

# Causal inference in multisensory perception

Adu Matory and Lukas Braun

Supervisor: Denis Alevi

Models of Higher Brain Function Pt. 2, WiSe 20

## Introduction

In this work we replicate parts of Körding et al. (2017). The paper uses an “ideal-observer” in a multisensory perception task, which is based on causal bayesian inference model, and compares it to human behaviour. Given an auditory and a visual cue, the task is to estimate the location of the cues and to infer whether the cues originate from the same cause i.e. from the same location. The model accurately predicts the nonlinear integration of cues by human subjects.

A simultaneously perceived auditory and visual cue can originate from the same or two distinct causes. For example, the ringing of a bell and flashing lights at an amusement park might come from two distinct attractions like a merry-go-round and a nearby rollercoaster or they might come from the same cause e.g. a garbage truck trying to pass. Integrating both pieces of sensory information such that the underlying cause(s) are correctly inferred is crucial for generating an understanding of the world surrounding us from multi-sensory perceptions.

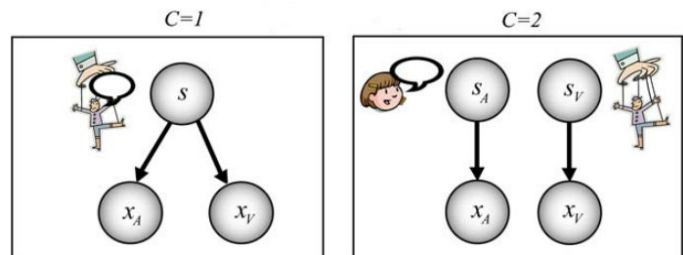
## Model

We distinguish between two conditions:

Either, the auditory ( $s_a$ ) and visual ( $s_v$ ) cues are generated by the same cause ( $C=1$ ), in which case they originate from the same position  $s=s_a=s_v$ ; or they are generated from two distinct causes ( $C=2$ ), in which case they originate from two different positions  $s_a \neq s_v$ .

The real position of the auditory cue ( $s_a$ ) and

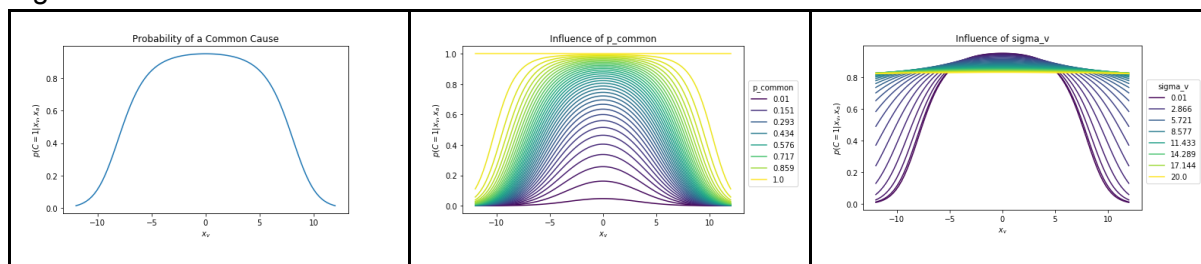
visual cue ( $s_v$ ) are perturbed by auditory noise ( $\sigma_a$ ) and visual noise ( $\sigma_v$ ) respectively, leading to the perceived position for the auditory cue  $x_a$  and the perceived position of the visual cue  $x_v$ .

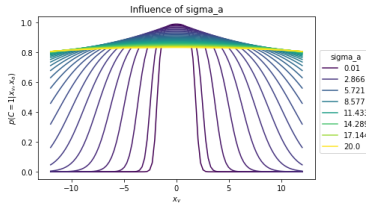
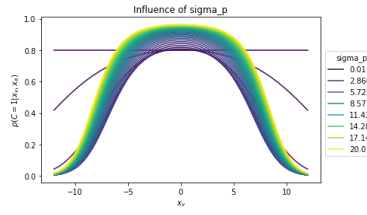


## Exercise 1: Understanding and implementing the model

Given the causal inference model (Eq. 2 from the paper), we can calculate the probability of a common cause, given percepts  $x_a$  and  $x_v$ . That is  $p(C=1|x_a, x_v)$ . In Figure 1. we investigate the influence of other model parameters. We assume, that  $x_a$  is constantly 0 and plot the probability for changing  $x_v$ . The parameter and default values are: The prior probability that there is a single cause ( $p_{\text{common}} = 0.8$ ), the visual noise level ( $\sigma_v = 0.6$ ), the auditory noise level ( $\sigma_a = 3.1$ ) and a bias to model that subjects tend to perceive stimuli straight ahead ( $\sigma_p = 15$ ).

