Bayesian Causal Inference Toolbox (BCIT) for MATLAB

Majed Samad, Kellienne Sita, Annie Wang, Ladan Shams

May 1, 2017

1 Abstract

BCIT is a software extension built for the MATLAB platform, intended to facilitate running simulations and model fitting using the Bayesian causal inference model, a statistical framework for determining whether to integrate or segregate information arriving to the nervous system from different sensory channels. This is of relevance for many multisensory phenomena that result from the tricking of the sensory-perceptual system such as ventriloquism, the flash-beep illusion, the rubber hand illusion, and the Mckgurk effect. These illusions can all be accounted for under the normative framework of the Bayesian causal inference model. The aims of this program are 1) to provide the user with a graphical user interface by which the inner workings of the model can be made intuitive and easy to understand, and 2) to provide the user with the necessary machinery to be able to run an optimization procedure to obtain optimal model fits to user-supplied datasets. Therefore, the primary intention with releasing this toolbox is to enhance the acquisition of the intuition behind the computational framework, which we hope to achieve by the implementation of user interface elements to control various parameters in the model, and to instantaneously observe the effect they have on the output from the model. Noting that the model is typically used to account for experimentally collected data, and would thus be intended to be fit to such data, we are also hoping to eliminate the barriers that prevent researchers from implementing the model and conducting fitting. Here, we present the toolbox and provide a description of the models that are implemented in it, and we also document a validation procedure demonstrating the convergence of the fitting procedure to the approximately correct parameters. Therefore, this toolbox provides a powerful platform for the rapid implementation of the Bayesian causal inference model.

2 Introduction

The Bayesian causal inference model is a well-established computational model of perception that performs a statistical inference to determine whether signals across different sense modalities originated from the same cause, and thus ought to be integrated, or otherwise, and thus ought to be segregated. It was established nearly a decade ago and has been widely used since then to account for a wide range of multisensory perception phenomena (Kording et al., 2007; Beierholm et al., 2009b,a; Wozny et al., 2010; Samad et al., 2015; Samad and Shams, 2016; Rohe and Noppeney, 2015; Kilteni et al., 2015).

Specifically, the causal inference model is a statistical inference model that performs an arbitration between integration and segregation, based on the spatiotemporal congruence of the signals – or indeed congruence along any suitably defined space – as well as a prior tendency to integrate/segregate. As such, this model represents a significant advance in the field of computational modeling from the method commonly known as Maximum Likelihood Estimation (MLE), which assumes that the signals of interest ought to always be combined, and thus, that they were generated by a common cause (Ernst and Banks, 2002). In contrast, the causal inference model makes no such assumptions but rather infers whether the situation of having been generated by a common cause or separate causes is more likely and then estimates the stimulus attributes accordingly.In this model, the variability in response from trial to trial is modeled by the variability in the mean of the likelihood function, which is sampled from a distribution with a mean equal to the true stimulus value (if we assume that the estimator is unbiased, for example the true location of the stimulus) and a variance which is equal to the variance of the likelihood function on each trial. Therefore, it is assumed that

the nervous system is aware of its own variability.

This model has been shown to account for a wide range of multisensory phenomena across many domains including numerosity judgments in the flash-beep illusion (Wozny et al., 2008), spatial localization judgments in an audiovisual task (Wozny et al., 2010) and a visuotactile task (Samad and Shams, 2016), the size-weight illusion (Peters, 2014) and the rubber hand illusion (Samad et al., 2015), and has even been shown to account for the Mcgurk-Mcdonald Effect (Magnotti et al., 2013). The dissemination and distribution of this toolbox will, therefore, provide a very important service to the field of perception, and by extension computational neuroscience. In recent years, we have seen an explosion of interest in this computational framework by research groups from all around the world. Providing them with an interface as user-friendly as ours will dramatically reduce the friction with which they will be able to make use of its powerful computations.

Across these different domains, three general forms of the model have been in use, and are therefore provided to the user with the current software release. Namely, we are referring to the models of localization and numerosity across a variety of modalities. These have been modeling using the Bayesian causal inference model that is characterized over a continuous or a discrete space. A third variant was introduced by Samad et al. (2015) in order to account for the rubber hand illusion and thus operates over a two dimensional (spatiotemporal) continuous space. Therefore, in what follows, we will concern ourselves with these three forms.

Mathematical Formulation

The model utilizes the following form of Bayes Rule:

$$p(C|x_1, x_2) = \frac{p(x_1, x_2|C)p(C)}{p(x_1, x_2)} \tag{1}$$

where x_1 and x_2 are two signals received by the nervous system, and C is a binary variable denoting the number of causes in the environment, 1 or 2.

Therefore, the posterior probability of the signals having a single cause in the environment is computed as:

$$p(C = 1|x_1, x_2) = \frac{p(x_1, x_2|C = 1)p(C = 1)}{p(x_1, x_2|C = 1)p(C = 1) + p(x_1, x_2|C = 2)(1 - p(C = 1))} \tag{2}$$

where the likelihood probability is:

$$p(x_1, x_2|C = 1) = \iint p(x_1, x_2|X)p(X)dX \tag{3}$$

and $p(C = 1)$ is the prior probability of a common cause. X denotes the attributes of the stimuli in the dimension of relevance, and which gives rise to the neural representations $\{x_1, x_2\}$. It is modeled as a continuous random variable and has the following prior: $\mathcal{N}(\mu_X, \sigma_X)$, where $\mathcal{N}(\mu, \sigma)$ stands for a normal distribution with mean μ and standard deviation σ. Equation A.2 shows that two factors contribute to the inference of a common cause: the likelihood (the first term in the numerator) and the prior (the second term in the numerator). A high likelihood (Equation A.3) occurs if the sensory signals are similar. The prior probability of a common cause, $p(C = 1)$, on the other hand, is independent of the present sensations, and depends on the observer's prior experience.

Note that the rest of the full mathematical formulation is left out from this document as it has appeared previously in print, and we direct the inquisitive reader thereto for further elucidation. In the tables which follow, any ambiguous formulation will be paired with an Equation number in Wozny and Shams (2011) that describes the formulation in full detail.

