Trial-by-trial data analysis using computational models

Nathaniel D. Daw

August 27, 2009

[manuscript for *Attention & Performance XXIII*.]

1 Introduction

In numerous and high-profile studies, researchers have recently begun to integrate computational models into the analysis of data from experiments on reward learning and decision making (Platt and Glimcher, 1999; O'Doherty et al., 2003; Sugrue et al., 2004; Barraclough et al., 2004; Samejima et al., 2005; Daw et al., 2006; Li et al., 2006; Frank et al., 2007; Tom et al., 2007; Kable and Glimcher, 2007; Lohrenz et al., 2007; Schonberg et al., 2007; Wittmann et al., 2008; Hare et al., 2008; Hampton et al., 2008; Plassmann et al., 2008). As these techniques are spreading rapidly, but have been developed and documented somewhat sporadically alongside the studies themselves, the present review aims to clarify the toolbox (see also O'Doherty et al., 2007). In particular, we discuss the rationale for these methods and the questions they are suited to address. We then offer a relatively practical tutorial about the basic statistical methods for their answer and how they can be applied to data analysis. The techniques are illustrated with fits of simple models to simulated datasets. Throughout, we flag interpretational and technical pitfalls of which we believe authors, reviewers, and readers should be aware. We focus on cataloging the particular, admittedly somewhat idiosyncratic, combination of techniques frequently used in this literature, but also on exposing these techniques as instances of a general set of tools that can be applied to analyze behavioral and neural data of many sorts.

A number of other reviews (Daw and Doya, 2006; Dayan and Niv, 2008) have focused on the scientific conclusions that have been obtained with these methods, an issue we omit almost entirely here. There are also excellent books that cover statistical inference of this general sort with much greater generality, formal precision, and detail (MacKay, 2003; Gelman et al., 2004; Bishop, 2006; Gelman and Hill, 2007).

2 Background

Much work in this area grew out of the celebrated observation (Barto, 1995; Schultz et al., 1997) that the firing of midbrain dopamine neurons (and also the BOLD signal measured via fMRI in their primary target, the striatum; Delgado et al., 2000; Knutson et al., 2000; McClure et al., 2003; O'Doherty et al., 2003) resembles a "prediction error" signal used in a number of computational algorithms for reinforcement learning (RL, i.e. trial and error learning in decision problems; Sutton and Barto, 1998). Although the original empirical articles reported activity averaged across many trials, and the mean behavior of computational simulations was compared to these reports, in fact, a more central issue in learning is how behavior (or the underlying neural activity) changes *trial by trial* in response to feedback. In fact, the computational theories are framed in just these terms, and so more recent work on the system (O'Doherty et al., 2003; Bayer and Glimcher, 2005) has focused on comparing their predictions to raw data timeseries, trial by trial: measuring, in effect, the theories' goodness of fit to the data, on average, rather than their goodness of fit to the averaged data.

This change in approach represents a major advance in the use of computational models for experimental design and analysis, which is still unfolding. Used this way, computational models represent exceptionally detailed, quantitative hypotheses about how the brain approaches a problem, which are amenable to direct experimental test. As noted, such trial-by-trial analyses are particularly suitable to developing a more detailed and dynamic picture of learning than was previously available.

In a standard experiential decision experiment, such as a "bandit" task (Sugrue et al., 2004; Lau and Glimcher, 2005; Daw et al., 2006), a subject is offered repeated opportunities to choose between multiple options (e.g. slot machines) and receives rewards or punishments according to her choice on each trial. Data might consist of a series of choices and outcomes (one per trial). In principle, any arbitrary relationship might obtain between the entire list of past choices and outcomes, and the next one. Computational theories constitute particular claims about some more restricted function by which previous choices and feedback give rise to subsequent choices. For instance, standard RL models (such as "Q learning"; Watkins, 1989) envision that subjects track the expected reward for each slot machine, via some sort of running average over the feedback, and it is only through these aggregated "value" predictions that past feedback determines future choices.

This example points to another important feature of this approach, which is that the theories purport to quantify, trial-by-trial, variables such as the reward expected for a choice (and the "prediction error," or difference between the received and expected rewards). That is, the theories permit the estimation of quantities (expectations, expectation violations) that would otherwise be *subjective*; this, in turn, enables the search for neural correlates of these estimates (Platt and Glimcher, 1999; Sugrue et al., 2004).

By comparing the model's predictions to trial-by-trial experimental data, such as choices or BOLD signals, it is possible using a mixture of Bayesian and classical statistical techniques to answer two sorts of questions about a model, which are discussed in Sections 3 and 4 below. The art is framing questions of scientific interest in these terms.

The first question is *parameter estimation*. RL models typically have a number of free parameters — measuring quantities such as the "learning rate," or the degree to which subjects update their beliefs in response to feedback. Often, these parameters characterize (or new parameters can be introduced so as to characterize) factors that are of experimental interest. For instance, Behrens et al. (2007) tested predictions about how particular task manipulations would affect the learning rate.

The second type of question that can be addressed is *model comparison*. Different computational models, in effect, constitute different hypotheses about the learning process that gave rise to the data. These hypotheses may be tested against one another on the basis of their fit to the data. For example, Hampton et al. (2008) use this method to compare which of different approaches subjects use for anticipating an opponent's behavior in a multiplayer competitive game.

Learning and observation models: In order to appreciate the extent to which the same methods may be applied to different sets of data, it is useful to separate a computational theory into two parts. The first, which we will call the *learning model*, describes the dynamics of the model's internal variables such as the reward expected for each slot machine. The second part, which we will call the *observation model*, describes how the model's internal variables are reflected in observed data: for instance, how expected values drive choice or how prediction errors produce neural spiking. Essentially, the observation model regresses the learning model's internal variables onto the observed data; it plays a similar role as (and is often, in fact, identical to) the "link function" in generalized linear modeling. In this way, a common learning process (a single *learning model*) may be viewed as giving rise to distinct observable data streams in a number of different modalities (e.g., choices and BOLD, through two separate *observation models*). Thus, although we describe the methods in this tutorial primarily in terms of choice data, they are directly applicable to other modalities simply by substituting a different observation model.

Crucially, whereas the learning model is typically deterministic, the observation models are *noisy*: that is, given the internal variables produced by the learning model, an observation model assigns some *probability* to any possible observations. Thus the "fit" of different learning models, or their parameters, to any observed data can be quantified statistically in terms of the probability they assign to the data, a procedure at the core of the methods that follow.

3 Parameter estimation

Model parameters can characterize a variety of scientifically interesting quantities, from how quickly subjects learn (Behrens et al., 2007) to how sensitive they are to different rewards and punishments (Tom et al., 2007). Here we consider how to obtain statistical results about parameters' values from data. We first consider the general statistical rationale underlying the problem; then develop the details for an example RL model before considering various pragmatic factors of actually performing these analyses on data. Finally, having discussed these details in terms of choice data, we discuss how the same methods may be applied to other sorts of data.

Suppose we have some model M, with a vector of free parameters θ_M. The model (here, the composite of our learning and observation models) describes a probability distribution, or *likelihood* function, $P(D \mid M, \theta_M)$ over possible data sets D. Then, Bayes' rule tells us that having observed a data set D,

$$P(\theta_M \mid D, M) \propto P(D \mid M, \theta_M) \cdot P(\theta_M \mid M) \tag{1}$$

That is, the posterior probability distribution over the free parameters, given the data, is proportional to the product of two factors, (1) the likelihood of the data, given the free parameters, and (2) the *prior* probability of the parameters. This equation famously shows how to start with a theory of how parameters (noisily) produce data, and invert it into an theory by which data (noisily) reveal the parameters that produced it. Classically, we seek a point estimate of the parameters θ_M rather than a posterior distribution over all possible values; if we neglect (or treat as flat) the prior over the parameters $P(\theta_M|M)$, then the most probable value for θ_M is the *maximum likelihood estimate*: the setting of the parameters that maximizes the likelihood function, $P(D \mid M, \theta_M)$. We denote this $\hat{\theta}_M$.

3.1 Maximum likelihood estimation for RL

An RL model: We may see how the general ideas play out in a simple reinforcement learning setting. Consider a simple game in which on each trial t, a subject makes a choice c_t (= L or R) between a left and a right slot machine, and receives a reward r_t (= \$1 or \$0) stochastically. According to a simple Q-learning model (Watkins, 1989), on each trial the subject assigns an expected value to each machine: $Q_t(L)$ and $Q_t(R)$. We initialize these values to (say) 0, and then on each trial, the value for the chosen machine is updated as

$$Q_{t+1}(c_t) = Q_t(c_t) + \alpha \cdot \delta_t \tag{2}$$

where $0 \leq \alpha \leq 1$ is a free learning rate parameter, and $\delta_t = r_t - Q_t(c_t)$ is the *prediction error*. Equation 2 is our *learning model*. To explain the choices c_t in terms of the values Q_t we assume an *observation model*. In RL, it is often assumed that that subjects choose probabilistically according to a *softmax* distribution:

$$P(c_t = L \mid Q_t(L), Q_t(R)) = \frac{\exp(\beta \cdot Q_t(L))}{\exp(\beta \cdot Q_t(R)) + \exp(\beta \cdot Q_t(L))} \tag{3}$$

Here, β is a free parameter known in RL as the inverse temperature parameter. However, note that Equation 3 is also equivalent to standard logistic regression where the dependent variable is the binary choice variable c_t and there is one predictor variable, the difference in values $Q_t(L) - Q_t(R)$. Therefore, β can also be as viewed the regression weight connecting the Qs to the choices. More generally, when there are more than two choice options, the softmax model corresponds to a generalization of logistic regression known as conditional logit regression (McFadden, 1974).

The model of Equations 2 and 3 is only a representative example of the sorts of algorithms used to study reinforcement learning. Since our focus here is on the methodology for estimation given a model, a full review of the many candidate models is beyond the scope of the present article (see Bertsekas and Tsitsiklis, 1996; Sutton and Barto, 1998 for exhaustive treatments). That said, most models in the literature are variants on the example shown here. Another commonly used (Daw et al., 2006; Behrens et al., 2007) and seemingly rather different family of learning methods is Bayesian models such as the Kalman filter (Kakade

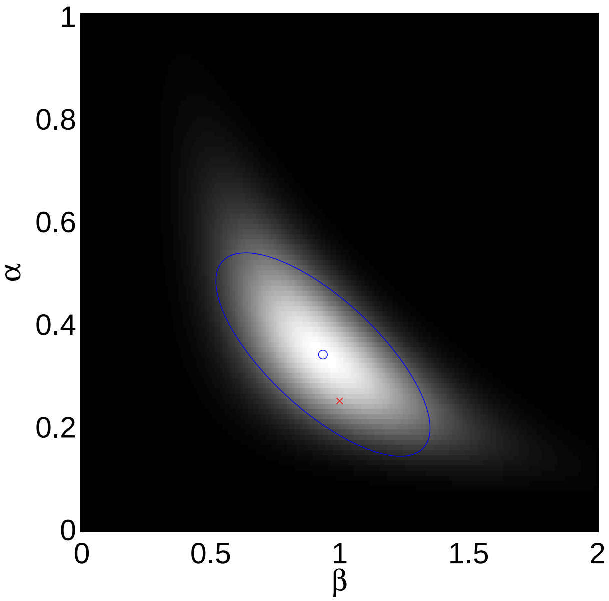

Figure 1: Likelihood surface for simulated reinforcement learning data, as a function of two free parameters. Lighter colors denote higher data likelihood. The maximum likelihood estimate is shown as an "o" surrounded by an ellipse of one standard error (a region of about 90% confidence); the true parameters from which the data were generated are denoted by an "x".

and Dayan, 2002). In fact, the Q-learning rule of Equation 2 can be seen as a simplified case of the Kalman filter: the Bayesian model uses the same learning rule but has additional machinery that determines the learning rate parameter α on a trial-by-trial basis (Kakade and Dayan, 2002; Behrens et al., 2007; Daw et al., 2008).

