CHAPTER 13

Collecting and Interpreting Judgments about Perceived Simultaneity: A Model-Fitting Tutorial

Kielan Yarrow

1 Introduction

In this chapter, I consider the simultaneity judgement (SJ) as a measure of the relative time perceived between two events, outlining the basic experimental design, the kind of data it generates, and how these data can be interpreted via the parameters of fitted models. After a brief overview of data collection methods, I outline the steps involved in both generating model predictions for plausible observer models and determining a single set of best-fitting model parameters, which maximise the likelihood that the model produced the data. I do so in a way intended to make sense to the competent programmer with limited formal mathematical expertise, making reference to accompanying Matlab code (see book's GitHub repository). Although I will focus on fitting simple detection-theoretic models, I also consider alternative approaches to treating SJ data, and briefly review the ways in which more complex models can be conceived and tested. I subsequently extend my discussion to consider a ternary choice, where participants can indicate either simultaneity or one of two possible orders, and also a novel task requiring a choice about which of two intervals contains the *most* simultaneous stimulus pair.

2 Chapter Architecture

In the following sections, I will discuss various abstract concepts that are often made concrete within a set of accompanying Matlab code (see book's GitHub repository). To facilitate understanding, I have adopted a cross referencing scheme. At various points in the chapter, I include footnotes to code references, which are shown within triangular brackets, e.g., <SimultaneityNoisyCriteria 112>. These indicate that the concept that is being discussed has been implemented in a Matlab function with that name. The number is the line number at which the relevant code begins. Given that I am not a mathematician, and that this chapter is intended to be comprehensible to non-mathematicians,

© KIELAN YARROW, 2018 | DOI 10.1163/9789004280205_014

I have generally avoided including formal equations in the text, except where they seem particularly helpful to assist understanding (or with implementing the ideas that are being discussed).

3 Judging Relative Time

As timing researchers, we are often interested in how observers perceive the timing of events. For example, we might wish to assess how the timing between a brief sound and a flash of light is experienced. There are many ways in which we might operationalise this assessment, but a classic approach is to provide multiple trials containing different relative timings between events (hereafter referred to as *stimulus onset asynchronies* or *SOAs*) and have our observer make a simple judgement on each trial. The temporal order judgement (TOJ; e.g., "which came first") was popular for many years (e.g., Sternberg & Knoll, 1973) but recently the simultaneity or SJ (e.g., "were they simultaneous?") has become increasingly popular. This may reflect the comparative ease with which observers perform these two tasks: Participants tend to say that the TOJ task is harder than the SJ task (Love, Petrini, Cheng, & Pollick, 2013) and also make more errors than one would predict in the TOJ based only on estimates of sensory precision derived from other timing tasks (García-Pérez & Alcalá-Quintana, 2012; Yarrow et al., 2016). Here, I will primarily address the SJ task in considerable detail, but also briefly introduce some variant tasks towards the end of the chapter.

■ Reference García-Pérez & Alcalá-Quintana (2012) is cited in the text but not provided in the reference list. Please check.

4 The Simultaneity Judgement Experiment

The basic SJ design is simple: Present observers with pairs of stimuli on each trial, specifying the SOA between them. Across the experiment, present many different SOAs in a random order, and see how often participants judge each SOA to be simultaneous. However, we will need to define the range of SOAs that will be used, and how often each is presented. A classic approach is to use the method of constant stimuli, in which each possible SOA is presented an equal number of times across the experiment. In this case, we must still select the set of SOAs to test. Given a finite number of trials, there is an inevitable trade-off between the resolution implied by the step sizes we use and the range of SOAs we wish to cover. In the SJ task, it is important to adequately sample both of the transition points from perceived succession to perceived simultaneity. For example, in an audiovisual (AV) task, we need to capture the boundary where observers change from perceiving A then V (i.e., perceived succession) to perceiving synchrony, and also the boundary where they change from perceiving synchrony to perceiving V then A (i.e., back to perceived succession once again, but now in the opposite direction). For many participants, this implies sampling a rather wide range of SOAs. One potential problem is that in order to capture observers who report synchrony over a wide range, we end up sampling many times at extreme SOAs, which may appear trivially non-synchronous to experienced observers.

For this reason, some researchers prefer to use adaptive methods to select the SOA on each trial. For the SJ, these approaches generally attempt to place most trials near to the transition boundaries (from succession to synchrony and back again) while still adequately sampling the regions lying both between them and at the extremes. For example, Yarrow et al. (2013) used an approach loosely based on Rosenberger and Grill (1997) where the distribution from which trials are selected starts off being uniform, but is modified after each decision based on how the participant responds. The aim is to end up with a bimodal distribution that peaks over both transition boundaries. A similar goal can also be achieved in various ways via modified and/or interleaved staircases (e.g., Arnold, Petrie, Gallagher, & Yarrow, 2015; García-Pérez, 2014).

No approach is perfect. The method of constant stimuli can be wasteful, and is likely to establish a Bayesian prior that might bias perception towards the centre of the tested range (Miyazaki, Yamamoto, Uchida, & Kitazawa, 2006).[1] Some adaptive approaches imply a statistical dependency between successive stimuli, which is not really desirable, and the distribution of SOAs is likely to be uneven, and also to vary between conditions, particularly where they induce different biases. Different researchers will weigh these concerns differently. Hence, the only advice on which I suspect all researchers in this area would agree is that some pilot work with the population of interest is crucial before finalising the method for selecting SOAs across trials.

A further issue that should be considered closely before formal data collection begins is the accuracy and precision with which the desired SOAs are being generated by the lab hardware and software. Achieving precise stimulus timing is generally not trivial despite the assumed capacities of modern computers. This chapter is not the place to make a detailed comparison of different rigs and their technical limitations. Instead, I offer some brief advice: Check the timing of your stimuli over a fairly large number of trials using an oscilloscope or some similar method, and *never* assume that your computer is simply doing what you think you told it to do.

[1] This will also be true of adaptive methods, but here the centre of the test range is more likely to conform to a participant's natural bias.

5 Interpreting SJ Data with Observer Models

Having run an experiment as outlined above, on each trial you will typically have: 1) an SOA and 2) a decision (i.e., synchronous or not). Such trial-by-trial data from SJs are commonly summarised as the proportion of times each SOA was judged synchronous.[2] You will then be confronted by a set of data similar to those plotted in Figure 13.1. These data form the basis of a psychometric function, plotting performance (proportion judged synchronous) against the tested SOA. Often, it is helpful to further summarise the data, for example to produce one or more dependent variables for inferential statistical analyses. How should this be achieved? Although I will briefly consider some alternatives in Section 9 of this manuscript, the approach I focus on in the majority of this chapter is the use of parametric observer models to summarise SJ data.

We could summarise data with any mathematical function that looks about right, and indeed this is the approach that has often been taken in characterizing SJ data, with the function of choice being a (truncated and/or vertically rescaled) Gaussian (Stone et al., 2002; Vroomen & Keetels, 2010). However, this arbitrariness comes at a cost. Firstly, if the function has not been derived from any meaningful observer model, there is little reason to believe that it will adequately summarise data in a wide range of situations. Secondly, and relatedly, the parameters of the model will have only a superficial descriptive meaning. By contrast, the parameters of an observer model have meanings that are clearly defined, being tied to the latent processes that have been hypothesised to generate the observations. Furthermore, they can be compared with the same parameters derived when similar observer models are defined and fitted to other tasks (such as the TOJ).

What do I mean by an "observer model"? In short, I mean a model in which a series of well-defined (but often quite abstract) processing steps have been hypothesised to intervene between perception and response. Here, I will be working with fairly simple observer models. They are based on the assumption that each of the two sensory signals to be compared must pass along a neural pathway to a decision hub. The latency with which they do so is considered to be a random variable (i.e., to vary from trial to trial about some mean value

[2] This kind of summary is less useful where each SOA is only sampled once, as sometimes occurs in situations where SOAs have a random component. An example would be experiments comparing the time of an action to an event (where the event is presented around the time that the action is expected to occur, but this cannot be known for certain in advance; e.g., Yarrow, Sverdrup-Stueland, Roseboom, & Arnold, 2013). Another example would be data generated using an adaptive procedure without a fixed step size. Note that although such data will be more difficult to graph, the model-fitting procedures outlined in this chapter will still work perfectly well.