3 Program Description

BCIT is structured into two main types of operations: simulation and model fitting. The former permits the user to simulate from a selection of the three most commonly used variants of the model. These will be described in much greater detail below. The model fitting aspect permits users to use the `fminsearchbnd.m` optimization method on their own data sets, or alternatively, on a sample data set created from the

`create_data.mat` file that is included. Thus, the user interface is structured into a main panel that allows the user to choose from among different model types, and whether simulation or fitting is desired. From there, the user navigates through some additional panels to achieve the desired computation, which will be described in the sections that follow.

The graphical user interface

The main menu that the user sees upon first opening up the program is illustrated in Figure A.1. Here the user is presented with a choice from amongst the three most commonly used variants of the model: one dimensional continuous space, one dimensional discrete space, and two dimensional continuous space.

The user is also provided with the option to run simulations using these all three of the variants and can launch separate windows to view these simulations by having selected the desired model and pressing the "Simulate" button. In addition, the user is provided with the ability to conduct a model fitting procedure for a dataset of choice, using either of the one dimensional models, by selected one of them and pressing the "Fit Model" button. Note that a model fitting routine for the two dimensional model was not provided as this has not yet been performed due to the difficulty with acquiring a suitable dataset.

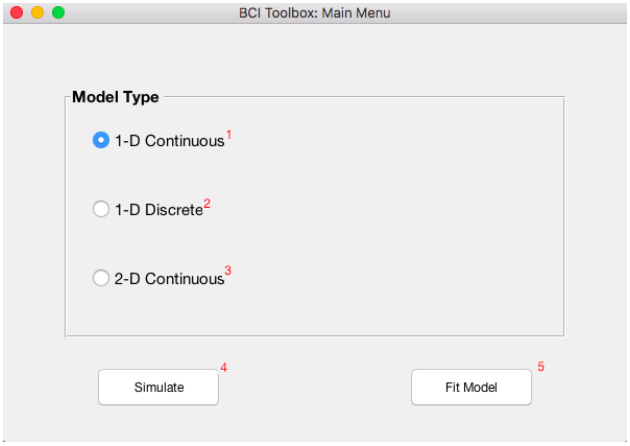

Figure 1: Main Menu of the GUI

The simulation panels

The primary aim of this project is to provide the user with an interface by which the inner workings of the Bayesian causal inference model can be made intuitive and easy to understand. Therefore, the primary intention with releasing this toolbox is to enhance the acquisition of the intuition behind the computational framework, and is thus primarily to be used as an educational tool. To that end, we hope to be able to provide the user with interface elements to control various parameters in the model, and instantaneously be able to observe the effect they have on the output from the model.

The simulation panels are split into three parts.

Model Elements	Response Distribution	The model output: a distribution of the estimates of the positions of the stimuli based on the likelihood and prior
	Stimulus Encoding	The probability density functions representing the encoding of the stimuli, modeled as Gaussian distributions. See equations 1 and 2 in Wozny and Shams (2011)
	Spatial Prior	The probability density function representing the expected stimulus location, modeled as a Gaussian distribution. See equation 3 in Wozny and Shams (2011)
Model Estimates	Mode	Most probable estimated response
	Mean	Mean estimated response
Strategies	Selection	Model selection is when the observer selects the most likely causal structure and estimates the stimulus location wholly on the basis of the selected model. See equation 16 in Wozny and Shams (2011)
	Averaging	Model averaging is when the observer weights the estimates of the stimulus locations by the inferred probabilities of their causal structure. Considered the most optimal strategy. See equation 15 in Wozny and Shams (2011)
	Matching	Probability matching is a strategy that choses the estimates from either causal structure based on their inferred probabilities. Although this method is suboptimal, it appears to be the most frequently used in cognitive tasks. See equation 17 in Wozny and Shams (2011)

Table 1: **Simulation Panels:** Overview of Settings

Stimulus Position	Stimulus 1	The true position of the stimulus (modality 1)
	Stimulus 2	The true position of the stimulus (modality 2)
Parameters	P(C=1)	The prior probability that both signals can be attributed to one cause
	SD(1)	The standard deviation of the Gaussian distribution of the likelihood for modality 1
	SD(1)	The standard deviation of the Gaussian distribution of the likelihood for modality 2
	SD(Prior)	The standard deviation of the Gaussian distribution of the prior (the anticipated location of the stimuli)
	Mean(Prior)	The mean of the Gaussian distribution of the prior
	Additional Parameters	Additional parameters specific to the models will be discussed in the detailed model descriptions below
Bottom Panel Buttons	Screenshot	Saves a copy of the screenshot (figure, all parameters) with user-set filename to the current directory
	Reset	Resets all parameters and figure to default settings
	Return	Returns to the main menu

Table 2: **Simulation Panels:** Overview of UI Elements

One Dimensional Continuous

This model is one dimensional and continuous. This represents the most basic form of the model and is most akin to the form that was introduced in the seminal paper in 2007 (Kording et al., 2007). Over the years, we have produced several variants of it that were tailored for particular tasks and domains. But it is best we begin our discussion of the core computations with reference to this initial form.

