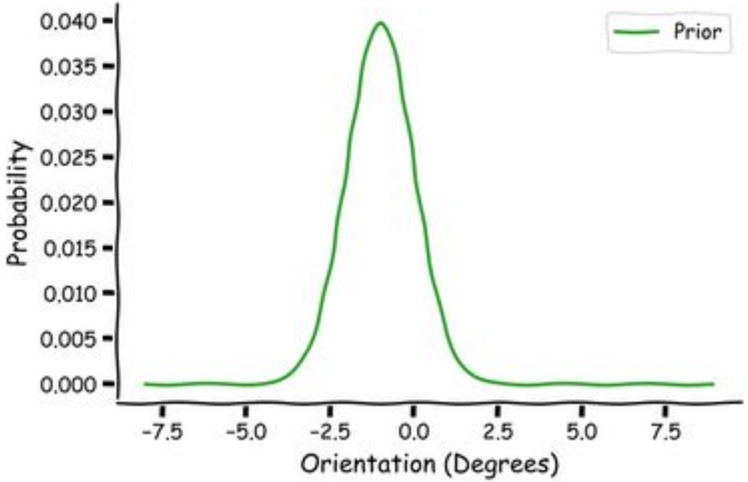
Tutorial #1 Explanations

Gaussian Distribution

Gaussians also have some mathematical/probabilistic (sum=1) properties that permit simple closed-form solutions to several important problems.

Gaussians have two parameters. The **mean** μ, which sets the location of its center. Its "scale" or spread is controlled by its **standard deviation** σ or its square, the **variance** σ^2.

$$\mathcal{N}(\mu, \sigma^2) = \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left(\frac{-(x-\mu)^2}{2\sigma^2}\right)$$



Bayes Rule -

$$\text{posterior} = \frac{\text{likelihood} \times \text{prior}}{\text{normalisation constant}}$$

Converting to guassians -

$$\mathcal{N}(\mu_l, \sigma_l^2) \qquad \mathcal{N}(\mu_p, \sigma_p^2)$$

$$\text{posterior} \propto \mathcal{N}(\mu_l, \sigma_l^2) \times \mathcal{N}(\mu_p, \sigma_p^2)$$

$$= \mathcal{N}\left(\frac{\sigma_l^2 \mu_p + \sigma_p^2 \mu_l}{\sigma_p^2 + \sigma_l^2}, \frac{\sigma_p^2 \sigma_l^2}{\sigma_p^2 + \sigma_l^2}\right)$$

(works for any distribution)

$$\mathcal{N}(\mu, \sigma) = \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left(\frac{-(x-\mu)^2}{2\sigma^2}\right)$$

Closed form :

W2D1_pod 031

Bayes theorem

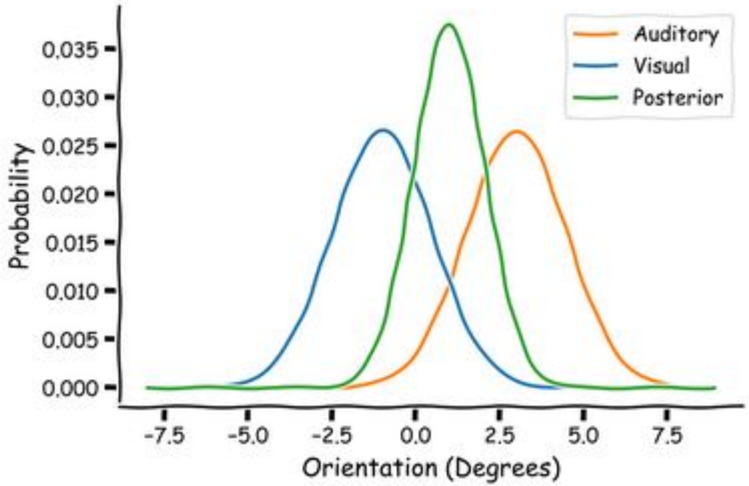
TASK: estimate location of a noise-emitting object. To estimate its position, combine two sources of information:

1. new noisy auditory information (the likelihood)
2. prior visual expectations of where the stimulus is likely to come from (visual prior).

use Gaussian distributions to represent the auditory likelihood (in red), and a Gaussian visual prior (expectations - in blue). Using Bayes rule, combine them into a posterior distribution that summarizes the probability that the object is in each location.

> Bayes' rule tells how to combine two sources of information: the prior (e.g., a noisy representation of our expectations about where the stimulus might come from) and the likelihood (e.g., a noisy representation of the stimulus position on a given trial), to obtain a posterior distribution taking into account both pieces of information.

W1D5_pod 031



prior : Noisy representation of our expectations about where stimulus might come from

likelihood : Noisy representation of stimulus position on a given trial

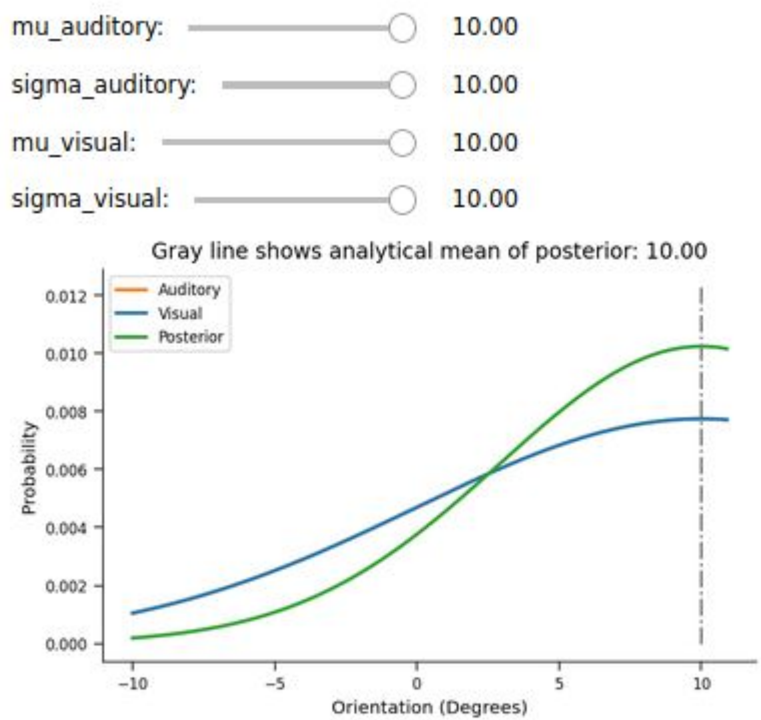
→ new noisy auditory information

visual prior : prior visual expectations of where stimulus is likely to come from!

combine them into posterior distribution that summarises the probability that object is in each location.

Gaussian Parameter variance

Vary the parameters of Gaussians to see how changing the prior and likelihood affect the posterior.