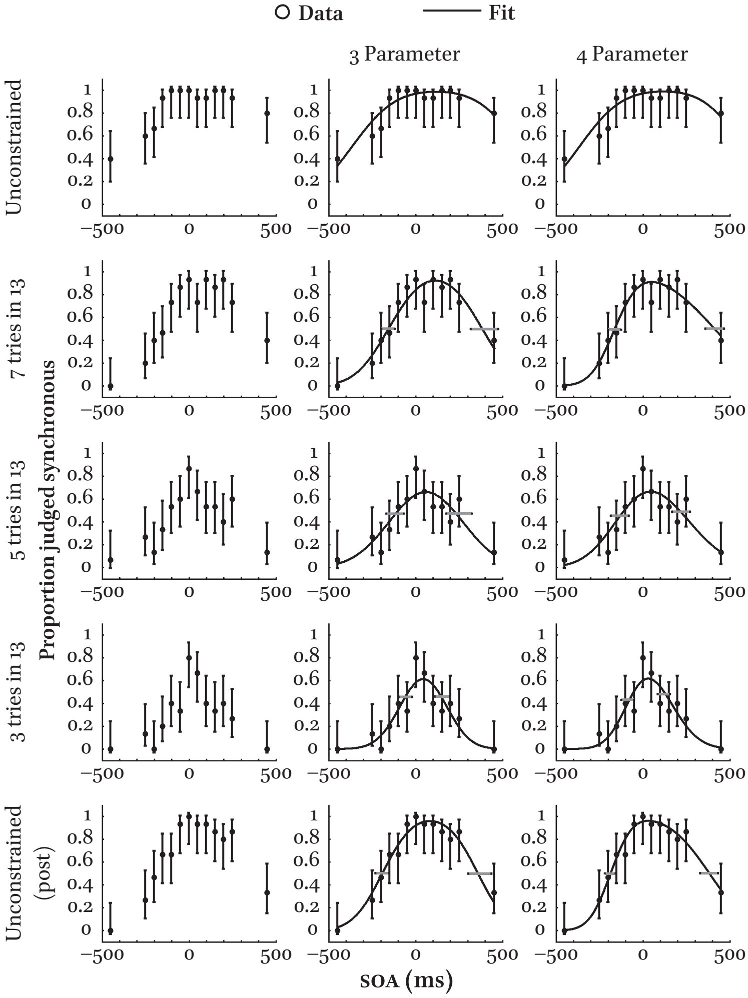

FIGURE 13.1 *Example SJ data. Each row shows data collected under different instructions (see main text). Left column: Data alone. Middle column: Data along with the predictions of a best-fitting three-parameter observer model. Right column: Data fitted with a four-parameter observer model, allowing asymmetry (i.e., a varied slope on each side of the psychometric function). Vertical error bars show 95% binomial confidence intervals on the data. Horizontal (grey) error bars surround estimates of two model parameters, and show 95% bootstrap confidence intervals. These two parameters represent the transition points from judgements of simultaneity to judgements of succession (or vice versa).*

according to a known distribution). The decision hub receives both signals, and, thus, has access to the subjective difference in arrival times between them (Δt), which is corrupted by their latency noise. Hence, Δt is also a random variable, with a distribution that depends on its two contributors. If they are each distributed in a Normal/Gaussian way, Δt is also Gaussian, with a variance equal to the sum of the two contributing signals' variances. The observer then interprets Δt on each trial by placing decision criteria, typically one below and one above true synchrony, so that values that fall between them can be judged synchronous. These ideas are illustrated schematically in Figure 13.2. If, after reviewing it, you find that you are struggling with these concepts, I would suggest that you find out about the basics of signal detection theory, for example in Macmillan and Creelman (2005), before studying this chapter again.

If the two decision criteria that an observer uses to define simultaneity can be held perfectly constant across trials, this model predicts a psychometric function that is the *difference of two cumulative Gaussians*, each having the same variance but a different mean (Schneider & Bavelier, 2003). Hence both cumulative Gaussians can be described using just three parameters:

$$P\text{ ``simultaneous''} = \Phi\left(C_{\text{High}}, \text{SOA}, \sigma\right) - \Phi\left(C_{\text{Low}}, \text{SOA}, \sigma\right)^{3} \qquad (1)$$

where Φ is the normal cumulative density function. The two means represent the positions of the decision criteria (C_{Low} and C_{High}) on the SOA axis, and the single standard deviation (σ) shared by both represents the variability in Δt.

What about if we doubt that our observer can hold their decision criteria perfectly constant across trials? If the positions of the two criteria are additionally considered to be Gaussian random variables (Yarrow, Jahn, Durant, & Arnold, 2011), the psychometric function becomes the difference of two cumulative Gaussians with different means and variances:

$$P\text{ ``sim.''} = \Phi\left(C_{\text{High}}, \text{SOA}, \sigma_{\text{High}}\right) - \Phi\left(C_{\text{Low}}, \text{SOA}, \sigma_{\text{Low}}\right)^{4} \qquad (2)$$

[3] <SimultaneityNoisyCriteria 12–22>.

[4] <SimultaneityNoisyCriteria 23–33>. Note that this formula is an approximation and will break down if the criteria are close together and one is noisier than the other (because the cumulative Gaussians will overlap, producing negative predictions for judgements of simultaneity). One possible fix is to implement a simulation in such cases and assume that when noise in the criteria makes their order illogical, observers default to just using the less noisy criterion <SimultaneityNoisyCriteria 46>.

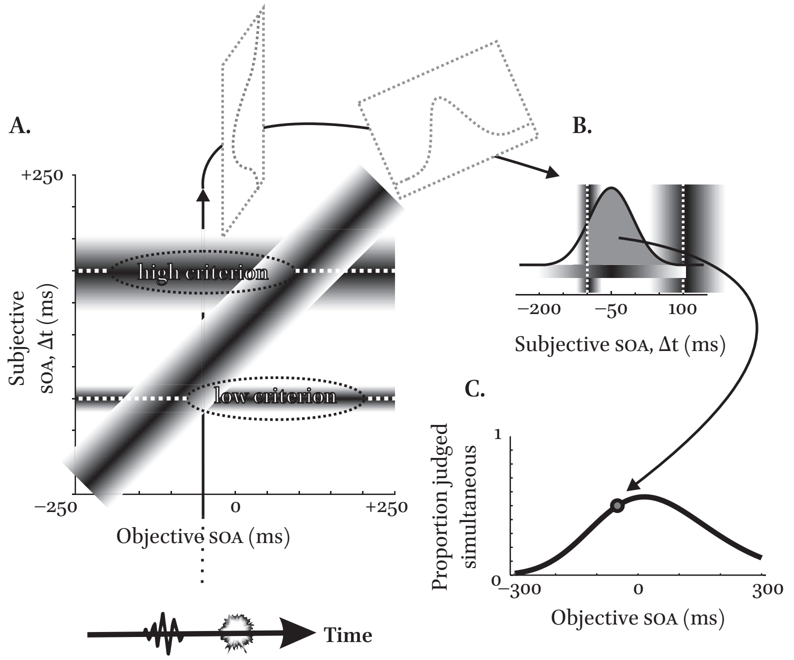

FIGURE 13.2 *Schematic of the four-parameter observer model outlined in the main text. A. Each SOA is presented many times. Each trial yields a noisy internal response (Δt, the subjective SOA). Hence the relationship between objective and subjective SOAs is depicted as linear (and in this case unbiased) but shading is used to denote the likelihood of each Δt value (darker shading denoting higher probabilities). Cutting vertically through this function for any given objective SOA yields the Gaussian distribution of resulting Δt values across trials. B. This probability density function (PDF) is shown for a −50 ms SOA. An observer will judge the trial synchronous when Δt falls between two decision criteria (shaded region between white dashed lines). The area under a PDF (to the left of some point) is given by a cumulative density function, so the shaded region is estimated as the difference of two cumulative Gaussians, one integrating all the way to the rightmost criterion, the other integrating only to the leftmost one. Shading around the criteria denotes additional criterion noise; criterion likelihood is highest where the shading is darkest. C. Resulting psychometric function (asymmetrical, due to differential criterion noise) with the point defined by the shaded region in Part B highlighted. Other points on the function are similarly obtained by integrating the region between the two criteria that is obtained at different SOAs.*

Hence, four parameters are needed to describe the predictions from this model. The means retain the same interpretation as before, corresponding to the mean positions of the decision criteria for synchrony, while the variance of each cumulative Gaussian now represents the sum of two sources of variability. Both cumulative Gaussians contain the variance in Δt, but each also uniquely contains the variance in the placement of the corresponding criterion. To expand slightly: Psychologically, we are now envisaging several contributing sources of noise – from variability in signal transmission times, and from variability in the placement of two decision boundaries. However, when we implement the model mathematically it becomes apparent that these psychological constructs are degenerate should we attempt to have a free parameter for each one.[5] To make it possible to recover model parameters (as outlined later, in Section 6) we must create composite parameters (σ_{Low} and σ_{High}), which can vary independently from each other, and from all other model parameters, but represent a somewhat complex combination of different theoretical sources of sensory/decision noise.