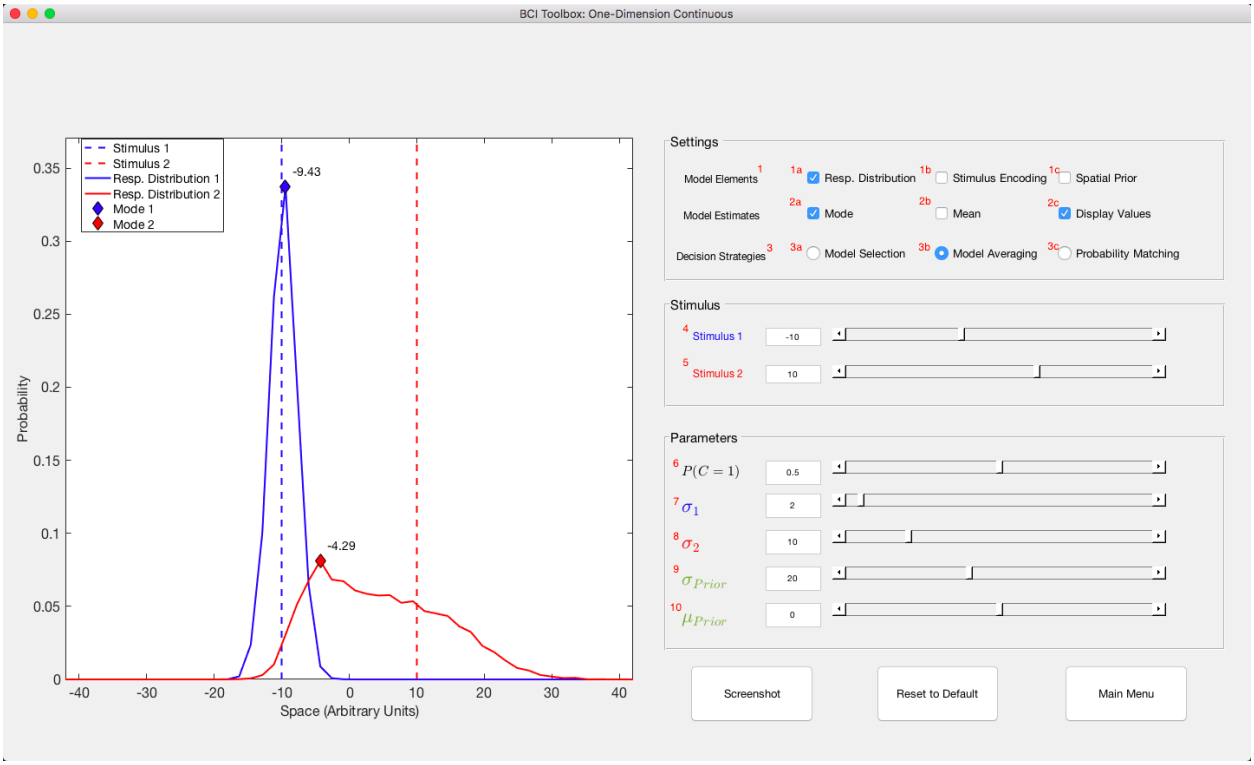

Figure 2: Simulation Panel 1: One Dimensional Continuous

For all parameters: Both the boxes and sliders can be used to manipulate values. Sliders can be manipulated either by using the arrow keys attached to the left and right or by pressing the spaces to the left or right of the value indicator, and the increments of the sliders are given below.

Model Elements (1)	Response Distribution (1a)	Indicated by the solid blue and red lines
	Stimulus Encoding (1b)	Indicated by the dotted blue and red lines
	Spatial Prior (1c)	Indicated by the dotted green line
Model Estimates (2)	Mode (2a)	Value indicated by red and blue diamonds
	Mean (2b)	Value indicated by red and blue squares
	Display Values (2c)	Shows the value of the model estimate of probability on the figure
Strategies (3)	Selection (2a)	See explanations in "Description" section, under "Simulation Panels"
	Averaging (3b)	
	Matching (3c)	
Stimulus Position	Stimulus 1 (4)	Stimulus position bounds range from -40 to 40. Sliders increment by 1
	Stimulus 2 (5)	
Elements Parameter	P(C=1) (6)	Probability values range from 0 to 1. Slider increments by 0.01
	SD(1) (7)	Standard deviation of X1 signal ranges from 0.1 to 50. Slider increments by 0.1
	SD(1) (8)	Standard deviation of X2 signal ranges from 0.1 to 50. Slider increments by 0.1
	SD(Prior) (9)	Standard deviation of prior signal ranges from 1 to 50. Slider increments by 0.1
	Mean(Prior) (10)	Average of prior signal ranges from -40 to 40. Slider increments by 1

Table 3: **One-Dim Continuous Simulation Panel:** Description of UI Elements

One Dimensional Discrete

This model is also one dimensional and discrete.

For all parameters: Both the boxes and sliders can be used to manipulate values. Sliders can be manipulated either by using the arrow keys attached to the left and right or by pressing the spaces to the left or right of the value indicator. The increments of the sliders are given below.

