Adaptive stimulus selection for multi-alternative psychometric functions with lapses

Ji Hyun Bak

School of Computational Sciences,
Korea Institute for Advanced Study, Seoul, Korea
Department of Physics, Princeton University,
Princeton, NJ, USA

Jonathan W. Pillow

Department of Psychology and Princeton Neuroscience Institute,
Princeton University, Princeton, NJ, USA

Psychometric functions (PFs) quantify how external stimuli affect behavior, and they play an important role in building models of sensory and cognitive processes. Adaptive stimulus-selection methods seek to select stimuli that are maximally informative about the PF given data observed so far in an experiment and thereby reduce the number of trials required to estimate the PF. Here we develop new adaptive stimulus-selection methods for flexible PF models in tasks with two or more alternatives. We model the PF with a multinomial logistic regression mixture model that incorporates realistic aspects of psychophysical behavior, including lapses and multiple alternatives for the response. We propose an information-theoretic criterion for stimulus selection and develop computationally efficient methods for inference and stimulus selection based on adaptive Markov-chain Monte Carlo sampling. We apply these methods to data from macaque monkeys performing a multi-alternative motion-discrimination task and show in simulated experiments that our method can achieve a substantial speed-up over random designs. These advances will reduce the amount of data needed to build accurate models of multi-alternative PFs and can be extended to high-dimensional PFs that would be infeasible to characterize with standard methods.

Introduction

Understanding the factors governing psychophysical behavior is a central problem in neuroscience and psychology. Although accurate quantification of the behavior is an important goal in itself, psychophysics provides an important tool for interrogating the mechanisms governing sensory and cognitive processing in the brain. As new technologies allow direct manipulations of neural activity in the brain, there is a growing need for methods that can characterize rapid changes in psychophysical behavior.

In a typical psychophysical experiment, an observer is trained to report judgments about a sensory stimulus by selecting a response from among two or more alternatives. The observer is assumed to have an internal probabilistic rule governing these decisions; this probabilistic map from stimulus to response is called the observer's *psychometric function*. Because the psychometric function is not directly observable, it must be inferred from multiple observations of stimulus–response pairs. However, such experiments are costly due to the large numbers of trials typically required to obtain good estimates of the psychometric function. Therefore, a problem of major practical importance is to develop efficient experimental designs that can minimize the amount of data required to accurately infer an observer's psychometric function.

Bayesian adaptive stimulus selection

A powerful approach for improving the efficiency of psychophysical experiments is to design the data-collection process so that the stimulus is adaptively selected on each trial by maximizing a suitably defined objective function (MacKay, 1992). Such methods are known by a variety of names, including active learning, adaptive or sequential optimal experimental design, and closed-loop experiments.

Bayesian approaches to adaptive stimulus selection define optimality of a stimulus in terms of its ability to improve a posterior distribution over the psychometric function, for example by reducing variance or entropy of the posterior. The three key ingredients of a

Citation: Bak, J. H., & Pillow, J. W. (2018). Adaptive stimulus selection for multi-alternative psychometric functions with lapses. *Journal of Vision*, *18*(12):4, 1–25, https://doi.org/10.1167/18.12.4.

https://doi.org/10.1167/18.12.4 Received February 6, 2018; published November 6, 2018 ISSN 1534-7362 Copyright 2018 The Authors

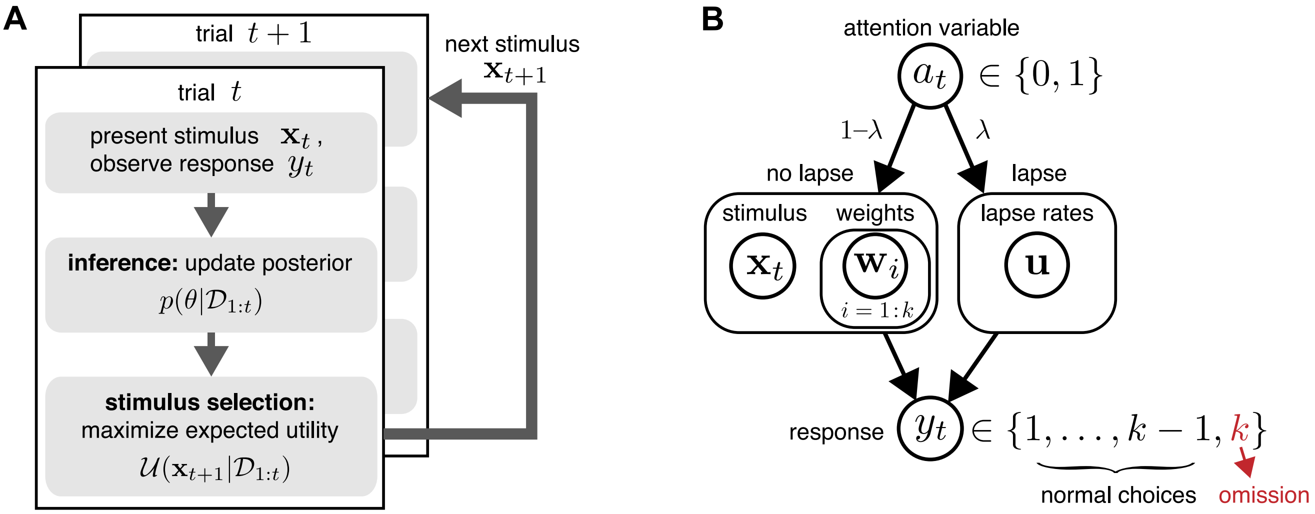

Figure 1. (A) Schematic of Bayesian adaptive stimulus selection. On each trial, a stimulus is presented and the response observed; the posterior over the parameters θ is updated using all data collected so far in the experiment $\mathcal{D}_t$; and the stimulus that maximizes the expected utility (in our case, information gain) is selected for the next trial. (B) A graphical model illustrating a hierarchical psychophysical-observer model that incorporates lapses as well as the possibility of omissions. On each trial, a latent attention or lapse variable a_t is drawn from a Bernoulli distribution with parameter λ, to determine whether the observer attends to the stimulus $\mathbf{x}_t$ on that trial or lapses. With probability $1-\lambda$, the observer attends to the stimulus ($a_t = 0$) and the response y_t is drawn from a multinomial logistic regression model, where the probability of choosing option i is proportional to $\exp(\mathbf{w}_i^\top \mathbf{x}_t)$. With probability λ, the observer lapses ($a_t = 1$) and selects a choice from a (stimulus-independent) response distribution governed by parameter vector $\mathbf{u}$. So-called omission trials, in which the observer does not select one of the valid response options, are modeled with an additional response category $y_t = k$.

Bayesian adaptive stimulus-selection method are (Chaloner & Verdinelli, 1995; Pillow & Park, 2016):

- **model**, which parametrizes the psychometric function of interest;
- **prior**, which captures initial beliefs about model parameters; and
- **utility function**, which quantifies the usefulness of a hypothetical stimulus–response pair for improving the posterior.

Sequential algorithms for adaptive Bayesian experiments rely on repeated application of three basic steps: data collection (stimulus presentation and response measurement); inference (posterior updating using data from the most recent trial); and selection of an optimal stimulus for the next trial by maximizing expected utility (see Figure 1A). The inference step involves updating the posterior distribution over the model parameters according to Bayes's rule with data from the most recent trial. Stimulus selection involves calculating the expected utility (i.e., the expected improvement in the posterior) for a set of candidate stimuli, averaging over the responses that might be elicited for each stimulus, and selecting the stimulus for which the expected utility is highest. Example utility functions include the negative trace of the posterior covariance (corresponding to the sum of the posterior variances for each parameter) and the mutual information or information gain (which corresponds to minimizing the entropy of the posterior).