Data likelihood: Given the model described above, the probability of a whole dataset D (i.e., a whole sequence of choices $\mathbf{c} = c_{1...T}$ given the rewards $\mathbf{r} = r_{1...T}$) is just product of their probabilities from Equation 3,

$$\prod_t P(c_t = L \mid Q_t(L), Q_t(R)) \quad (4)$$

Note that the terms Q_t in the softmax are determined (via equation 2) by the rewards $r_{1...t-1}$ and choices $c_{1...t-1}$ on trials prior to t.

Together, Equations 2 and 3 constitute a full likelihood function $P(D \mid M, \theta_M)$, and we can estimate the free parameters ($\theta_M = \langle \alpha, \beta \rangle$) by maximum likelihood. Figure 1 illustrates the process. 1,000 choice trials were simulated according to the model (with parameters $\alpha = .25$ and $\beta = 1$, red x). The likelihood of the observed data was then computed for a range of parameters, and plotted (with brighter colors for higher likelihood) on a 2-D grid. In this case, the maximum likelihood point ($\hat{\alpha} = .34$ and $\hat{\beta} = .93$, blue circle) was near the true parameters.

Confidence intervals: Of course, in order actually to test a hypothesis about the parameters' values, we need to be able to make statistical claims about the quality of the estimate $\hat{\theta}_M$. Intuitively, the degree to which our estimate can be trusted depends on how much better it accounts for the data than other nearby parameter estimates, that is on how sharply peaked is the "hill" of data likelihoods in the space of parameters. Such peakiness is characterized by the second derivative (the *Hessian*) of the likelihood function with respect to the parameters. The Hessian is a square matrix (here, 2x2) with a row and column for each parameter. Evaluated at the peak point $\hat{\theta}_M$, the elements of the Hessian are larger the more rapidly the likelihood function is dropping off away from it in different directions, which corresponds to a more reliable estimate of the parameters. Conversely, the matrix inverse of the Hessian (like the reciprocal of a scalar) is larger for poorer estimates, like error bars. More precisely, if H is the Hessian of the *negative log* of the likelihood function at the maximum likelihood point $\hat{\theta}_M$, then a standard estimator for the *covariance* of the parameter estimates is its matrix inverse H^{-1} (MacKay, 2003).

The diagonal terms of H^{-1} correspond to variances for each parameter separately, and their square roots measure one standard error on the parameter. Thus, for instance, 95% confidence intervals around the maximum likelihood estimate may be estimated as $\hat{\theta}$ plus or minus 1.96 standard errors.

Covariance between parameters: The off-diagonal terms of of H^{-1} measure *covariance* between the parameters, and are useful for diagnosing model fit. In general, large off-diagonal terms are a symptom of a poorly specified model or some kinds of bad data. In the worst case, two parameters may be redundant, so that there is no unique optimum. The Q learning model has a more moderate coupling between the parameters. As can be seen by the elongated, tilted shape of the "ridge" in Figure 1, estimates of α and β tend to be inversely coupled in this model. By increasing β while decreasing α (or vice versa: moving northwest or southeast in the figure), a similar likelihood is obtained. This is because the reward r_t is multiplied by both α (in Equation 2 to update Q_t) and then by β (in Equation 3) before affecting the choice likelihood on the next trial. As a result of this, either parameter individually cannot be estimated so tightly by itself (the "ridge" is a bit wide if you cross it horizontally in β or vertically in α), but their product is well estimated (the hill is most narrow when crossed from northeast to southwest). The blue oval in the figure traces out a one-standard error ellipse in the two parameters jointly, derived from H^{-1}; its tilt follows the contour of the ridge.

Often in applications such as logistic regression, a corrected covariance estimator is used that is thought to be more robust to problems such as mismatch between the true and assumed models. This "Huber-White" or "sandwich" estimator (Huber, 1967; Freedman, 2006) is $H^{-1}BH^{-1}$ where $B = \sum_t g(c_t)^T g(c_t)$, and $g(c_t)$, in turn, is the gradient (vector of first partial derivatives with respect to the parameters) of the negative log likelihood of the tth data point c_t, evaluated at $\hat{\theta}_M$. This is harder to compute in practice, since it involves keeping track of g, which is laborious. However, as discussed below, g can also be useful when searching for the maximum likelihood point.

3.2 Pragmatics

Above, we developed the general equations for maximum likelihood parameter estimation in an RL model; how can these be implemented in practice for data analysis?

First, although we have noted an equivalence between Equation 3 and logistic regression, it is not possible simply to use an off-the-shelf regression package to estimate the parameters. This is because although the observation stage of the model represents a logistic regression from values Q_t to choices c_t, the values are not fixed but themselves depend on the free parameters (here, α) of the learning process. As these do not enter the likelihood linearly they cannot be estimated by a generalized linear model. Thus, we must search for the full set of free parameters that optimize the likelihood function.

Likelihood function: At the heart of optimizing the likelihood function is computing it. It is straightforward to write a function that takes in a dataset (a sequence of choices and rewards) and a candidate setting of the free parameters, loops over the data computing equations 2 and 3, and returns the aggregate likelihood of the data. Importantly, the product in Equation 4 is often an exceptionally small number; it is thus numerically more stable to compute its log, i.e. the *sum* over trials of the log of the choice probability from equation 3, which is $\beta \cdot Q_t(c_t) - \log(\exp(\beta \cdot Q_t(L)) + \exp(\beta \cdot Q_t(R)))$. Since log is a monotonic function, this quantity has the same optimum but is less likely to underflow the minimum floating point value representable by a computer. (Another numerical trick is to note that Equation 3 is invariant to the addition or subtraction of any constant to all of the Q values. The chance of the exponential under- or overflowing can thus be reduced by evaluating the log probability for Q values after first subtracting their mean.)

How, then, to find the optimal likelihood? In general, it may be tempting to discretize the space of free parameters, compute the likelihood everywhere, and simply search for the best, much as is illustrated in Figure 1. We recommend against this approach. First, most models of interest have more than two parameters, and exhaustively testing all combinations in a higher dimensional space becomes intractable.

Second, and not unrelated, discretizing the parameters too coarsely, or searching within an inappropriate range, can lead to poor results; worse yet, since the parameters are typically coupled (as in Figure 1), a poor search on one will also corrupt the estimates for other parameters.

Nonlinear optimization: To avoid errors related to discretization, one may use routines for nonlinear function optimization that are included in many scientific computing languages. These functions do not discretize the parameter space, but instead search continuously over candidate values. In Matlab, for instance, this can be accomplished by functions such as `fmincon` or `fminsearch`; similar facilities exist in such packages as R and the SciPy toolbox for Python. Given a function to compute the likelihood, and also starting settings for the free parameters, these functions search for the optimal parameter settings (the parameters that minimize the function, which means that the search must be applied to the *negative* log likelihood). Because these functions conduct a local search — e.g., variants on hill climbing — and because in general, likelihood surfaces may not be as well behaved as the one depicted in Figure 1 but may instead have multiple peaks, these optimizers are not guaranteed to find a global optimum. A solution to this problem is to call them many times from different starting points (e.g., randomly chosen or systematically distributed on a grid), and use the best answer (lowest negative log likelihood) found. It is important to remember that the goal is to find the single *best* setting of the parameters, so it is only the best answer (and not, for instance, some average) that counts.

Gradients and Hessians: Nonlinear optimization routines such as `fmincon` are also, often, able to estimate the Hessian of the log likelihood function numerically, and can return this matrix for use in computing confidence intervals. Alternatively, it is possible (albeit laborious) to compute both the Hessian H and the gradient g of the likelihood analytically alongside the likelihood itself. This involves repeatedly using the chain rule of calculus to work out, for instance, the derivative of the log likelihood of the data with respect to α (the first element of g) in terms of the derivatives of the log likelihoods for each choice, $\partial \log(P(c_t))/\partial\alpha$ which in turn depend, through Equation 3, on c_t, $\partial P(c_t)/\partial Q_t$, and $\partial Q_t/\partial\alpha$. The last of these is vector valued, $[\partial Q_t(L)/\partial\alpha, \partial Q_t(R)/\partial\alpha]$, and depends through the derivative of Equation 2 on r_t and recursively on $\partial Q_{t-1}/\partial\alpha$ and so on back to r_1 and $\partial Q_1/\partial\alpha = [0, 0]$. (It is easier to envision computing this forward, alongside the data likelihood: i.e., computing $\partial \log(P(c_t))/\partial\alpha$ along with $\log(P(c_t))$ and $\partial Q_t/\partial\alpha$ along with Q_t at each step while looping through each trial's data in order.) If g is computed, it can be provided to the optimization function so as to guide the search; this often markedly improves the optimizer's performance.

Boundaries: With some function optimizers (e.g. Matlab's `fmincon`), it is possible to place boundaries on the free parameters. We suggest that these be used with caution. On one hand, some free parameters have semantics according to which values outside some range appear senseless or may be unstable. For instance, the learning rate α in Equation 2 is a fractional stepsize and thus naturally ranges between zero and one. Moreover, for $\alpha > 1$, the Qs can grow rapidly and diverge. Thus, it makes some sense to constrain the parameter within this range. Similarly, negative values of β seem counterproductive while very large values of β lead to large exponentiated terms in computing the log of Equation 3, and ultimately program crashes due to arithmetic overflow. Also, for some subjects in this model, the maximum likelihood parameter estimates can occur at a very large β and a very small α (Schonberg et al., 2007). For reasons like these, it can be tempting to constrain β also to stay within some range.

However, if optimal parameters rest at these boundaries, this is cause for concern, since it indicates that the true optima lie outside the range considered and the estimates found depend rather arbitrarily on the boundary placement. Also, note again that since the parameters are often interrelated, constraining one will impact the others as well. Although fits outside a sensible range can be simply due to noise, this can also suggest a problem with the model, a bad data set, or a programming bug. There may, for instance, be features of the data that can only be captured within the model in question by adopting seemingly nonsensical parameters. (For instance, the issue with abnormally large α and small β, mentioned above, arises for a subset of subjects whose choices aren't well explained by this model.) It should also be noted that most of the statistical inference techniques discussed in this article — including the use of the inverse Hessian to estimate error bars, and the model comparison techniques discussed in Section 4 — are ultimately based on approximating the likelihood function around $\hat{\theta}_M$ by a Gaussian "hill." Thus, none of these methods is formally justified when estimated parameters lie on a boundary, since the hill will be severely truncated,

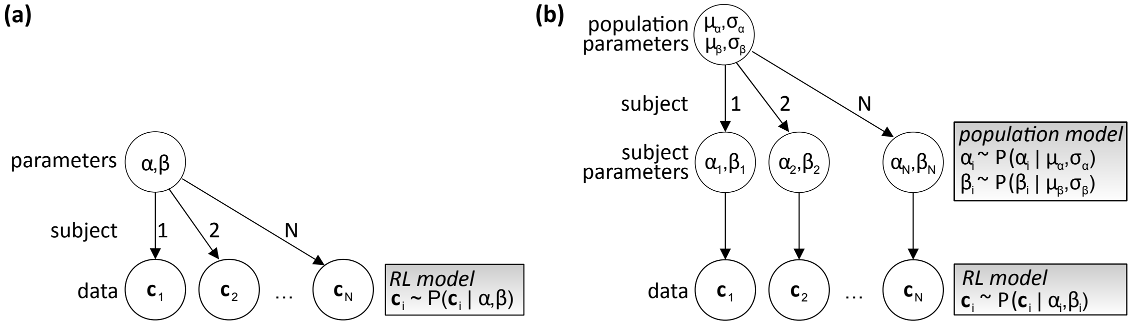

Figure 2: Models of population data. (a) Fixed effects: Model parameters are shared between subjects. (b) Random effects: Each subject's parameters are drawn from a common population distribution.

and the point being examined may not even actually be its peak.