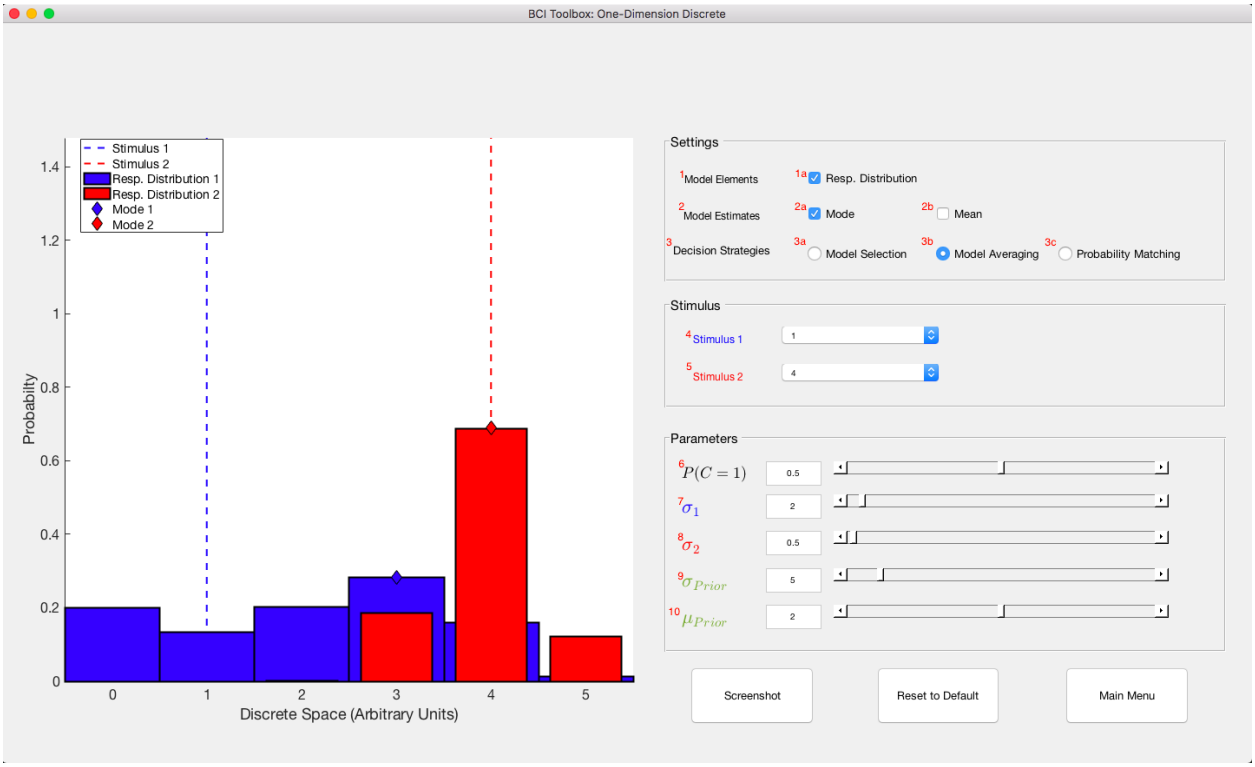

Figure 3: Simulation Panel 2: One Dimensional Discrete

Model Elements (1)	Response Distribution (1a)	Model estimate indicated by red and blue bars
Model Estimates (2)	Mode (2a)	Value indicated by red and blue diamonds
	Mean (2b)	Value indicated by red and blue squares
Strategies (3)	Selection (2a)	See explanations in "Description" section, under "Simulation Panels"
	Averaging (3b)	
	Matching (3c)	
Stimulus Position	Stimulus 1 (4)	Discrete model only allows for the stimuli to be in discrete positions defined by number: 0, 1, 2, 3 or 4
	Stimulus 2 (5)	
Elements Parameter	P(C=1) (6)	Probability values range from 0 to 1. Slider increments by 0.01
	SD(1) (7)	Standard deviation of X1 likelihood ranges from 0.1 to 50. Slider increments by 0.1
	SD(1) (8)	Standard deviation of X2 likelihood ranges from 0.1 to 50. Slider increments by 0.1
	SD(Prior) (9)	Standard deviation of prior ranges from 1 to 50. Slider increments by 0.1
	Mean(Prior) (10)	Mean of prior ranges from -40 to 40. Slider increments by 1

Table 4: **One-Dim Discrete Simulation Panel:** Description of UI Elements

Two Dimensional Continuous

This model is used for spatiotemporal causal inference.

For all parameters: Both the boxes and sliders can be used to manipulate values. Sliders can be manip-

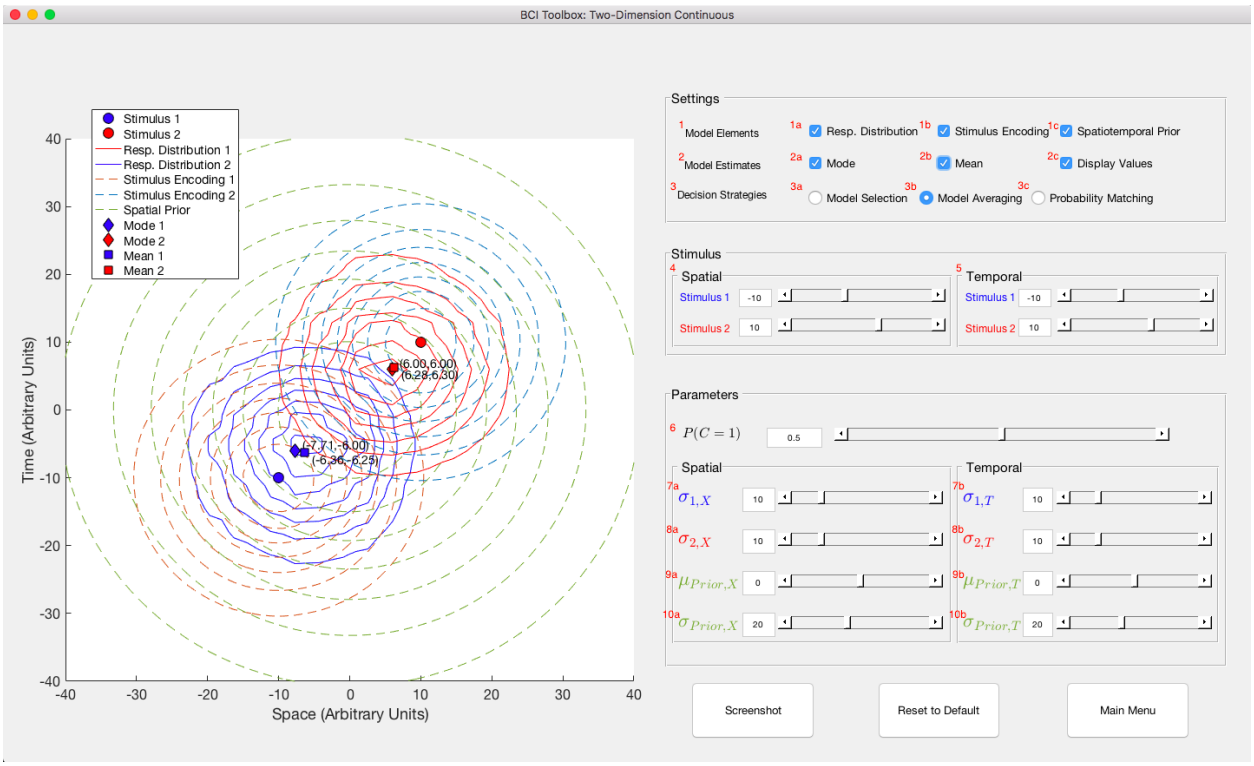

Figure 4: Simulation Panel 3: Two Dimensional Continuous

ulated either by using the arrow keys attached to the left and right or by pressing the spaces to the left or right of the value indicator. The increments of the sliders are given below.