Methods for Bayesian adaptive stimulus selection have been developed over several decades in a variety of different disciplines. If we focus on the specific application of estimating psychometric functions, the field goes back to the QUEST (A. B. Watson & Pelli, 1983) and ZEST (King-Smith, Grigsby, Vingrys, Benes, & Supowit, 1994) algorithms, which are focused on the estimation of discrimination thresholds, and to the simple case of 1-D stimulus and binary responses (Treutwein, 1995). The Ψ method (Kontsevich & Tyler, 1999) uses Bayesian inference for estimating both the threshold and slope of a psychometric function, which have been extended to 2-D stimuli by Kujala and Lukka (2006). Further development of the method allowed for adaptive estimation of more complex psychometric functions, where the parameters are no longer limited to a threshold and a slope (Barthelmé & Mamassian, 2008; Kujala & Lukka, 2006; Lesmes, Lu, Baek, & Albright, 2010; Prins, 2013) and may exhibit statistical dependencies (Vul, Bergsma, & MacLeod, 2010). Models with multidimensional stimuli have also been considered (DiMattina, 2015; Kujala & Lukka, 2006; A. B. Watson, 2017).

Different models have been used to describe the psychometric function. Standard choices include the logistic regression model (Chaloner & Larntz, 1989; DiMattina, 2015; Zocchi & Atkinson, 1999), the

Weibull distribution function (A. B. Watson & Pelli, 1983), and the Gaussian cumulative distribution function (Kontsevich & Tyler, 1999). More recent work has considered Gaussian process regression models (Gardner, Song, Weinberger, Barbour, & Cunningham, 2015). Most previous work, however, has been limited to the case of binary responses.

Bayesian methods for inferring psychometric functions (Kuss, Jäkel, & Wichmann, 2005; Prins, 2012; Wichmann & Hill, 2001a, 2001b) have enlarged the space of statistical models that can be used to describe psychophysical phenomena based on (often limited) data. A variety of recent advances have arisen in sensory neuroscience or neurophysiology, driven by the development of efficient inference techniques for neural encoding models (Lewi, Butera, & Paninski, 2009; M. Park, Horwitz, & Pillow, 2011) or model comparison and discrimination methods (Cavagnaro, Myung, Pitt, & Kujala, 2010; DiMattina & Zhang, 2011; Kim, Pitt, Lu, Steyvers, & Myung, 2014). These advances can in many cases be equally well applied to psychophysical experiments.

Our contributions

In this article, we develop methods for adaptive stimulus selection in psychophysical experiments that are applicable to realistic models of human and animal psychophysical behavior. First, we describe a psychophysical model that incorporates multiple response alternatives and lapses, in which the observer makes a response that does not depend on the stimulus. This model can also incorporate omission trials, where the observer does not make a valid response (e.g., breaking fixation before the go cue), by considering them as an additional response category. Second, we describe efficient methods for updating the posterior over the model parameters after every trial. Third, we introduce two algorithms for adaptive stimulus selection, one based on a Gaussian approximation to the posterior and a second based on Markov-chain Monte Carlo (MCMC) sampling. We apply these algorithms to simulated data and to real data analyzed with simulated closed-loop experiments and show that they can substantially reduce the number of trials required to estimate multi-alternative psychophysical functions.

Psychophysical-observer model

Here we describe a flexible model of psychometric functions (PFs) based on the multinomial logistic (MNL) response model (Glonek & McCullagh, 1995). We show how omission trials can be naturally incorporated into a model as one of the multiple response alternatives. We then develop a hierarchical extension of the model that incorporates lapses (see Figure 1B).

Multinomial logistic response model

We consider the setting where the observer is presented with a stimulus $\mathbf{x} \in \mathbb{R}^d$ and selects a response $y \in \{1, \ldots, k\}$ from one of k discrete choices on each trial. We will assume the stimulus is represented internally by some (possibly nonlinear) feature vector $\boldsymbol{\phi}(\mathbf{x})$, which we will write simply as $\boldsymbol{\phi}$ for notational simplicity.

In the MNL model, the probability p_i of each possible outcome $i \in \{1, \ldots, k\}$ is determined by the dot product between the feature $\boldsymbol{\phi}$ and a vector of weights $\mathbf{w}_i$ according to

$$p_i = \frac{\exp(\mathbf{w}_i^\top \boldsymbol{\phi})}{\sum_{j=1}^{k} \exp(\mathbf{w}_j^\top \boldsymbol{\phi})}, \quad (1)$$

where the denominator ensures that these probabilities sum to 1, $\sum_{i=1}^{k} p_i = 1$. The function from stimulus to a probability vector over choices, $\mathbf{x} \mapsto (p_1, \ldots, p_k)$, is the psychometric function, and the set of weights $\{\mathbf{w}_i\}_{i=1}^{k}$ its parameters. Note that the model is overparameterized when written this way, since the requirement that probabilities sum to 1 removes one degree of freedom from the probability vector. Thus, we can without loss of generality fix one of the weight vectors to zero, for example $\mathbf{w}_k = \mathbf{0}$, so that the denominator in Equation 1 becomes $z = 1 + \sum_{j=1}^{k} \exp(\mathbf{w}_j^\top \boldsymbol{\phi})$ and $p_k = 1/z$.

We consider the feature vector $\boldsymbol{\phi}$ to be a known function of the stimulus $\mathbf{x}$, even when the dependence is not written explicitly. For example, we can consider a simple form of feature embedding, $\boldsymbol{\phi}(\mathbf{x}) = [1, \mathbf{x}^\top]^\top$, corresponding to a linear function of the stimulus plus an offset. In this case, the weights for the ith choice would correspond to $\mathbf{w}_i = [b_i, \mathbf{a}_i^\top]^\top$, where b_i is the offset or bias for the ith choice and $\mathbf{a}_i$ are the linear weights governing sensitivity to $\mathbf{x}$. The resulting choice probability has the familiar form $p_i \propto \exp(b_i + \mathbf{a}_i^\top \mathbf{x})$. Nonlinear stimulus dependencies can be incorporated by including nonlinear functions of $\mathbf{x}$ in the feature vector $\boldsymbol{\phi}(\mathbf{x})$ (Knoblauch & Maloney, 2008; Murray, 2011; Neri & Heeger, 2002). Dependencies on the trial history, such as the previous stimulus or reward, may also be included as additional features in $\boldsymbol{\phi}$ (see, e.g., Bak, Choi, Akrami, Witten, & Pillow, 2016).