To the extent that inadmissible parameter estimates arise not due to poor model specification, but instead simply due to the inherent noise of maximum likelihood estimation, one possible alternative to a hard constraint on parameters is a *prior.* In particular, equation 1 suggests that prior information about the likely range of the parameters could enter via the term $P(\theta_M|M)$, and would serve to regularize the estimates. In this case we would use a *maximum a posteriori* estimator for $\hat{\theta}_M$: i.e., optimize the (log) product of both terms on the right hand side of Equation 1, rather than only the likelihood function. Apart from the change of objective function, the process of estimation remains quite similar. Indeed, hard constraints, such as $0 \leq \alpha \leq 1$, are equivalent to a uniform prior over a fixed range, but soft constraints (which assign, say, decreasing prior likelihood to larger parameter values in a graded manner) are equally possible in this framework. Of course, it is hard to know how to select a prior in an objective manner. One empirical source for a prior at the level of the individual is the behavior of others in the population from which the individual was drawn, a point to which we return in the next section.

In summary, parameter estimation via nonlinear function approximation is feasible but finicky. Program crashes and odd parameter estimates are common due to issues such as numerical stability and parameter boundaries. We have discussed a number of practical suggestions for minimizing problems, but the most important point is simply that the process is not as automatic as it sounds: it requires ongoing monitoring and tuning.

3.3 Intersubject variability and random effects

Typically, a dataset will include a number of subjects. Indeed, often questions of scientific interest involve characterizing a population — do students, on average, have nonzero learning rates in a subliminal learning task (Pessiglione et al., 2008)? — or comparing two populations — for instance, do Parkinson's disease patients exhibit a lower learning rate than healthy controls (Frank et al., 2004)? How do we extend the methods outlined above to incorporate multiple subjects and answer population-level questions?

Fixed effects analysis: One obvious approach, which we generally do not advocate, is simply to aggregate likelihoods not just across trials, but also across subjects, and to estimate a single set of parameters $\hat{\theta}_M$ that optimize the likelihood of the entire dataset. Such an analysis treats the data from all subjects as though they were just more data from a single subject. That is, it treats the estimated parameters as *fixed effects*, quantities that do not vary across subjects (Figure 2a). Of course, this is unlikely to be the case. Indeed, the variability *between* subjects in a population is precisely what is relevant to answering statistical questions about the population (do college students have a nonzero learning rate? do Parkinson's patients learn slower than controls?). For population-level questions, treating parameters as fixed effects and thereby conflating within- and between-subject variability can lead to serious problems such as overstating the true significance of results. This issue is familiar in fMRI data analysis (Holmes and Friston, 1998; Friston et al., 2005) but less so in other areas of psychology and neuroscience.

Summary statistics approach: An often more appropriate but equally simple procedure is, for n subjects, separately to estimate a set of maximum likelihood parameters for each subject. Then we may test the mean value of a parameter or compare groups using (e.g.) a one- or two-sample t-test on the estimates. Intuitively, such a procedure seems reasonable. It treats each parameter estimate as a random variable (a *random effect*), and essentially asks what value one would expect to estimate if one were to draw a new subject from the population, then repeat the entire experiment and analysis. This *summary statistics* procedure is widely used, and this use is at least partly justified by a formal relationship with the more elaborate statistical model laid out next (Holmes and Friston, 1998; Friston et al., 2005). However, note one odd feature of this procedure: it ignores the within-subject error bars on each subject's parameter estimates.

Hierarchical model of population: We can clarify these issues by extending our approach of modeling the data generation process explicitly to incorporate a model of how parameters vary across the population (Figure 2b; Penny and Friston, 2004). Suppose that when we recruit a subject i from a population, we also draw a set of parameters (e.g., α_i and β_i) according to some statistical distributions that characterize the distribution of parameters in the population. Perhaps β_i is Gaussian-distributed with some mean μ_β and standard deviation σ_β. We denote this Gaussian as $P(\beta_i \mid \mu_\beta, \sigma_\beta)$. Similarly α_i would be given by some other probability distribution. In the examples below, we also assume that $P(\alpha_i \mid \mu_\alpha, \sigma_\alpha)$ is also Gaussian. An alternative reflecting the (potentially problematic) assumption of bounds on α (i.e., $0 < \alpha < 1$) is instead to assume a distribution with support only in this range. The parameter might be distributed, for instance, as a beta distribution or as a Gaussian $x_i \sim N(\mu_\alpha, \sigma_\alpha)$ transformed through a logistic function, $\alpha_i = 1/(1+\exp(-x_i))$. Structural questions about which sort of distribution to use are ultimately *model selection* questions, which can be addressed through the methods discussed in Section 4.

Adopting a model of the parameters in the population gives us a two-level *hierarchical* model of how a full dataset is produced (Figure 2): Each subject's parameters are drawn from population distributions, then the Q values and the observable choice data are generated, as before, according to an RL model with those parameters. Usually, the parameters of interest are those characterizing the population (e.g. μ_α, σ_α, μ_β, and σ_β); for instance, it is these that we would like to compare between different populations to study whether Parkinson's disease affects learning. The full equation that relates these population-level parameters to a particular subject's choices, $\mathbf{c}_i$, is then the probability given to them by the RL model, here abbreviated $P(\mathbf{c}_i \mid \alpha_i, \beta_i)$, averaged over all possible settings of the individual subject's parameters according to their population distribution:

$$P(\mathbf{c}_i \mid \mu_\alpha, \mu_\beta, \sigma_\alpha, \sigma_\beta) = \int d\alpha_i d\beta_i P(\alpha_i \mid \mu_\alpha, \sigma_\alpha) P(\beta_i \mid \mu_\beta, \sigma_\beta) P(\mathbf{c}_i \mid \alpha_i, \beta_i) \tag{5}$$

This formulation emphasizes that individual parameters α_i and β_i intervene between the observable quantity and the quantity of interest, but from the perspective of drawing inferences about the population parameters, they are merely nuisance variables to be averaged out. The probability of a full dataset consisting of choice sets $\mathbf{c}_1 \ldots \mathbf{c}_N$ for N subjects is just the product over subjects:

$$P(\mathbf{c}_1 \ldots \mathbf{c}_N \mid \mu_\alpha, \mu_\beta, \sigma_\alpha, \sigma_\beta) = \prod_i P(\mathbf{c}_i \mid \mu_\alpha, \mu_\beta, \sigma_\alpha, \sigma_\beta) \tag{6}$$

We can then use Bayes' rule to recover the population parameters in terms of the full dataset:

$$P(\mu_\alpha, \mu_\beta, \sigma_\alpha, \sigma_\beta \mid \mathbf{c}_1 \ldots \mathbf{c}_N) \propto P(\mathbf{c}_1 \ldots \mathbf{c}_N \mid \mu_\alpha, \mu_\beta, \sigma_\alpha, \sigma_\beta) P(\mu_\alpha, \mu_\beta, \sigma_\alpha, \sigma_\beta) \tag{7}$$

Estimating population parameters in a hierarchical model: Equation 7 puts us in in a position, in principle, to estimate the population parameters from the set of all subjects' choices, using maximum likelihood or maximum a posteriori methods exactly as discussed for individual subjects in the previous section. Confidence intervals on these parameters (from the inverse Hessian) allow between-group comparisons. This would require programming a likelihood function that returns the (log) probability of given choices over a population of subjects, given the four population-level parameters (Equation 6). This, in turn, requires averaging, for each individual subject, over possible sets of values for that subject's parameters according

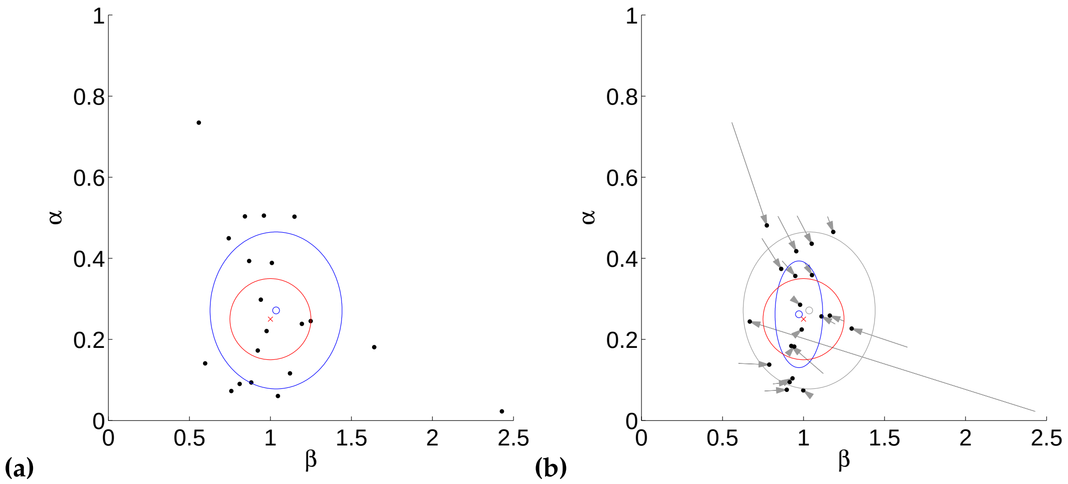

Figure 3: (a) Estimating population parameters via summary statistics. Learning parameters for 20 simulated subjects from the bivariate Gaussian distribution with mean and standard deviation shown in red. 1,000 choice trials for each subject were simulated, and the model fit to each individual by maximum likelihood. The individual fits are shown as dots, and their summary statistics are shown in blue. Here, the population mean is well estimated, but the population variance is inflated. (b) Parameters for the simulated subjects were re-estimated by maximum a posteriori taking the population summary statistics from part a (gray ellipse) as a prior. Estimates are pulled toward the population mean. Summary statistics for the new estimates are shown in blue.

to Equation 5 and then aggregating results over subjects. Such a function could then be optimized as before, using a nonlinear function optimizer such as `fmincon`.

The difficulty with this approach in practice is that the integral in Equation 5 is intractable, so it must be approximated in some way to compute the likelihood. One possibility is via sampling, e.g. by drawing k (say, 10,000) settings of the parameters for each subject according to the distributions $P(\alpha_i \mid \mu_\alpha, \sigma_\alpha)$ and $P(\beta_i \mid \mu_\beta, \sigma_\beta)$, then averaging over these samples to approximate the integral as $1/k \cdot \sum_{j=1}^{k} P(\mathbf{c}_i \mid \alpha_j, \beta_j)$. One practical issue here is that optimization routines such as `fmincon` require the likelihood function to change smoothly as they adjust the parameters. Sampling a set of individual subject parameters anew for each setting of the population parameters the optimizer tries can therefore cause problems, but this can generally be addressed by using the same underlying random numbers at each evaluation (i.e., resetting the random seed to the same value each time the likelihood function is evaluated; Bhat, 2001; Ng and Jordan, 2000) .

Estimating population parameters via summary statistics: Suppose that we know the true values of the individual subject parameters α_i and β_i: for instance, suppose we could estimate these perfectly from the choices. In this case, we could estimate the population parameters directly from the subject parameters, since Equation 7 reduces to $P(\mu_\alpha, \sigma_\alpha \mid \alpha_1 \ldots \alpha_N) \propto \prod_i [P(\alpha_i \mid \mu_\alpha, \sigma_\alpha)] \cdot P(\mu_\alpha, \sigma_\alpha)$ and similarly for β_i. Moreover, assuming the distributions $P(\alpha_i \mid \mu_\alpha, \sigma_\alpha)$ and $P(\beta_i \mid \mu_\beta, \sigma_\beta)$ are Gaussian, then finding the population parameters for these expressions is just the familiar problem of estimating a Gaussian distribution from samples. In particular, the population means and variances can be estimated in the normal way by the the sample statistics. Importantly, we could then compare the estimated mean parameters between groups or (within a group) against a constant using standard t-tests. Note that in this case, since the parameter estimates arise from an average of samples, confidence intervals can be derived from the sample standard deviation divided by the square root of the number of samples, i.e. the familiar standard error of the mean in Gaussian estimation. We need not use the Hessian of the underlying likelihood function in this case.