Model Elements (1)	Response Distribution (1a)	Indicated by the solid blue and red lines
	Stimulus Encoding (1b)	Indicated by the dotted blue and red lines
	Spatiotemporal Prior (1c)	Indicated by the dotted green line
Model Estimates (2)	Mode (2a)	Value indicated by red and blue diamonds
	Mean (2b)	Value indicated by red and blue squares
	Display Values (2c)	Shows the value of the model estimate of probability on the figure
Strategies (3)	Selection (2a)	See explanations in "Description" section, under "Simulation Panels"
	Averaging (3b)	
	Matching (3c)	
Stimulus Position	Spatial Dimension (4)	Increasing or decreasing this value changes the location of stimulus 1 and stimulus 2 along the x-axis
	Temporal Dimension (5)	Increasing or decreasing this value changes the location of stimulus 1 and stimulus 2 across the y-axis
Probability	P(C=1) (6)	Probability values range from 0 to 1. Slider increments by 0.01
Spatial	SD_X(1) (7a)	Standard deviation of X1 likelihood ranges from 0.1 to 50. Slider increments by 0.1
	SD_X(2) (8a)	Standard deviation of X1 likelihood ranges from 0.1 to 50. Slider increments by 0.1
	Mean_X(Prior) (9a)	Average of the prior, in terms of space, ranges from -40 to 40. Slider increments by 1
	SD_X(Prior) (10a)	Standard deviation of the prior, in terms of space, ranges from 0.1 to 50. Slider increments by 1
Temporal	SD_T(1) (7b)	Standard deviation of modality 2 likelihood, in terms of time, ranges from 0.1 to 50. Slider increments by 0.1
	SD_T(2) (8b)	Standard deviation of modality 2 likelihood, in terms of time, ranges from 0.1 to 50. Slider increments by 0.1
	Mean_T(Prior) (9b)	Average of the prior, in terms of time, ranges from -40 to 40. Slider increments by 1
	SD_T(Prior) (10b)	Standard deviation of the prior, in terms of time, ranges from 0.1 to 50. Slider increments by 1

Table 5: **Two-Dim Continuous Simulation Panel:** Description of UI Elements

The fitting panel

Noting that the model is typically used to account for experimentally collected data, and would thus be intended to be fit to such data, we secondarily intend to provide our users with the necessary machinery to be able to achieve this stated purpose of the model, given that the data they have collected conforms to our nominal conventions, which are to be specified in the documentation. While there are a great many variants of the model that have been implemented over the years for fitting purposes, we will provide a restricted set of the eight most commonly used parameters and give the user control over which of them to include in the fitting procedure, as well as the three most commonly used decision strategies by which the model estimates are read out.