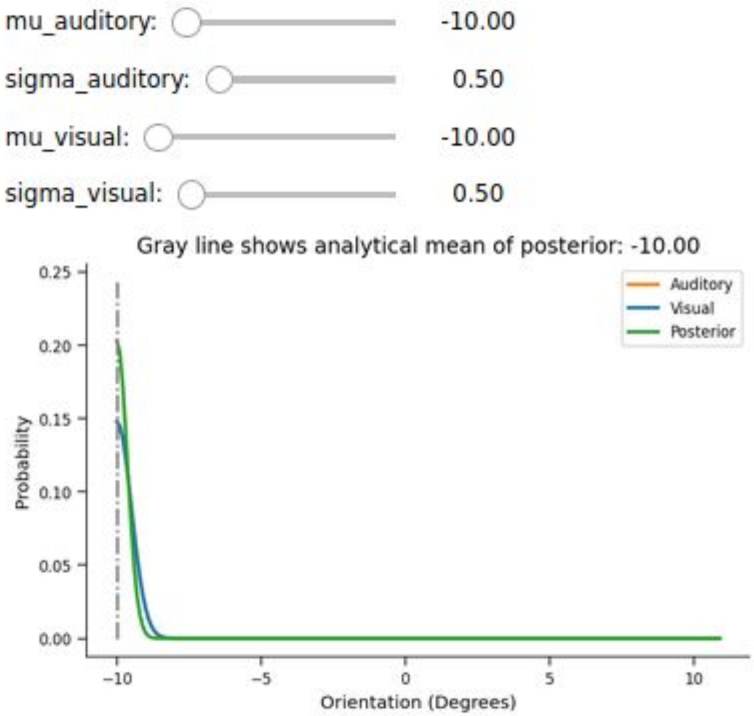
mu_auditory: 10.00

sigma_auditory: 10.00

mu_visual: 10.00

sigma_visual: 10.00

Gray line shows analytical mean of posterior: 10.00



mu_auditory: -10.00

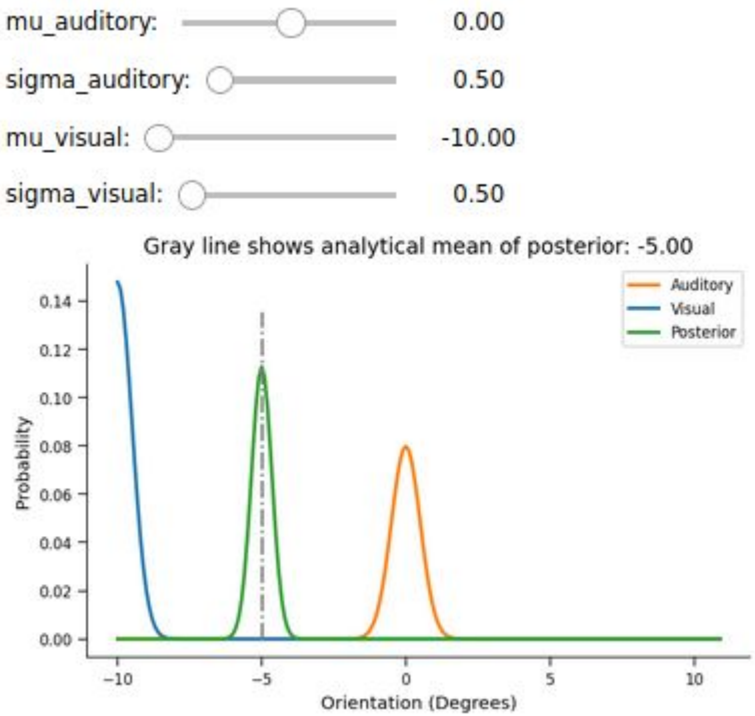
sigma_auditory: 0.50

mu_visual: -10.00

sigma_visual: 0.50

Gray line shows analytical mean of posterior: -10.00

mu_auditory



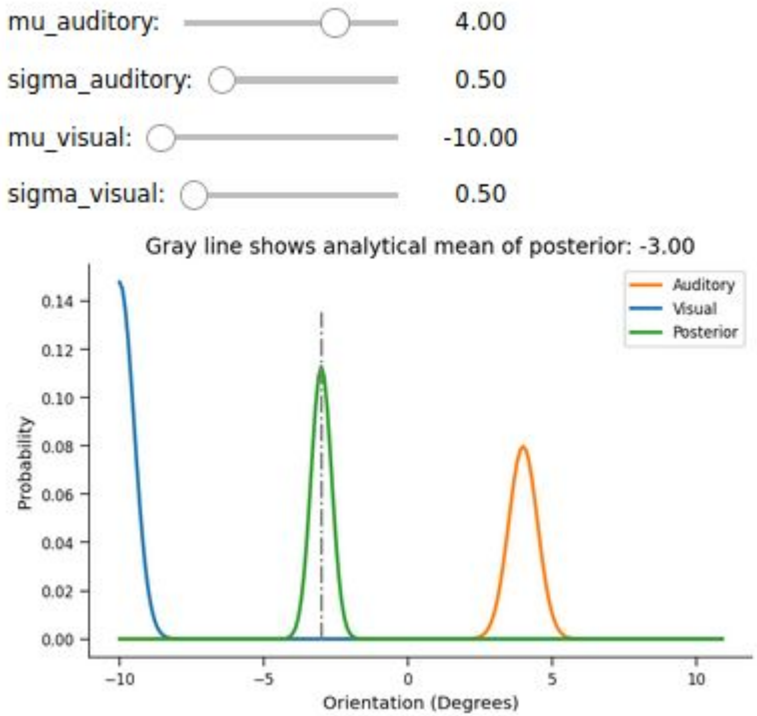
mu_auditory: 0.00

sigma_auditory: 0.50

mu_visual: -10.00

sigma_visual: 0.50

Gray line shows analytical mean of posterior: -5.00



mu_auditory: 4.00

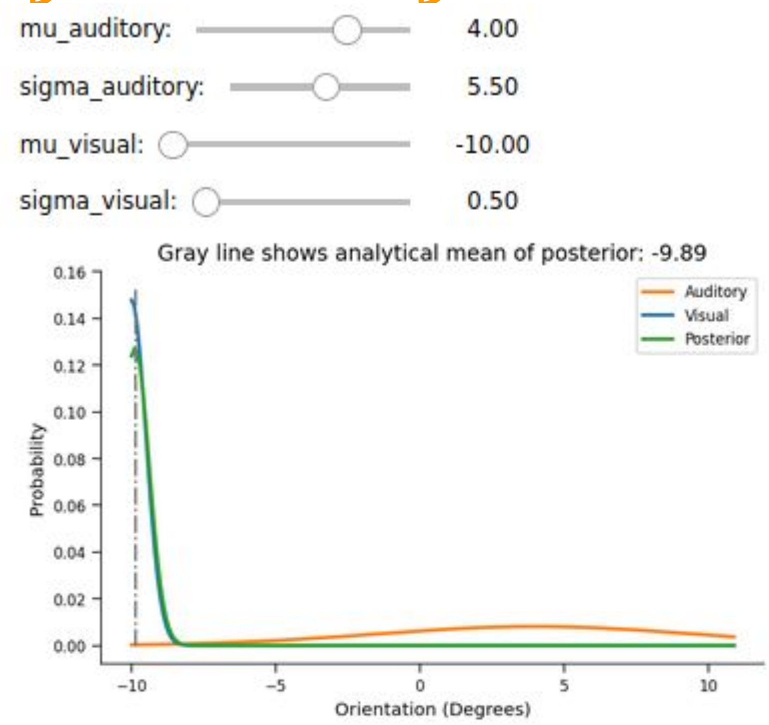
sigma_auditory: 0.50

mu_visual: -10.00

sigma_visual: 0.50

Gray line shows analytical mean of posterior: -3.00

sigma_auditory



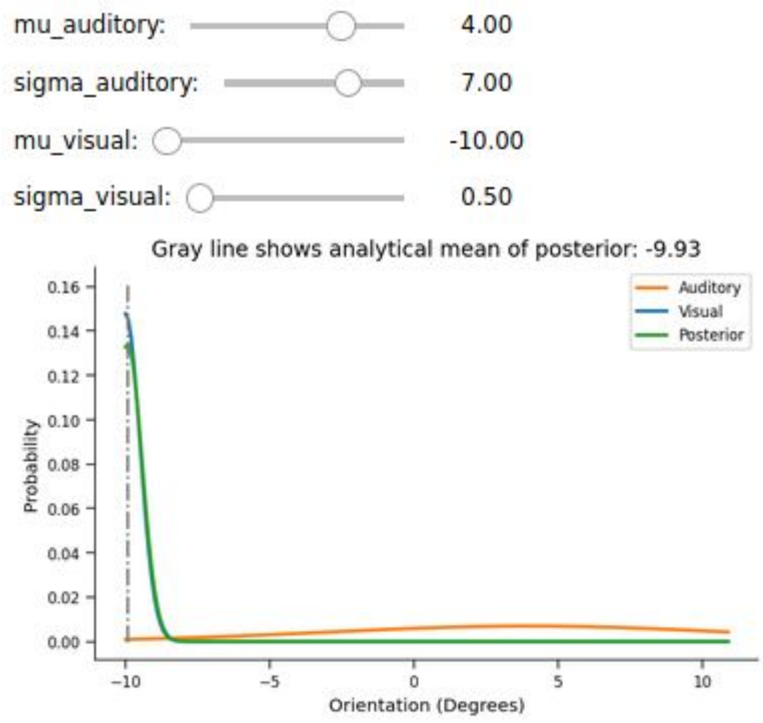
mu_auditory: 4.00

sigma_auditory: 5.50

mu_visual: -10.00

sigma_visual: 0.50

Gray line shows analytical mean of posterior: -9.89



mu_auditory: 4.00

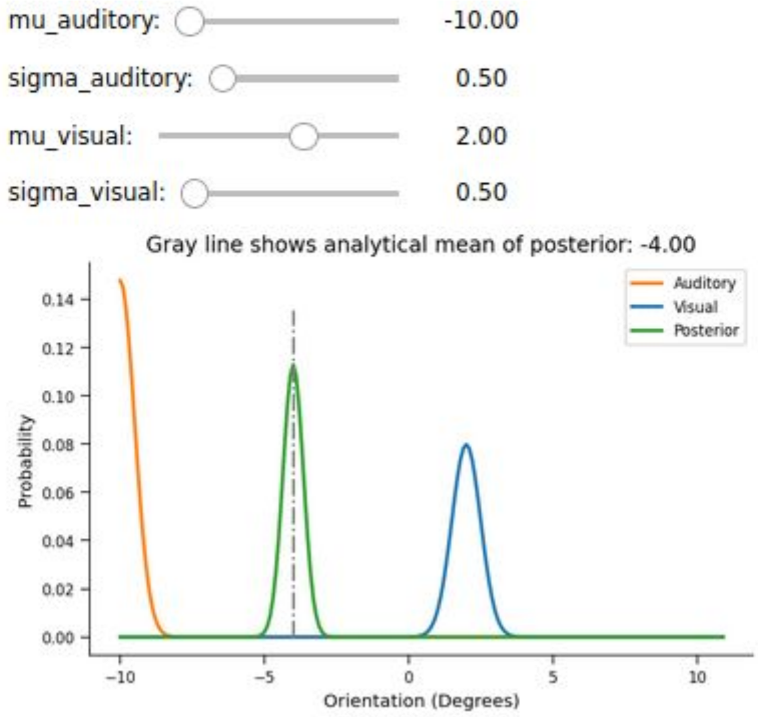
sigma_auditory: 7.00

mu_visual: -10.00

sigma_visual: 0.50

Gray line shows analytical mean of posterior: -9.93

mu_visual



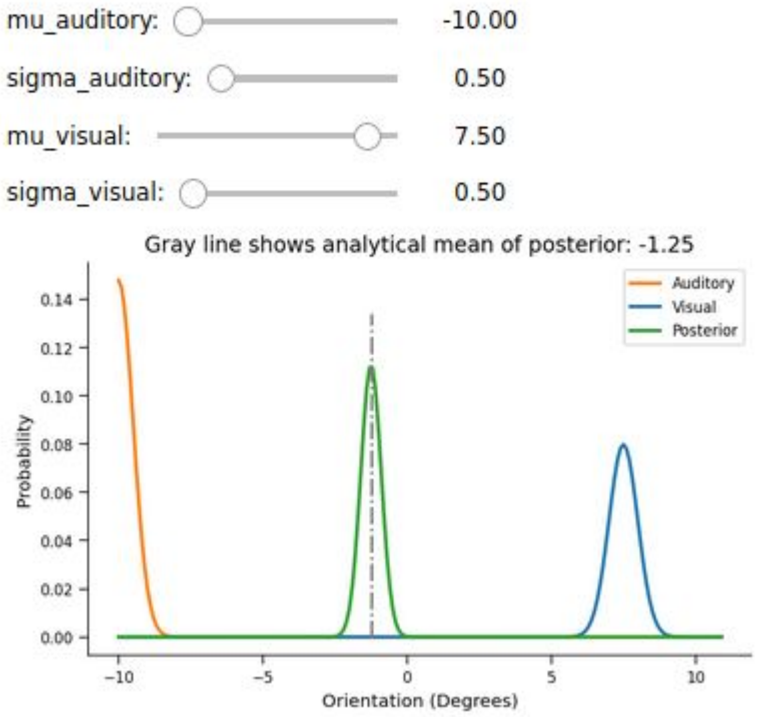
mu_auditory: -10.00

sigma_auditory: 0.50

mu_visual: 2.00

sigma_visual: 0.50

Gray line shows analytical mean of posterior: -4.00



mu_auditory: -10.00

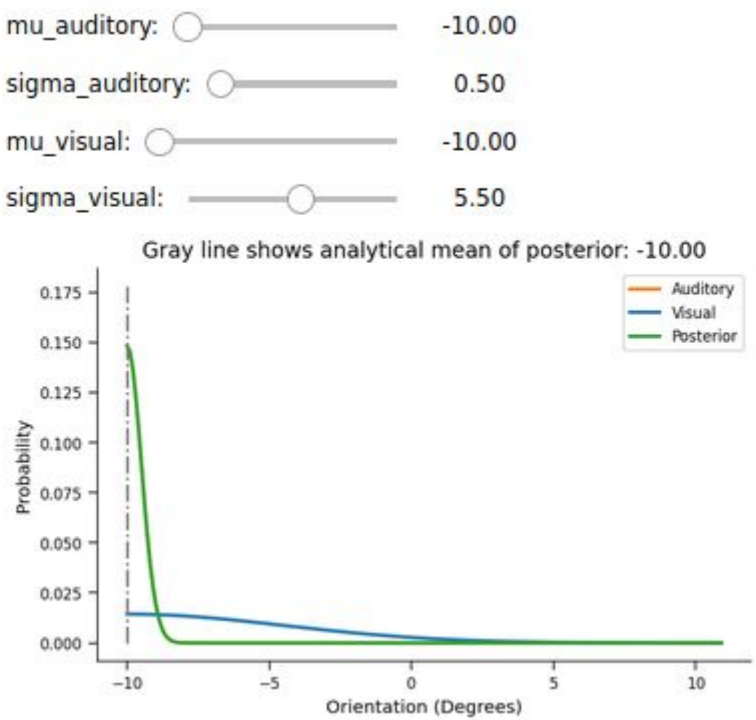
sigma_auditory: 0.50

mu_visual: 7.50

sigma_visual: 0.50

Gray line shows analytical mean of posterior: -1.25

sigma_visual



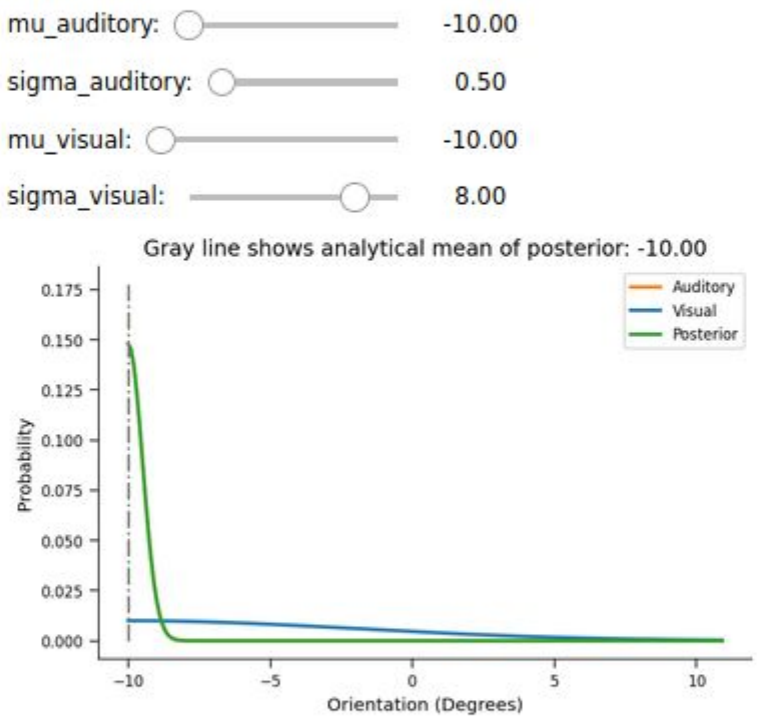
mu_auditory: -10.00

sigma_auditory: 0.50

mu_visual: -10.00

sigma_visual: 5.50

Gray line shows analytical mean of posterior: -10.00



mu_auditory: -10.00

sigma_auditory: 0.50

mu_visual: -10.00

sigma_visual: 8.00

Gray line shows analytical mean of posterior: -10.00

Gaussians

The product of two Gaussian distributions, like our prior and likelihood, remains a Gaussian, regardless of the parameters. We can directly compute the parameters of that Gaussian from the means and variances of the prior and likelihood.

When does the prior have the strongest influence over the posterior? When is it the weakest?

Mu-values have the strongest influence over posterior. When combined with sigma (lower mu contributions) or when sigma is largest, prior seems to have a lesser influence over posterior.

Conjugate-ness

Conjugate distributions or *conjugate priors* (for a particular likelihood) hold the following properties:

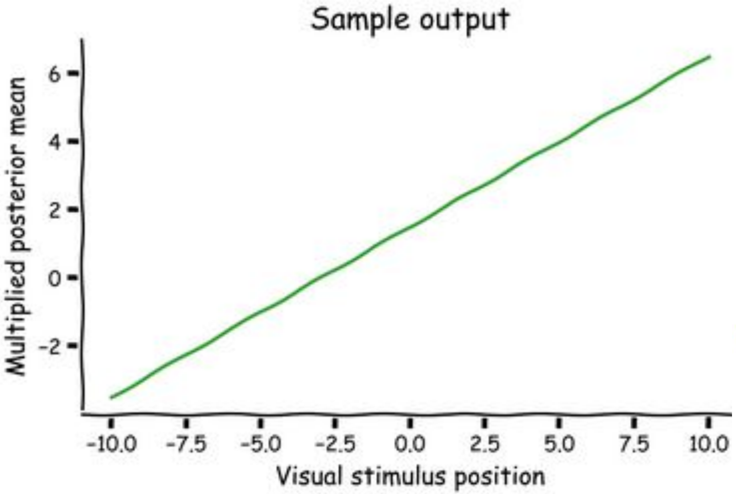
- The posterior has the same form (here, a normal distribution) as the prior, and
- There is simple, closed-form expression for its parameters.

Working with conjugate distributions is very convenient; otherwise, it is often necessary to use computationally-intensive numerical methods to combine the prior and likelihood.

product of gaussians → gaussian.

Compute parameters of that gaussian from means/ variances of that prior & likelihood!

$$\mu_p = \frac{\mu_{aud} \cdot \frac{1}{\sigma^2_{aud}} + \mu_{visual} \frac{1}{\sigma^2_{visual}}}{1/\sigma^2_{aud} + 1/\sigma^2_{visual}}$$



Verifying conjugate properties.

① auditory likelihood constant

② compute posterior distribution

→ find mean

$\int_x p(x)\,dx$ or $\sum_x x \cdot p(x)$.

③ Compute analytical posterior mean (from auditory & visual)

④ plot mean estimates

Tutorial #1 Bonus Explanations