We can thus interpret the two-stage summary statistics procedure discussed above as an approximate estimation strategy for the hierarchical model of Figure 2b, and an alternative to the strategy of direct maximum likelihood estimation discussed above. In particular, the procedure would be correct for Gaussian distributed parameters, if the uncertainty about the within-subject parameters were negligible. What is the effect of using this as an approximation when this uncertainty is instead substantial, as when the parameters were estimated from individual subject model fits? Intuitively, the within-subject estimates will be jittered with respect to their true values due to estimation noise. We might imagine (and in some circumstances, it is indeed the case) that in computing the population means, μ_α and μ_β, this jitter will average out and the resulting estimates will be unbiased. However, the estimation noise in the individual parameter estimates will *inflate* the estimated population variances beyond their true values (Figure 3a).

What mostly matters for our purposes is the validity of t-tests and confidence intervals on the estimated population means. For some assumptions about the first-level estimation process, Holmes and Friston (1998) demonstrate that for t-tests and confidence intervals, the inflation in the population variance is expected to be of just the right amount to compensate for the unaccounted uncertainty in the subject-level parameters. While this argument is unlikely to hold exactly for the sorts of computational models considered here, it also seems that this procedure is relatively insensitive to violations of the assumptions (Friston et al., 2005). Thus, these considerations provide at least partial justification for use of the summary statistics procedure.

Estimating individual parameters in a hierarchical model: Though we have so far treated them as nuisance variables, in some cases, the parameter values for an individual in a population may be of interest. The hierarchical model also provides insight into estimating these while taking into account the characteristics of the population from which the individual was drawn (Friston and Penny, 2003). Assuming we know the population level parameters, Bayes' rule specifies that

$$P(\alpha_i, \beta_i \mid \mathbf{c}_i, \mu_\alpha, \mu_\beta, \sigma_\alpha, \sigma_\beta) \propto P(\mathbf{c}_i \mid \alpha_i, \beta_i) P(\alpha_i, \beta_i \mid \mu_\alpha, \mu_\beta, \sigma_\alpha, \sigma_\beta) \qquad (8)$$

Here again we might make use of this equation as an approximation even if we have only an estimate of the population parameters, ignoring the uncertainty in that estimate.

Equation 8 takes the form of Equation 1: $P(\mathbf{c}_i \mid \alpha_i, \beta_i)$ is just the familiar likelihood of the subject's choice sequence given the parameters, but playing the role of the prior is $P(\alpha_i, \beta_i \mid \mu_\alpha, \mu_\beta, \sigma_\alpha, \sigma_\beta) = P(\alpha_i \mid \mu_\alpha, \sigma_\alpha) P(\beta_i \mid \mu_\beta, \sigma_\beta)$, the population distribution of the parameters. This equation thus suggests one *empirical* source for a prior over a subject's parameters: the distribution of the parameters in the population from which the subject was drawn.

If we estimate (or re-estimate) the subject's parameters by maximizing Equation 8, then the estimated parameters will be drawn toward the group means, reflecting the fact that the data for other subjects in a population are also relevant to estimating a subject's parameters (Figure 3b). Thus, in the context of a hierarchical model, the combination of prior and likelihood in Bayes rule specifies exactly how to balance this population information with data about the individual in estimating parameters.

3.4 Summary and recommendations

For most questions of interest, it is important to treat model parameters as random effects, so as to enable statistical conclusions about the population from which the subjects were drawn. Given such a model, the easy way to estimate it is the summary statistics procedure, and since this is so simple, transparent, and fairly well behaved, we recommend it as a starting point. The more complex alternative — the fit of a full hierarchical model based on an approximation to Equation 5 — is better justified and seems a promising avenue for improvement.

3.5 Extensions

The basic procedures outlined above admit of many extensions. Of particular importance is how they may be applied to types of data other than choices.

Other types of data: We have conducted the discussion so far in terms of choice data. But an entirely analogous process can be (and has been) applied to many other data sources, such as neural spiking (Platt and Glimcher, 1999; Sugrue et al., 2004; Barraclough et al., 2004) or fMRI measurements (O'Doherty et al., 2003; Wittmann et al., 2008). All that is required, in principle, is to replace the observation model of Equation 3 with one appropriate to the data modality. The viewpoint is that a common learning model (here, Equation 2), may be observed, indirectly, through many different sorts of measurements that it impacts. The observation model provides a generative account of how a particular sort of measurement may be impacted by the underlying learning.

For instance, it is natural to assume that spike rates (or perhaps log spike rates) taken from an appropriate window reflect model variables such as δ_t corrupted by Gaussian noise, e.g.

$$s_t = \beta_0 + \beta_1 \delta_t + N(0, \sigma) \tag{9}$$

or similarly for value variables, e.g. $Q_t(L)$ or $Q_t(c_t)$. This replaces the logistic regression of model values onto choices with *linear* regression of model values onto spike rates, and unifies the learning model-based approach here with the ubiquitous use of regression to characterize spike responses. Thus, here again, we could estimate not just the magnitude and significance of spiking correlates (β_1 in Equation 9) but also the underlying learning rate parameter α that best explains a timeseries of per-trial spike rates. Analogous to the case of choice data, while the observation stage of this model terminates in linear regression, parameters in the learning model (here, α) affect the data nonlinearly, so the entire model cannot be fit using a standard regression routine, but instead a nonlinear search must be used. This model of spiking could also be extended hierarchically, analogously to the discussion above, to reason about the characteristics of a population of neurons recorded from a particular region; or even, through another level of hierarchy, regions in multiple animals.

The same linear (or log-linear) observation model can also be used to examine whether reaction times or other behavioral data such as pupilommetry are modulated by reward expectancy, and if so, to examine the underlying learning process. Approaches of this sort can be used to model learning in behavioral data obtained from Pavlovian experiments (i.e., those involving reward or punishment expectancy without choices; O'Doherty et al., 2003; Seymour et al., 2007). Finally, by fitting a model through both behavioral and neural modalities, it is possible to conduct a neurometric/psychometric comparison (Kable and Glimcher, 2007; Tom et al., 2007; Wittmann et al., 2008).

fMRI: A very similar observation model is also common in analyzing fMRI data. This typically involves assuming an underlying timeseries with impulses of height given by the variable of interest (e.g.,δ_t) at appropriate times (e.g. when reward is revealed, and $\delta_t = 0$ otherwise). To produce the BOLD timeseries measured in a voxel, it is assumed that this impulse timeseries is convolved with a hemodynamic response filter, and finally scaled and corrupted by additive Gaussian noise, as in equation 9. The full model might be written

$$b_t = \beta_0 + \beta_1(\mathrm{HRF} \star \delta_t) + N(0, \sigma) \tag{10}$$

In fact, this observation model (augmented with a hierarchical random effects model over the regression weights, such as β_1, across the population) is identical to the general linear model used in standard fMRI analysis packages such as SPM. Thus, a standard approach is simply to enter model-predicted timeseries (e.g., δ or Q) as parametric regressors in such a package (O'Doherty et al., 2003, 2007).

Since these packages implement only the linear regression stage of the analysis, inference in fMRI tends to focus on simply testing the estimated regression weights (e.g. β_1 from Equation 10) against a null hypothesis of 0, to determine *whether* and *where in the brain* a variable like δ_t is significantly reflected in BOLD timeseries. Thus, the predictor timeseries are typically generated with the parameters of the underlying learning model (here, the learning rate α) fixed, e.g., having previously been fit to behavior.

One practical point is that, in our experience (Daw et al., 2006; Schonberg et al., 2007), multisubject fMRI results from analyses of this sort are, in practice, more robust if a single α (and a single set of any other parameters of the learning model) is used to generate regressors for all subjects. A single set of parameters might be obtained from a fixed effect analysis of behavior, or from the population level parameter means in a random effects analysis. This issue may arise because maximum likelihood parameter estimates are relatively noisy, or because differences in learning model parameters can effectively produce differences between subjects in the scaling of predictor timeseries, which inflate the variability in their regression weights across the population and suppress the significance of population-level fMRI results.

The approach of using fixed learning parameters rather than fitting them to BOLD data is mandated by efficiency: for whole-brain analyses, a linear regression to estimate β_1 for fixed α is feasible, but conducting a laborious *nonlinear* estimate of α at each of hundreds of thousands of voxels in the brain is computationally infeasible. Of course, if a BOLD timeseries for a particular voxel or area of interest were isolated, then a full model could be estimated from it using nonlinear optimization, much as described above for spikes, reaction times, or choices.

Linearized parameter estimation: Short of a nonlinear fit to a targeted area, it is also possible to use a whole-brain linear regression approach to extract at least some information about the learning model parameters that best explain the BOLD signal (Wittmann et al., 2008). Suppose we compute the prediction error timeseries δ_t for some relevant choice of α, such as that found from behavior, say 0.5. Denote this $\delta_t(0.5)$. Now we may also compute a second timeseries $\partial \delta_t / \partial \alpha$: the partial derivative of the prediction error timeseries with respect to α. For any choice of α, this is another timeseries. We evaluate it at the same point, here $\alpha = 0.5$ and denote the result as $\delta'_t(0.5)$.

The partial derivative measures how the original regressor timeseries would change as you move the parameter α infinitesimally away from its starting value of 0.5. Indeed, one can approximate the partial derivative as the difference $(\delta_t(0.5 + \Delta) - \delta_t(0.5))/\Delta$ between the original regressor and that recomputed for a slightly larger learning rate, $\alpha = (0.5 + \Delta)$ for some small Δ; or equivalently, express the regressor for a larger α, $\delta_t(0.5 + \Delta)$ as $\delta_t(0.5) + \Delta \delta'_t(0.5)$. (The true derivative is just the limit of the difference as $\Delta \to 0$, and can also be computed exactly with promiscuous use of the chain rule of calculus, in a similar manner to the gradient of the likelihood function discussed above.)

We can now approximate $\delta_t(\alpha)$ for any learning rate α as

$$\delta_t(\alpha) \approx \delta_t(0.5) + (\alpha - 0.5)\delta'_t(0.5) \tag{11}$$

This *linear* approximation of $\delta_t(\alpha)$ is, formally, the first two terms of a Taylor expansion of the function. Since this approximation is linear in α, we can estimate it using linear regression, e.g. using a standard fMRI analysis package. In particular, if we include the partial derivative as an additional regressor in our analysis, so that we are modeling the BOLD timeseries in a voxel as $b_t = \beta_0 + \beta_1(\mathrm{HRF} \star \delta_t(0.5)) + \beta_2(\mathrm{HRF} \star \delta'_t(0.5)) + N(0, \sigma)$, then the estimate of β_2 — the coefficient for the partial derivative of the regressor — is an estimate of α under the linearized approximation, since it plays the same role as $(\alpha - 0.5)$ in Equation 11. Normalizing by the overall effect size, the estimate of α is $\hat{\beta}_2/\hat{\beta}_1 + 0.5$.

However, since the linear approximation is poor, it would be unwise to put too much faith in the particular numerical value estimated by this method. Instead, this approach is most useful for simply testing whether the α that best explains the data is greater or less than the chosen value (here, 0.5, by testing whether β_2 is significantly positive or negative. It can also be used for testing whether neural estimates of α covary, across subjects, with some other factor (Wittmann et al., 2008).

A very similar approach is often used in fMRI to capture the (nonlinear) effects of intersubject variation in the hemodynamic response filter (Friston et al., 1998).

Other learning models: Clearly, the learning model itself (Equation 2) could also be swapped with another as appropriate to a task or hypothesis, for instance one with more free parameters or a different learning rule. Again the basic strategy described above is unchanged. In Section 4, we consider the question of comparing between models which is a better account for data.

Other population models: Above, we assumed that individual subject parameters followed simple distributions such as a unimodal Gaussian, $P(\beta_i \mid \mu_\beta, \sigma_\beta)$. This admits of many extensions. First, subjects might cluster into several types. This can be captured using a multimodal *mixture model* of the parameters (Camerer and Ho, 1998), e.g., in the two-type case: $\pi_1 N(\mu_{\alpha 1}, \sigma_{\alpha 1}) N(\mu_{\beta 1}, \sigma_{\beta 1}) + (1 - \pi_1) N(\mu_{\alpha 2}, \sigma_{\alpha 2}) N(\mu_{\beta 2}, \sigma_{\beta 2})$. Parameters $\mu_{\alpha 1}$ and so on can be estimated to determine what the modes are that best fit the data; π_1 controls the predominance of subject type 1; and the question how many types of subjects do the data support is a model selection question, answerable by the methods discussed in Section 4.