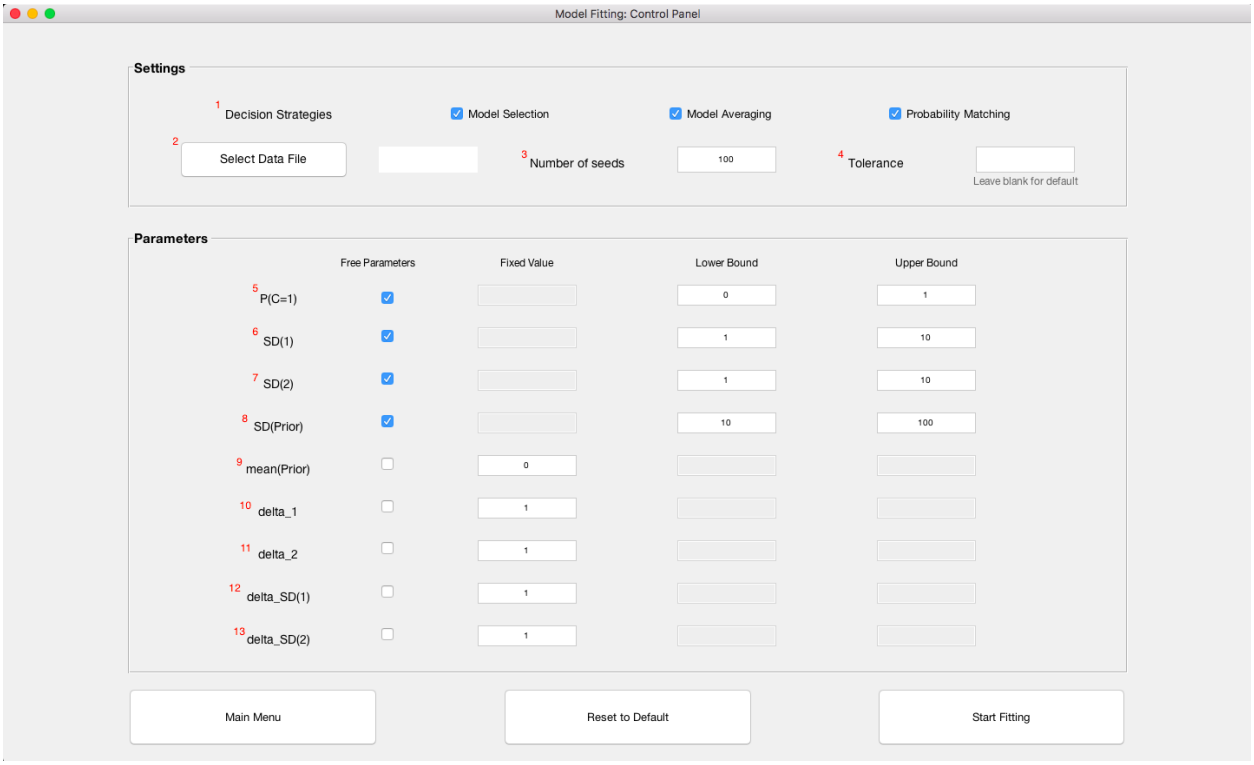

Figure 5: Fitting Panel

Strategies (1)	Selection	For strategy descriptions, see explanations in "Description" section, under "Simulation Panels"
	Averaging	
	Matching	
User Inputs	Subject List (2)	Users can upload their data using a specific layout in a .mat format (see A Note on How to Format Data)
	Number of Seeds (3)	Sets the number of seeds used to analyze user data, increasing this number will require more processing time. Number of seeds will be consistent and independent for all strategies the user runs (i.e. setting this to 100 would use the same 100 seeds for Averaging, Matching, and Selection if all strategies were selected)
	Tolerance (4)	The lower bound on changes in error that the optimizer uses as a criterion for convergence
Parameters The user can designate all parameters as either "Free Parameters" or "Fixed Values". Free parameters will be allowed to vary between a user-specified lower bound and upper bound. Using more free parameters will increase the model fit to the data, however it will increase the processing time required.	P(C=1) (5)	The prior probability that both signals can be attributed to one cause
	SD(X1) (6)	The standard deviation of the Gaussian distribution of the likelihood for modality 1
	SD(X2) (7)	The standard deviation of the Gaussian distribution of the likelihood for modality 2
	SD(Prior) (8)	The standard deviation of the Gaussian distribution of the prior (observers' anticipated location of the stimuli)
	Mean(Prior) (9)	The mean of the Gaussian distribution of the prior
	Delta_X1 (10)	A constant added to the mean of the Gaussian distribution for the likelihood for modality 1
	Delta_X2 (11)	A constant added to the mean of the Gaussian distribution for the likelihood for modality 2
	Delta_SD(X1) (12)	A multiplicative factor that scales the standard deviation of the Gaussian distribution for the likelihood for modality 1 as a function of the space
	Delta_SD(X2) (13)	A multiplicative factor that scales the standard deviation of the Gaussian distribution for the likelihood for modality 2 as a function of the space

Table 6: **Fitting Panel:** Description of UI Elements

Description of Model Fitting Procedure

Our implementation of model fitting relies on the use of the included function `fminsearchbnd.m`, which is based on the built-in Matlab function `fminsearch.m`. Briefly, this function implements the Nelder & Mead Simplex algorithm for derivative-free optimization, that is, for use with objective functions that are difficult or impossible to calculate gradients for. In the case of this toolbox, the objective function is based on the calculation of the negative log likelihood of the data to be fit given the predicted response distribution that is estimated from the model parameters. Given the complexity of the model structure, it is not readily apparent how a gradient of this error quantity with respect to the parameters can be computed, thus making `fminsearchbnd.m` a good choice of optimization algorithm. The difference between `fminsearchbnd.m` and `fminsearch.m` is that the former adds a way to constrain the space within which the algorithm searches for parameters, thus requiring the user to specify these bounds individually for each parameters to be optimized. This is an important and desired property for our purposes because it prevents the optimizer from diverging too wildly in its estimates and helps minimize the variance of the optimized parameters.

As with `fminsearch.m`, `fminsearchbnd.m` requires the user to input initial values from which the optimizer begins its descent down the error hill. These are chosen by random sampling from the uniform distribution encompassing the space defined by the bounds on each parameter. The greater the number of such initial random **Seeds**, as they are called, the less likely that the optimizer will be stuck in a local minimum, and thus the greater will be the modeler's faith that it will converge onto a global optimum. Another factor to consider here is also the criterion on error changes that the optimizer uses to terminate the procedure and consider its parameters converged. This is often referred to as the **Tolerance** and a careful setting of its value can greatly aid in the efficient optimization of a set of parameters. If it is set too low, the optimizer will waste computational resources chasing negligible reductions in error that do not add significant improvements, but in contrast, if the value is too high, the optimizer will terminate very rapidly without generating a satisfactory fit.

A heuristic to employ when setting the **Tolerance** parameter is to let the program first run with the default setting, then to examine the FunVal output in the command window. Place the tolerance within a comparable range to the FunVal values. For example, if the FunVals are around 1000 to 2000, the tolerance can be 1000 for a liberal (i.e. quick) fit, or 100 for a tight (i.e. more accurate, but more time-consuming) fit.

Similarly, the number of **seeds** parameter can be set by establishing the amount of time and computing power available to the user. It is recommended to first run the program with the default setting, and in subsequent trials the user can adjust the number of seeds based upon how fast the program progresses through each seed. For example, if the default setting was 10 seeds and the program requires approximately 1 minute per seed, the user may find this acceptable as the entire run would be complete in 10 minutes. If additional time is available, then setting higher numbers of seeds will provide more accurate model estimates.

A Note on How to Format Data

If researchers wish to use their own data for conducting model fitting using this toolbox, a few very important considerations regarding the format of the data will arise. In order for the built-in model fitting routines to properly function, the data format conventions that we have outlined below must be adhered to. The included file in the toolbox `create_data.mat`, can be examined for a demonstration of how this format is implemented. In brief, the data must be stored as a MATLAB data structure, the required details of which will be explained in more detail in what follows.

Firstly, the data structure variable must be assigned the name "data", and it is critical that the data structure contain the following required fields, which the model fitting routine expects and utilizes for setting up and performing the fitting procedure. (1) A field with the name "N" indicating the number of Monte Carlo samples to take in computing the model estimates – setting this equal to 10,000 will suffice in most cases. (2) A field with the name "space" that contains the discretized continuum upon which the data and model fits will be represented – it is often useful to construct this so as to yield 1 degree of visual angle per discrete unit. (3) A field with the name "stim_locs" that contains the stimulus positions within the space from where stimuli can be presented – note that the first element must be a NaN so as to indicate the possibility of unisensory conditions. (4) A field with the name "conds" whose columns specify all the possible unique stimulus combinations – the two rows represent the stimulus positions for the two modalities, respectively, with a NaN indicating the lack of the presentation of a stimulus from that modality, thus, a