Multimodal Priors

Gaussian prior: Stimulus is expected to come from a single location, though they might not know precisely where.

Multimodal priors: sound might come from one of two distinct locations.

We could model this using a Gaussian prior with a large σ that covers both locations, but that would also make every point in between seem likely too. A better approach is to adjust the form of the prior so that it better matches the experiences/expectations by building a bimodal (2-peaked) prior out of Gaussians and examine the resulting posterior and its peaks.

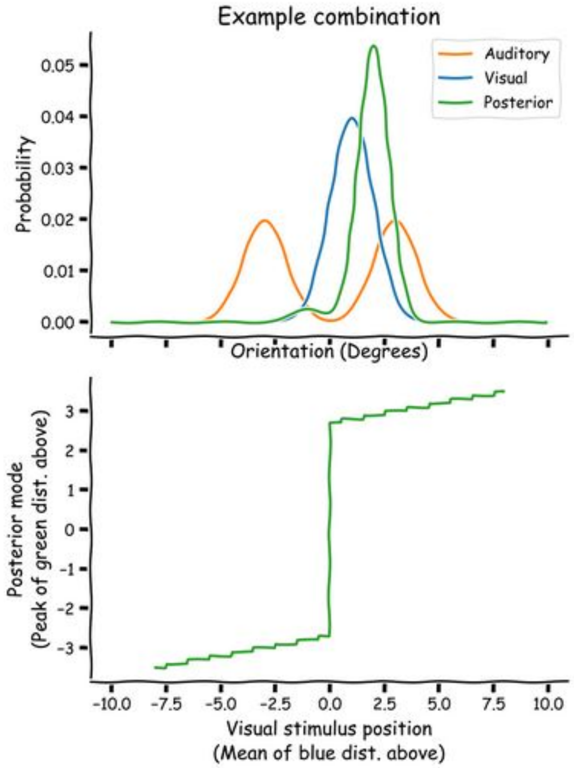
Note: normalize the result so it is a proper probability distribution.

Implementation and test!

Previous implementations do not help us answer questions like: What is the mean of our new prior? Is it a particularly likely location for the stimulus? Instead, we will use the posterior **mode** to summarize the distribution. The mode is the location of the most probable part of the distribution.

Observe what happens to the posterior as the likelihood gets closer to the different peaks of the prior.

Notice what happens to the posterior when the likelihood is exactly in between the two modes of the prior.



Tutorial #2
Explanations

W2D1_pod 031

Experimentation methodology

Study the effect of change as the distance between the visual stimulus (and the auditory stimulus increases/decreases

present only the auditory stimulus at varying locations and report where the source of the sound is located.using two pieces of information:

- The prior information about sound localization, learned during the trials before the curtain fell.
- Their noisy sensory estimates about where a particular sound originates.

The eventual goal is to predict the subjects' responses which implicitly requires building a prior that captures knowledge and expectations;

Mixture of Gaussian Priors

single Gaussian prior -> could represent one of these possibilities.

A broad Gaussian with a large σ -> could represent sounds originating from nearly anywhere,

while a narrow Gaussian with μ near zero -> could represent sounds originating from the puppet.

Combine those into a mixture-of-Gaussians probability density function (PDF) that captures both possibilities

control Gaussian mixtures by summing them together with a 'mixing' or weight parameter pcommon, set to a value between 0 and 1

W2D1_pod 031

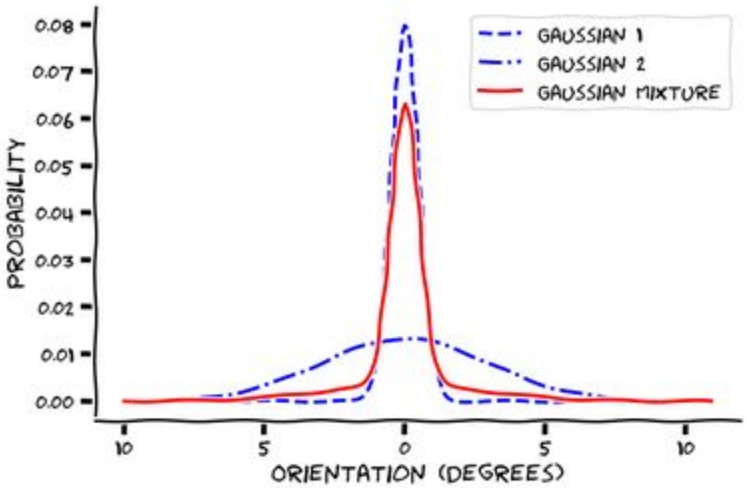
$$\text{Mixture} = p_c \times \overset{\text{(common)}}{\mathcal{N}(\mu_c, \sigma_c)} + \left[(1 - p_c) \times \overset{\text{independent}}{\mathcal{N}(\mu_i, \sigma_i)} \right]$$

probability that auditory stimulus shares a common source with the latent visual input

Sounds independent of the first source

Note:

$$\left. p_c + p_i = 1 \right\} \text{law of total probability}$$



gaussian_1 ≡ gaussian_common($\mu = 0, \sigma = 0.5$)

gaussian_2 ≡ gaussian_independent($\mu = 0, \sigma = 3$)

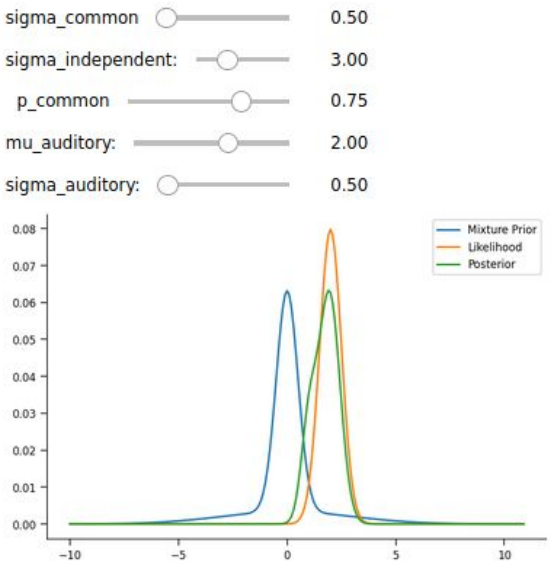
Combine ① & ② to make new prior by mixing }
with mixing parameter $p_c = 0.75$

(peakier common cause gaussian
with 75% of the weight)

Bayes theorem with Complex posteriors

We will compute the posterior by using Bayes Theorem to combine the mixture-of-gaussians prior and varying auditory Gaussian likelihood.