These kinds of models have been developed by several authors for different tasks (Allan, 1975; Baron, 1969; Gibbon & Rutschmann, 1969; Schneider & Bavelier, 2003; Sternberg & Knoll, 1973; Ulrich, 1987; Yarrow et al., 2011). It is worth noting that the exact processes that lead Δt to be a Gaussian random variable (i.e., independent *latency noise* in the two signals) can be incorrect without invalidating this whole approach. Consider that this kind of model also leads to the prediction of a (single) cumulative Gaussian psychometric function for TOJs. However, the cumulative Gaussian function is actually used very widely in psychophysics (whenever a judgement is made that effectively divides a continuous decision variable into two halves). This is because all that is really being assumed for this fit to be sensible is that the internal response that informs the decision (here Δt, but in other applications contrast, intensity, orientation, or whatever) has *in some way* accumulated Gaussian noise. The central limit theorem[6] of classical probability theory makes this a fairly appealing conjecture for many sensory domains, regardless of the exact processing steps that might precede a sensory judgement.

So far, I have described an observer model with two variants. The first, with three parameters, produces a symmetric psychometric function for SJs. The second uses four parameters and can additionally capture an asymmetry in the data. I will comment on formal methods for model selection in a later section. For now, I want to discuss the meaning of the model parameters, partly just to help make the models more interpretable, and partly in order to clarify what I

[5] By “degenerate”, I mean that such parameters could trade off perfectly with one another, such that different combinations lead to the exact same model prediction. This creates a problem in model fitting known as *non-identifiability*.

[6] If you take a large number of random variables and add them together, the distribution of their sum will be Gaussian (even if the contributing variables are not).

think are some misconceptions that have arisen out of the common decision to fit an arbitrary function (the Gaussian) to SJ data.

To get us going, consider the first column of Figure 13.1. These data come from five blocks (of 195 trials each) of an SJ experiment, performed by a novice observer (using the method of constant stimuli, and evaluating synchrony between an LED flash and a 1000 Hz beep, both 10 ms in duration). In the first block, they were simply told to report simultaneity if that is what they perceived. In the second, third and fourth blocks, they were instead told to try and successfully guess the stimuli that were truly simultaneous, but given a maximum number of attempts (7, 5, and 3 in every set of 13 trials) on which they could make use of the "simultaneous" response option. In the fifth block, the original (standard) simultaneity instruction was repeated.

What effect did these altered instructions have on the psychometric function? When unconstrained, this participant, like many others I have tested, made extensive use of the synchronous response, so that they reported synchrony almost 100% of the time across a range of SOAs. If these data were fitted with a Gaussian, we might be tempted to interpret its standard deviation (or some linear transform of this value, such as the full width half height) as a measure of sensitivity to asynchrony. We might further be tempted to consider this parameter equivalent to the slope of the fitted function (or its inverse, the just noticeable difference) in a different task, like the TOJ.

However, consider what happens when instructions require the participant to be more conservative with their use of the synchronous response (Figure 13.1 rows 2–4). If they were simply insensitive across the range of SOAs that they originally reported as synchronous (see row 1) they would still perceive synchrony across this full range. Any constraint on the number of "synchronous" responses that could be made would yield a psychometric function with a flat plateau across this range, but with a ceiling at a proportion lower than 1.0 (because the limited responses would now have to be shared out at random across this region). This is not what occurs. Instead, the synchronous responses increasingly cluster close to true synchrony. The observer model I have outlined in this chapter describes these patterns of data quite naturally, as the result of *changing decision criteria.* Initially, the participant adopts quite loose criteria regarding what is synchronous, but the instructions force them to adaptively tighten them up in response to task requirements.[7] Fits are shown in the

[7] It is easy to envisage other situations that might alter a participant's decision strategy, for example the speed with which they are expected to respond, or their beliefs about the proportion of stimuli that are actually simultaneous.

middle column for the three-parameter observer model I outlined earlier. The first two model parameters capture the position of the decision criteria.

If the width of the SJ function is a poor measure of sensitivity, what is a good measure? The answer is the *slope* of the function (on either side), which, under the three-parameter observer model, is determined directly by sensory noise in the Δt distribution. This measure remains rather similar down the rows of Figure 13.1, consistent with our expectation that a change in instructions has not somehow radically adjusted the participant's levels of sensory precision. The σ parameter of the SJ function, when implemented as I have described (as the difference of two cumulative Gaussians), is exactly equivalent to the σ parameter of a sigmoidal psychometric function applied to TOJ data (when considered as the prediction of the same model). Note that fitting an arbitrary Gaussian provides no such way of dissociating the width of the SJ function from the steepness of the SJ function. The practical importance of this limitation is up for debate (it may, for example, be the case that slope and width of the SJ function are typically highly correlated, perhaps because noisy observers tend to choose liberal criteria; c.f. Magnotti, Ma, & Beauchamp, 2013). However, conceptually this distinction seems a sensible one to maintain, being closely related to the distinction between d-prime (d') and c made famous in classical signal detection theory (Green & Swets, 1966).

I have also discussed a four-parameter model, which is fitted in the right-hand column of Figure 13.1. Here, the asymmetry in the function arises from unequal variance in the placement of the two decision criteria (low and high) over the trials of the experiment. These sources of variance would sum with sensory noise in the Δt distribution, uniquely at each decision boundary. If we believe that criterion noise may be present, it complicates our interpretation slightly, because we can never fully separate criterion noise from sensory noise in order to determine the magnitude of either one. All we can do is place an upper limit on sensory noise (being the smaller of the σ^2 estimates associated with the two sides of the SJ function). Note that this conflation of sensory noise and criterion noise applies equally to the interpretation of sigmoidal functions in TOJ and other (non-temporal) tasks: If both kinds of noise are assumed to be Gaussian, only their sum can be estimated from a psychometric function.

Before I conclude this section, it is important to consider a measure that is often derived in timing studies which I have not yet touched upon: The point of subjective simultaneity or PSS. Classically, this measure is estimated from TOJ tasks, being the SOA at which the two order responses are equally likely (implying maximum uncertainty about stimulus order). An analysis couched in terms of the kinds of observer model I have described here illustrates that this SOA represents the combination of a sensory bias (for example, if stimuli from one modality in an AV task must travel a shorter neural pathway than those from the other to reach the decision hub) and a decision bias (in placing a criterion to demarcate the two order responses; Sternberg & Knoll, 1973).

In the SJ task, a similar ambiguity is present (Yarrow et al., 2011). If a sensory bias exists, it seems plausible that the two criteria demarcating synchrony from asynchrony would simply be placed at equal distances from "subjective zero." In that case, we can simply average them to recover a single PSS with a purely sensory interpretation. Note, however, that such an "equal distance" assumption may not be applicable in many situations, particularly if the two stimuli are substantially different from one another. For example, two stimuli may persist to different extents within the brain, and this might influence how decision criteria are set (e.g., "I will call them simultaneous if I experience no gap between them"). A weaker claim would be that the PSS is very likely to lie *somewhere between* the two criteria. My personal preference is generally to report the two criteria, which may in any case provide greater insights about changes across conditions than a single inferred PSS (Yarrow et al., 2013; Yarrow et al., 2011). However, reviewers often request (quite reasonably) that the PSS also be reported for easier comparison with the previous literature. In this case, I would suggest that averaging the criteria is a good compromise.

6 Fitting Models to SJ Data

So far, I have alluded to the general notion of fitting observer models to SJ data in order to derive meaningful parameters, and described two variants of what I believe to be a sensible observer model for this purpose. If you are happy that the observer models I have suggested serve your experimental needs, then you may already have most of what you need from this chapter, because the Matlab code to fit these models is available.[8] With an intuitive understanding of the models, you can fit them and interpret their parameters (rather like having only an intuition about the maths underlying ANOVA is more than sufficient to apply this statistical tool). However, it is possible that you may want to fit variants of the models I have outlined (e.g., Yarrow, Minaei, & Arnold, 2015) or other models entirely, or that you are simply inquisitive about how models are fitted to data and wish for a deeper understanding of this process. In this section, I will provide a whistle-stop tour. Realistically, I can only touch on the topic of model fitting. If you would like to know more, I would highly

[8] <SimultaneityDiffCumGaussMultistart> and <SimultaneityNoisyCriteriaMultistart>.

recommend that you take a look at Lewandowsky and Farrell (2010), an excellent and readable book on this topic from which much of what I will say has been gleaned. The Matlab functions provided as part of this chapter owe a large debt to the structure that Lewandowsky and Farrell introduce and the examples that they provide in their code snippets. Other very useful sources for what follows are the now classic papers by Wichmann and Hill (2001a, b) on fitting sigmoidal psychometric functions, and Myung's (2003) short tutorial on maximum likelihood estimation.