It is useful to always work with a normalized stimulus space, in which the mean of each stimulus component x_α over the stimulus space is $\langle x_\alpha \rangle = 0$ and

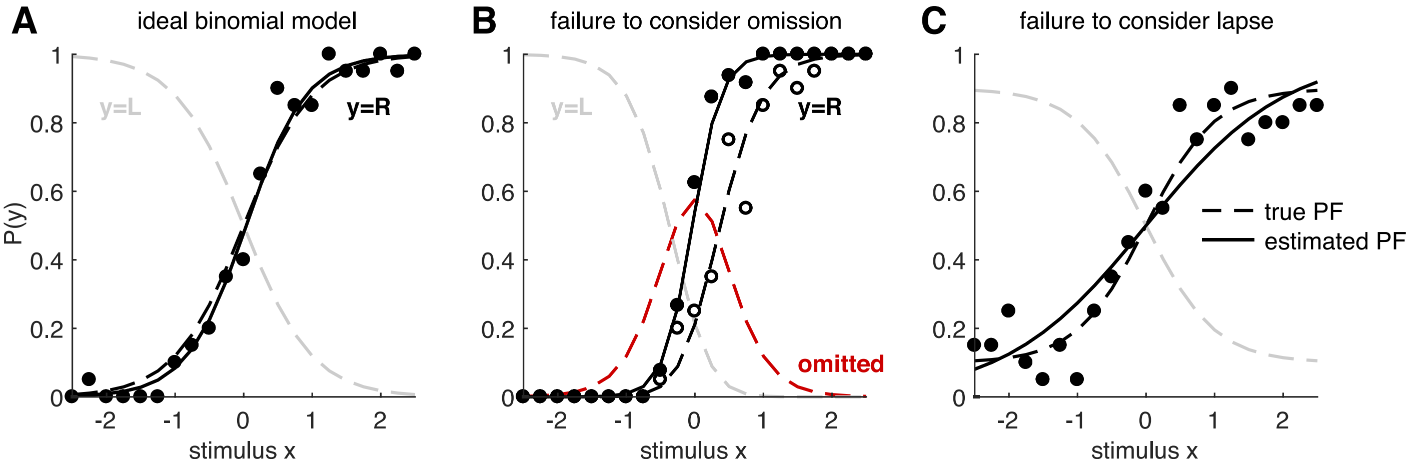

Figure 2. Effects of omission and lapse. Here we illustrate the undesirable effects of failing to take into account omission and lapse. (A) If the psychometric function (PF) follows an ideal binomial logistic model, it can be estimated very well from data. The black dashed line shows the true PF for one of the two responses (say $y = R$) and the gray dashed line shows the true PF for the other (say $y = L$), such that the two dashed curves always add up to 1. The black dots indicate the mean probability of observing this response $y = R$ at each stimulus point x. We drew 20 observations per stimulus point, at each of the 21 stimulus points along the one-dimensional axis. The resulting estimate for $P(y = 1)$ is shown by the solid black line. The inference method is not important for the current purpose, but we used the maximum a posteriori estimate. (B) Now suppose that some trials fell into the implicit third choice, which is omission (red dashed line). The observed probability of $y = R$ at each stimulus point (open black circles) follows the true PF (black dashed line). But if the omitted trials are systematically excluded from analysis, as in common practice, the estimated PF (solid black line) reflects a biased set of observations (filled black circles) and fails to recover the true PF. (C) When there is a finite lapse rate (we used a total lapse of $\lambda = 0.2$, uniformly distributed to the two outcomes), the true PF (dashed black line) asymptotes to a finite offset from 0 or 1. If the resulting observations (black dots) are fitted to a plain binomial model without lapse, the slope of the estimated PF (solid black line) is systematically biased.

the standard deviation $\mathrm{std}(x_\alpha) = 1$. This normalization ensures that the values of the weight parameters are defined in more interpretable ways. The zero-mean condition ensures that the bias b is the expectation value of log probability over all possible stimuli. The unit-variance condition means that the effect of moving a certain distance along one dimension of the weight space is comparable to moving the same distance in another dimension, again averaged over all possible stimuli. In other words, we are justified in using the same unit along all dimensions of the weight space.

Including omission trials

Even in binary tasks with only two possible choices per trial, there is often an implicit third choice, which is to make no response, make an illegal response, or interrupt the trial at some point before the allowed response period. For example, animals are often required to maintain an eye position or a nose poke or wait for a "go" cue before reporting a choice. Trials on which the animal fails to obey these instructions are commonly referred to as *omissions* or *violations* and are typically discarded from analysis. However, failure to take these trials into account may bias the estimates of the PF if these trials are more common for some stimuli than others (see Figure 2B).

The multinomial response model provides a natural framework for incorporating omission trials because it accommodates an arbitrary number of response categories. Thus we can model omissions explicitly as one of the possible choices the observer can choose from, or as response category $k + 1$ in addition to the k valid responses. One could even consider different kinds of omissions separately—for example, allowing choice $k + 1$ to reflect fixation-period violations and choice $k + 2$ to reflect failure to report a choice during the response window. Henceforth, we will let k reflect the total number of choices including omission, as illustrated in Figure 1B.

This formulation can also be useful for the rated yes/no task in human psychophysics, where a "not sure" response is explicitly presented (C. S. Watson, Kellogg, Kawanishi, & Lucas, 1973). Although such a model was considered for adaptive stimulus selection (Lesmes et al., 2015), the third alternative was not handled as a fully independent choice, as the goal was only to estimate the two detection thresholds separately: one for a strict yes, another for a collapsed response of either yes or not sure. Our model treats each of the multiple alternatives equivalently.

Modeling lapses with a mixture model

Another important feature of real psychophysical observers is the tendency to occasionally make errors that are independent of the stimulus. Such choices, commonly known as *lapses* or *button-press errors*, may reflect lapses in concentration or memory of the response mapping (Kuss et al., 2005; Wichmann & Hill, 2001a). Lapses are most easily identified by errors on easy trials—that is, trials that should be performed perfectly if the observer is paying attention.

Although lapse rates can be negligible in highly trained observers (Carandini & Churchland, 2013), they can be substantially greater than zero in settings involving nonprimates or complicated psychophysical tasks. Lapses affect the PF by causing it to saturate above 0 and below 1, so that perfect performance is never achieved even for the easiest trials. Failure to incorporate lapses into the PF model may therefore bias estimates of sensitivity, as quantified by PF slope or threshold (illustrated in Figure 2C; also see Prins, 2012; Wichmann and Hill, 2001a, 2001b).

To model lapses, we use a mixture model that treats the observer's choice on each trial as coming from one of two probability distributions: a stimulus-dependent one (governed by the MNL model) or a stimulus-independent one (reflecting a fixed probability of choosing any option when lapsing). Simpler versions of such mixture model have been proposed previously (Kuss et al., 2005).

Figure 1B shows a schematic of the resulting model. On each trial, a Bernoulli random variable $a \sim \mathrm{Ber}(\lambda)$ governs whether the observer lapses: With probability λ the observer lapses (i.e., ignores the stimulus), and with probability $1-\lambda$ the observer attends to the stimulus. If the observer lapses ($a=1$), the response is drawn according to the fixed-probability distribution $(c_1, \ldots, c_k)$ governing the probability of selecting options 1 to k, where $\sum c_i = 1$. If the observer does not lapse ($a=0$), the response is selected according to the MNL model. Under this model, the conditional probability of choosing option i given the stimulus can be written as

$$p_i = (1-\lambda)q_i + \lambda c_i, \qquad q_i = \frac{\exp(\mathbf{w}_i^\top \boldsymbol{\phi})}{\sum_j \exp(\mathbf{w}_j^\top \boldsymbol{\phi})}, \quad (2)$$

where q_i is the lapse-free probability under the classical MNL model (Equation 1).