Explore how a mixture-of-Gaussians prior and Gaussian likelihood interact



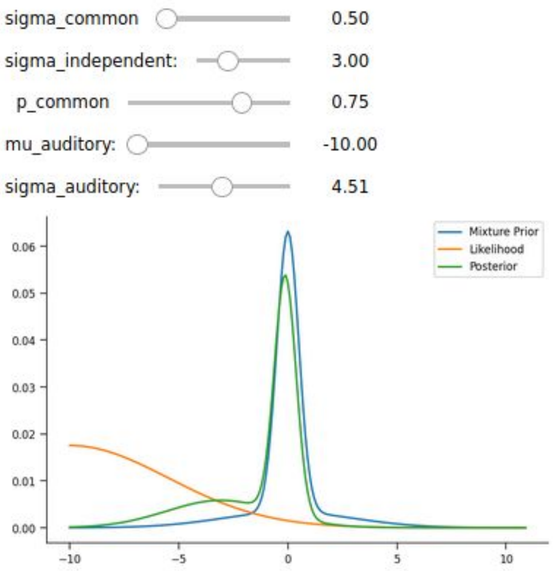
sigma_common 0.50

sigma_independent: 3.00

p_common 0.75

mu_auditory: 2.00

sigma_auditory: 0.50



sigma_common 0.50

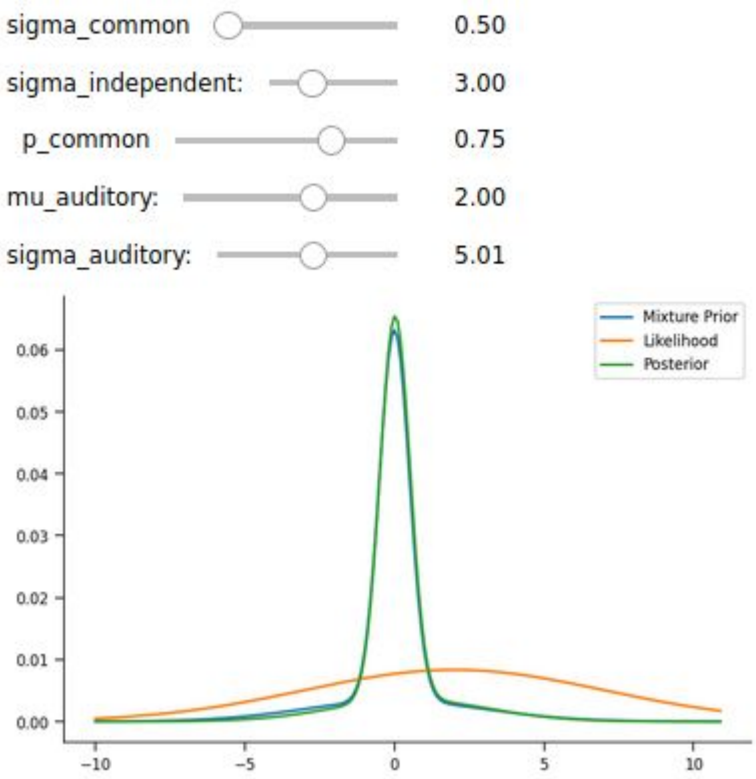
sigma_independent: 3.00

p_common 0.75

mu_auditory: -10.00

sigma_auditory: 4.51

W2D1_pod 031



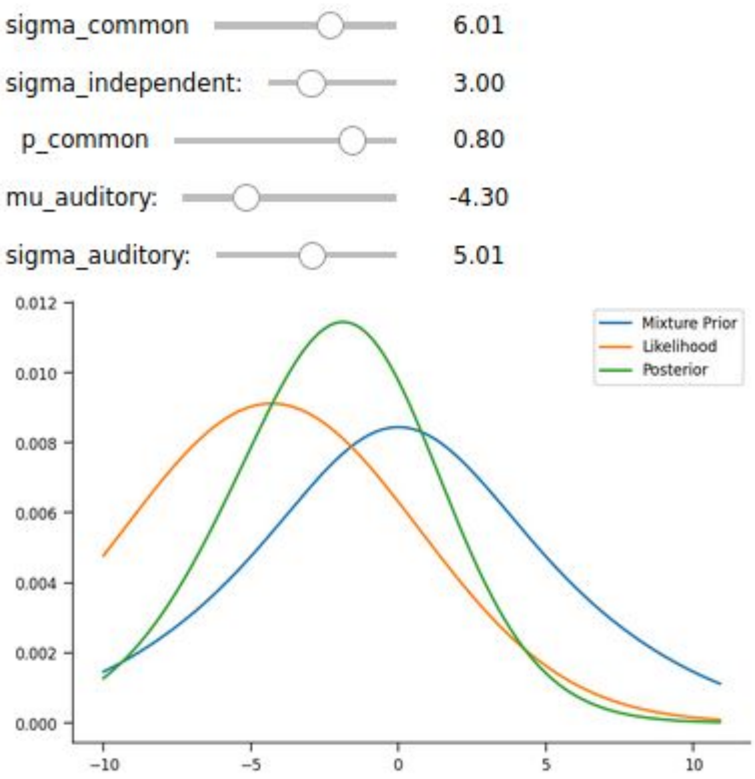
sigma_common 0.50

sigma_independent: 3.00

p_common 0.75

mu_auditory: 2.00

sigma_auditory: 5.01



sigma_common 6.01

sigma_independent: 3.00

p_common 0.80

mu_auditory: -4.30

sigma_auditory: 5.01

W2D1_pod 031

Gaussian behavior Analysis

The mixture of Gaussian prior creates some interesting behaviour:

1. We observe multiple modes (i.e. peaks) in our posterior (the common and independent causes).
2. The mode of the posterior jumps between stimulus locations. These correspond to the participant switching from the independent to the common parts (i.e. causes) of the prior.

A similar discontinuity (ie. 'jump') in the posterior mode would happen in the case of cue combination illusion with both auditory sources.
The same-source illusion breaks-down when the voice stimulus is presented too far away from the visual input.

Tutorial #3
Explanations

Gaussian behavior Analysis

compute all the necessary steps to perform model inversion (estimate the model parameters such as pcommon that generate data based on a mixture of Gaussian prior (common + independent priors) and a Gaussian likelihood similar to that of a participant).

for multiple possible stimulus inputs:

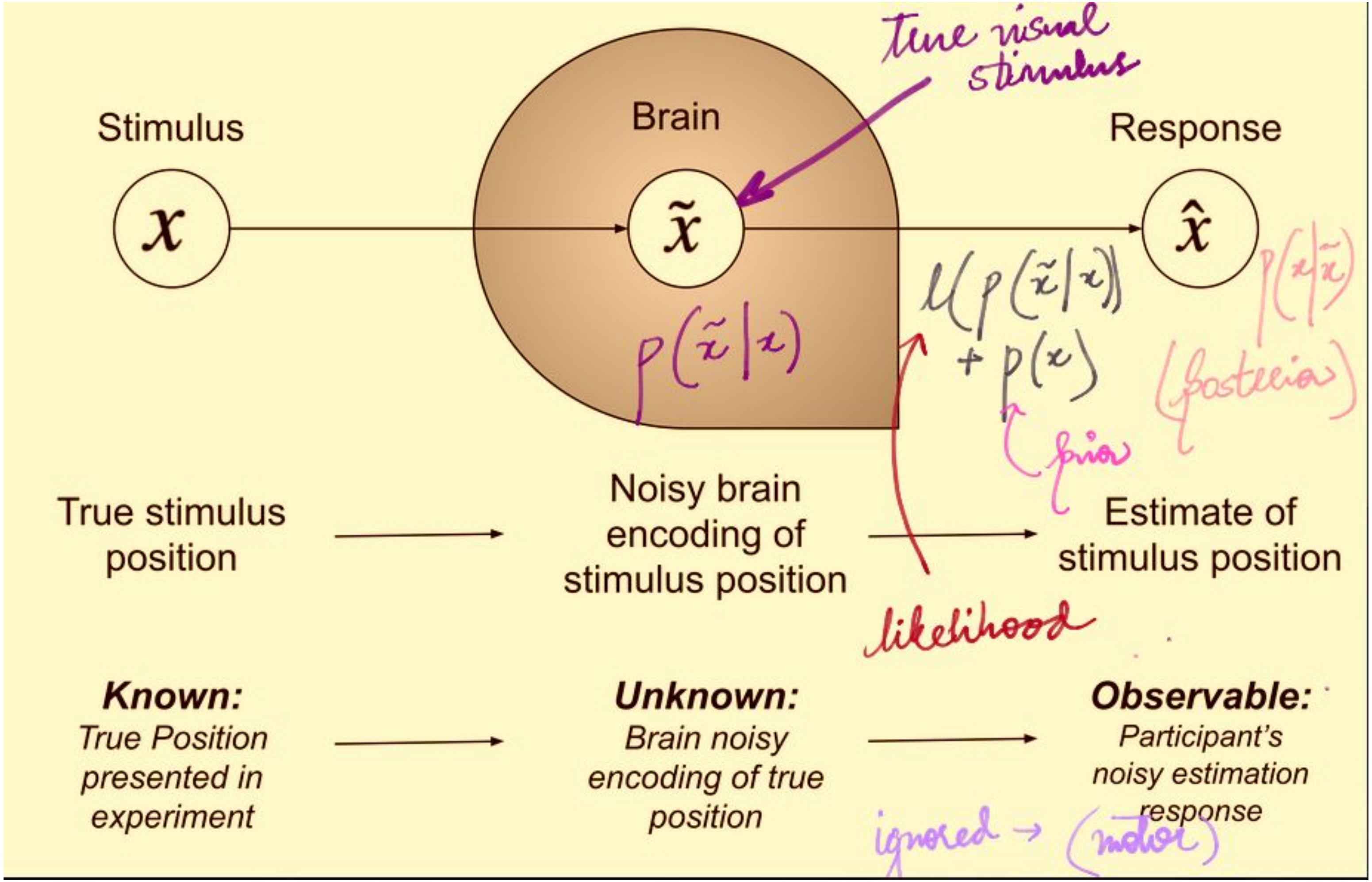
generative model

1. create mixture of gaussian prior
2. generate likelihood
3. estimate posterior as a f(stimulus_input)
4. estimate participant response given posterior.

model fitting/ inversion!