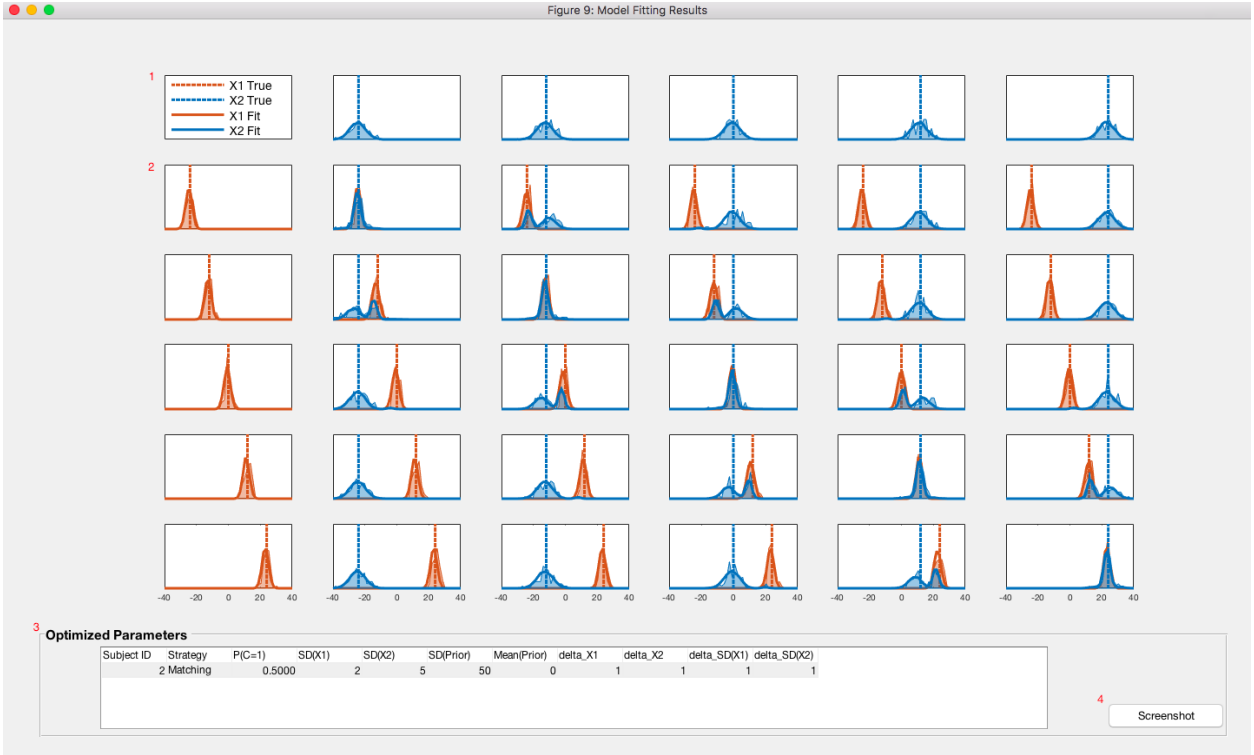

Figure 6: Model Fits and Optimized Parameters

unisensory condition. (5) A field with the name "stim" indicating the order of trials, it is essentially a pseudorandomly generated permutation of the columns of the "conds" field – thus it has as many columns as there were trials in the experiment and two rows for the two modalities that stimuli could be presented from. (6) A field with the name "resp", which has the same dimensions as the "stim" field, but which stores the responses on each of those trials that are indicated therein. (7) A field with the name "cond_resps" and containing an array that stores all the responses sorted into columns according to the conditions indicated by the "conds" field. Thus "cond_resps" has as many columns as "conds", as many rows as there were repetitions for each particular stimulus configuration, and has dimension 2 along the third index separating the data based on whether the responses were from modality 1 or modality 2. (8) A field with the name "subject" that specifies a numeric identifier for the subject that will be used to identify the fitting results.

Legend (1)	X1 True	True position of the first stimulus represented by the red dotted line
	X2 True	True position of the second stimulus represented by the blue dotted line
	X1 Fit	Model estimate for first stimulus location probability represented by solid red line
	X2 Fit	Model estimate for second stimulus location probability represented by solid red line
Plots (2)	Axes	Axes mirror that of the simulation panels for the 1-dimension continuous model, the X corresponding to spatial location ranging from -40 to 40 degrees and the Y corresponding to the normalized probability of perceived location of the stimuli
	Layout	The plots are displayed such that moving horizontally rightward corresponds to shifting the true position of the second stimuli rightward. Conversely moving downwards vertically corresponds to shifting the true position of the first stimuli rightward
Optimized Parameters (3)	Subject ID	The subject number for which the parameters were optimized for
	Strategy	The optimal strategy used for parameters (calculated in model fitting based on the selected strategies in the control panel)
	Parameters	The optimized parameter values
Buttons (4)	Screenshot	The screenshot button will save both the plots as well as the optimization parameters as a 300 dpi .png file (named by user) to the current directory. The button will disappear as the image is being saved and then reappear after the process is complete

Table 7: **Model Fitting Results:** Description of Figure Output

4 Fitting Validation

Method

To demonstrate the standard operation of the fitting procedure that is built-in to the toolbox, as well as to validate its proper functioning, we first created some example data using the `create_data.mat` file that is included. This file utilizes the one dimensional continuous Bayesian causal inference model, supplied with some parameters that the user is able to input. For this simulation, we used the following parameters: $\{p(C = 1) = 0.5,\ \sigma_{X1} = 2,\ \sigma_{X2} = 5,\ \text{prior} \sim \mathcal{N}(0, 15)\}$, and used a strategy of probability matching. The `create_data.mat` file simulated 2030 trials of localization that included both unisensory and bisensory trials. There were five candidate positions where stimuli from both modalities could be presented at, which were separated by 12°, thus providing for 35 possible stimulus configurations ($(5 \times 5 = 25$ bisensory pairs) and $(5+5 = 10$ unisensory stimuli) $= 35$ total). The script runs through a pseudorandomized and balanced order of stimulus configurations and generates simulated response distributions using the supplied parameters. This distribution is then sampled in accordance to its probability distribution in order to generate the dataset.