It is convenient to reparameterize this model so that λc_i, the conditional probability of choosing the ith option due to a lapse, is written

$$\lambda c_i = \frac{\exp(u_i)}{1+\sum_j \exp(u_j)}, \quad (3)$$

where each auxiliary lapse parameter u_i is proportional to the log probability of choosing option i due to lapse. The lapse-conditional probabilities c_i of each choice and the total lapse probability λ are respectively

$$c_i = \frac{\exp(u_i)}{\sum_j \exp(u_j)}, \qquad \lambda = \sum_i \frac{\exp(u_i)}{1+\sum_j \exp(u_j)}. \quad (4)$$

Because each u_i lives on the entire real line, fitting can be carried out with unconstrained optimization methods, although adding reasonable constraints may improve performance in some cases. The full parameter vector of the resulting model is $\boldsymbol{\theta} = [\mathbf{w}^\top,\ \mathbf{u}^\top]^\top$, which includes k additional lapse parameters $\mathbf{u} = \{u_1, \ldots, u_k\}$. Note that in some cases it might be desirable to assume that lapse choices obey a uniform distribution, where the probability of each option is $c_i = 1/k$. For this simplified uniform-lapse model we need only a single lapse parameter u. Note that we have unified the parameterizations of the lapse rate (deviation of the upper asymptote of the PF from 1; in this case, $\lambda - \lambda c_i$) and the guess rate (deviation of the lower asymptote from 0; in this case, λc_i), which have often been modeled separately in previous works with two-alternative responses (Schütt, Harmeling, Macke, & Wichmann, 2016; Wichmann & Hill, 2001a, 2001b). Here they are written in terms of a single family of parameters $\{u_i\}$ and extended naturally to multi-alternative responses.

Our model provides a general and practical parameterization of PFs with lapses. Although previous work has considered the problem of modeling lapses in psychophysical data, much of it assumed a uniform-lapse model, where all options are equally likely during lapses. Earlier approaches have often assumed either that the lapse probability was known a priori (Kontsevich & Tyler, 1999) or was fitted by a grid search over a small set of candidate values (Wichmann & Hill, 2001a). Here we instead infer individual lapse probabilities for each response option, similar to recent approaches described by Kuss et al. (2005), Prins (2012, 2013), and Schütt et al. (2016). Importantly, our method infers the full parameter $\boldsymbol{\theta}$ that includes both the weight and lapse parameters, rather than treating the lapse separately. In particular, our parameterization (Equation 3) has the advantage that there is no need to constrain the support of the lapse parameters u_i. These parameters' relationship to lapse probabilities c_i takes the same (softmax) functional form as the MNL model, placing both sets of parameters on an equal footing.

Before closing this section, we would like to reflect briefly on the key differences between omissions and lapses. First, although omissions and lapses both reflect errors in decision making, omissions are defined as invalid responses and are thus easily identifiable from the data; lapses, on the other hand, are indistinguishable from normal responses, and are identifiable only from the fact that the psychometric function does not saturate at 0 or 1. Second, modeling omissions as a response category under the MNL model means that the probability of omission is stimulus dependent (e.g., more likely to arise on trials with high difficulty, or generally when the evidence for other options is low). Even if the omissions are not stimulus dependent, and are instead governed entirely by a bias parameter, the probability of omission will still be higher when the evidence for other choices is low or lower when the evidence for other choices is high. Omissions that arise in a purely stimulus-independent fashion, on the other hand, will be modeled as arising from the lapse parameter associated with the omission response category. Omissions can thus arise in two ways under the model: as categories selected under the multinomial model or as lapses arising independent of the stimulus and other covariates.

Posterior inference

Bayesian methods for adaptive stimulus selection require the posterior distribution over model parameters given the data observed so far in an experiment. The posterior distribution results from the combination of two ingredients: a prior distribution $p(\boldsymbol{\theta})$, which captures prior uncertainty about the model parameters $\boldsymbol{\theta}$, and a likelihood function $p(\{y_s\}|\{\mathbf{x}_s\}, \boldsymbol{\theta})$, which captures information about the parameters from the data $\{(\mathbf{x}_s, y_s)\}$, where $s = 1, \ldots, t$ consists of stimulus–response pairs observed up to the current time bin t.

Unfortunately, the posterior distribution for our model has no analytic form. We therefore describe two methods for approximate posterior inference: one relying on a Gaussian approximation to the posterior, known as the Laplace approximation, and a second one based on MCMC sampling.

Prior

The prior distribution specifies our beliefs about model parameters before we have collected any data, and serves to regularize estimates obtained from small amounts of data—for example, by shrinking estimated weights toward zero. Typically we want the prior to be weak enough that the likelihood dominates the posterior for reasonable-sized data sets. However, the choice of prior is especially important in adaptive stimulus-selection settings, because it determines the effective volume of the search space (M. Park & Pillow, 2012; M. Park, Weller, Horwitz, & Pillow, 2014). For example, if the weights are known to exhibit smoothness, then a correlated or smoothness-inducing prior can improve the performance of adaptive stimulus selection because the effective size (or entropy) of the parameter space is much smaller than under an independent prior (M. Park & Pillow, 2012).

In this study, we use a generic independent, zero-mean Gaussian prior over the weight vectors

$$p(\mathbf{w}_i) = \mathcal{N}(\mathbf{0}, \sigma^2 I), \quad (5)$$

for all $i \in (1, \ldots, k)$, with a fixed standard deviation σ. This choice of prior is appropriate when the regressors $\{\mathbf{x}\}$ are standardized, since any single weight can take values that allow for a range of PF shapes along that axis, from flat ($w = 0$) to steeply decreasing ($w = -2\sigma$) or increasing ($w = +2\sigma$). We used $\sigma = 3$ in the simulated experiments in Results. For the lapse parameters $\{u_i\}$, we used a uniform prior over the range [log(0.001), 0] with the natural log, so that each lapse probability λc_i is bounded between 0.001 and 1/2. We set the lower range constraint below $1/N$, where $N = 100$ is the number of observed trials in our simulations, since we cannot reasonably infer lapse probabilities with precision finer than $1/N$. The upper range constraint gives maximal lapse probabilities of $1/(k+1)$ if all u_i take on the maximal value of 0. Note that our prior is uniform with respect to the rescaled lapse parameters $\{u_i\}$ rather than to the actual lapse rates; projected to the space of the lapse probabilities, given the bounds, the prior increases toward smaller lapse. For a comprehensive study of the effect of different priors on lapse, see Schütt et al. (2016).

PF likelihood

The likelihood is the conditional probability of the data as a function of the model parameters. Although we have thus far considered the response variable y to be a scalar taking values in the set $\{1, \ldots, k\}$, it is more convenient to use a "one-hot" or "1-of-k" representation, in which the response variable $\mathbf{y}$ for each trial is a vector of length k with one 1 and $k - 1$ zeros; the position of the 1 in this vector indicates the category selected. For example, in a task with four possible options per trial, a response vector $\mathbf{y} = [0\ 0\ 1\ 0]$ indicates a trial on which the observer selected the third option.