6.1 *Introduction to Model Fitting: Regression*

If you are attempting a chapter like this, I am going to assume that you are somewhat familiar with simple linear regression, so I will start there. You will recall that regression fits a straight-line model with two (or more) parameters to data. The parameters, for simple regression, are the slope (s) and intercept (c) of the line defined by the function $y = sx + c$. I am going to begin with an even simpler model, where c is fixed to zero. Hence, the model I am working with has just one free parameter (s) and the model's predictions are captured in the equation $y = sx$.

Data for a regression-style problem come in the form of a vector of values of x ($\mathbf{x}$) and the associated vector of values of y ($\mathbf{y}$) so that $x_{1 \ldots n}$ and $y_{1 \ldots n}$ are matched pairs, for example the height and weight of a set of n participants. At this point I have a set of data and a parametric model with predictions defined by an equation. How should I go about finding the value of my parameter s, which maximises the fit of the model to the data? For this, I need to consider something called the discrepancy function. In the case of regression, the discrepancy function is based on summed squared error. If I pick a value of s, I can use my model equation $y = sx$ to find a prediction (about y) for each value of x in my data set. Then I can look at the actual value of y associated with each value of x in the dataset. Finally, I can subtract each predicted y from the corresponding y in the data, square this difference, and sum these values up to produce the summed squared error for the model (associated specifically with the particular value of s that I have just picked and tested). If I were to repeat this process with many values of s (for example stepping up from $s = 0$ to $s = 2$ in small increments) I could save each error value and plot out a discrepancy function, showing how discrepant the model predictions are from the data for different values of s. What I want to find is the minimum of this function, because that will be the value of s that provides the best fit of model to data. These ideas are illustrated in Figure 13.3.

For my reduced regression problem (or indeed for much more realistic and complex linear regression problems) I wouldn't actually bother to generate a

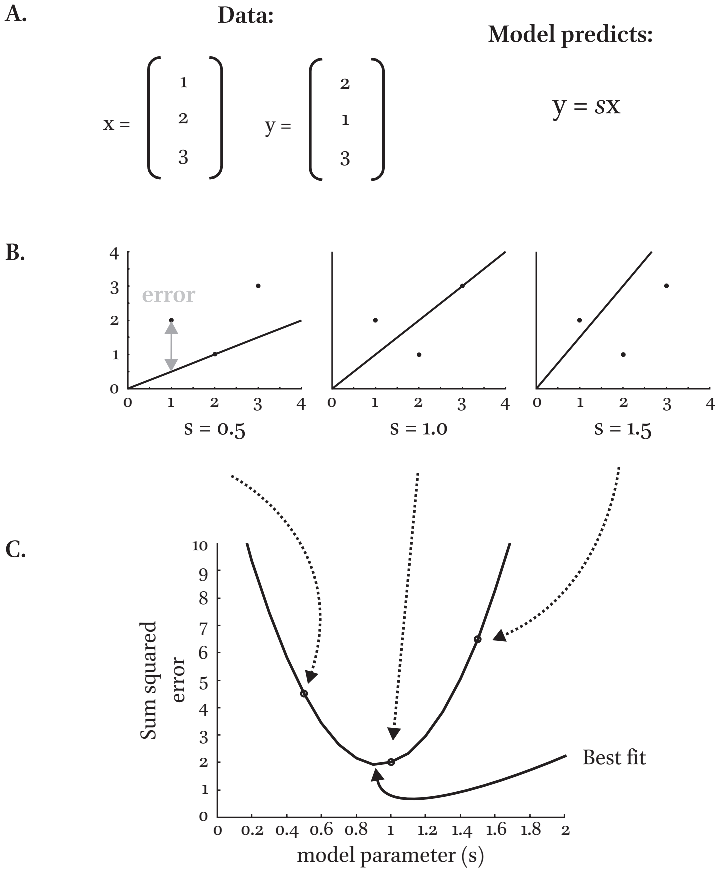

FIGURE 13.3 *Schematic of process for the generation of a discrepancy function. Here, a toy regression problem is illustrated. A. Three data points are shown, along with the equation that captures the model's predictions. B. The model's parameter,* s, *is varied. For each value of* s, *error is determined as the distance between the model prediction and each data point. C. To provide a metric of model fit, errors for each data point are summed and squared. For example, when* s *= 0.5, errors are 1.5, 0, and 1.5 (for the three data points), so squared errors are 2.25, 0 and 2.25, and the sum of squared error (SSE) = 4.5. Calculating SSE for all values of the model parameter* s *allows us to plot a discrepancy function. The best-fitting model parameter is the value of* s *that minimises this function.*

discrepancy function in this tedious iterative manner. For the toy example, I could just about crunch through the necessary maths, using calculus, in order to reach an analytic solution (by first deriving a formal expression for the discrepancy function, then differentiating it to find its slope, and finally setting this derivative to zero to find the minimum). For a more realistic regression problem, I could thank my lucky stars that competent mathematicians have already derived analytic solutions, and simply plug my data into those to find the best-fitting parameters in a single step. However, in the case of the SJ models that are our main interest here, we will actually end up doing something nearly as crude as the iterative search I have outlined above, because finding an expression for the discrepancy function in terms of the model parameters is not as trivial as just looking it up in a statistics textbook. Before getting to that, however, we need to touch on another important concept that is required when we move to fitting SJs: Maximum likelihood estimation.

6.2 *Maximum Likelihood Estimation and the Binomial Data Model*

Summed squared error (or the equivalent mean squared error) is an intuitive measure of model fit. We can clearly visualise how a model fits poorly if its predictions fall at a greater distance from the data. Furthermore, the squaring operation seems a sensible way to punish both positive and negative "prediction errors" (rather than having them cancel each other out when we sum over data points). However, this goodness-of-fit statistic is not generally applicable. Rather, it is a special case of a more generally applicable metric (with summed squared error giving the same answer when data are distributed normally and with equal variance at each level of prediction).[9] In general, to find best-fitting parameters, what we want to do is to find model parameters, which *maximise the likelihood* that the model at hand generated the data (known as maximum likelihood estimation or MLE).

Recall that for linear regression, we attempted to find parameters that minimised the summed squared error. In order to do so, we first had to be able to measure the summed squared error obtained with a particular parameter value. Analogously, in order to find a fit that maximises likelihood, we first need to able to measure the likelihood that a model generated our data given particular parameter values. With regression, we broke this process down by measuring error at each data point (and then squaring and summing them). With MLE, we can also begin at the level of a single data point.

In fact, we will begin by considering a single data point and a model with a single parameter. In doing so, we are (almost inadvertently) introducing an important concept in MLE fitting – the *data model*. The data model is our best guess about the statistical process that makes our data noisy. In the case of regression, the data model is Gaussian. We assume that our measurements are being corrupted by Gaussian noise. However, for synchrony judgements this would be the wrong data model. In an SJ experiment, the observer can

[9] Which you will probably recognise as one of the assumptions for linear regression.

only select one of two options on any given trial. The observer model, which I described earlier generates predictions about the probability with which they will say "simultaneous" at each SOA. Hence, at each SOA, our experiment can be considered a *Bernoulli process* (like repeatedly flipping a coin, with a particular probability of coming up heads). When you sum the number of times one or other outcome is obtained from a Bernoulli process across a set number of trials, you get a *binomial distribution*. Hence, for an SJ experiment, at each SOA we expect our data to follow a binomial distribution, with a probability parameter that can be predicted by our observer model. For binomial data (denoted *X* here) the probability of getting exactly *k* successes (e.g. heads, or "synchronous" decisions) in *n* trials with a probability of success of *p* is:

$$p(X=k)=\binom{n}{k}p^{k}(1-p)^{n-k} \tag{3}$$ [10]

where:

$$\binom{n}{k}=\frac{n!}{k!(n-k)!} \tag{4}$$

Now we are ready to appreciate what it means to measure the likelihood that model *a* with parameter *p* generated data point X. Make model *a* a coin toss with parameter p = 0.5 (a fair coin). If data point X showed 7 heads out of n = 10, we can make a precise quantitative statement about how likely it is that this model generated those data. We do this by plugging the numbers into the formula for the binomial distribution above. The answer, as it happens, is 0.1172. If we adjusted the probability parameter *p* of our binomial distribution to 0.1, you will probably guess that the model with this parameter will not do such a good job of predicting our data, and indeed the calculation returns a value of only 0.00008. What this number is telling us is that given the data we observed, it is rather unlikely that it was generated from a binomial distribution with a probability parameter of 0.1. On the other hand, with a probability parameter of 0.7 (i.e. where the prediction looks very much like the data) we obtain a likelihood of 0.2668, because this combination of model and parameter value is much more likely.