5. Create distribution for input as f(possible inputs)
6. marginalisation
7. generate data using model (steps 1-4)
8. model inversion / model fitting using generated data } to check for recovery of original parameters.

W2D1_pod 031



True stimulus position → Noisy brain encoding of stimulus position → Estimate of stimulus position

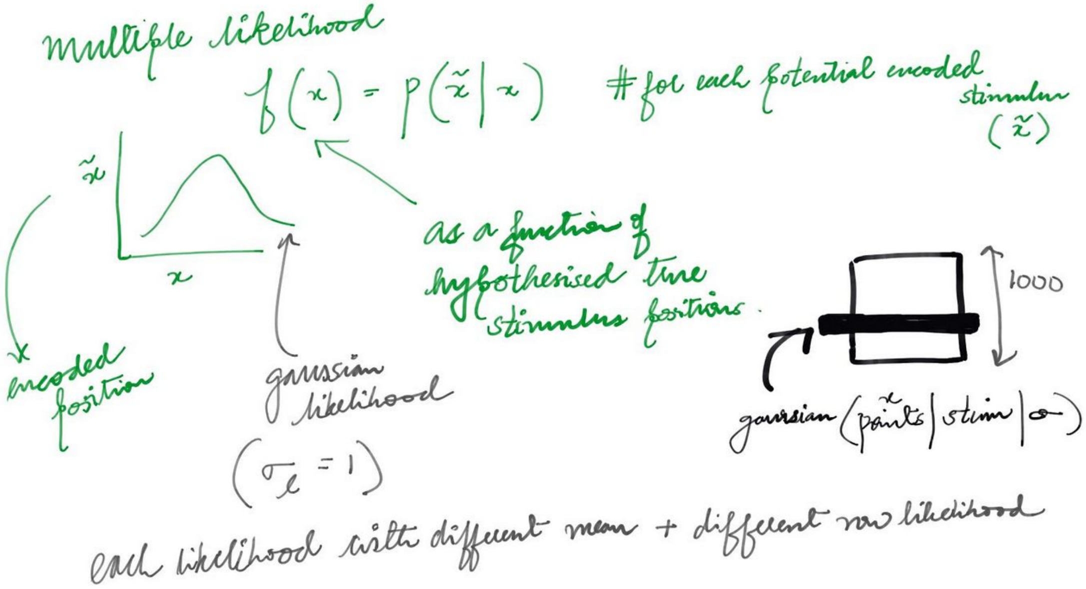
Known: *True Position presented in experiment* → ***Unknown:*** *Brain noisy encoding of true position* → ***Observable:*** *Participant's noisy estimation response*

ignored → (motor)

W2D1_pod 031

Likelihood array

- *to consider all possible brain encodings*



multiple likelihood

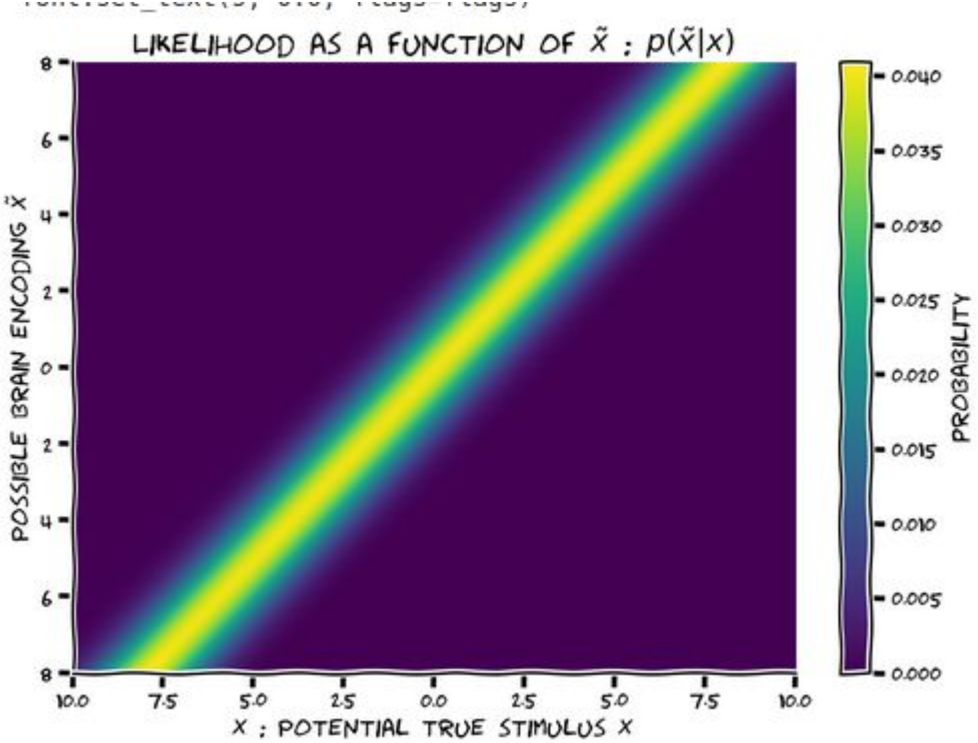
$f(x) = p(\tilde{x} \mid x)$ # for each potential encoded stimulus $(\tilde{x})$

as a function of hypothesised true stimulus positions.

gaussian(points^{x} | stim | σ)

each likelihood with different mean + different row likelihood

W2D1_pod 031


LIKELIHOOD AS A FUNCTION OF $\tilde{x}$: $p(\tilde{x}|x)$
POSSIBLE BRAIN ENCODING $\tilde{x}$
8
6
4
2
0
2
4
6
8
10.0
7.5
5.0
2.5
0.0
2.5
5.0
7.5
10.0
X : POTENTIAL TRUE STIMULUS X
0.040
0.035
0.030
0.025
0.020
0.015
0.010
0.005
0.000
PROBABILITY

Causal mixture of gaussian prior

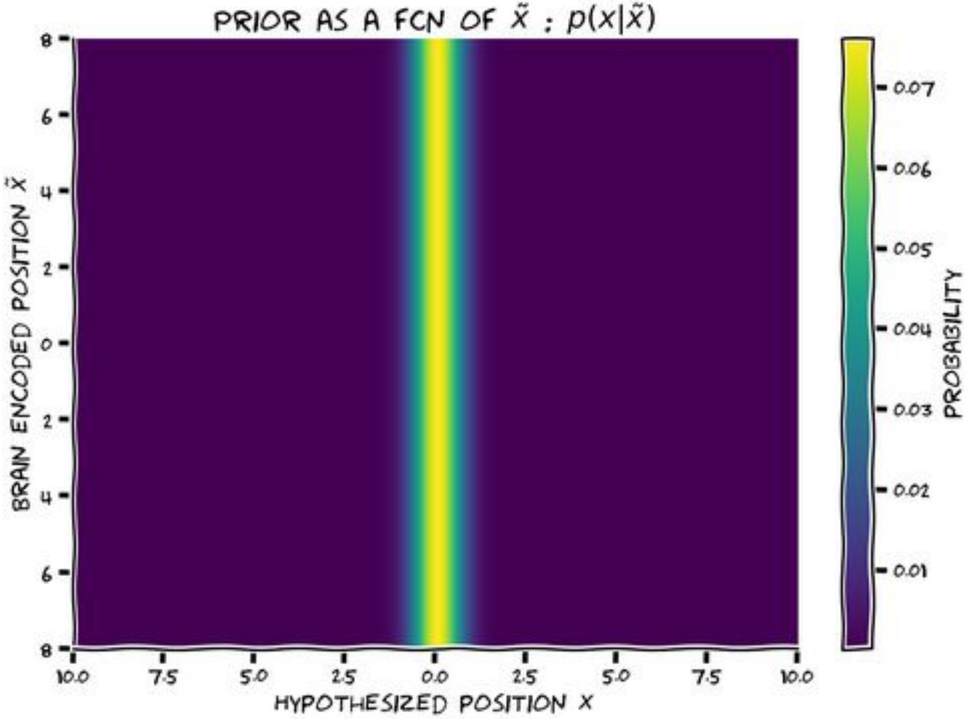
Create mixture of gaussian prior as f(brain encoded stimulus $\tilde{x}$)

(prior doesn't change as $f(\tilde{x}) \Rightarrow$ identical $\forall$ row)

① create gaussian_c with $\mu = 0, \sigma = 0.5$.

② create gaussian_i with $\mu = 0, \sigma = 10$.