A separate question is whether intersubject parametric variability can be explained or predicted via factors other than random variation (Wittmann et al., 2008). For instance, perhaps IQ predicts learning rate. If we have separately measured each subject's IQ, IQ_i, then we might test this hypothesis by estimating a hierarchical model with additional IQ effects in the generation of learning rates, such as $P(\alpha_i \mid \mu_\alpha, \sigma_\alpha, k_{IQ}, IQ_i) = N(\mu_\alpha + k_{IQ} IQ_i, \sigma_\alpha)$. Here, the parameter k_{IQ} controls the strength of the hypothesized linear effect. This parameter can be estimated (and the null hypothesis that it equals zero can be tested) using the same methods discussed above.

Parametric nonstationarity: Finally, we have assumed that model parameters are stationary throughout an experimental session, which is unlikely actually to be the case in many experiments. For instance, in a two-armed bandit problem in which one option pays off 40% of the time and the other pays off 60% of the time, subjects may figure out which option is better and then choose it more or less exclusively. In the model we have considered, a constant setting of the parameters often cannot account for such behavior — for instance, a high learning rate promotes rapid acquisition but subsequent instability; a learning rate slow enough to explain why subjects are asymptotically insensitive to feedback would also predict slow acquisition. In all, fast acquisition followed by stable choice of the better option might be modeled with a decrease over trials in the learning rate, perhaps combined with an increase in the softmax temperature. (This example suggests that, here again, the interrelationship between estimated parameters introduces complexity, here in characterizing and analyzing their separate change over time.)

Three approaches have been used to deal with parametric nonstationarity. The approach most consistent with the outlook of this review is to specify a computational theory of the dynamics of free parameters, perhaps itself parameter-free or expressed in terms of more elementary parameters that are expected to be stationary (Behrens et al., 2007). Such a theory can be tested, fit and compared using the same methods discussed here. For instance, as already mentioned, the Kalman filter model (Kakade and Dayan, 2002) generalizes the model of Equation 2 and specifies a particular learning rate for each trial. The coupling between softmax temperature and learning rate is one pitfall in testing such a theory, since if we treat the temperature as constant in order to test the model's account of variability in the learning rate, then changes in the temperature will not be accounted for and may masquerade as changes in the learning rate.

In lieu of a bona fide theory from first principles of a parameter's dynamics, one can specify a more generic parametrized account of changing parameters (Camerer and Ho, 1998). Note, of course, that we can't simply fit a separate free temperature β_t for each trial, since that would involve as many free parameters as data points. But we can specify some functional form for change using a few parameters. For instance, perhaps β ramps up or down linearly, so that $\beta_t = \beta_{start} + (\beta_{end} - \beta_{start}) t/T$. Here, the constant parameter β is replaced with two parameters, which can be fit as before. Another possibility (Samejima et al., 2004) is to use a Gaussian random walk, e.g. $\beta_{t=1} = \beta_{start}$; $\beta_{t+1} = \beta_t + \epsilon_t$; $\epsilon_t \sim N(0, \sigma_\epsilon)$, which also has two free parameters, β_{start} and σ_ϵ. Note, however, that this model is difficult to fit. Given only a setting for the free parameters, β_t is not determined since its dynamics are probabilistic. Thus, computing the data likelihood for any individual subject requires averaging over many different possible random trajectories of $\beta_1 \ldots \beta_T$, much as we did for different subject-specific parameters in Equation 5.

A final approach to dealing with parametric nonstationarity is to design tasks in an attempt to to minimize it (Daw et al., 2006). Returning to the example of the bandit problem, if the payoff probabilities for the two bandits were not fixed, but instead diffused slightly and randomly from trial to trial, then, ideally, instead of settling on a machine and ceasing to learn, subjects would have to continue learning about the value of the machines on each trial. Intuitively, if the speed of diffusion is constant from trial to trial, this might motivate relatively smooth learning, i.e. a nearly constant learning rate. Formally, ideal observer models such as the Kalman filter (Kakade and Dayan, 2002) predict that learning rates should be asymptotically stable in tasks similar to this one.

4 Model comparison

So far, we have assumed a fixed model of the data and sought to estimate its free parameters. A second question that may be addressed by related methods is to what extent the data support different candidate models.

In neural studies, model comparisons have often played a supporting role for parametric analyses of the sort discussed in Section 3, since comparing a number of candidate models can help to validate or select the model whose parameters are being estimated. More importantly, many questions of scientific interest are themselves naturally framed in terms of model selection. In particular, models like that of Equation 2 and its alternatives constitute different hypotheses about the mechanisms or algorithms that the brain uses to solve RL problems. These hypotheses can be compared against one another based on their fit to data.

Such an analysis can formally address questions about methods of valuation: for instance, do subjects really make decisions by directly learning a net value for each action, in the manner of Equation 2, or do they instead evaluate actions indirectly by learning more fundamental facts about the task and reasoning about them? (In RL, the latter approach is known as "model-based" learning; Daw et al., 2005.) They can also assess refinements to a model or its structure: for instance, are there additional influences on action choice above the effect of reward posited by Equation 2? Analogous analyses may also be applied to assess parts of the data model other than the learning itself; for instance the observation model (are spike counts well described as linear or do they saturate?) or the population model (are there multiple subtypes of subject or a single cluster?).

How well a model fits data depends on the settings of its free parameters; moreover, blindly following the approach to parameter optimization from the previous section will not produce a useful answer to the model-selection question, since in general, the more free parameters a model has, the better will be its fit to data at the maximum likelihood point. The methods discussed below address this problem.

In some cases, questions of interest might be framed either in terms of parameter estimation or model selection, and thus addressed using either the methods of the previous or the current situation. For instance, categorical structural differences can sometimes be recast as graded parametric differences (as in "automatic relevance determination"; MacKay, 2003). In general, since the methods in both sections all arise from basically similar reasoning (mainly, the fanatical use of Bayes' rule) in a common framework, similar questions framed in both ways should yield similar results.

4.1 Examples from RL

We illustrate the issues of model selection using some simple alternatives to the model of choice behavior discussed thus far. In fact, the practical ingredients for model evaluation are basically the same as those for parameter estimation; as before, what is needed is simply to compute data likelihoods under a model, optimize parameters, and estimate Hessians.

Policy and value models: One fundamental issue in RL is the *representational* question what is actually learned that guides behavior. In this respect, one important distinction is between *value-based* models such as Equation 2 that learn about the *values* of actions, vs. another family of *policy-based* algorithms that learn directly about what choice strategies work best (Dayan and Abbott, 2001). In the simple choice setting here, the latter replaces Equation 2 with an update such as:

$$\pi_{t+1}(c_t) = \pi_t(c_t) + (r_t - \bar{r}) \tag{12}$$

then chooses as before with

$$P(c_t = L \mid \pi_t(L), \pi_t(R)) = \frac{\exp(\beta \cdot \pi_t(L))}{\exp(\beta \cdot \pi_t(R)) + \exp(\beta \cdot \pi_t(L))}$$

The learning equation tracks a new variable π_t measuring preference over the alternatives ($\bar{r}$ is a comparison constant often taken to be an average over all received rewards; for simplicity, in the simulations below we took it to be fixed at 0.5, which was the true average). The latter equation is just the softmax choice rule of Equation 3, rewritten in terms of π_t instead of Q_t.

Equation 12 hypothesizes a different learning process (that is, a differently constrained form for the relationship between feedback and subsequent choices) than does Equation 2. The interpretation of this difference may seem obscure in this particular class of tasks, where the difference is indeed subtle. The key point is that the model of the previous section estimates the average reward Q expected for each choice, and chooses actions based on the comparison between these value estimates; whereas a model like Equation 12 is obtained by treating the parameters π as generic "knobs" controlling action choice, and then attempts to set the knobs so as to attain as much reward as possible. (See Dayan and Abbott, 2001, chapter 9, for a full discussion.)

One prominent observable difference arising from this distinction is that the policy-based algorithm will ultimately tend to turn the action choice "knobs" as far as possible toward exclusive choice of the richer option ($\pi \rightarrow \infty$ for a better than average option), whereas the values Q in the value model asymptote at the true average reward (e.g., 60 cents for an option paying off a dollar 60% of the time). Depending on β (assumed to be fixed), these asymptotic learned values may imply less-than-complete preference for the better option over the worse. This particular prediction is only one aggregate feature of what are, in general, different trial-by-trial hypotheses about learning dynamics. Thus, while it might be possible simply to examine learning curves for evidence that choices asymptote short of complete preference, the difference between models can be assessed more robustly and quantitatively by comparing their fit to raw data in the manner advocated here.

Choice autocorrelation: Note also that these models contain different numbers of free parameters: the Q learning model has two (α and β), while the policy model has only β (treating $\bar{r}$ as given or determined by the received rewards). As already noted, this introduces some difficulty in comparing them. This difficulty is illustrated more obviously by another simple alternative to the Q learning model, which can be expressed by replacing the softmax rule of Equation 3 with

$$P(c_t = L \mid Q_t(L), Q_t(R), L_{t-1}, R_{t-1}) = \frac{\exp(\beta \cdot Q_t(L) + \kappa \cdot L_{t-1})}{\exp(\beta \cdot Q_t(R) + \kappa \cdot R_{t-1}) + \exp(\beta \cdot Q_t(L) + \kappa \cdot L_{t-1})} \qquad (13)$$

Here, L_{t-1} and R_{t-1} are binary indicator variables that take on the values 1 or 0 according to whether the choice on trial $t-1$ was L or R. The motivation for models of this sort is the observation that whereas the Q learning model predicts that choice is driven only by reward history, in choice datasets, there is often significant additional choice autocorrelation (e.g., switching or perseveration) not attributable to the rewards (Lau and Glimcher, 2005). Equation 13 thus includes a simple effect of the previous choice, scaled by the new free parameter, κ, for which positive values promote sticking and negative values promote alternation.

4.2 Classical model comparison

How can we determine how well each model fits the data? By analogy with parameter fitting, we might consider the probability of the data for some model M_1, evaluated at the maximum likelihood parameters: $P(D \mid M_1, \hat{\theta}_{M_1})$. The upside of this is that this is a quantity we know how to compute (it was the entire focus of Section 3); the downside is that it provides an inflated measure of how well a model predicts a dataset.

To see this, consider comparing the original Q-learning model (M_1: Equations 2 and 3) with the version that includes previous-choice effects (M_2: Equations 2 and 13). Note that M_1 is actually a special case of M_2, for $\kappa = 0$. (The models are known as *nested*.) Since every setting of parameters in M_1 is available in M_2, the maximum likelihood point for M_2 is necessarily at least as good as that for M_1. In particular, even for a dataset that is actually generated according to M_1 (i.e., with $\kappa = 0$), it is highly likely that, due to noise in any particular set of choices (Equation 3) there will be some accidental bias toward perseveration or switching, and thus that the data will be slightly better characterized with a positive or negative θ, producing a higher likelihood for M_2 (Figure 4). This phenomenon is known as *overfitting*: in general, a more complex model will fit data better than a simpler model, by capturing noise in the data. Of course, we could fit a 300-choice data set perfectly and trivially with a 300-parameter "model" (one parameter for each choice), but clearly such a model is a poor predictive account of the data.