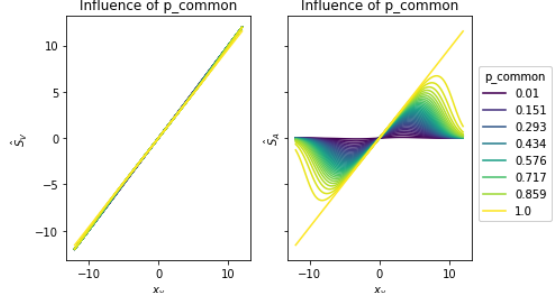
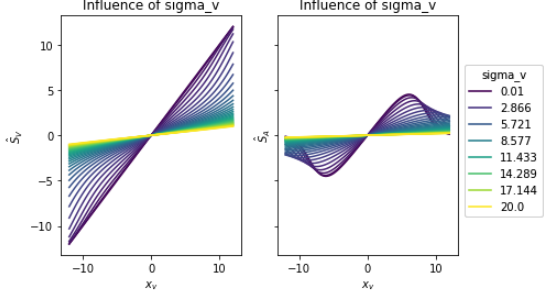
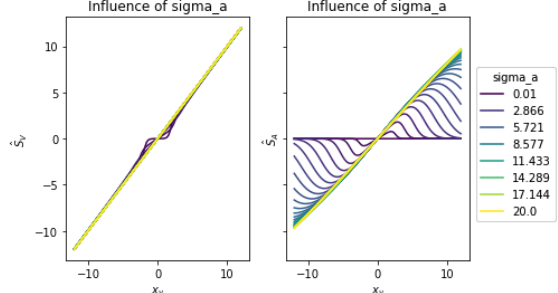
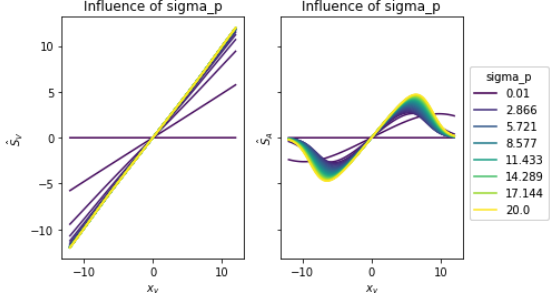
Figure 1



<p>The probability for a single cause, given the percepts <math>x_a</math> and <math>x_v</math>, decreases as the disparity between the percepts increases (remember that <math>x_a</math> is constantly kept at 0).</p>	<p>Increasing the probability of a common cause (<math>p_{\text{common}}</math>), increases the peak of <math>p(C=1 x_v, x_a)</math> around <math>x_v=0</math> constantly and for larger deviations of <math>x_a</math> and <math>x_v</math> i.e. the curve is widened.</p>	<p>Increasing the visual noise (<math>\sigma_v</math>) widens the curve, until it fully flattens at 0.8, which is the value of <math>p_{\text{common}}</math> and thus spatial information is neglected.</p>
 <p>Plot titled 'Influence of sigma_a' showing the probability <math>p(C=1 x_v, x_a)</math> on the y-axis (0.0 to 1.0) versus <math>x_v</math> on the x-axis (-10 to 10). Multiple curves are shown for different values of <math>\sigma_a</math>: 0.01, 2.866, 5.721, 8.577, 11.433, 14.289, 17.144, and 20.0. As <math>\sigma_a</math> increases, the curves become wider and flatter, with the peak at <math>x_v=0</math> decreasing.</p>	 <p>Plot titled 'Influence of sigma_p' showing the probability <math>p(C=1 x_v, x_a)</math> on the y-axis (0.0 to 1.0) versus <math>x_v</math> on the x-axis (-10 to 10). Multiple curves are shown for different values of <math>\sigma_p</math>: 0.01, 2.866, 5.721, 8.577, 11.433, 14.289, 17.144, and 20.0. As <math>\sigma_p</math> increases, the curves become narrower and taller, with the peak at <math>x_v=0</math> increasing.</p>	
<p>Increasing the auditory noise (<math>\sigma_a</math>) widens the curve. For very low noise values the curve is more narrow than in the varying visual noise condition, reflecting the lower visual noise baseline of <math>\sigma_v = 0.6</math>.</p>	<p>Increasing the bias term (<math>\sigma_p</math>) slowly narrows and increases the curve, indicating that the higher the bias, the higher the probability for the percepts to be perceived as coming from the same source.</p>	