③ combine ① & ② to make new prior (mixing parameter $p_i = 0.05$) } peakier gaussian has 95% of the weight


PRIOR AS A FCN OF $\tilde{x}$: $p(x|\tilde{x})$
BRAIN ENCODED POSITION $\tilde{x}$
8
6
4
2
0
2
4
6
8
10.0
7.5
5.0
2.5
0.0
2.5
5.0
7.5
10.0
HYPOTHESIZED POSITION X
PROBABILITY
0.07
0.06
0.05
0.04
0.03
0.02
0.01

Bayes rule and posterior array

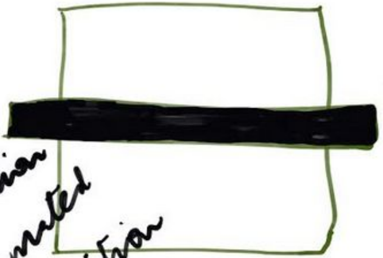
HADAMARD PRODUCT

NumPy operations can often process an entire matrix in a single "vectorized" operation. This approach is often much faster and much easier to read than an element-by-element calculation.

$$\text{posterior}[i,:] \propto \text{likelihood}[i,:] \odot \text{Prior}[i,:)$$

for each encoded position $\tilde{x}$

hadamard product
(element wise multiplication)



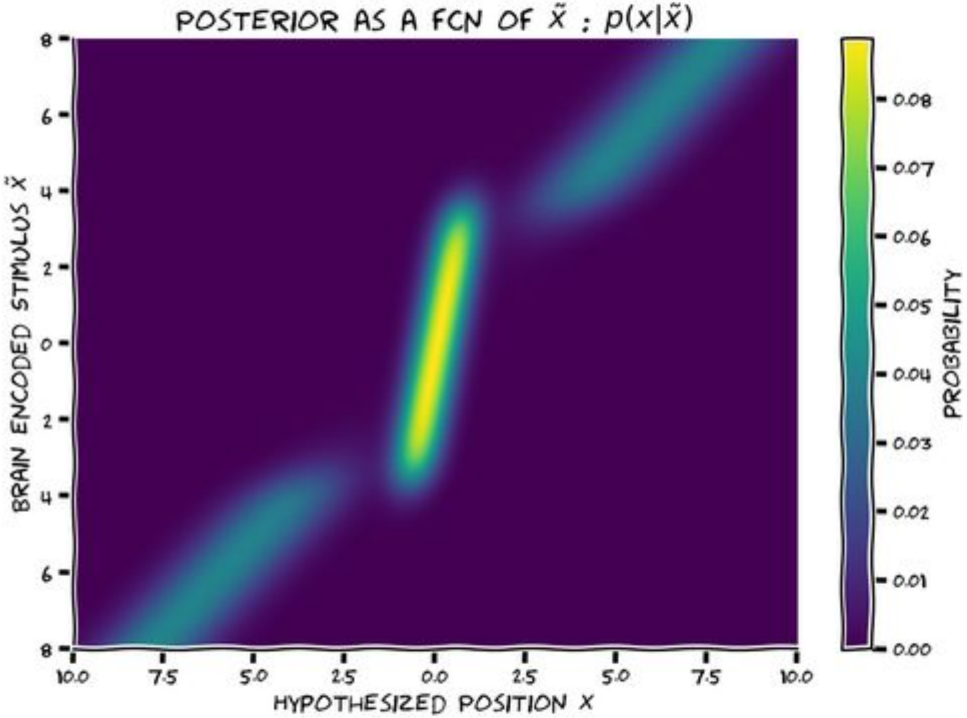
} represents posterior density

encodes decision for given encoding

mean of posterior = decision rule
(assumption)

posterior distribution represents estimated stimulus position $p(x|\tilde{x})$

{ estimate/response of sound location }
$\hat{x}(f(\tilde{x}))$


POSTERIOR AS A FCN OF $\tilde{x}$: $p(x|\tilde{x})$
BRAIN ENCODED STIMULUS $\tilde{x}$
8
6
4
2
0
2
4
6
8
10.0
7.5
5.0
2.5
0.0
2.5
5.0
7.5
10.0
HYPOTHESIZED POSITION X
PROBABILITY
0.08
0.07
0.06
0.05
0.04
0.03
0.02
0.01
0.00

Compute the binary decision array for each possible encoding

- calculate how likely each possible encoding is}
  for given true stimulus

→ create gaussian centered around true presented stimulus $(\sigma = 1)$

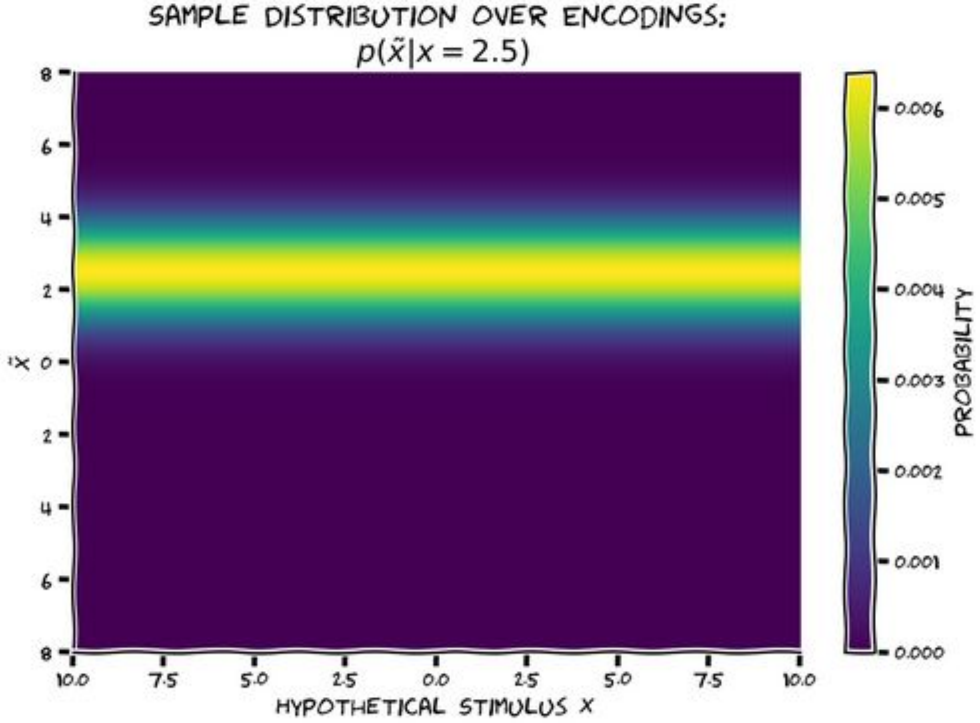
repeat across as f(potentially encoded value $\tilde{x}$)

→ Column gaussian centered around true presented stimulus

repeat across as f(hypothetical stimulus values x)

encodes distribution of brain encoded stimulus
& enable us to link true stimulus x to potential encoding $\tilde{x}$.

W2D1_pod 031


SAMPLE DISTRIBUTION OVER ENCODINGS:
$p(\tilde{x}|x = 2.5)$
8
6
4
2
$\tilde{x}$ 0
2
4
6
8
10.0
7.5
5.0
2.5
0.0
2.5
5.0
7.5
10.0
HYPOTHETICAL STIMULUS X
0.006
0.005
0.004
0.003
0.002
0.001
0.000
PROBABILITY

true stimulus x ⟷ potential encoding

↓ calculate

distribution of encodings.

↓

estimate

Marginalisation :

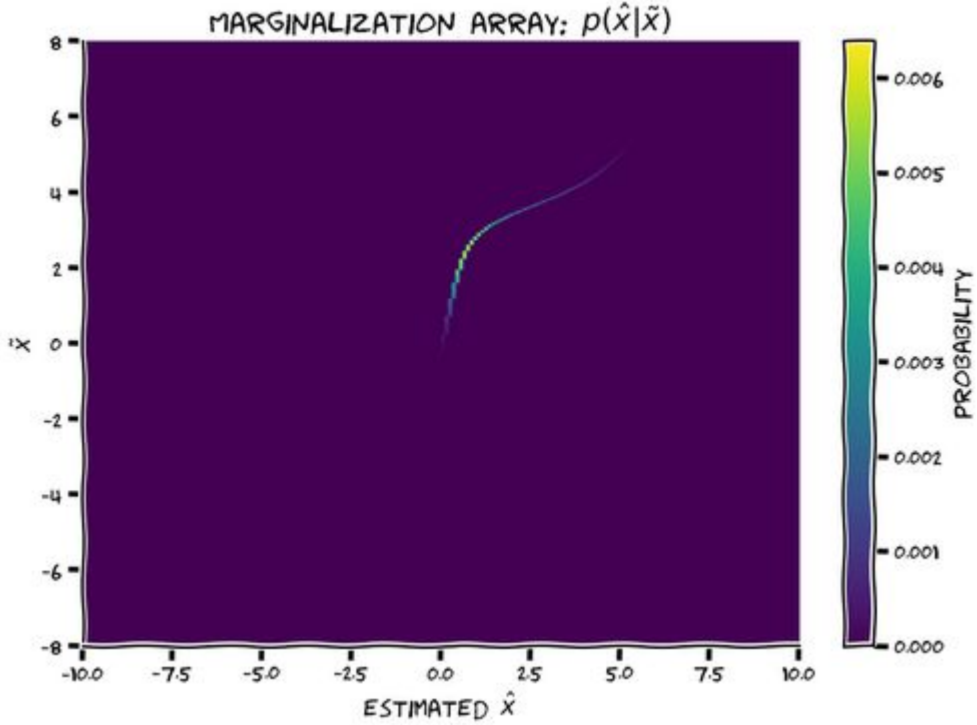
$$\text{Marginalisation\_Array} = \text{input array} \odot \text{binary decision array}$$

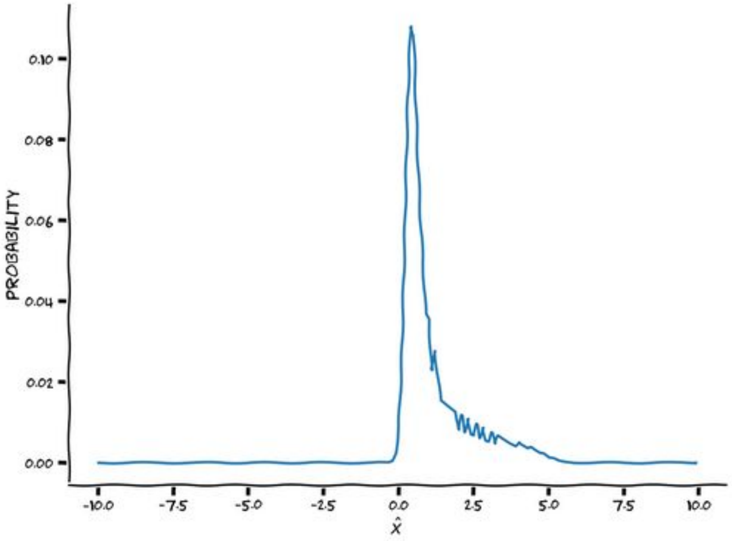
(MA)