Cross validation: But in what sense is an overfit model worse, if it actually assigns a higher probability to the fit data? One way to capture the problem is to fit a model to one dataset, then compute the likelihood under the previously fit parameters for a *second* data set generated independently from the same

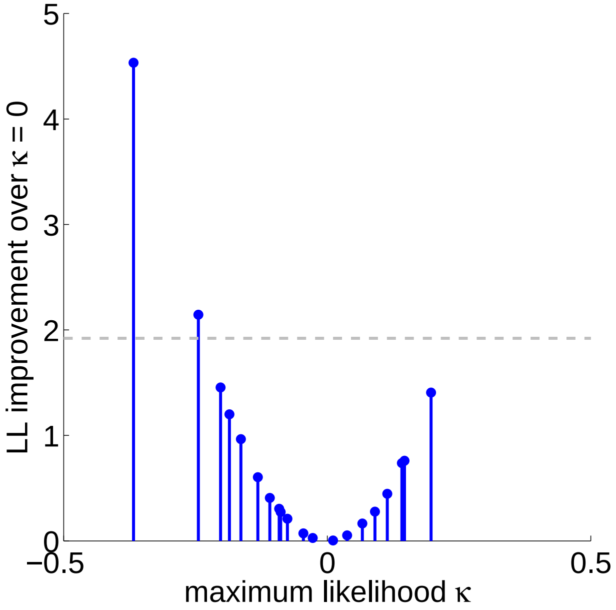

Figure 4: Overfitting: 300 trials each from 20 subjects were simulated using the basic Q learning model, then fit with the model including an additional parameter κ capturing choice autocorrelation. For each simulated subject, the maximum likelihood estimate for κ is plotted against the difference in log-likelihood between this model and the best fit of the true model with $\kappa = 0$. These differences are always positive, demonstrating overfitting, but rarely exceed the level expected by chance (95% significance level from likelihood ratio test, gray dashed line).

distribution as the original. Intuitively, to the extent a model fit is simply capturing noise in the original data set, this will hurt rather than help in predicting the second ("holdout", "cross validation") data set: by definition, the noise is unlikely to be the same from one dataset to the next. Conversely, a model fits well exactly to the extent that it captures the repeatable aspects of the data, allowing good predictions of additional datasets. This procedure allows models with different numbers of parameters to be compared on equal footing, on the basis of the likelihood of the holdout data: the holdout likelihood score is not inflated by the number of parameters in the model, since in any case *zero* parameters are fit to the second dataset.

In some areas of neuroscience — notably multivoxel fMRI pattern analyses (Norman et al., 2006) — this "cross validation" procedure is the predominant method to assess and compare model fit. In contrast, it has rarely been used in studies of reinforcement learning (though see Camerer and Ho, 1999). We do not recommend it in this area, mainly because it is difficult in timeseries data to define a second dataset that is truly independent of the first. Additionally, there are concerns about whether split datasets (e.g., train on early trials and test on late trials) are really identically distributed due to the possibility that subjects' parameters are changing.

Likelihood ratio test: Let us consider again the likelihood of a single dataset, using best-fitting parameters. It turns out that while this metric is inflated, it is still useful because the degree of inflation can in some cases be quantified. Specifically, it is possible to assess how likely is a particular level of improvement in a model's fit to data, if this were due to adding only superfluous parameters and fitting noise (Figure 4, dashed line). For the particular case of nested models, this allows us to estimate the probability of the observed data under the null hypothesis that the data are actually due to the simpler model, and thus (if this p-value is low) reject simpler model with confidence. The resulting test is called the likelihood ratio test, and is very common in regression analysis. To carry it out, fit both a complex model, M_2, and a simpler nested model, M_1, to the same data set, and compute twice the difference in log likelihoods, $d = 2 \cdot [\log P(D \mid M_2, \hat{\theta}_{M_2}) - \log P(D \mid M_1, \hat{\theta}_{M_1})]$. (Since M_2 nests M_1, this difference will be positive or zero.) The probability of a particular difference d arising under M_1 follows a chi-square distribution with a number of degrees of freedom equal to the number, n, of *additional* parameters in M_2 ; so the p-value of the test (a difference d or larger arising due to chance) is one minus the chi-square cumulative distribution at d. (In Matlab, the p value is `1-chi2cdf(d,n)` and the critical d value for 95% significance is `chi2inv(.95,n)`.)

This test cannot be used to compare models that are not nested in one another, such as the value and policy RL models of equations 2 and 12. For this application, and to develop more intuition for the problems of overfitting and model comparison, we turn to Bayesian methods.

4.3 Bayesian model comparison in theory

Model evidence: In general, we wish to determine the posterior probability of a model M, given data D. By Bayes rule:

$$P(M \mid D) \propto P(D \mid M)P(M) \tag{14}$$

The key quantity here is $P(D \mid M)$, known as the *model evidence*: the probability of the data under the model. Importantly, this expression does not make reference to any particular parameter settings such as $\hat{\theta}_M$: since in asking how well a model predicts data we are not given any particular parameters. This is why the score examined above, $P(D \mid M, \hat{\theta}_M)$, is inflated by the number of free parameters: it takes as *given* parameters that are in fact *fit* to the data. That is, in asking how well a model predicts a dataset, it is a fallacy, having seen the data, to retrospectively choose the parameters that would have best fit it. This overstates the ability of the model to predict the dataset. Comparing models according to $P(D \mid M)$, instead, avoids overfitting.

Instead, the possible values of the model parameters are in this instance (as in others we have seen before) nuisance quantities that must be averaged out according to their probability *prior to examining the data*, $P(\theta_M \mid M)$. That is:

$$P(D \mid M) = \int d\theta_M P(D \mid M, \theta_M) P(\theta_M \mid M) \tag{15}$$

The "automatic Occam's razor": Another way to see that comparing models according to Equation 14 is immune to overfitting is to note that this equation incorporates a preference for simpler models. One might assume that this could be incorporated by simply *assuming* such a preference in $P(M)$, the prior over models, but in fact, it arises automatically due to the other term, $P(D \mid M)$ (MacKay, 2003). $P(D \mid M)$ is a probability distribution, so over all possible data sets, it must sum to 1: $\int dD \cdot P(D \mid M) = 1$. This means that a more flexible model (one with more parameters that is able to achieve good fit to many data sets with different particular parameter settings) must correspondingly assign lower $P(D \mid M)$ to all of them since a fixed probability of 1 is divided among them all. Conversely, an inflexible model will fit only a few datasets well, and $P(D \mid M)$ will be higher for those datasets. Effectively, the normalization of $P(D \mid M)$ imposes a penalty on more complex and flexible models (MacKay, 2003).

Bayes factors: The result of a Bayesian model comparison is a statistical claim about the relative fit of one model over another. When comparing two models, a standardized measure of their relative fit is the *Bayes factor*, defined as ratio of their posterior probabilities (Kass and Raftery, 1995):

$$\frac{P(M_1 \mid D)}{P(M_2 \mid D)} = \frac{P(D \mid M_1)P(M_1)}{P(D \mid M_2)P(M_2)} \tag{16}$$

(Here the denominator from Bayes' rule, which we have anyway been ignoring, actually cancels out.) The log of the Bayes factor is symmetric: positive values favor M_1 and negative values favor M_2. Although Bayes factors are not the same as classical p values, they can loosely be interpreted in a similar manner. A Bayes factor of 20 (or a log Bayes factor of about 3) corresponds to 20:1 evidence in favor of M_1, which is similar to $p = .05$. Kass and Raftery (1995) present a table of conventions for interpreting Bayes factors; note that their logs are taken in base-10 rather than base-*e*.

4.4 Bayesian model comparison in practice

The theory of Bayesian model selection is very useful conceptual framework; for instance, it clarifies why the maximum likelihood score is an inappropriate metric for model comparison. However, actually using these methods in practice poses two problems. The first is one we have already encountered repeatedly: the integral in Equation 15 is intractable, and it must be approximated, as discussed below.

Priors: The second problem, which is different here, is the centrality of the prior over parameters, $P(\theta_M \mid M)$ to the analysis. We have mostly ignored priors thus far, because their subjective nature arguably makes them problematic in the context of objective scientific communication. However, in the analysis above, the prior over parameters controls the average in Equation 15. What it *means*, on the view we have described, to ask how well a model predicts data, parameter-free, is to ask how well it predicts data, averaged and weighted over the possible parameter settings. For this purpose, specifying a model *necessarily* includes specifying the admissible range of parameters for this average and their weights, i.e. the prior. The choice also affects the answer: the "spread" of the prior controls the degree of implicit penalty for free parameters that the automatic Occam's razor imposes (see MacKay, 2003, chapter 28, for a full discussion). For instance, a fully specified parameter (equivalent to a prior with support at only one value) is not free and does not contribute a penalty; as the prior admits of more possible parameter settings, the model becomes more complex. Moreover, because we are taking a weighted average over parameter settings, and not simply maximizing over them, simply ignoring the prior as before is often not mathematically well behaved.

Thus, most of the methods discussed below do require assuming a prior over parameters. Only the simplest method, BIC, ignores this.

Sampling: One approach to approximating the integral of 15 is, as before, by sampling. In the simplest case, one would draw candidate parameter settings according to $P(\theta_M \mid M)$; compute the data likelihood $P(D \mid M, \theta_M)$ for each, and average. This process does not involve any optimization, only evaluating the likelihood at randomly chosen points. Naive sampling of this sort can perform poorly if the number of model parameters is large. See MacKay (2003) and Bishop (2006) for discussion of more elaborate sampling techniques that attempt to cope with this situation.

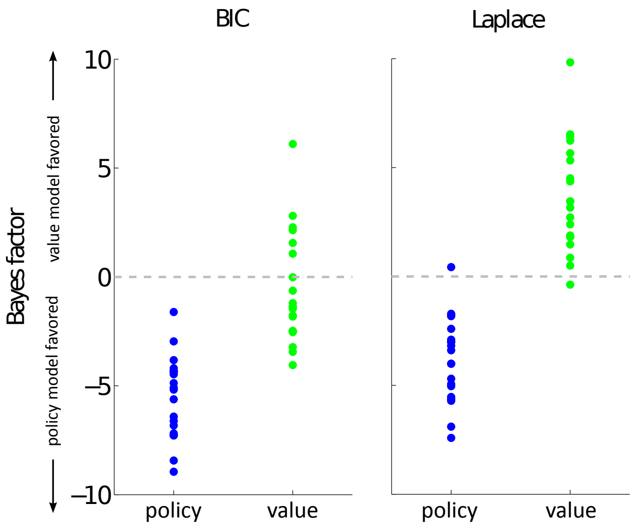

Figure 5: Model comparison with complexity penalties: 300 choice trials each from 20 subjects were simulated using the one-parameter policy model (blue dots) and the two-parameter value model (green dots); the choices were then fit with both models and Bayes factors comparing the two models were computed according to both BIC and Laplace approximations to the model evidence. (For Laplace, the prior over parameters were taken as uniform over a large range.) BIC (left) overpenalizes the value model for its additional free parameter, and favors the simpler policy model even for many of the simulated value model subjects (green dots below the dashed line); the Laplace approximation not only sets the penalty more appropriately, but it also separates the two sets of subjects more effectively because it takes into account not just the raw number of parameters but also how well they were actually fit for each subject.

Laplace approximation: A very useful shortcut for the integral of Equation 15 is to approximate the function being integrated with a Gaussian, for which the integral can then be computed analytically. In particular, we can characterize the likelihood surface around the maximum a posteriori parameters $\hat{\theta}_M$ as a Gaussian centered on that point. (This is actually the same approximation that motivates the use of the inverse Hessian H^{-1} for error bars on parameters in Section 3.)

This *Laplace approximation* results in the following expression:

$$log(P(D \mid M)) \approx log(P(D \mid M, \hat{\theta}_M)) + log(P(\hat{\theta}_M \mid M)) + \frac{n}{2}\log(2\pi) - \frac{1}{2}\log|H| \tag{17}$$

where n is the number of parameters in the model and $|H|$ is the determinant of the Hessian (which captures the covariance of the Gaussian). The great thing about this approximation is that we already know how to compute all the elements; they are just what we used in Section 3. One bookkeeping issue here is that this equation is in terms of the MAP parameter estimate (including the prior) rather than the maximum likelihood. In particular, here $\hat{\theta}_M$ refers to the setting of parameters that maximizes the first two terms of Equation 17, not just the first one. Similarly, H is the Hessian of the function being optimized (minus the sum of the first two terms of Equation 17), evaluated at the MAP point, not the Hessian of just the log likelihood.

Equation 17 can thus be viewed as the maximum (actually MAP) likelihood score, but penalized with an additional factor (the last two terms) that corrects for the inflation of this quantity that was discussed in Section 4.2.