With this parameterization, the log-likelihood function for a single trial can be written

$$\begin{aligned}\log p(\mathbf{y}|\mathbf{x}, \boldsymbol{\theta}) &= \sum_i y_i \log p_i(\mathbf{x}, \boldsymbol{\theta}) \\ &= \mathbf{y}^\top \log \mathbf{p}(\mathbf{x}, \boldsymbol{\theta}), \quad (6)\end{aligned}$$

where $p_i(\mathbf{x}, \boldsymbol{\theta})$ denotes the probability $p(y_i = 1|\mathbf{x}, \boldsymbol{\theta})$ under the model (Equation 1), and $\mathrm{p}(\mathbf{x}, \boldsymbol{\theta}) \equiv [p_1(\mathbf{x}, \ \boldsymbol{\theta}), \ldots, p_k(\mathbf{x}, \ \boldsymbol{\theta})]^\top$ denotes the vector of probabilities for a single trial.

In the classical (lapse-free) MNL model, where $\boldsymbol{\theta} = \{\mathbf{w}_i\}$, the log likelihood is a concave function of $\boldsymbol{\theta}$, which guarantees that numerical optimization of the log likelihood will find a global optimum. With a finite lapse rate, however, the log likelihood is no longer concave (see Appendix A).

Posterior distribution

The log-posterior can be written as the sum of log prior and log likelihood summed over trials, plus a constant:

$$\log p(\boldsymbol{\theta}|\mathcal{D}_t) = \log p(\boldsymbol{\theta}) + \sum_{s=1}^{t} \log p(\mathbf{y}_s|\mathbf{x}_s, \boldsymbol{\theta}) + c, \quad (7)$$

where $\mathcal{D}_t \equiv \{\mathbf{x}_s, y_s\}_{s=1}^{t}$ denotes the accumulated data up to trial t and $c = -\log\left(\int p(\boldsymbol{\theta}) \prod_s p(\mathbf{y}_s|\mathbf{x}_s) d\boldsymbol{\theta}\right)$ is a normalization constant that does not depend on the parameters $\boldsymbol{\theta}$. Because this constant has no tractable analytic form, we rely on two alternate methods for obtaining a normalized posterior distribution.

Inference via Laplace approximation

The Laplace approximation is a well-known Gaussian approximation to the posterior distribution, which can be derived from a second-order Tayler series approximation to the log posterior around its mode (Bishop, 2006).

Computing the Laplace approximation involves a two-step procedure. The first step is to perform a numerical optimization of $\log p(\boldsymbol{\theta}|\mathcal{D}_t)$ to find the posterior mode, or maximum a posteriori (MAP) estimate of $\boldsymbol{\theta}$. This vector, given by

$$\widehat{\boldsymbol{\theta}}_t = \arg\max_{\boldsymbol{\theta}} \log p(\boldsymbol{\theta}) + \sum_{s=1}^{t} \log p(\mathbf{y}_s|\mathbf{x}_s, \boldsymbol{\theta}), \quad (8)$$

provides the mean of the Laplace approximation. Because we can explicitly provide the gradient and Hessian of the log likelihood (see Appendix A) and log prior, this optimization can be carried out efficiently via Newton–Raphson or trust-region methods.

The second step is to compute the second derivative (the Hessian matrix) of the log posterior at the mode, which provides the inverse covariance of the Gaussian. This gives us a local Gaussian approximation of the posterior, centered at the posterior mode:

$$p(\boldsymbol{\theta}|\mathcal{D}_t) \approx \mathcal{N}(\widehat{\boldsymbol{\theta}}_t, C_t), \quad (9)$$

where covariance $C_t = -H_t^{-1}$ is the inverse Hessian of the log posterior, $H_t(i,j) = \partial^2(\log p(\boldsymbol{\theta}|\mathcal{D}_t)/(\partial\theta_i\partial\theta_j)$, evaluated at $\widehat{\boldsymbol{\theta}}_t$.

Note that when the log posterior is concave (i.e., when the model does *not* contain lapses), numerical optimization is guaranteed to find a global maximum of the posterior. Log concavity also strengthens the rationale for using the Laplace approximation, since the true and approximate posterior are both log-concave densities centered on the true mode (Paninski et al., 2010; Pillow, Ahmadian, & Paninski, 2011). When the model incorporates lapses, these guarantees no longer apply globally.

Inference via MCMC sampling

A second approach to inference is to generate samples from the posterior distribution over the parameters via MCMC sampling. Sampling-based methods are typically more computationally intensive than the Laplace approximation but may be warranted when the posterior is not provably log concave (as is the case when lapse rates are nonzero) and therefore not well approximated by a single Gaussian.

The basic idea in MCMC sampling is to set up an easy-to-sample Markov chain that has the posterior as its stationary distribution. Sampling from this chain produces a dependent sequence of posterior samples $\{\boldsymbol{\theta}_m\} \sim p(\boldsymbol{\theta}|\mathcal{D}_t)$, which can be used to evaluate posterior expectations via Monte Carlo integrals:

$$\mathbb{E}[f(\boldsymbol{\theta})] \approx \frac{1}{M}\sum_{m=1}^{M} f(\boldsymbol{\theta}_m), \quad (10)$$

for any function $f(\boldsymbol{\theta})$. The mean of the posterior is obtained from setting $f(\boldsymbol{\theta}) = \boldsymbol{\theta}$, although for adaptive stimulus selection we will be interested in the full shape of the posterior.

The Metropolis–Hastings algorithm is perhaps the simplest and most widely used MCMC sampling method (Metropolis, Rosenbluth, Rosenbluth, Teller, & Teller, 1953). It generates samples via a proposal distribution centered on the current sample (see Appendix B). The choice of proposal distribution is critical to the efficiency of the algorithm, since this governs the rate of mixing, or the number of Markov-chain samples required to obtain independent samples from the posterior distribution (Rosenthal, 2011). Faster mixing implies that fewer samples M are required to obtain an accurate approximation to the posterior.

Here we propose a semiadaptive Metropolis–Hastings algorithm, developed specifically for the current context of sequential learning. Our approach is based on an established observation that the optimal width of the proposal distribution should be proportional to the typical length scale of the distribution being