true presented stimulus

$$\text{marginal} = \int_{\tilde{x}} MA$$

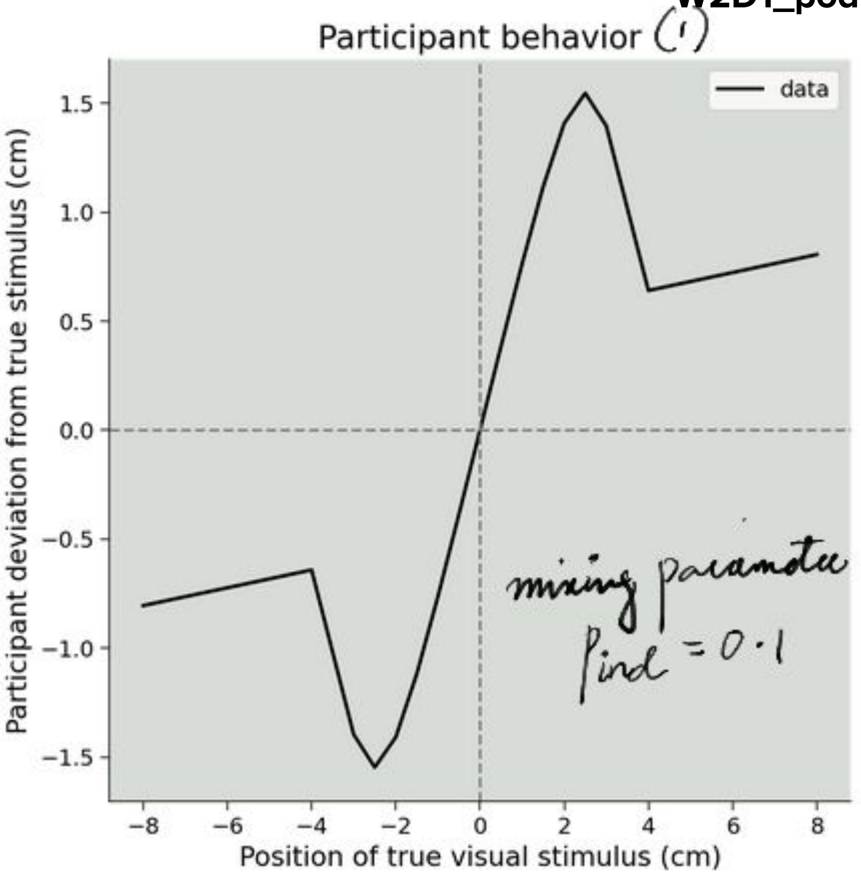
all hypothetical values


MARGINALIZATION ARRAY: $p(\hat{x}|\tilde{x})$
8
6
4
2
0
-2
-4
-6
-8
$\tilde{x}$
-10.0
-7.5
-5.0
-2.5
0.0
2.5
5.0
7.5
10.0
ESTIMATED $\hat{x}$
0.006
0.005
0.004
0.003
0.002
0.001
0.000
PROBABILITY


0.10
0.08
0.06
0.04
0.02
0.00
PROBABILITY
-10.0
-7.5
-5.0
-2.5
0.0
2.5
5.0
7.5
10.0
$\hat{x}$

Data generation

Parameter recovery experiments are a powerful method for planning and debugging Bayesian analyses--if you cannot recover the given parameters, implementation has malfunctioned! Note that this value for pindependent is not quite the same as our prior, which used pindependent=0.05.



Model fitting algorithm -

① implement prior matrix
(recompute posterior/input/marginal) } depends on p_{ind}

↓

$p(\tilde{x}|x)$ { not likelihood as it doesn't depend on p_{ind}

② compute negative log likelihood ∀ trials

find p_{ind} that minimises -ve (ll) / max +ve (ll)

assumption : trials are independent

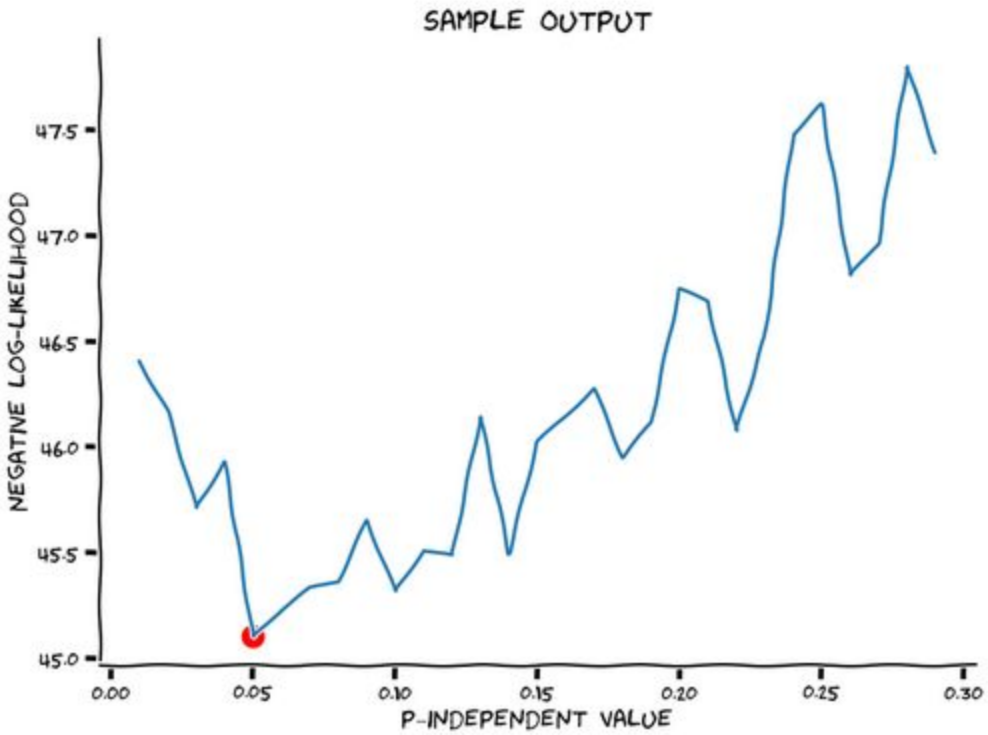
trial i }

$$-LL = -\sum_{i} \log p(\hat{x}_i \mid x_i)$$

$\hat{x}_i$ → participant response

x_i → presented stimulus

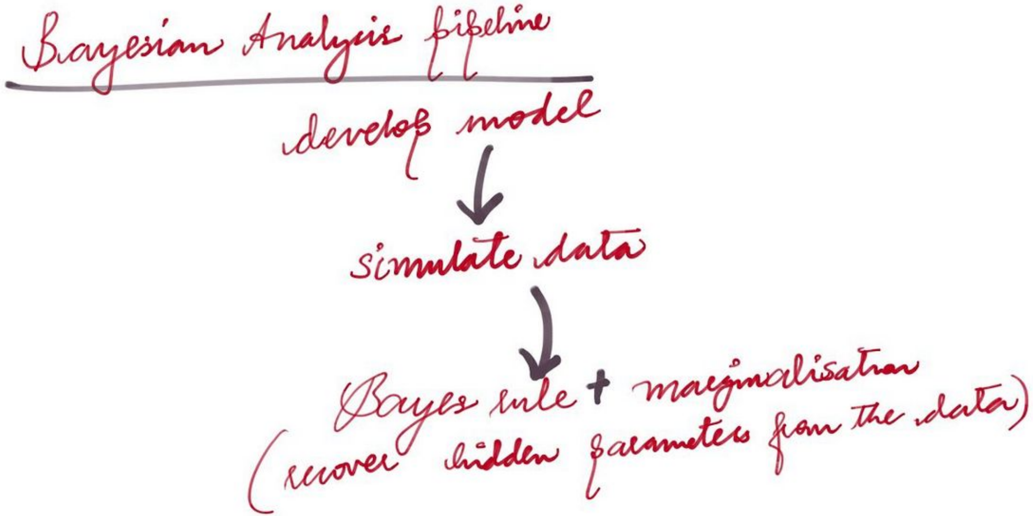
W2D1_pod 031


SAMPLE OUTPUT
NEGATIVE LOG-LIKELIHOOD
47.5
47.0
46.5
46.0
45.5
45.0
0.00
0.05
0.10
0.15
0.20
0.25
0.30
P-INDEPENDENT VALUE

Conclusion

Importance of $p_{independent}$ = describes how much weight subjects assign to the same-cause vs. independent-cause origins of a sound

posterior: describes beliefs based on a combination of current evidence and prior experience.



Bayesian Analysis pipeline

develop model

simulate data

Bayes rule + marginalisation

(recover hidden parameters from the data)

Tutorial #4 (Bonus) Explanations

Agenda

1. Implement three commonly-used cost functions: mean-squared error, absolute error, and zero-one loss
2. Discover the concept of expected loss, and
3. Choose optimal locations on the posterior that minimize these cost functions and verify that these locations can be found analytically as well as empirically.

Bayesian Decision theory

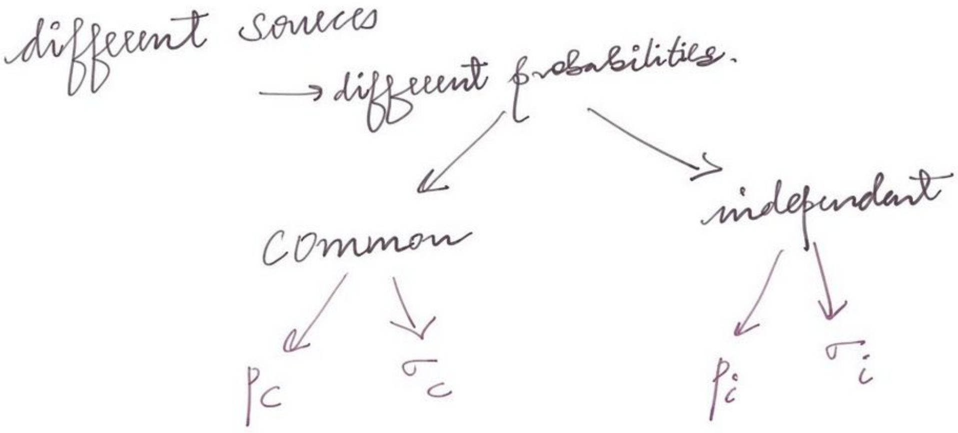
*combines the posterior with **cost functions** that allow us to quantify the potential impact of making a decision or choosing an action based on that posterior.*

$$\text{probabilities} \xrightarrow{\text{cost function}} \text{actions}$$

Mean of posterior $p(x|\tilde{x})$ as proxy for response $\hat{x}$ for the participants

Using mean/median/mode of posterior distribution } as decision rule!