BIC and cousins: A simpler approximation, which can be obtained from Equation 17 in a limit of large data, is the Bayesian Information Criterion (BIC; Schwarz, 1978). This is:

$$log(P(D \mid M)) \approx log(P(D \mid M, \hat{\theta}_M)) - \frac{n}{2}\log m$$

where m is the number of datapoints (e.g., choices). This is also a penalized likelihood score (the penalty is given by the second term), but it does not depend on the prior over parameters and can instead be evaluated for $\hat{\theta}_M$ being the maximum likelihood parameters. The neglect of a prior, while serendipitous from the perspective of scientific communication, seems also somewhat dubious given the entirely crucial role of the prior discussed above. Also, counting datapoints m and particularly free parameters n can be subtle; importantly, the fit of a free parameter should really only be penalized to the extent it actually contributes to explaining the data (e.g., a parameter that has no effect on observable data is irrelevant; other parameters may be only loosely constrained by the data; MacKay, 2003). The last term of the Laplace approximation accounts properly for this by factoring in the uncertainty in the posterior parameter estimates, while parameter-counting approaches like BIC or the likelihood ratio test do not. This can produce notably better results (Figure 5).

Finally, other penalized scores for model comparison exist. The most common is the Akaike information criterion (AIC; Akaike, 1974), $log(P(D \mid M, \hat{\theta}_M)) - n$. Although this has a similar form to BIC, we do not advocate its use since it does not arise from an approximation to $log(P(D \mid M))$, and thus cannot be used to approximate Bayes factors (Equation 16), which seem the most reasonable and standard metric to report.

4.5 Summary and recommendations

Models may be compared to one another on the basis of the likelihood they assign to data; however, if this likelihood is computed at parameters chosen to optimize it, the measure must be corrected for overfitting to allow a fair comparison between models with different numbers of parameters. In practice, when models are nested, we suggest using a likelihood ratio test, since this permits reporting a classical p-value and is well accepted. When they are not, an approximate Bayes factor can be computed instead; BIC is a simple and widely accepted choice for this, but its usage rests more on convention than correctness. If one is willing to define a prior, and defend it, we suggest exploring the Laplace approximation, which is almost as simple but far better founded.

One important aspect of the Laplace approximation, compared to BIC (and also the likelihood ratio test), is that it does not rely simply on counting parameters. Even if two candidate models have the same number of parameters — and thus scores like BIC are equivalent to just comparing raw likelihoods — the complexity penalty implied by Equation 15 may not actually be the same between them if the two sets of parameters are differently constrained, either a priori or by the data. As in Figure 5, this more accurate assessment can have salutary effects.

4.6 Model comparison and populations

So far we have described model comparison mostly in the abstract, with applications to choice data at the single subject level. But how can we extend them to multisubject data of the sort discussed in Section 3.3? There are a number of possibilities, of which the simplest will often suffice.

A first question is whether we treat the choice of model as itself a fixed or random effect. Insofar as the model is a categorical claim about how the brain works, it may often seem natural to assume that there is no variability across subjects in the model *identity* (as opposed to in its parameters). Thus, model identity is often taken as a fixed effect across subjects. (Note that even if we assume a lack of variability in the *true* model underlying subjects' behavior, because of noise in the parameters and the choices, every subject might not always *appear* to be using the same model when analyzed individually.) By Bayes rule:

$$P(M \mid c_1 \ldots c_N) \propto P(c_1 \ldots c_N \mid M) P(M) \tag{18}$$

Then one simple approach is to neglect the top level of the subject parameter hierarchy of Figure 2, and instead assume that individual parameters are drawn independently according to some fixed (known or ignored) prior. In this case, the right hand side of Equation 18 decomposes across subjects, and inference can proceed separately, similar to the summary statistics procedure for parameter estimation:

$$\log[P(c_1 \ldots c_N \mid M)P(M)] = \sum_i [\log P(c_i \mid M)] + \log P(M)$$

That is, we can just aggregate the probability of the data given the model over each subject's fit (single-subject BIC scores or Laplace-approximated model evidence, for instance: not raw data likelihoods) to compute the model evidence for the full dataset. These aggregates can then be compared between two models to compute a Bayes factor over the population. In this case, it is useful also to report the number of subjects for whom the individual model comparison would give the same answer as that for the population, in order to help verify the assumption that the model is a fixed effect.

The more involved approach to population variability discussed in Section 3.3 was to integrate out single subject parameters according to Equation 5 (e.g., by sampling them), in order to estimate the top-level parameters using the full model of Figure 2. In principle, it is possible to combine this approach with model selection — indeed, this is the only possibility if the models in question are at the population level (e.g., if one is asking how many clusters of subjects there are). In this case,

$$P(c_1 \ldots c_N \mid M) = \int d\theta_{pop} P(c_1 \ldots c_N \mid M, \theta_{pop}) P(\theta_{pop} \mid M)$$

where θ_{pop} are the population-level parameters, $\langle \mu_\alpha, \mu_\beta, \sigma_\alpha, \sigma_\beta \rangle$. This integral has the form of Equation 14 and can be approximated in the same way, e.g., with BIC or a Laplace approximation; this, in turn, will involve also computing $P(c_1 \ldots c_N \mid M, \theta_{pop})$, which by Equations 6 and 5 involves another integral (over the individual subject parameters) whose approximation was discussed in Section 3.3. Note that in this case, the prior that must be assumed is over the population parameters, not directly over the individual parameters.

Finally, one could take the identity of the model as varying over subjects, i.e. as a random effect (Stephan et al., 2009). This involves adding another level to the hierarchy of Figure 2, according to which, for each subject, one of a set of models is drawn with some according to a multinomial distribution (given by new free parameters) and then the model's parameters and the data are drawn as before. Inference about the probability could then proceed analogously to that for other population-level parameters.

Note that one alternative sometimes observed in the literature (Stephan et al., 2009) is summary statistics reported and tested for individual subject Bayes factors (e.g., "across subjects, model M_1 is favored over M_2 by an average log Bayes factor of 4.1, which is significantly different from zero by a t-test"). Such an approach does not appear to have an obvious analogy with summary statistics for *parameter inference* that would justify its use in terms of a hierarchical population model like that of Figure 2. (The difference is that the hierarchical model of parameters directly specifies intersubject variation over the parameters, which can then be directly estimated by summary statistics on parameter estimates. A hierarchical model parameterizing intersubject variability in *model identity* would imply variability over estimated Bayes factors only in a complex and indirect fashion; thus, conversely, the summary statistics on the Bayes factors don't seem to offer any simple insight into the parameters controlling intersubject variability in model identity.)

5 Pitfalls and alternatives

We close this tutorial by identifying some pitfalls, caveats, and concerns with these methods that we think it is important for readers to appreciate.

Why not assess models by counting how many choices they predict correctly? We have stressed the use of a probabilistic *observation model* to connect models to data. For choice behavior, an approach that is sometimes used and that may initially seem more intuitive is to optimize parameters (and compare model fit) on the basis of the number of choices correctly predicted (Brandstatter et al., 2006). For instance, given the learning model of equation 2, we might ask, on each trial, whether the choice c_t is the one with the *maximal* Q value, and if so score this trial as "correctly predicted" by the model. We might then compare parameters or models on the basis of this score, summed over all trials.

This approach is poor in a number of respects. Because it eschews the use of an overt statistical observation model of the data, this scoring technique forgoes obvious connections to statistical estimation, which are what, we have stressed, permit actual statistical conclusions to be drawn. Similarly, whereas the use of observation models clarifies how the same underlying learning model might be applied in a consistent manner to multiple data modalities, there is no obvious extension of the trial-counting scoring technique to BOLD or spiking data.

Finally, even treated simply as a score for evaluating models or parameters, and not as a tool for statistical estimation, the number of "correctly predicted" choices is still evidently inferior to the probabilistic data likelihood. In particular, because the data likelihood admits the possibility of noise in the choices, it considers a model or a set of parameters to be better when they *come closer* to predicting a choice properly (for instance, if they assign that choice 45% rather than 5% probability, even when the other choice is nevertheless viewed as more likely). Counting "correct" predictions does not distinguish between these cases.

How can we isolate between-group differences in parameters if parameter estimates are correlated? As mentioned, even if learning parameters such as temperature and learning rate are actually independent from one another (i.e., in terms of their distribution across a population), the *estimates* of those parameters from data may be correlated due to their having similar expected effects on observable data (Figure 1). This may pose interpretational difficulty for comparing populations, as when attempting to isolate learning deficits due to some neurological disease. For instance, if PD patients actually have a lower learning rate than controls but a normal softmax temperature, and we estimate population distributions for both parameters as discussed in Section 3.3, it is possible that their deficit might also partly masquerade as a decreased softmax temperature. However, the question what subset of parameters is fixed or varying across groups is more naturally framed as a structural, model-selection question, which may also be less prone to ambiguities in parameter estimation. Such an analysis would ask whether the pattern of behavior across both populations can best be explained by assuming only one or the other parameter varies (on average) between groups while the other one is shared. In this case, three models (shared mean learning rate, shared mean softmax temperature, neither shared) might be compared.

How well does the model fit? This is a common question. One suspects it is really meant as another way of asking the question discussed next — is there some other, better model still to be found? — to which there is really no answer. Nevertheless, there are a number of measures of model performance that may be useful to monitor and report. Although it is difficult to conclude much in absolute terms from these measures, if this reporting becomes more common, the field may eventually develop better intuitions about their interpretation.

Data likelihoods (raw or BIC-corrected) are often reported, but these measures are more interpretable if standardized in various ways. First, it is easy to compute the log data likelihood under pure chance. This allows reporting the fractional reduction in this measure afforded by the model (toward zero, i.e. $P(D|\theta_M, M) = 1$ or perfect prediction), a statistic known as "pseudo-r^2" (Camerer and Ho, 1999). If R is the log data likelihood under chance (e.g., for 100 trials of a two-choice task, $100 \cdot \log(.5)$) and L is the log likelihood under the fit model, then pseudo-r^2 is $1 - L/R$. Second, since likelihood measures are typically aggregated across trials, it can be more interpretable to examine the average log likelihood per trial, i.e. L/T for T trials. For choice data, exponentiating this average log likelihood, $\exp(L/T)$ produces a probability that is easily interpreted relative to the chance level.

Finally, it is easy to conduct a simple statistical verification that a model fits better than chance. Since every model nests the 0-parameter empty model (which assumes all data are due to chance) a likelihood ratio test can be used to verify that any model exceeds the performance of this one. Better still is to compare the full model against a submodel that contains only any parameters modeling mean response tendencies, nuisance variables, or biases. This is commonly done in regression analysis.

Is there another explanation for a result? There certainly could be. We may conclude that a model fits the data better than another model (or, assuming a particular model, estimate its parameters) but it seems impossible entirely to rule out the possibility that there is yet another model so far unexamined that would explain the data still better (though see Lau and Glimcher, 2005, for one approach). Although this issue is certainly not unique to this style of analysis, in our experience, authors and readers may be less likely to appreciate it in the context of a model-based analysis, perhaps because of the relatively novel and technical nature of the process.

In fMRI particularly, the problem of correlated regressors is extremely pernicious. As discussed, many studies have focused on testing whether, and where in the brain, timeseries generated from computational models correlate significantly with BOLD timeseries (O'Doherty et al., 2007). These model-generated regressors are complicated and rather opaque objects and may very well be correlated with other factors (for instance, reaction time, time on task, or amount won), which might, in turn, suffice to explain the neural activity.

It is thus important to identify possible confounds and exclude them as factors in explaining the neural signal (e.g., by including them as nuisance regressors). Also, almost certainly, a model-generated signal will be correlated with similar timeseries that might be generated from other similar models. This points again to the fact that these methods are suited to drawing *relative* conclusions comparing multiple hypotheses (the data support model A over model B). It is tempting to instead employ them in more confirmatory fashion (e..g, interpreting a finding that some model-generated signal loads significantly on BOLD as evidence supporting the correctness of model A in an absolute sense). Sadly, confirmatory reasoning of this sort is common in the literature, but it should be treated with suspicion.