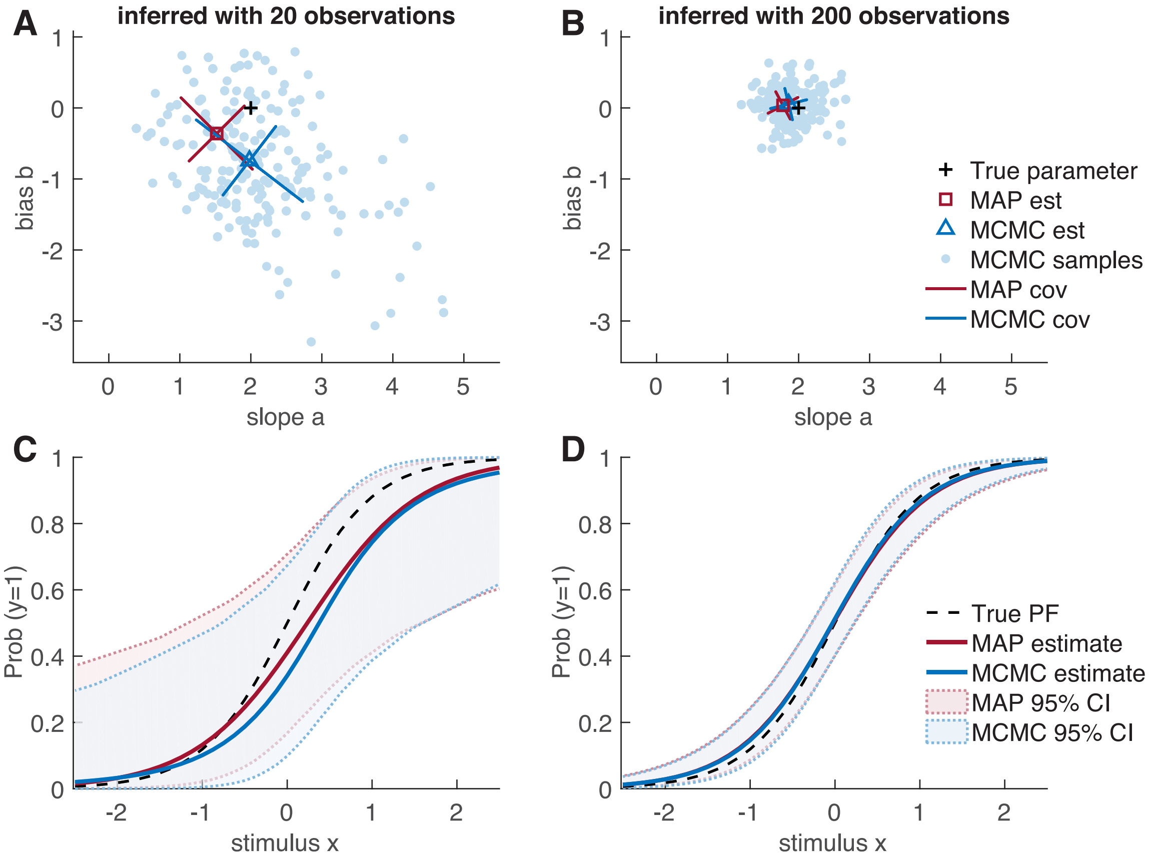

Figure 3. Inferring the psychometric function. Example of a psychometric problem, with a lapse-free binomial logistic model $f(v) = e^v/(1 + e^v)$. Given a 1-D stimulus, a response was drawn from a "true" model $P(y = 1) = f(b + ax)$ with two parameters, slope $a = 2$ and bias $b = 0$. (A–B) On the parameter space, the posterior distributions become sharper (and closer to the true parameter values) as the data-set size N increases. (A) $N = 20$ (small); (b) $N = 200$ (large). For the maximum a posteriori estimate, the mode of the distribution is marked with a square and the two standard deviations ("widths") of its Gaussian approximation with bars. For the Markov-chain Monte Carlo sampling method, all $M = 500$ samples of the chain are shown with dots, the sample mean with a triangle, and the widths with bars. The widths are the standard deviations along the principal directions of the sampled posterior (eigenvectors of the covariance matrix; not necessary aligned with the a–b axes). (C–D) The accuracy of the estimated psychometric function improves with the number of observations N, using either of the two posterior inference methods (MAP or MCMC). (C) $N = 20$ (small); (D) $N = 200$ (large). The two methods are highly consistent in this simple case, especially when N is large enough.

sampled (Gelman, Roberts, & Gilks, 1996; Roberts, Gelman, & Gilks, 1997). Our algorithm is motivated by the adaptive Metropolis algorithm (Haario, Saksman, & Tamminen, 2001), where the proposal distribution is updated at each proposal within a single chain; here we adapt the proposal not within chains but rather after each trial. Specifically, we set the covariance of a Gaussian proposal distribution to be proportional to the covariance of the samples from the previous trial, using the scaling factor of Haario et al. (2001). See Appendix B for details. The adaptive algorithm takes advantage of the fact that the posterior cannot change too much between trials, since it changes only by a single-trial likelihood term on each trial.

Adaptive stimulus-selection methods

As data are collected during the experiment, the posterior distribution becomes narrower due to the fact that each trial carries some additional information about the model parameters (see Figure 3). This narrowing of the posterior is directly related to information gain. A stimulus that produces no expected narrowing of the posterior is, by definition, uninformative about the parameters. On the other hand, a stimulus that (on average) produces a large change in the current posterior is an informative stimulus. Selecting informative stimuli will reduce the number of stimuli required to obtain a narrow posterior, which is the essence of adaptive stimulus-selection methods. In this section, we introduce a precise measure of information gain between a stimulus and the model parameters, and propose an algorithm for selecting stimuli to maximize it.

Infomax criterion for stimulus selection

At each trial, we present a stimulus $\mathbf{x}$ and observe the outcome $\mathbf{y}$. After t trials, the expected gain in information from a stimulus $\mathbf{x}$ is equal to the mutual information between $\mathbf{y}$ and the model parameters $\boldsymbol{\theta}$, given the data $\mathcal{D}_t$ observed so far in the experiment. We denote this conditional *mutual information*:

$$I_t(\boldsymbol{\theta};\mathbf{y}|\mathbf{x}) = \iint d\boldsymbol{\theta}\, d\mathbf{y}\ p(\boldsymbol{\theta},\mathbf{y}|\mathbf{x},\mathcal{D}_t) \times \log \frac{p(\boldsymbol{\theta},\mathbf{y}|\mathbf{x},\mathcal{D}_t)}{p(\boldsymbol{\theta}|\mathcal{D}_t)p(\mathbf{y}|\mathbf{x},\mathcal{D}_t)}, \quad (11)$$

where $p(\boldsymbol{\theta},\mathbf{y}|\mathbf{x},\mathcal{D}_t)$ is the joint distribution of $\boldsymbol{\theta}$ and $\mathbf{y}$ given a stimulus $\mathbf{x}$ and dataset $\mathcal{D}_t$; $p(\boldsymbol{\theta}|\mathcal{D}_t)$ is the current posterior distribution over the parameters from previous trials; and $p(\mathbf{y}|\mathbf{x},\mathcal{D}_t) = \int d\boldsymbol{\theta}\, p(\mathbf{y}|\mathbf{x},\boldsymbol{\theta})p(\boldsymbol{\theta}|\mathcal{D}_t)$ is known as the posterior-predictive distribution of $\mathbf{y}$ given $\mathbf{x}$.

It is useful to note that the mutual information can equivalently be written in two other ways involving Shannon entropy. The first is given by

$$I_t(\boldsymbol{\theta};\mathbf{y}|\mathbf{x}) = H_t(\boldsymbol{\theta}) - H_t(\boldsymbol{\theta}|\mathbf{y};\mathbf{x}), \quad (12)$$

where the first term is the entropy of the posterior at time t,

$$H_t(\boldsymbol{\theta}) = -\int d\boldsymbol{\theta}\, p(\boldsymbol{\theta}|\mathcal{D}_t)\log p(\boldsymbol{\theta}|\mathcal{D}_t), \quad (13)$$

and the second is the conditional entropy of $\boldsymbol{\theta}$ given $\mathbf{y}$,

$$\begin{aligned} H_t(\boldsymbol{\theta}|\mathbf{y};\mathbf{x}) &= -\mathbb{E}_{\boldsymbol{\theta},\mathbf{y}}[\log p(\boldsymbol{\theta}|\mathbf{y},\mathbf{x},\mathcal{D}_t)] \\ &= -\iint d\boldsymbol{\theta}\, d\mathbf{y}\ p(\boldsymbol{\theta},\mathbf{y}|\mathbf{x},\mathcal{D}_t) \\ &\qquad \times \log p(\boldsymbol{\theta}|\mathbf{y},\mathbf{x},\mathcal{D}_t), \quad (14) \end{aligned}$$

which is the entropy of the updated posterior *after* having observed $\mathbf{x}$ and $\mathbf{y}$, averaged over draws of $\mathbf{y}$ from the posterior-predictive distribution. Written this way, the mutual information can be seen as the expected reduction in posterior entropy from a new stimulus–response pair. Moreover, the first term $H_t(\boldsymbol{\theta})$ is independent of the stimulus and response on the current trial, so infomax stimulus selection is equivalent to picking the stimulus that minimizes the expected posterior entropy $H_t(\boldsymbol{\theta}|\mathbf{y};\mathbf{x})$.