Under the assumption of a mean squared error on the observer's observation of the visual ( $\hat{s}_v$ ) and auditory positions ( $\hat{s}_a$ ), equations 9. and 10. then define the estimates that minimize the error: The optimal-observer. In Figure 2., we investigate how the model parameters influence the observer's behaviour. Again, we assume that  $x_a$  is constantly 0 and plot the position estimate for changing  $x_v$ .

Figure 2

 <p>Two plots titled 'Influence of p_common' showing the estimate <math>\hat{s}_v</math> on the y-axis (-10 to 10) versus <math>x_v</math> on the x-axis (-10 to 10). The left plot shows the estimate for <math>\hat{s}_a</math> and the right plot shows the estimate for <math>\hat{s}_v</math>. Multiple curves are shown for different values of <math>p_{\text{common}}</math>: 0.01, 0.151, 0.293, 0.434, 0.576, 0.717, 0.859, and 1.0. As <math>p_{\text{common}}</math> increases, the curves shift towards zero.</p>	 <p>Two plots titled 'Influence of sigma_v' showing the estimate <math>\hat{s}_v</math> on the y-axis (-10 to 10) versus <math>x_v</math> on the x-axis (-10 to 10). The left plot shows the estimate for <math>\hat{s}_a</math> and the right plot shows the estimate for <math>\hat{s}_v</math>. Multiple curves are shown for different values of <math>\sigma_v</math>: 0.01, 2.866, 5.721, 8.577, 11.433, 14.289, 17.144, and 20.0. As <math>\sigma_v</math> increases, the curves shift towards zero.</p>
<p>While the estimate of the visual position is only slightly perturbed from the perfect estimate (diagonal) by a changing <math>p_{\text{common}}</math>, the estimate of the auditory position is perturbed from the optimal to estimates closer to the position straight ahead of the observer for a decreasing <math>p_{\text{common}}</math>.</p>	<p>Increasing the visual noise shifts the estimate of the position of the visual cue from optimal (diagonal) to closer around zero. Interestingly the estimate of the auditory cue, causing stimuli with high disparity in <math>x_v</math> and <math>x_a</math></p>
 <p>Two plots titled 'Influence of sigma_a' showing the estimate <math>\hat{s}_v</math> on the y-axis (-10 to 10) versus <math>x_v</math> on the x-axis (-10 to 10). The left plot shows the estimate for <math>\hat{s}_a</math> and the right plot shows the estimate for <math>\hat{s}_v</math>. Multiple curves are shown for different values of <math>\sigma_a</math>: 0.01, 2.866, 5.721, 8.577, 11.433, 14.289, 17.144, and 20.0. As <math>\sigma_a</math> increases, the curves shift towards zero.</p>	 <p>Two plots titled 'Influence of sigma_p' showing the estimate <math>\hat{s}_v</math> on the y-axis (-10 to 10) versus <math>x_v</math> on the x-axis (-10 to 10). The left plot shows the estimate for <math>\hat{s}_a</math> and the right plot shows the estimate for <math>\hat{s}_v</math>. Multiple curves are shown for different values of <math>\sigma_p</math>: 0.01, 2.866, 5.721, 8.577, 11.433, 14.289, 17.144, and 20.0. As <math>\sigma_p</math> increases, the curves shift towards zero.</p>
<p>Increasing the auditory noise (<math>\sigma_a</math>) widens the curve. For very low noise values the curve is more narrow than in the varying visual noise condition, reflecting the lower visual noise baseline of <math>\sigma_v = 0.6</math>.</p>	<p>Increasing the bias term (<math>\sigma_p</math>) slowly narrows and increases the curve, indicating that the higher the bias, the higher the probability for the percepts to be perceived as coming from the same source.</p>

To evaluate the integral in equation 13, we must solve the function  $p(s_{\hat{v}} | x_v; x_a)$ , the conditional probability of the predicted distribution of estimated visual positions on two priors  $x_v, x_a$ . For a solution using bayes rule, we'd know we need the fixed gaussian probabilities  $p(x_v)$ ,  $p(x_a)$ , and  $p(s_v)$ , as well as  $p(x_v | x_{\hat{v}})$  and  $p(x_a | s_{\hat{v}})$ , the probability distributions of the stimulus positions on the estimated visual position. This function is a gaussian probability density, which means analytically, the larger integral just a gaussian (resulting from a product of gaussians) and the integral can be parameterized nicely.