W2D1_pod 031



different sources

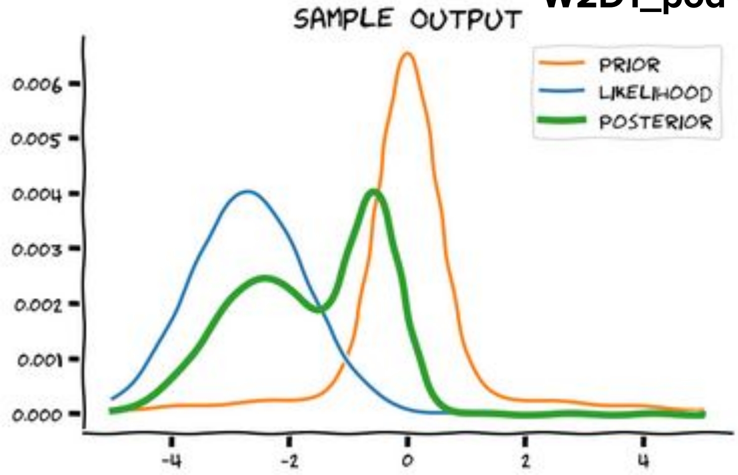
→ different probabilities.

common

p_c σ_c

independent

p_i σ_i

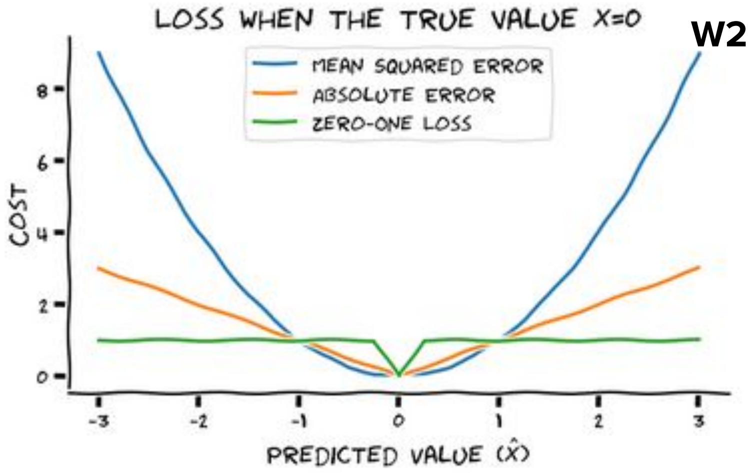


$$\text{prior} = \begin{cases} N_c(0, 0.5) & 95\%\ \text{weight} \\ N_i(0, 3) & 5\%\ \text{weight} \end{cases}$$

$$\text{likelihood} = N(-2.7, 1)$$

Cost functions

W2D1_pod 031



Cost functions.

→ determines cost/penalty of estimating $\hat{x}$ when true/correct quantity = x

→ cost of error b/w true stimulus x & estimate $\hat{x}$

① Mean squared error = $(x - \hat{x})^2$

② Absolute error = $|x - \hat{x}|$

③ Zero-one loss = $\begin{cases} 0 & \text{if } x = \hat{x} \\ 1 & \text{otherwise} \end{cases}$

Expected loss function

A posterior distribution tells us about the confidence or credibility we assign to different choices. A cost function describes the penalty we incur when choosing an incorrect option. These concepts can be combined into an *expected loss* function

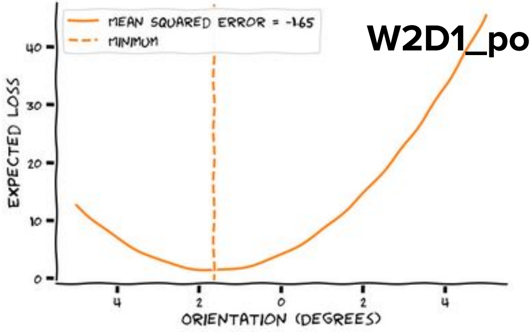
$$\text{Expected loss } E\left[\text{Loss} \,\middle|\, \hat{x}\right] = \int \underbrace{L\left[\hat{x}, x\right]}_{\text{loss function}} \underbrace{\odot}_{\substack{\text{hadamard product} \\ \text{(element wise multiplication)}}} \underbrace{p(x|\tilde{x})}_{\text{bimodal posterior}} \, dx$$

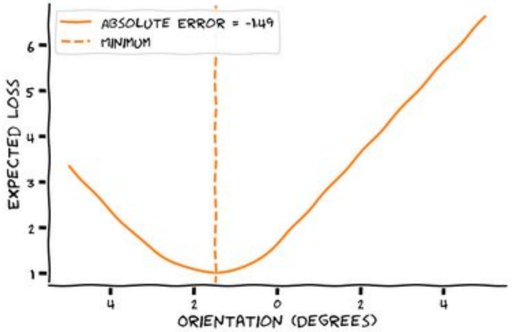
W2D1_pod 031

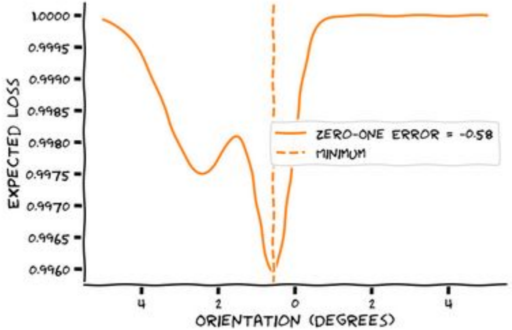
Expected loss outputs

minimum expected loss via brute-force: we searched over all possible values of X and found the one that minimized each of our loss functions.

This is feasible but can quickly become intractable.







Summary

the three loss functions are minimized at specific points on the posterior, corresponding to the itss mean, median, and mode. To verify this property, replot the loss functions with the posterior on the same scale beneath. The mean, median, and mode are marked on the posterior.

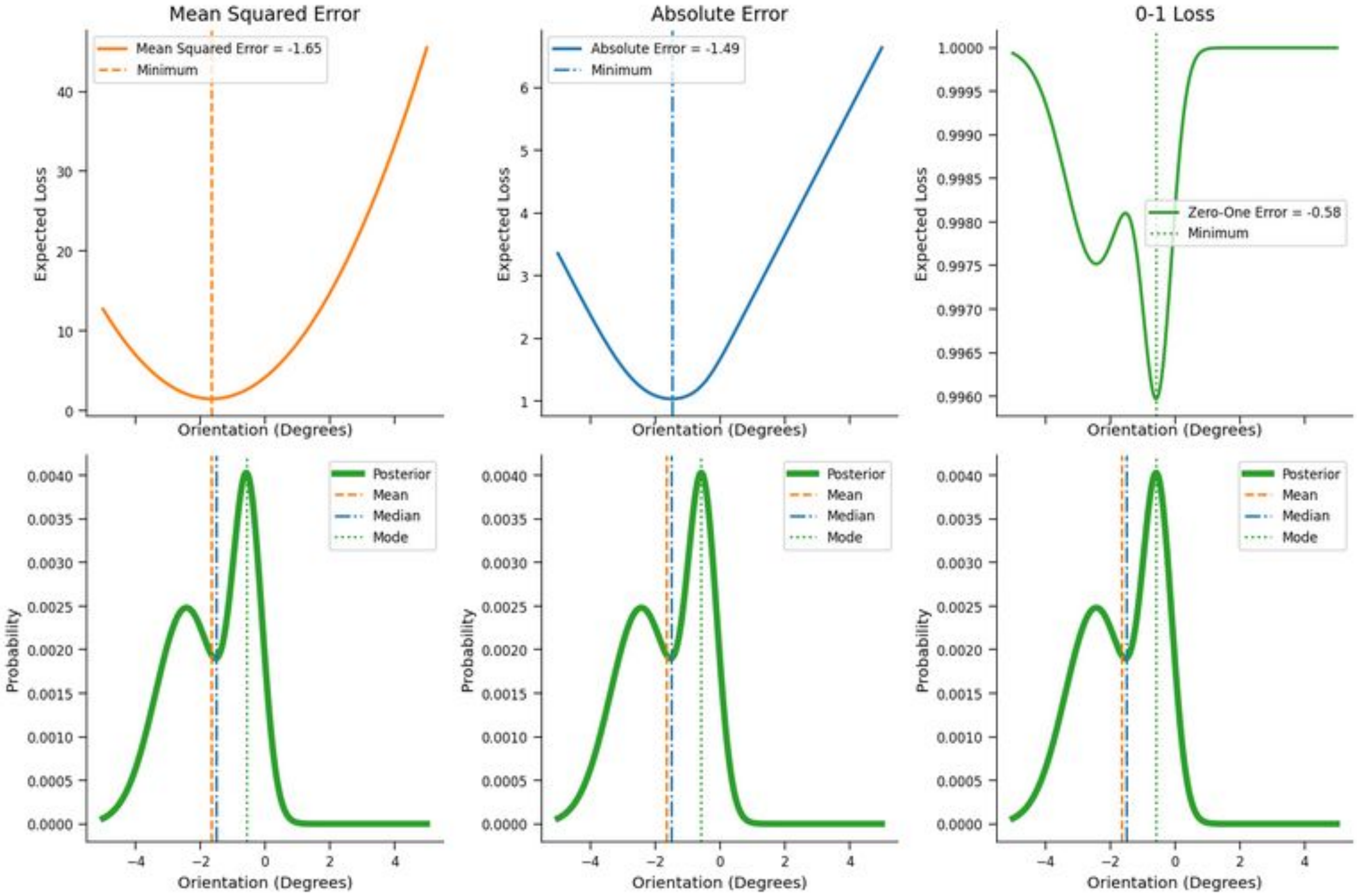
We used expected loss to quantify the results of making a decision, and showed that optimizing under different cost functions led us to choose different locations on the posterior. Finally, we found that these optimal locations can be identified analytically, sparing us from a brute-force search.

OBSERVATIONS: The mean minimizes the mean-squared error.

Absolute error is minimized by the median,

while zero-one loss is minimized at the posterior's mode.

W2D1_pod 031


Mean Squared Error
Mean Squared Error = -1.65
Minimum
Expected Loss
0
10
20
30
40
Orientation (Degrees)
Absolute Error
Absolute Error = -1.49
Minimum
Expected Loss
1
2
3
4
5
6
Orientation (Degrees)
0-1 Loss
Zero-One Error = -0.58
Minimum
Expected Loss
0.9960
0.9965
0.9970
0.9975
0.9980
0.9985
0.9990
0.9995
1.0000
Orientation (Degrees)
Posterior
Mean
Median
Mode
Probability
0.0000
0.0005
0.0010
0.0015
0.0020
0.0025
0.0030
0.0035
0.0040
−4
−2
0
2
4
Orientation (Degrees)

W2D1_pod 031

Food for thought

- *Suppose your professor offered to grade your work with a zero-one loss or mean square error.*
    - *When might you choose each?*
    - *Which would be easier to learn from?*

- *All of the loss functions we considered are symmetrical. Are there situations where an asymmetrical loss function might make sense? How about a negative one?*

Question #1 Answer

Firstly, the mean squared error is close to the variance, however you average the value of variance out by the number of the observations. In a way, it is a mean//average//expected value of the variance//dispersion of the data values.

The sum of your losses would no longer represent accuracy in this case, but rather the total "cost" of misclassification. The 0-1 loss function is unique in its equivalence to accuracy, since all you care about is whether you got it right or not, and not how the errors are made.

So, prefer mean squared errors while it's probably easier to learn from 0-1 loss as it's more straightforward.

Question #2 Answer

Not all loss is the same. So, we weight different losses in the loss functions giving rise to an asymmetrical loss metric.

It is the case that we often use loss functions that become equal to zero when the fit of the model to the training data is perfect, but the optimization algorithms don't care about this, and they drive the loss function to algebraically more negative values, and not towards zero.

References

http://www.lenstrnad.com/blog/2018/09/asymmetric_loss_function

https://discuss.pytorch.org/t/what-happens-when-loss-are-negative/47883/3

https://stats.stackexchange.com/questions/284028/0-1-loss-function-explanation/284062