What I have just described is exactly what we need to do at each data point (corresponding to each tested SOA) when we conduct an MLE fit to SJ data.

[10] <BinomialLikelihood 9>.

We need to take the probability predicted by our model, and use it, along with the number of synchronous responses and the number of trials at that SOA, to obtain the likelihood of obtaining those data given binomially distributed data with the predicted probability. However, we still need to scale this calculation up in two ways. Firstly, we need to make this assessment for all data points, as I outline next, in order to generate the likelihood that the model generated the *complete* data set. Secondly, we need to perform this whole evaluation repeatedly, for the sets of probability values predicted by our observer model as we change that model's parameters. In this way, we can create a likelihood function that can be used as a discrepancy function. I outline that process in the next section.

How do we move from the likelihood that this predicted probability yielded this many synchronous judgements out of this many trials (i.e., a single prediction and a single data point) to the likelihood that a full set of predicted probabilities (one per SOA) gave rise to a full set of data points? We need to perform the calculation at each data point and then multiply the obtained probabilities together (because the probability of several independent events all occurring is simply their product). However, there are practical reasons for doing this in a slightly different way, not least the fact that when you multiply lots of probabilities together you soon end up with a very small number indeed, which can be tough for computers to represent. You may or may not recall that the logarithm of a product of one or more numbers equals the sum of the logarithm of each. Hence it is standard practice to calculate *log* probabilities, and *sum* them across data points.[11] We could then convert this back to a probability for the overall prediction, but given that we are looking for the maximum likelihood value, and log likelihood increases monotonically with increasing likelihood, it's more typical to simply carry on working with the log-likelihood values when we search for a best-fitting set of model parameters.[12]

[11] E.g., <SimultaneityNoisyCriteriaWrapperForFMin 103–113>.

[12] Actually, working with log likelihoods derived from the log of the equation for a binomial distribution imposes an unnecessary computational burden, because one of the terms ("N choose K") depends only on the data, not on the model's predictions, so will never vary as model parameters are changed. Hence it is not going to be relevant to finding the best-fitting parameters. In practice, we therefore tend to drop this term to speed things up. This is known as a "kernel" log-likelihood calculation. Using a kernel won't matter at all if you limit yourself to making comparisons between models fitted to the same data using the same kernel, but it will matter if you want to interpret the absolute value of log likelihood (or likelihood) for the best-fitting parameters. Fortunately, we don't generally need to do that. See e.g., <SimultaneityNoisyCriteriaWrapperForFmin 121–124>.

Before I move on to this process, I want to briefly touch upon one important feature of likelihood as a metric of goodness of fit: It is very sensitive to deviations from extreme predictions. What I mean by this is that if a model predicts probabilities of virtually zero or virtually 1 (as observer models often do) the likelihood of observing even a single trial at odds with this prediction is vanishingly small. What this means in practice is that a single lapse by an observer (say, pressing the wrong button by mistake) will have a disproportionately large effect on the resulting model fit (Wichmann & Hill, 2001a). For this reason, it is often worth considering incorporating a "lapse rate" parameter into our observer models. However, extra parameters are generally undesirable for various reasons, so a compromise position is to simply fix a small but reasonable lapse rate, which then adjusts model predictions at extreme SOAs. The code that accompanies this chapter incorporates a fixed lapse rate of 1%.[13] Essentially, model predictions are tweaked slightly to range from a very small probability of saying simultaneous to a very high probability of doing so without ever getting as low as 0 or as high as 1.

6.3 *Finding Best-Fitting Parameters*

So far, I have tried to explain how we determine the (log) likelihood that a set of model predictions (i.e. predicted probabilities at each SOA) generated a set of corresponding data. However, the probability of saying "synchronous" that an observer model predicts at each SOA depends on the parameters fed into the model. What we want is the set of parameters that generates the set of predicted probabilities that maximise the likelihood that the model generated the data. One very labour-intensive way to go about finding them would be to iteratively modify each parameter at all levels of the other parameters, determine log likelihood, and repeat to sample the entire parameter space. This approach, known as a grid search, is very similar to the one I outlined in my toy regression example. That model had just one parameter, and hence generated a discrepancy function that could be visualised in two dimensions (Figure 13.3). If my observer model had just one parameter, I could do something very similar and generate a log-likelihood function amenable to a 2D plot. The main difference would be that I would be looking for the maximum, rather than the minimum, of this function.

Even with a single parameter, this approach is slow, particularly if you want a high-resolution search and have little idea about the range within which your best-fitting parameter lies. However, with two parameters, it is necessary to iterate through all reasonable values of one parameter *at each* reasonable value

[13] E.g., <SimultaneityNoisyCriteriaWrapperForFmin 93–94>.

of the other (i.e., all combinations of two parameters). I then end up with a discrepancy *surface* that must be visualised in 3D. With more parameters, my discrepancy function becomes very hard to visualise and, more importantly, the number of points that must be searched in a grid search grows exponentially. Hence a grid search is not very practical for the observer models I discuss here, with three and four parameters. Fortunately, many algorithms exist to help with searches of this kind. The most famous is the Nelder-Mead simplex search (Nelder & Mead, 1965).

The intuition for this approach is simple. Set a starting point (i.e. a reasonable guess for each parameter) and establish error of fit. Then, test a few more points in the vicinity to find their errors. Apply some geometric rules to try and figure out the slope of the discrepancy surface in this small region. Then, crawl down this slope, testing new parameter combinations as you go and re-applying the rules, in order to move towards a minimum. For a log-likelihood search, the approach needs to be tweaked slightly before the simplex algorithm will work, because we are seeking a maximum, not a minimum. However, simply inverting the obtained log-likelihoods is sufficient.[14] The set of rules embedded in the algorithm will then guide it to a best-fitting solution without having to sample the discrepancy surface exhaustively.[15]

You don't really need to know any more than that to perform a simplex search, as functions to implement it are readily available. However, you might want to find out a bit more in order to appropriately set the various options that these functions let you vary. One important fact to bear in mind is that a simplex search may struggle when the discrepancy surface is not smooth and well behaved. In particular, if there are *local minima* that vary from the *global minimum*, the simplex is likely to home in on the local minimum in the region where it started to search and get stuck there. A good sanity check for any search procedure is to use the model that is to be fitted to generate some data (based on a known combination of parameters) and then see if the fitting procedure recovers the model parameters successfully when initiated from various different start positions. In my experience, simplex searches often fail in this regard. In an attempt to overcome this issue, the code associated with this chapter actually combines grid-search and simplex-search approaches, by initiating a separate simplex search from each parameter combination defined by a grid search.[16]

[14] E.g., <SimultaneityNoisyCriteriaWrapperForFmin 126 & 47>.

[15] E.g., <SimultaneityNoisyCriteriaWrapperForFmin 44>.

[16] E.g., <SimultaneityNoisyCriteriaMultistart 138–182>.

6.4 *Confidence Intervals around Model Parameters*

For comparisons involving groups of participants, recovering a set of best-fitting parameters for each participant in each condition is usually sufficient, and the standard error can then be computed across the sample (usually as an implicit part of common approaches to statistical inference like ANOVA). However, sometimes we wish to have an idea about how well parameters are being estimated *for each participant*. I will outline a couple of popular approaches.

Firstly, we can make use of a result from asymptotic statistical theory (i.e., theory that is true when our sample size in infinite), which (basically) tells us that there is a close relationship between the curvature of the log-likelihood surface at the point where we obtained the best-fitting parameters and the standard errors of those parameters. The intuition is that if changing a parameter by just a little bit makes the fit a lot worse, the parameter is tightly constrained and probably well estimated. In this situation, the point of best fit effectively sits in a steep-sided hole on the (negative) log-likelihood discrepancy surface (hence curvature is high). Formally, the curvature of the log-likelihood surface is captured by something called the Hessian matrix (a matrix of second-order partial derivatives). We can't work that out exactly without (at least) an analytic expression for the discrepancy function, and we don't have one for the kinds of observer model I have described here. However, we can approximate the Hessian using numerical methods (by measuring changes in log-likelihood in the best-fitting region via a series of small steps). Having done so, the inverse of the Hessian provides a covariance matrix for the model's best-fitting parameters, and the main diagonal values can, thus, be square rooted to estimate standard errors (which can then be straightforwardly converted to confidence intervals).[17]

It's questionable whether results from asymptotic statistical theory are actually going to hold for psychophysics experiments with fairly low numbers of data points and trials (Wichmann & Hill, 2001a). Hence a popular alternative approach is to estimate confidence intervals via bootstrapping. Bootstrapping theory, described in detail in Efron and Tibshirani (1994), tells us (roughly) that if we resample from a data set repeatedly *with replacement* to generate a new "bootstrap" data set of the same size, calculate a statistic of interest, and then repeat many times, the resulting distribution will allow us to make inferences about the standard error of that statistic. In the case of SJ models, a reasonable approach is to resample the data (known as non-parametric bootstrapping), fit the model to each resample, and record the parameters on each of around

[17] E.g., <SimultaneityNoisyCriteriaWrapperForFmin 53–63> and <SimultaneityNoisyCriteriaMultistart 219–224>.