## Exercise 2: Fitting the computer model to experimental data

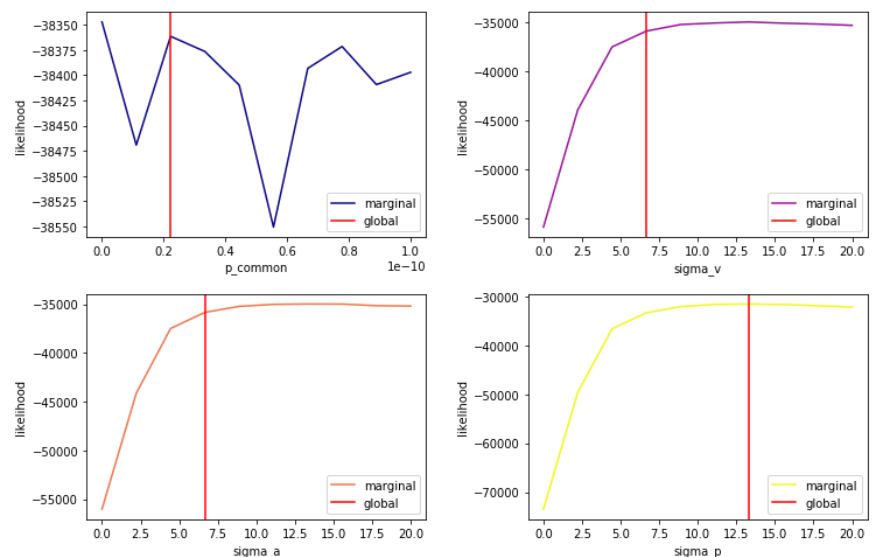
In this part of the exercise, we generated 'experimental data' and performed both maximum likelihood estimation and Monte Carlo Markov Chain (MCMC) sampling to find parameters that had the highest likelihood of explaining our experimental data. By comparing the parameters found by these search methods to the parameters that were used to generate the experimental data, we could observe how well each search method performed at actually finding optima. The following is a discussion of the implementation of these methods and how well they performed.

For maximum likelihood estimation, we first took the likelihood function of our experimental data probability density. The probability density was generated by sampling the space 10 times more than the experimental data, vitally giving us a distribution closer to the actual probability distribution. We then applied a logarithm, whose monotonically increasing values allow for easy comparison between parameter combinations of low and high likelihood. In general, MLE doesn't scale well with more complex likelihood functions and doesn't work well with small samples. It was suitable for this case given our 10,000 experimental samples and our simple likelihood function.

NaN and infinity can appear in the calculation of the likelihood if there is zero or near zero probability of a particular condition, because the log of zero is undefined. This tells us that the likelihood manifold is convex in parameter space and has a global minimum. Therefore we can use gradient-based sampling methods to search for parameters with the maximum likelihood of fitting the data. For a solution to this, we added a small number epsilon (i.e. =  $1e-10$ ) that ensures that the logarithm doesn't return negative infinity or NaN for zero or near-zero probabilities, therefore allow our calculation to continue without errors.

Figure 3.

Likelihood of model parameters  
(calculated with brute force search)



We implemented a MLE brute force parameter search by generating  $10^4$  sets of parameters, with 10 degrees of freedom for each relevant parameter and then calculated the log-likelihood of each set. Figure 3. shows the marginal probability over each parameter. We noted a difference in the maximum likelihood values globally and marginally. This was expected, as it would not have been

possible to find the maximum values with only the marginals because the interaction priors between the terms dictate that the global maximum likelihood parameters cannot be reliably obtained from the marginal.