A second equivalent expression for the mutual information, which will prove useful for our sampling-based method, is

$$I_t(\boldsymbol{\theta};\mathbf{y}|\mathbf{x}) = H_t(\mathbf{y};\mathbf{x}) - H_t(\mathbf{y}|\boldsymbol{\theta};\mathbf{x}), \quad (15)$$

which is the difference between the marginal entropy of the response distribution conditioned on $\mathbf{x}$,

$$H_t(\mathbf{y};\mathbf{x}) = -\int d\mathbf{y}\, p(\mathbf{y}|\mathbf{x},\mathcal{D}_t)\log p(\mathbf{y}|\mathbf{x},\mathcal{D}_t), \quad (16)$$

and the conditional entropy of the response $\mathbf{y}$ given $\boldsymbol{\theta}$, conditioned on the stimulus,

$$H_t(\mathbf{y}|\boldsymbol{\theta};\mathbf{x}) = -\iint d\mathbf{y}\, d\boldsymbol{\theta}\, p(\boldsymbol{\theta},y|\mathbf{x},\mathcal{D}_t) \times \log p(\mathbf{y}|\mathbf{x},\boldsymbol{\theta}). \quad (17)$$

This formulation shows the mutual information to be equal to the difference between the entropy of the marginal distribution of $\mathbf{y}$ conditioned on $\mathbf{x}$ (with $\boldsymbol{\theta}$ integrated out) and the average entropy of $\mathbf{y}$ given $\mathbf{x}$ and $\boldsymbol{\theta}$, averaged over the posterior distribution of $\boldsymbol{\theta}$. The dual expansion of the mutual information has also been used by Kujala and Lukka (2006).

In a sequential setting where t is the latest trial and $t + 1$ is the upcoming one, the optimal stimulus is the information-maximizing ("infomax") solution:

$$\mathbf{x}_{t+1} = \underset{\mathbf{x}}{\operatorname{argmax}}\, I_t(\boldsymbol{\theta};\mathbf{y}|\mathbf{x}). \quad (18)$$

Figure 4 shows an example of a simulated experiment where the stimulus was selected adaptively following the infomax criterion. Note that our algorithm takes a "greedy" approach of optimizing one trial at a time. For work on optimizing beyond the next trial, see for example Kim, Pitt, Lu, and Myung (2017).

Selecting the optimal stimulus thus requires maximizing the mutual information over the set of all possible stimuli $\{\mathbf{x}\}$. Since each evaluation of the mutual information involves a high-dimensional integral over parameter space and response space, this is a highly computationally demanding task. In the next sections, we present two algorithms for efficient infomax stimulus selection based on each of the two approximate inference methods described previously.

Infomax with Laplace approximation

Calculation of the mutual information is greatly simplified by a Gaussian approximation of the posterior. The entropy of a Gaussian distribution with covariance C is equal to $\frac{1}{2}\log|C|$ up to a constant factor. If we expand the mutual information as in Equation 12 and recall that we need only minimize the expected posterior entropy after observing the re-

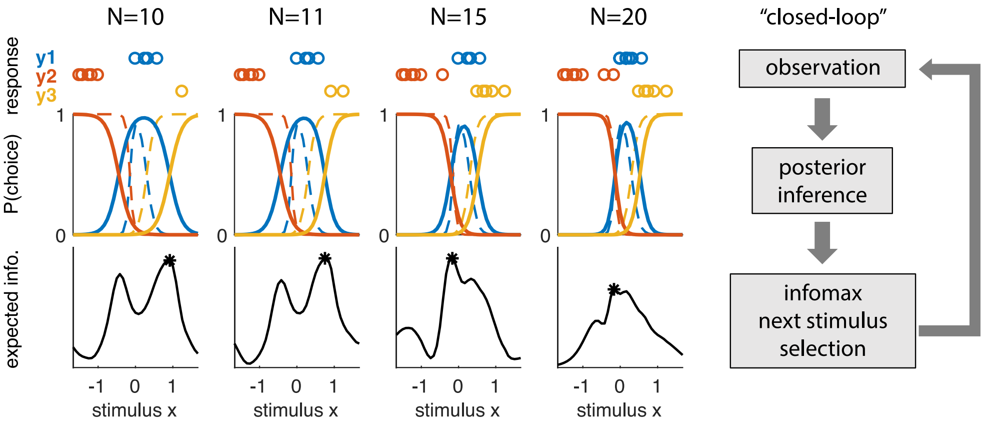

Figure 4. Example of infomax adaptive stimulus selection, simulated with a three-alternative lapse-free model on 1-D stimuli. The figure shows how, given a small set of data (the stimulus–response pairs shown in the top row), the psychometric functions are estimated based on the accumulated data (middle row) and the next stimulus is chosen to maximize the expected information gain (bottom row). Each column shows the instance after the N observations in a single adaptive stimulus-selection sequence, for $N = 10$, 11, 15, and 20, respectively. In the middle row, the estimated psychometric functions (solid lines) quickly approach the true functions (dashed lines) through the adaptive and optimal selection of stimuli. This example was generated using the Laplace approximation–based algorithm, with an independent Gaussian prior over the weights with mean zero and standard deviation $\sigma = 10$.

sponse, the optimal stimulus for time step $t + 1$ is given by

$$\mathbf{x}^*_{t+1} = \underset{\mathbf{x}}{\operatorname{argmin}} \int d\mathbf{y}\, p(\mathbf{y}|\mathbf{x}, \mathcal{D}_t) \log |\tilde{C}(\mathbf{x}, \ \mathbf{y})|, \quad (19)$$

where $\tilde{C}(\mathbf{x}, \mathbf{y})$ is the covariance of the updated (Gaussian) posterior after observing stimulus–response pair ($\mathbf{x}$, $\mathbf{y}$). To evaluate the updated covariance $\tilde{C}(\mathbf{x}, \mathbf{y})$ under the Laplace approximation, we would need to numerically optimize the posterior for $\boldsymbol{\theta}$ for each possible response $\mathbf{y}$ for any candidate stimulus $\mathbf{x}$, which would be computationally infeasible. We therefore use a fast approximate method for obtaining a closed-form update for $\tilde{C}(\mathbf{x}, \mathbf{y})$ from the current posterior covariance C_t, following an approach developed by Lewi et al. (2009). See Appendix C for details. Note that this approximate sequential update is only used for calculating the expected utility of each candidate stimulus by approximating the posterior distribution at the next trial. For obtaining the MAP estimate of the current model parameter $\boldsymbol{\theta}_t$, numerical optimization needs to be performed using the full accumulated data $\mathcal{D}_t$ each time.