Next, we ran the model fitting routine in order to observe whether the model would converge on the parameters that we used in generating the simulations. We conducted this using 100 random initial values that we used as seeds to the fitting procedure. These initial values were selected by random sampling using a uniform distribution between the lower and upper bounds. For this test, the bounds used were $\{p(C = 1) \in [0, 1],\ \sigma_{X1} \in [1, 10],\ \sigma_{X2} \in [1, 10],\ \text{prior} \sim \mathcal{N}(\mu \in [-40, 40], \sigma \in [10, 100])\}$. Here, we should hasten to add that we ran this test with all three decision strategies selected. Thus, the optimization routine

will use each randomly generated initial seed three times as it attempts the model fitting for all three strategies in turn, and selects the best fitting set of parameters and decision strategy at the end.

In addition, we set the tolerance on error changes that would be used as a criterion for convergence to 100. Note that since this error is computed as negative log likelihood of the data under the simulated response distribution, the absolute value of the tolerance depends strongly on the number of points that the model attempts to fit and would therefore require modification for the particular dataset at hand. The value of 100 herein was chosen so as to optimally balance speed of fitting with accuracy of fitted parameter values.

Results

The model fitting procedure took 10 minutes to complete on a Macbook (Retina, 15-inch, Mid 2015, 2.8 GHz Intel Core i7, 16 GB 1600 MHz DDR3). In the table below, we report the optimized parameters from the 8 runs of model fitting that we conducted. As can be clearly seen, despite a little variation across runs, the parameters appear to converge satisfactorily onto the true values.

	$\mathbf{p(C=1)}$	$\sigma_{\mathbf{X1}}$	$\sigma_{\mathbf{X2}}$	$\mu_{\mathbf{prior}}$	$\sigma_{\mathbf{prior}}$	**Strategy**	**LL-Error**
True Values	0.5	2	5	0	15	Probability Matching	
Fitting Run 1	0.67	2.00	4.51	0.00	15.73	Probability Matching	7156.1
Fitting Run 2	0.45	2.03	5.21	0.00	14.05	Probability Matching	7136.9
Fitting Run 3	0.40	1.91	5.17	0.00	16.04	Probability Matching	7142.2
Fitting Run 4	0.38	1.96	4.70	0.00	13.68	Probability Matching	7154.7
Fitting Run 5	0.54	2.09	5.71	0.00	14.74	Probability Matching	7156.3
Fitting Run 6	0.45	1.91	4.84	0.00	14.14	Probability Matching	7144.0
Fitting Run 7	0.53	2.11	5.26	0.00	18.02	Probability Matching	7143.2
Fitting Run 8	0.52	2.10	5.04	0.00	15.01	Probability Matching	7130.3

Table 8: **Results:** Optimized Parameter Fits

5 Outlook

Here we present a new computational tool designed for the purpose of aiding researchers to better understand and implement the Bayesian causal inference model as an explanatory framework where they might have data suitable for this purpose. Many of the paradigms of multisensory research are highly amenable to being modeled by this framework and we, therefore, expect this tool to have widespread utility across the field. Aside from its educational function as an intuition-building software package, this tool also provides researchers the ability to generate fits of the model to their own data, gaining insights into the behavior of subjects through the optimized parameters. In the past decade of work on this model, it still remains a complex framework to grasp, and there is often a steep barrier to its utilization by research groups interested in doing so. We, therefore, expect this tool to be of use to researchers in a variety of disciplines such as experimental psychology, computational neuroscience and cognitive science, who may wish to implement it for the study of multisensory phenomena across a wide range of paradigms.

References

Beierholm, U., Kording, K. P., Shams, L., and Ma, W. J. (2009a). Comparing Bayesian models of multisensory cue combination without mandatory integration. In *Advances in neural information processing systems*, volume 20, pages 81–88. MIT Press, Cambridge, MA.

Beierholm, U. R., Quartz, S. R., and Shams, L. (2009b). Bayesian priors are encoded independently from likelihoods in human multisensory perception. *Journal of Vision*, 9(5):23–23.

Ernst, M. O. and Banks, M. S. (2002). Humans integrate visual and haptic information in a statistically optimal fashion. *Nature*, 415(6870):429–433.

Kilteni, K., Maselli, A., Kording, K. P., and Slater, M. (2015). Over my fake body: body ownership illusions for studying the multisensory basis of own-body perception. *Frontiers in Human Neuroscience*, 9.

Kording, K. P., Beierholm, U., Ma, W. J., Quartz, S., Tenenbaum, J. B., and Shams, L. (2007). Causal Inference in Multisensory Perception. *PLoS ONE*, 2(9).

Magnotti, J. F., Ma, W. J., and Beauchamp, M. S. (2013). Causal inference of asynchronous audiovisual speech. *Frontiers in Psychology*, 4:798.

Peters, M. A. K. (2014). *Hierarchical Bayesian Causal Inference and Natural Statistics Explain Heaviness Perception*. University of California, Los Angeles.

Rohe, T. and Noppeney, U. (2015). Cortical Hierarchies Perform Bayesian Causal Inference in Multisensory Perception. *PLoS Biol*, 13(2):e1002073.

Samad, M., Chung, A. J., and Shams, L. (2015). Perception of Body Ownership Is Driven by Bayesian Sensory Inference. *PLoS ONE*, 10(2):e0117178.

Samad, M. and Shams, L. (2016). Visual-Somatotopic Interactions in Spatial Perception. *Neuroreport*, (27):180–185.

Wozny, D. R., Beierholm, U. R., and Shams, L. (2008). Human trimodal perception follows optimal statistical inference. *Journal of Vision*, 8(3):24–24.

Wozny, D. R., Beierholm, U. R., and Shams, L. (2010). Probability Matching as a Computational Strategy Used in Perception. *PLoS Computational Biology*, 6(8).

Wozny, D. R. and Shams, L. (2011). Computational Characterization of Visually Induced Auditory Spatial Adaptation. *Frontiers in Integrative Neuroscience*, 5.