There is no absolute answer to these difficulties, other than paying careful attention to identifying and ruling out confounds in designing and analyzing studies, rather than adopting a confirmatory stance. We also find it particularly helpful, in parallel, to analyze our data using more traditional (non-model-based) methods such as averaging responses over particular kinds of trials, and also to fit more generic models such as pure regression models to test the assumptions of our more structured models (Lau and Glimcher, 2005). Although these methods all have their own serious limitations, they are in some sense more transparent and visualizable; they are rooted in rather different assumptions than model-based analyses and so provide a good double-check; and the mere exercise of trying to identify how to test a model and visualize data by traditional means is useful for developing intuitions about what features of the data a model-based analysis may be picking up.

Ultimately, in our view, the methods of computational model fitting discussed here are exceptionally promising and flexible tools for asking many novel questions at a much more detailed and quantitative level than previously possible. The preceding review aimed to provide readers the tools to apply these methods to their own experimental questions. But like all scientific methods, they are most useful in the context of converging evidence from a range of approaches.

Acknowledgements

The author is supported by a Scholar Award from the McKnight Foundation. I am very grateful to Peter Dayan, John O'Doherty, Yael Niv, Aaron Bornstein, Sam Gershman, Dylan Simon, and Larry Maloney for many helpful conversations about the issues covered here.

References

H. Akaike. A new look at the statistical model identification. *IEEE transactions on automatic control*, 19(6): 716–723, 1974.

D. J. Barraclough, M. L. Conroy, and D. Lee. Prefrontal cortex and decision making in a mixed-strategy game. *Nat Neurosci*, 7(4):404–410, 2004.

A. G. Barto. Adaptive critics and the basal ganglia. In J. C. Houk, J. L. Davis, and D. G. Beiser, editors, *Models of Information Processing in the Basal Ganglia*, pages 215–232. MIT Press, Cambridge, MA, 1995.

H. M. Bayer and P. W. Glimcher. Midbrain dopamine neurons encode a quantitative reward prediction error signal. *Neuron*, 47:129–141, 2005.

T. E. J. Behrens, M. W. Woolrich, M. E. Walton, and M. F. S. Rushworth. Learning the value of information in an uncertain world. *Nat Neurosci*, 10(9):1214–1221, 2007.

D. P. Bertsekas and J. N. Tsitsiklis. *Neuro-Dynamic Programming*. Athena Scientific, Belmont, MA, 1996.

C. Bhat. Quasi-random maximum simulated likelihood estimation of the mixed multinomial logit model. *Transportation Research Part B*, 35(7):677–693, 2001.

C. Bishop. *Pattern recognition and machine learning*. Springer New York:, 2006.

E. Brandstatter, G. Gigerenzer, and R. Hertwig. The priority heuristic: Making choices without trade-offs. *Psychological Review*, 113(2):409–432, 2006.

C. Camerer and T. Ho. Experience-weighted attraction learning in coordination games: Probability rules, heterogeneity, and time-variation. *Journal of Mathematical Psychology*, 42(2-3):305–326, 1998.

C. Camerer and T. Ho. Experience-weighted attraction learning in games: A unifying approach. *Econometrica*, 67(4):827–74, 1999.

N. D. Daw and K. Doya. The computational neurobiology of learning and reward. *Current Opinion in Neurobiology*, 16:199–204, 2006.

N. D. Daw, Y. Niv, and P. Dayan. Uncertainty-based competition between prefrontal and dorsolateral striatal systems for behavioral control. *Nature Neuroscience*, 8:1704–1711, 2005.

N. D. Daw, J. P. O'Doherty, P. Dayan, B. Seymour, and R. J. Dolan. Cortical substrates for exploratory decisions in humans. *Nature*, 441(7095):876–879, 2006.

N. D. Daw, A. C. Courville, and P. Dayan. Semi-rational models: The case of trial order. In N. Chater and M. Oaksford, editors, *The Probabilistic Mind*. Oxford University Press, 2008.

P. Dayan and L. F. Abbott. *Theoretical Neuroscience: Computational and Mathematical Modeling of Neural Systems*. MIT Press, Cambridge, MA, 2001.

P. Dayan and Y. Niv. Reinforcement learning: the good, the bad and the ugly. *Curr Opin Neurobiol*, 18(2): 185–196, 2008.

M. R. Delgado, L. E. Nystrom, C. Fissell, D. C. Noll, and J. A. Fiez. Tracking the hemodynamic responses to reward and punishment in the striatum. *Journal of Neurophysiology*, 84:3072–3077, 2000.

M. J. Frank, L. C. Seeberger, and R. C. O'Reilly. By carrot or by stick: cognitive reinforcement learning in parkinsonism. *Science*, 306(5703):1940–1943, 2004.

M. J. Frank, A. A. Moustafa, H. M. Haughey, T. Curran, and K. E. Hutchison. Genetic triple dissociation reveals multiple roles for dopamine in reinforcement learning. *Proc Natl Acad Sci U S A*, 104(41):16311–16316, Oct 2007. doi: 10.1073/pnas.0706111104. URL `http://dx.doi.org/10.1073/pnas.0706111104`.

D. Freedman. On the so-called "Huber sandwich estimator" and" robust standard errors". *American Statistician*, 60(4):299–302, 2006.

K. J. Friston and W. Penny. Posterior probability maps and spms. *Neuroimage*, 19(3):1240–1249, 2003.

K. J. Friston, P. Fletcher, O. Josephs, A. Holmes, M. D. Rugg, and R. Turner. Event-related fmri: characterizing differential responses. *Neuroimage*, 7(1):30–40, 1998.

K. J. Friston, K. E. Stephan, T. E. Lund, A. Morcom, and S. Kiebel. Mixed-effects and fmri studies. *Neuroimage*, 24(1):244–252, 2005.

A. Gelman and J. Hill. *Data analysis using regression and multilevel/hierarchical models*. Cambridge University Press New York, 2007.

A. Gelman, J. Carlin, and H. Stern. *Bayesian data analysis*. CRC press, 2004.

A. N. Hampton, P. Bossaerts, and J. P. O'Doherty. Neural correlates of mentalizing-related computations during strategic interactions in humans. *Proc Natl Acad Sci U S A*, 105(18):6741–6746, 2008.

T. A. Hare, J. O'Doherty, C. F. Camerer, W. Schultz, and A. Rangel. Dissociating the role of the orbitofrontal cortex and the striatum in the computation of goal values and prediction errors. *J Neurosci*, 28(22):5623–5630, 2008.

A. Holmes and K. Friston. Generalisability, random effects & population inference. *Neuroimage*, 7, 1998.

P. Huber. The behavior of maximum likelihood estimates under nonstandard conditions. In *Proceedings of the fifth Berkeley symposium in mathematical statistics*, volume 1, pages 221–233, 1967.

J. W. Kable and P. W. Glimcher. The neural correlates of subjective value during intertemporal choice. *Nat Neurosci*, 10(12):1625–1633, 2007.

S. Kakade and P. Dayan. Acquisition and extinction in autoshaping. *Psychological Review*, 109:533–544, 2002.

R. E. Kass and A. E. Raftery. Bayes factors. *Journal of the American Statistical Association*, 90:7730–795, 1995.

B. Knutson, A. Westdorp, E. Kaiser, and D. Hommer. FMRI visualization of brain activity during a monetary incentive delay task. *NeuroImage*, 12:20–27, 2000.

B. Lau and P. W. Glimcher. Dynamic response-by-response models of matching behavior in rhesus monkeys. *Journal of the Experimental Analysis of Behavior*, 2005. (in press).

J. Li, S. M. McClure, B. King-Casas, and P. R. Montague. Policy adjustment in a dynamic economic game. *PLoS One*, 1:e103, 2006.

T. Lohrenz, K. McCabe, C. F. Camerer, and P. R. Montague. Neural signature of fictive learning signals in a sequential investment task. *Proc Natl Acad Sci U S A*, 104(22):9493–9498, 2007.

D. MacKay. *Information theory, inference, and learning algorithms*. Cambridge Univ Pr, 2003.

S. M. McClure, G. S. Berns, and P. R. Montague. Temporal prediction errors in a passive learning task activate human striatum. *Neuron*, 38(2):339–346, 2003.

D. McFadden. Conditional logit analysis of qualitative choice behavior. In P. Zarembka, editor, *Frontiers in Econometrics*, pages 105–142, New York, 1974. Academic Press.

A. Ng and M. Jordan. PEGASUS: A policy search method for large MDPs and POMDPs. In *Proceedings of the sixteenth Conference on Uncertainty in Artificial Intelligence*, pages 406–415, 2000.

K. A. Norman, S. M. Polyn, G. J. Detre, and J. V. Haxby. Beyond mind-reading: multi-voxel pattern analysis of fmri data. *Trends Cogn Sci*, 10(9):424–430, 2006.

J. P. O'Doherty, P. Dayan, K. Friston, H. Critchley, and R. J. Dolan. Temporal difference models and reward-related learning in the human brain. *Neuron*, 38(2):329–337, 2003.

J. P. O'Doherty, A. Hampton, and H. Kim. Model-based fmri and its application to reward learning and decision making. *Ann N Y Acad Sci*, 1104:35–53, 2007.

W. Penny and K. Friston. Hierarchical models. *Human brain function (2nd ed., pp. 851–863). London: Elsevier*, 2004.

M. Pessiglione, P. Petrovic, J. Daunizeau, S. Palminteri, R. J. Dolan, and C. D. Frith. Subliminal instrumental conditioning demonstrated in the human brain. *Neuron*, 59(4):561–567, 2008.

H. Plassmann, J. O'Doherty, B. Shiv, and A. Rangel. Marketing actions can modulate neural representations of experienced pleasantness. *Proc Natl Acad Sci U S A*, 105(3):1050–1054, 2008.

M. L. Platt and P. W. Glimcher. Neural correlates of decision variables in parietal cortex. *Nature*, 400(6741): 233–238, 1999.

K. Samejima, K. Doya, Y. Ueda, and M. Kimura. Estimating internal variables and parameters of a learning agent by a particle filter. *Advances in Neural Information Processing Systems*, 16, 2004.

K. Samejima, Y. Ueda, K. Doya, and M. Kimura. Representation of action-specific reward values in the striatum. *Science*, 310(5752):1337–1340, 2005.

T. Schonberg, N. D. Daw, D. Joel, and J. P. O'Doherty. Reinforcement learning signals in the human striatum distinguish learners from nonlearners during reward-based decision making. *J Neurosci*, 27(47):12860–12867, 2007.

W. Schultz, P. Dayan, and P. R. Montague. A neural substrate of prediction and reward. *Science*, 275: 1593–1599, 1997.

G. Schwarz. Estimating the dimension of a model. *Annals of Statistics*, 6:461–464, 1978.

B. Seymour, N. Daw, P. Dayan, T. Singer, and R. Dolan. Differential encoding of losses and gains in the human striatum. *J Neurosci*, 27(18):4826–4831, 2007.

K. E. Stephan, W. D. Penny, J. Daunizeau, R. J. Moran, and K. J. Friston. Bayesian model selection for group studies. *Neuroimage*, 46(4):1004–1017, 2009.

L. P. Sugrue, G. S. Corrado, and W. T. Newsome. Matching behavior and the representation of value in the parietal cortex. *Science*, 304:1782–1787, 2004.

R. S. Sutton and A. G. Barto. *Reinforcement Learning: An Introduction*. MIT Press, Cambridge, MA, 1998.

S. M. Tom, C. R. Fox, C. Trepel, and R. A. Poldrack. The neural basis of loss aversion in decision-making under risk. *Science*, 315(5811):515–518, 2007.

C. J. C. H. Watkins. *Learning from Delayed Rewards*. PhD thesis, Cambridge University, Cambridge, England, 1989.

B. C. Wittmann, N. D. Daw, B. Seymour, and R. J. Dolan. Striatal activity underlies novelty-based choice in humans. *Neuron*, 58(6):967–973, 2008.