1999 such iterations to form parameter distributions. If these distributions are symmetric, we can pretty much just read values straight out of them to form confidence intervals (e.g., the 50th and 1950th values out of 1999 will give us a roughly 95% confidence interval). If they are not, we must do something more complicated, with the best choice being the bias-corrected and accelerated (BCa) approach. Because of the large number of fits that are required, bootstrapping is fairly slow. If the experiment contains many trials, the BCa method makes it even slower (because it incorporates additional "jackknife" resampling, implying one further fitting iteration for almost every trial).[18]

The code accompanying this chapter offers options to generate confidence intervals on fitted parameters. Confidence intervals sometimes imply statistical inference, as for example when they fail to overlap some value and thus imply that our statistic differs significantly from that value. However, in SJ experiments we are more likely to want to ask a question such as whether a particular parameter differs between two conditions for a single observer. To answer this kind of question, you will need to modify or develop the code. If we take the example of whether parameters vary across conditions, my recommendation would be to adopt a permutation test approach.

To do so, take the trials from both conditions and think of each trial as a card in a deck of cards. Making sure you keep each trial intact (i.e., without breaking the link between SOAs and responses) shuffle the trials and then deal them at random into two new piles, each representing a pseudo-condition. If your original conditions contained different numbers of trials, make sure the two pseudo-conditions match the size of the original conditions. For each pseudo-condition, perform a model fit. Now calculate the difference between model parameters in the two pseudo-conditions. This is the value you want to retain. Now repeat this whole process many times. What you are forming is a null distribution of the expected difference between model parameters that would occur just by chance. You can then compare the difference you actually obtained against this null distribution to generate a p value for your difference of interest.

7 Variants of SJ Observer Models

In this chapter, I have presented two variants of a latency-based observer model applied to the SJ task. Both assume that a single ~~SOA~~ will generate an internal response (Δt) that is a Gaussian random variable. Both assume a simple

[18] E.g., <SimultaneityNoisyCriteriaMultistart 225–386>. Note that Matlab has inbuilt functions, which could have done most of this *if* you have the statistics toolbox extensions.

decision rule (“say synchronous if Δt > C_{low} and < C_{High}”, where C indicates decision criteria). The more complex variant also allows the two criteria to vary from trial to trial as Gaussian random variables. There are many variants of this kind of model that could be envisaged, some of which are considered in Sternberg and Knoll (1973) and Ulrich (1987).

This kind of model is generally presented as a consequence of two sensory signals travelling along independent pathways to a decision centre, with sensory noise reflecting variations in their latencies from trial to trial. However, the same predictions emerge if we assume the sensory noise accrues via some other process than latency variations (e.g., spike rate stochasticity) as long as the end result is a Gaussian Δt distribution. This is an attractive feature, because in fitting our data, we may not want to commit to anything more than the fairly defensible position that noisy representations are quite likely to be Gaussian (a hallmark of classical signal detection theory).

If we stick closer to the process model in which sensory noise *is* latency noise, it is reasonable to argue that the Gaussian assumption must be a simplification. Latencies cannot be negative, so modelling them as Gaussian cannot be completely correct (although if the variance of the latency distribution is fairly small relative to the length of the neural pathway, the density below zero would be negligible). An alternative observer model based on the same basic principles has been developed by García-Pérez and Alcalá-Quintana (2012a, b, see also Chapter 12, this volume) who use exponential latency noise in place of Gaussian noise for each signal. The result is a four-parameter model, which can generate an asymmetric psychometric function for SJs and thus capture the same sorts of features as the four-parameter model presented here, but via the mechanism of an asymmetric Δt distribution (rather than criterion noise). The authors are happy to provide fitting code for their model, which can also be scaled up to include extra parameters that deal with keying errors. They have a chapter in this volume.

Their model yields two noise parameters (one for each signal) and two further parameters, which seem distinct from the two criteria described here, but are in fact mathematically equivalent. García-Pérez and Alcalá-Quintana (2012a, b) describe τ, a processing delay parameter, basically what most researchers think of as the PSS, and δ, a resolution parameter, which defines the range of values judged synchronous. The two criteria I have described here map directly onto their parameters, being $\tau - \delta$ and $\tau + \delta$ (recall that I noted how a PSS could be recovered by averaging the positions of the two criteria). The differences in terminology seem to be driven by different theoretical positions. Whereas I view the decision criteria as being malleable components of the decision process, García-Pérez and Alcalá-Quintana (2012a, b) seem at least partly committed to a form of “low-threshold” or “triggered-moment” model where SOAs below some threshold cannot be recovered, so that an observer can only guess about order.[19] However, either kind of model can easily be fitted and interpreted from either theoretical perspective.

8 Choosing between Observer Models and Rejecting Participants

Two further reasonable questions one might ask are: 1) could my observer model have generated these data? and 2) does another observer model describe the data better? Model comparison is a large and complex topic, so once again, what I have to say here should be treated as a brief introduction rather than a comprehensive summary.

Let’s begin by considering a metric I have not yet mentioned: *Deviance*. Deviance (sometimes called G^2) is a measure based on log likelihood, but which looks rather more like summed squared error, in that it is zero for a perfectly fitting model and large/positive for a poorly fitting model. Formally, deviance is two times the difference in log likelihood between the *saturated* model and the model with our current set of parameters. A saturated model is one that exactly predicts the data (which can always be accomplished by a model that has one parameter per data point). Hence it represents the situation with the maximum possible log-likelihood when predicting this particular set of data. Deviance is closely related to a simpler calculation (–2 × log likelihood) that forms the basis of a couple of well-known metrics for model comparison (the Akaike information criterion, AIC, and the Bayesian information criterion, BIC) and indeed is occasionally defined this way. That’s because we are often only really interested in differences (in Deviance, or AIC, or BIC) between models, and the log-likelihood of the saturated model gets subtracted out in a comparison between two models (because it has contributed to the deviance in the same way for both) so calculating it is not necessary.

However, if you want to say something about the goodness of fit of a model *without* relating it to any other model, based on asymptotic statistical theory, you do need to calculate deviance properly. Asymptotically, it turns out that the deviance of a model fitted to data *when that model actually generated those data* follows a chi-square (χ^2) distribution, with degrees of freedom equal to

[19] García-Pérez and Alcalá-Quintana’s commitment to this account is a little unclear, because they often let δ vary across experimental conditions, suggesting flexibility more akin to a criterion-based account. It may be that they believe a low-threshold exists, but that synchrony is often additionally reported beyond this hard limit.

the number of data points minus the number of model parameters (note: for data points, think of the number of SOAs tested, not the number of trials!) Hence, if we want to know if our model might have generated our data, we could check the best-fitting deviance against such a distribution to see how improbable this is. Unfortunately, it seems that this asymptotic result may not always be accurate for data sets of a size typical in psychophysics experiments (Wichmann & Hill, 2001a).

For this reason, Wichmann and Hill (2001a) suggest using Monte-Carlo simulation to assess whether a model is plausible. The idea is as follows. First, find the best-fitting set of model parameters. Second, create a set of data based on a simulation of the experiment in which that model generates the data. Third, find a fit to that data, and record the deviance.[20] Fourth, repeat steps two and three many times to generate a distribution of deviances that you would expect *when that model actually generated those sets of data*. Finally, look to see where the deviance of your actual fit sits on this distribution in order to assess if the model is likely to have generated the data. This approach is not implemented in the code accompanying this chapter, but should be feasible for you to implement yourself if you are interested in assessing whether your data are under or over- dispersed relative to what would be expected. However, although certainly informative, I find it a rather high bar to set if you are, for example, deciding whether to use a model or to include a participant. After all, even the most ardent defender of a particular observer model would be unlikely to argue that it really represents a complete characterisation of the psychological processes that are being modelled. I think that a model fit can be informative even if the model is a simplification of absolutely everything that observers do in experiments. To paraphrase George Box: All models are wrong, but that doesn't mean that they are not useful (Box, 1979).

With this in mind, my preference is to ask a slightly different question: Does this observer model seem to fit the data better than some other simpler account? This question is well aligned with what we generally do during statistical inference. For example, a simple (i.e., two parameter) regression is generally considered significant if it explains the data significantly better than an even simpler one-parameter model (i.e., just the mean).