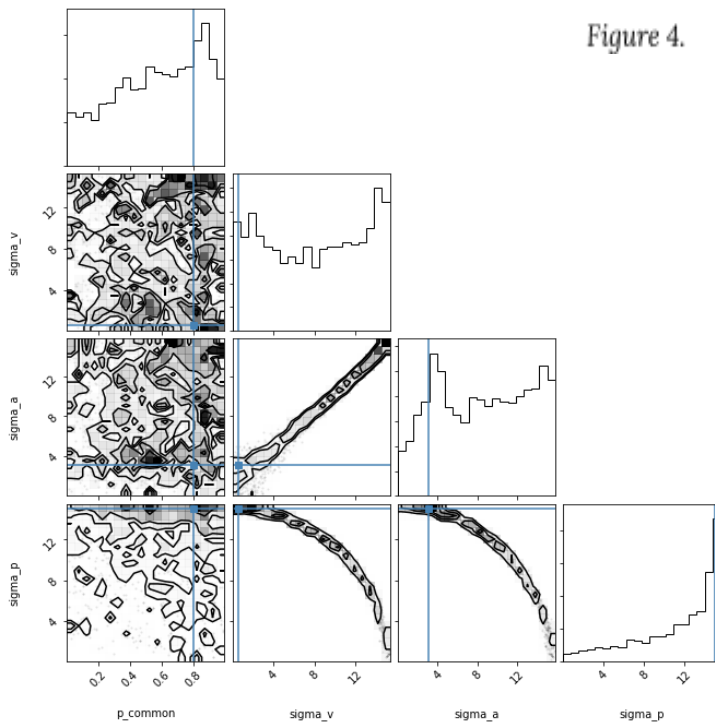


Figure 4.

In Figure 4., we have the posterior probability distributions of our parameters. In each plot on the diagonal, we see the marginal distribution for each parameter and in the others, the two-dimensional marginal distributions. The blue lines indicate the true parameter values that we used to generate the experimental data from which the MCMC sampler sampled. We see here how pairs of parameters covary in their explanation of the experimental data.

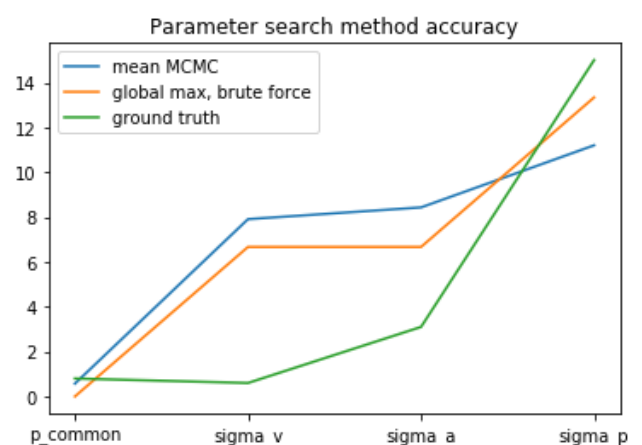
### Comparing search methods

In Figure 5., we see the mean value of the search space of the MCMC was more similar to from maximum values found by brute force search but was indeed further to the truth across most parameters.

The cost of brute force search is proportional to the number of possible solutions, which can grow wildly as the model over which it is search becomes more complex (i.e. increasing number of parameters). However, it is very simple and that can be preferable to more complex algorithms, especially when time is not a constraint. It can be helpful when one already has an idea of a limited set of parameters over which to search. Its simplicity also allows it to serve as a good benchmark for other search algorithms.

Using MCMC, we more intelligently look for regions in parameter space of maximum likelihood. Instead of discretely defining parameter values to search over we construct markov chain walkers that explore the parameter space by exploiting regions of high likelihood. This is a much more efficient search technique and it allows easier numerical approximation of integrals in very large, hierarchical bayesian models. However, the MCMC walkers may get stuck in regions of high local likelihood, without ever reaching region of maximum likelihood. The higher complexity of the method also means that the initial set up may take longer. In our case, implementing a functional MCMC algorithm with the package emcee took about 10 times as long as implementing the brute force search.

Figure 5.



### MCMC for mixed data

Figure 6. shows the results of applying the MCMC to generated data from two "subjects". The subjects' data only varied by the parameter  $\sigma_a$  used to generate them;  $\sigma_a,1 = 9$ ,

$\sigma_{a,2} = 2$ . The red lines represent the true generative parameters for subject 1 and the green, for subject 2.

We see fewer regions of high likelihood model, which implies the search was more constrained.

Superficially, this seems to be a good thing, but inspecting the results, we see there are no regions of high marginal likelihood for  $\sigma_a$  that come close. It seems the walkers simply never ventured into that territory because they found a region of much higher likelihood near  $\sigma_a = 9$ . This implies that an MCMC on experimental data may end up in regions of local optimality based on the results of just one subject out of many, making the algorithm susceptible to outliers.