What can we say about the deviance statistic as model complexity increases? Well, in general a complex model produces a better fit than a simple model whether it is correct or not, because more free parameters mean a greater ability to describe patterns that are actually just random noise

[20] It doesn't actually have to be deviance. Log likelihood, or -2 × log likelihood would be fine too.

(at least for nested models).[21] Hence, simply finding a decrease in deviance for a more complex model is not enough to show that it is better. We need to instead show that the decrease in deviance is greater than that expected by chance. Although the asymptotic result I outlined above for expectations about *absolute* deviance may be unreliable with psychophysical data sets, another rather similar result may be more robust even when N is low. The *change* in deviance from a simpler to a more complex model also follows a χ^2 distribution, but with degrees of freedom equal to the difference in free parameters between the models, as long as the models are nested.

The two observer models for SJs that I have discussed in this chapter are nested, so it's possible to make a decision about whether to use the more complicated one by comparing the deviances they each return. I have previously found the four parameter-variant to be justified for AV data with LED flashes and brief tones (Yarrow et al., 2011).

The code accompanying this chapter also includes an option, when fitting either of these models, to additionally fit a simpler model as a method of deciding whether to retain a participant as part of a group-level analysis. The logic here is that if a participant is simply guessing rather than taking the experiment seriously, they will be equally likely to say "synchronous" at any SOA, which can be captured by a straight horizontal line (effectively a model with just one parameter: their overall tendency to use one of the two keys). However, in SJ experiments we may also need to exclude participants who showed some ability to discriminate, but on only one side of the SJ function, implying that we failed to sample extreme enough SOAs to capture both of their transitions from synchrony to asynchrony. Although such an observer may have been concentrating well and following instructions, the model will return very poorly constrained and extreme parameter estimates. Hence, to look for this pattern, we should fit an intermediate model, a cumulative Gaussian, which can capture usable performance on one or other side of the SJ function, but not both. Only if the full SJ model provides a better fit relative to this partial performance model should the participant be retained (c.f. Yarrow et al., 2013).

I have now discussed what I believe is a reasonable approach to model comparison for nested models. I will finish this section by very briefly mentioning some possible approaches when models are not nested. Firstly, models can be compared using either AIC or BIC. Both of these statistics are equal to −2 times the log likelihood of the best-fitting model, but with a penalty applied

[21] Two models are nested if (basically) the more complex model can generate all the same sets of predictions that the simpler model can generate, plus a bit more. For example, stepwise regression compares nested models. Strictly, this approach requires that models are nested *and* that one of them is correct.

to the model with more free parameters. For AIC the penalty is simply 2 per parameter, whereas for BIC it is (generally) slightly greater per parameter and depends on the number of data points in the fit. BIC is actually an approximation to the Bayes factor, an (arguably) more sophisticated form of model comparison in which model performance is considered across all parameter combinations, not just at the best-fitting values. A second tactic would be to develop a Monte-Carlo simulation approach similar to that outlined above in order to produce a distribution of expected deviance improvements if the more complex model is fitted to data generated by the simpler model. As mentioned earlier, model comparison is a substantial and complex field, and there are several other approaches that could be considered beyond those touched on here.

9 Alternative Approaches to Interpreting SJ Data

Fitting a model is a nice way to summarise a set of SJ data with a few meaningful parameters. However, those parameters are only likely to tell you something useful if the model is (at some level) correct. The fact is, there is no consensus about whether any given observer model is correct, or about how literally parameter values should be interpreted. These considerations might lead us to consider doing away with any kind of parametric fit. For example, we could analyse the data without a pre-processing step, so that proportion judged simultaneous at each SOA is the dependent variable, or we could attempt a non-parametric fit to derive summary measures.

The former approach is used occasionally, sometimes as a supplement to a parametric fit. For example, Zampini et al. (2005) simply applied an ANOVA to proportion simultaneous data, incorporating their set of SOAs as a second factor (the first factor being the two conditions they were comparing). Interactions and main effects can then be interpreted to explain differences between conditions, although it may be somewhat challenging to explain what is going on in a succinct manner, particularly when many conditions are tested. Another concern is the application of ANOVA in a situation where data are clearly non-normal. Proportion/percentage data are likely to be skewed (and less variable) at the extremes (i.e., where most participants report synchrony not at all or all of the time). It might be possible to address this concern using a more complex variant of the generalized linear (mixed) model with an appropriate link function (in place of an ANOVA), an approach that has been applied successfully for data yielding sigmoidal psychometric functions (Moscatelli, Mezzetti, & Lacquaniti, 2012).

If summary measures akin to thresholds and PSSs are desirable, an alternative to a parametric fit would be to simply draw straight lines (or use some form of spline interpolation) between data points and make some informal estimates on that basis (e.g., a window where the proportion judged simultaneous falls above 0.75, or the point at which the highest proportion of simultaneity judgements is reached). However, noisy data tend to make this problematic, as the psychometric function may then appear non-monotonic on one or both sides. More sophisticated non-parametric approaches have been developed, but mainly for the more common situation of a sigmoidal psychometric function (e.g., Miller & Ulrich, 2001; Zchaluk & Foster, 2009). In some cases, it is possible to adapt these procedures to the SJ task (Lee & Noppeney, 2011).

10 Ternary Data

Before the SJ reached its current level of popularity, several authors had considered expanding the TOJ to a ternary task in which the two order responses where supplemented with a "simultaneous" response option to indicate uncertainty about order. In fact, latency-based observer models for this situation are formally identical to those I have discussed for the SJ. In early analyses, the ternary task was typically considered to permit two binary divisions of the data, each yielding a sigmoidal psychometric function. In the first such division, the psychometric function was constructed by plotting the proportion of times that observers report either "simultaneous" or "A then B" (i.e., the proportion of times they said anything other than "B then A"). In the second division, it was constructed by plotting the proportion of times that observers report only "A then B." These two psychometric functions are displaced from one another along the SOA axis. Their difference represents the occasions when the observer responded synchronous. Note that this provides an intuitive link regarding why the SJ function can be described as the difference of two cumulative Gaussians.

In fact, we can fit observer models directly to these data without rearranging them into a binary format. The observer models make predictions directly about a ternary division, which equates to predicting two out of three probability values at each SOA (with the third being defined by the fact that probabilities sum to 1.0). The code accompanying this chapter includes options to perform such a fit based on the two models (i.e., the three and four parameter variants) that I described in Section 4. From a practical perspective, there is only one conceptually challenging point of difference. It is the data model (discussed for a binary fit in Section 6.2). Because there are three possible choices, the appropriate data model (applied at each SOA) is no longer the binomial distribution, but rather the multinomial distribution, which can provide an exact likelihood of obtaining any particular combination of probabilities that divide N choices into three bins when the actual probabilities of selecting each bin are known (or rather, for fitting purposes, predicted).[22]

11 Dual-Presentation SJ Data

Several authors have investigated the use of a dual-presentation SJ task in which two bimodal stimuli are presented (one after another) and compared, for example by reporting which one was (most) synchronous (Allan & Kristofferson, 1974; Powers, Hillock, & Wallace, 2009; Roseboom, Nishida, Fujisaki, & Arnold, 2011). This is a form of what would, in classical signal detection theory, be described as a two-alternative forced choice (specifically the two-interval forced choice variant). However, that designation is ambiguous (about whether there are two presentations or two response categories) and has been applied to cases where either or both of the possible qualifying conditions are met, which is probably why the dual-presentation SJ task has ended up being given a variety of names (e.g., temporal 2AFC; forced-choice successiveness discrimination; 2IFC SJ, where the classic SJ is referred to as 2AFC SJ in the same paper). I will label it the *2xSJ*.

The simplest form of the 2xSJ would have a synchronous standard on every trial along with a non-synchronous test pair. Based on the kind of observer models discussed in this chapter, the resulting psychometric function (plotting the probability of judging the standard more synchronous than the test against the test's SOA) is U-shaped and centred over the PSS. This approach represents a reasonable way to derive estimates of inverse precision (i.e., $\sigma_{\Delta t}$) but a fairly poor way to estimate the PSS, because having a synchronous standard on every trial provides feedback about objective synchrony. A simple solution is to also include a range of standards as well as a range of tests, in a roving standard design.

The observer model can be fitted to data even when both standard and test are non-zero, as described in detail by Yarrow et al. (2016; see also García-Pérez & Peli, 2014). To present all of the data, it is necessary to plot a function for each standard SOA (using several standard plots, or a single 3D plot), which is somewhat cumbersome, but not a major obstacle to using the task. A simple

[22] <MultinomialLikelihood 9>.

observer model with three parameters captures PSS, sensory noise and an interval bias (i.e., a tendency to select one interval in preference to the other under uncertainty).

The 2xSJ task provides estimates that correlate fairly well with equivalent parameters estimated using TOJs, SJs, and ternary tasks. However, each trial takes longer than in those single-presentation tasks, which makes experiments more onerous. There are a few reasons why the roving-standard 2xSJ is still worth considering. Firstly, it asks about synchrony explicitly (unlike the TOJ) and by requiring relative judgements it reveals a point of maximal synchrony perception (whereas the SJ and ternary tasks often reveal a range of SOA values that are classified as synchronous). Secondly, it can be added in to a single-presentation task (as a follow-up question every two trials), which somewhat mitigates the burden of additional experimental time. Finally, a case can be made that it will be more resistant to some forms of decision-level bias (Morgan, Grant, Melmoth, & Solomon, 2015; Morgan, Melmoth, & Solomon, 2013). As with the other tasks I have described, code to fit data from the 2xSJ accompanies this chapter.[23] For further information, read the comments there and consult Yarrow et al. (2016).

12 Conclusion

In this chapter, I have outlined the benefits of fitting formal observer models to judgements about simultaneity, and described how this can be achieved using Matlab code (see book's GitHub repository). In doing so, I have presented one particular observer model in some detail, and highlighted the fundamentally subjective nature of the SJ task, which requires us to think carefully about how both the strategic decisions and perceptual sensitivity of a participant can affect their psychometric function. I have gone on to supply a brief overview of appropriate models for several closely related timing tasks. I hope I have also provided enough of a tutorial regarding bespoke model fitting and evaluation to allow the interested reader to go forward and explore their own models of perceived simultaneity. Modelling may seem intimidating, but in fact, a good understanding of just a few basic concepts (which is best gained through practical exploration) will take you a long way, providing tools to engage more fully with the timing literature. This is an endeavour I would very much encourage!

[23] <TwoAFCSimultaneity_3PEq_Multistart_rawdata>.

References

Allan, L.G. (1975). The relationship between judgments of successiveness and judgments of order. *Perception & Psychophysics, 18*(1), 29–36.

Allan, L.G., & A.B. Kristofferson (1974). Successiveness discrimination: Two models. *Perception & Psychophysics, 15*(1), 37–46.

Arnold, D.H., K. Petrie, R. Gallagher, & K. Yarrow (2015). An object-centered aftereffect of a latent material property: A squishiness visual aftereffect, not causality adaptation. *Journal of Vision, 15*(9), 4.

Baron, J. (1969). Temporal ROC curves and the psychological moment. *Psychological Science, 15*, 299–300.

Box, G.E.P. (1979). Robustness in the strategy of scientific model building. In Launer, R.L. & G.N. Wilkinson (Eds.), *Robustness in Statistics* (pp. 201–236). New York: Academic Press.

Efron, B., & R.J. Tibshirani (1994). *An introduction to the bootstrap.* CRC press.

García-Pérez, M.A. (2014). Adaptive psychophysical methods for nonmonotonic psychometric functions. *Attention, Perception, & Psychophysics, 76*(2), 621–641.

García-Pérez, M.A., & R. Alcalá-Quintana (2012a). On the discrepant results in synchrony judgment and temporal-order judgment tasks: A quantitative model. *Psychonomic Bulletin & Review, 19*(5), 820–846.

García-Pérez, M.A., & R. Alcalá-Quintana (2012b). Response errors explain the failure of independent-channels models of perception of temporal order. *Frontiers in Psychology, 3*, 94.

García-Pérez, M.A., & E. Peli (2014). The bisection point across variants of the task. *Attention, Perception, & Psychophysics, 76*(6), 1671–1697.

Gibbon, J., & R. Rutschmann (1969). Temporal order judgement and reaction time. *Science, 165*(891), 413–415.

Green, D.M., & J.A. Swets (1966). *Signal detection theory and psychophysics*. New York: Wiley.

Lee, H., & U. Noppeney (2011). Long-term music training tunes how the brain temporally binds signals from multiple senses. *Proceedings of the National Academy of Sciences of the United States of America, 108*(51), E1441–E1450.

Lewandowsky, S., & S. Farrell (2010). *Computational modeling in cognition: Principles and practice.* Sage.

Love, S.A., K. Petrini, A. Cheng, & F.E. Pollick (2013). A psychophysical investigation of differences between synchrony and temporal order judgments. *PloS One, 8*(1), e54798.

Macmillan, N.A., & C.D. Creelman (2005). *Detection theory: A user's guide* (2nd ed.). New York: Lawrence Erlbaum Associates.

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

Miller, J., & R. Ulrich (2001). On the analysis of psychometric functions: The spearman-kärber method. *Perception & Psychophysics, 63*(8), 1399–1420.

Miyazaki, M., S. Yamamoto, S. Uchida, & S. Kitazawa (2006). Bayesian calibration of simultaneity in tactile temporal order judgment. *Nature Neuroscience, 9*(7), 875–877.

Morgan, M., S. Grant, D. Melmoth, & J.A. Solomon (2015). Tilted frames of reference have similar effects on the perception of gravitational vertical and the planning of vertical saccadic eye movements. *Experimental Brain Research, 233*(7), 2115–2125.

Morgan, M., D. Melmoth, & J. Solomon (2013). Linking hypotheses underlying class A and class B methods. *Visual Neuroscience, 30*(5–6), 197–206.

Moscatelli, A., M. Mezzetti, & F. Lacquaniti (2012). Modeling psychophysical data at the population-level: The generalized linear mixed model. *Journal of Vision, 12*(11), 26.

Myung, I.J. (2003). Tutorial on maximum likelihood estimation. *Journal of Mathematical Psychology, 47*(1), 90–100.

Nelder, J.A., & R. Mead (1965). A simplex method for function minimization. *The Computer Journal, 7*(4), 308–313.

Powers, A.R., A.R. Hillock, & M.T. Wallace (2009). Perceptual training narrows the temporal window of multisensory binding. *The Journal of Neuroscience, 29*(39), 12265–12274.

Roseboom, W., S. Nishida, W. Fujisaki, & D.H. Arnold (2011). Audio-visual speech timing sensitivity is enhanced in cluttered conditions. *PloS One, 6*(4), e18309.

Rosenberger, W.F., & S.E. Grill (1997). A sequential design for psychophysical experiments: An application to estimating timing of sensory events. *Statistics in Medicine, 16*(19), 2245–2260.

Schneider, K.A., & D. Bavelier (2003). Components of visual prior entry. *Cognitive Psychology, 47*(4), 333–366.

Sternberg, S., & R.L. Knoll (1973). The perception of temporal order: Fundamental issues and a general model. In Kornblum, S. (Ed.), *Attention and performance IV* (pp. 629–686). London: Academic Press.

Stone, J.V., N.M. Hunkin, J. Porrill, R. Wood, V. Keeler, M. Beanland, M. Port, & N.R. Porter (2002). When is now? Perception of simultaneity. *Proceedings of the Royal Society of London Series B: Biological Sciences, 268,* 31–38.

Ulrich, R. (1987). Threshold models of temporal-order judgments evaluated by a ternary response task. *Perception & Psychophysics, 42*(3), 224–239.

Vroomen, J., & M. Keetels (2010). Perception of intersensory synchrony: A tutorial review. *Attention, Perception & Psychophysics 72*(4), 871–884.

Wichmann, F.A., & N.J. Hill (2001a). The psychometric function: I. Fitting, sampling, and goodness of fit. *Perception & Psychophysics, 63*(8), 1293–1313.

Wichmann, F.A., & N.J. Hill (2001b). The psychometric function: II. bootstrap-based confidence intervals and sampling. *Perception, & Psychophysics, 63*(8), 1314–1329.

Yarrow, K., I. Sverdrup-Stueland, W. Roseboom, & D.H. Arnold (2013). Sensorimotor temporal recalibration within and across limbs. *Journal of Experimental Psychology: Human Perception & Performance, 39*(6), 1678–1689.

Yarrow, K., N. Jahn, S. Durant, & D.H. Arnold (2011). Shifts of criteria or neural timing? The assumptions underlying timing perception studies. *Consciousness and Cognition, 20*, 1518–1531.

Yarrow, K., S.E. Martin, S. Di Costa, J.A. Solomon, & D.H. Arnold (2016). A roving dual-presentation simultaneity-judgment task to estimate the point of subjective simultaneity. *Frontiers in Neuroscience, 7*, 416.

Yarrow, K., S. Minaei, & D.H. Arnold (2015). A model-based comparison of three theories of audiovisual temporal recalibration. *Cognitive Psychology, 83,* 54–76.

Zampini, M., S. Guest, D.I. Shore, & C. Spence (2005). Audio-visual simultaneity judgments. *Perception & Psychophysics, 67*(3), 531–544.

Zchaluk, K., & D.H. Foster (2009). Model-free estimation of the psychometric function. *Attention, Perception, & Psychophysics, 71*(6), 1414–1425.