Once we have $\log\left|\tilde{C}(\mathbf{x}, \mathbf{y})\right|$ for each given stimulus–observation pair, we numerically sum this over a set of discrete counts $\mathbf{y}$ that are likely under the posterior-predictive distribution. This is done in two steps, by separating the integral in Equation 19 as

$$\int d\mathbf{y}\, p(\mathbf{y}|\mathbf{x}, \mathcal{D}_t) \ \log |\tilde{C}(\mathbf{x}, \mathbf{y})| = \int d\boldsymbol{\theta}_t\, p(\boldsymbol{\theta}_t|\mathcal{D}_t) \int d\mathbf{y}\, p(\mathbf{y}|\mathbf{x}, \boldsymbol{\theta}_t) \ \log |\tilde{C}(\mathbf{x}, \mathbf{y})|. \quad (20)$$

Note that the outer integral is over the current posterior $p(\boldsymbol{\theta}_t|\mathcal{D}_t) \approx \mathcal{N}(\widehat{\boldsymbol{\theta}}_t, C_t)$, which is to be distinguished from the future posterior $p(\boldsymbol{\theta}|\mathbf{y}, \mathbf{x}, \mathcal{D}_t) \approx \mathcal{N}(\widetilde{\boldsymbol{\theta}}(\mathbf{x}, \mathbf{y}), \tilde{C}(\mathbf{x}, \mathbf{y}))$, whose entropy we are trying to minimize. Whereas the inner integral is simply a weighted sum over the set of outcomes $\mathbf{y}$, the outer integral over the parameter $\boldsymbol{\theta}$ is in general challenging, especially when the parameter space is high dimensional. In the case of the standard MNL model that does not include lapse, we can exploit the linear structure of model to reduce this to a lower dimensional integral over the space of the linear predictor, which we evaluate numerically using Gauss–Hermite quadrature (Heiss & Winschel, 2008). (This integral is 1-D for classic logistic regression and has $k - 1$ dimensions for MNL regression with k classes; see Appendix C for details.) When the model incorporates lapses, the full parameter vector $\boldsymbol{\theta} = [\mathbf{w}^\top, \mathbf{u}^\top]^\top$ includes the lapse parameters in addition to the weights $\mathbf{w}$. In this case, our method with Laplace approximation may suffer from reduced accuracy due to the fact that the posterior may be less closely approximated by a Gaussian.

In order to exploit the convenient structure of the reduced integral over the weight space, we choose to maximize the *partial* information $I(\mathbf{w}; \mathbf{y}|\mathbf{x})$ between the observation and the psychophysical weights instead of the full information $I(\boldsymbol{\theta}; \mathbf{y}|\mathbf{x})$. This is a reasonable approximation in many cases where the stimulus-dependent behavior is the primary focus of the psychometric experiment (for a similar approach, see also Prins, 2013). However, we note that this is the only piece in this work where we treat the weights separately from the lapse parameters; posterior inference is still performed for the full parameter $\boldsymbol{\theta}$. Thus for Laplace-based infomax exclusively, the partial covariance $C_{\mathbf{ww}} = -(\partial^2(\log \mathcal{P})/\partial \mathbf{w}^2)^{-1}$ is used in place of the full covariance $C = -(\partial^2(\log \mathcal{P})/\partial \boldsymbol{\theta}^2)^{-1}$, where $\mathcal{P}(\boldsymbol{\theta})$ is the posterior distribution over the full parameter space. Because the positive semidefiniteness of the partial covariance is still not guaranteed, it needs to be approximated to the nearest symmetric positive semidefinite matrix when necessary (Higham, 1988). We can show, however, that the partial covariance is asymptotically positive semidefinite in the small-lapse limit (Appendix A).

Infomax with MCMC

Sampling-based inference provides an attractive alternative to the Laplace method when the model includes nonzero lapse rates, where the posterior may be less well approximated by a Gaussian. To compute mutual information from samples, it is more convenient to use the expansion given in Equation 15, so that it is expressed as the expected uncertainty reduction in entropy of the response $\mathbf{y}$ instead of a reduction in the posterior entropy. This will make it straightforward to approximate integrals needed for mutual information by Monte Carlo integrals involving sums over samples. Also note that we are back in the full parameter space; we no longer treat the lapse parameters separately, as we did for the Laplace-based infomax.

Given a set of posterior samples $\{\boldsymbol{\theta}_m\}$ from the posterior distribution $p(\boldsymbol{\theta}|\mathcal{D}_t)$ at time t, we can evaluate the mutual information using sums over "potential" terms that we denote by

$$L_{jm}(\mathbf{x}) \equiv p\big(y_j = 1|\mathbf{x}, \boldsymbol{\theta}_m\big). \quad (21)$$

This allows us to evaluate the conditional response entropy as

$$H_t(\mathbf{y}|\boldsymbol{\theta};\mathbf{x}) \approx -\frac{1}{M}\sum_{j,m} L_{jm}(\mathbf{x}) \log L_{jm}(\mathbf{x}), \quad (22)$$

and the marginal response entropy as

$$H_t(\mathbf{y};\mathbf{x}) \approx -\sum_j \left(\tfrac{1}{M}\sum_m L_{jm}(\mathbf{x})\right) \times \log\left(\tfrac{1}{M}\sum_m L_{jm}(\mathbf{x})\right), \quad (23)$$

where we have evaluated the posterior-predictive distribution as

$$p(y_j = 1|\mathbf{x}, \mathcal{D}_t) \approx \frac{1}{M}\sum_m L_{jm}(\mathbf{x}). \quad (24)$$

Putting together these terms, the mutual information can be evaluated as

$$I_t(\boldsymbol{\theta};\mathbf{y}|\mathbf{x}) = -\frac{1}{M}\sum_{j,m} L_{jm}(\mathbf{x}) \log \frac{L_{jm}(\mathbf{x})}{\sum_{m'} L_{jm'}(\mathbf{x})/M}, \quad (25)$$

which is straightforward to evaluate for a set of candidate stimuli $\{\mathbf{x}\}$. The computational cost of this approach is therefore linear in the number of samples, and the primary concern is the cost of obtaining a representative sample from the posterior.

Results

We consider two approaches for testing the performance of our proposed stimulus-selection algorithms: one using simulated data, and a second using an off-line analysis of data from real psychophysical experiments.

Simulated experiments

We first tested the performance of our algorithms using simulated data from a fixed psychophysical-observer model. In these simulations, a stimulus $\mathbf{x}$ was selected on each trial and the observer's response $\mathbf{y}$ was sampled from a "true" psychometric function, $p_{\text{true}}(\mathbf{y}|\mathbf{x}) = p(\mathbf{y}|\mathbf{x}, \boldsymbol{\theta}_{\text{true}})$.

We considered psychophysical models defined on a continuous 2-D stimulus space with four discrete response alternatives for every trial, corresponding to the problem of estimating the direction of a 2-D stimulus moving along one of the four cardinal directions (up, down, left, right). We computed expected information gain over a set of discrete stimulus values corresponding to a 21 × 21 square grid (Figure 5A). The stimulus plane is colored in Figure 5A to indicate the most likely response (one of the four alternatives) in each stimulus region. Lapse probabilities λc_i were set to either zero (the lapse-free case) or a constant value of 0.05, resulting in a total lapse probability of $\lambda = 0.2$ across the four choices (Figure 5B). We compared performance of our adaptive algorithms with a method that selected a stimulus uniformly at random from the grid on each trial. We observed that the adaptive methods tended to sample more stimuli near the boundaries between colored regions on the stimulus space (Figure 5C), which led to more efficient estimates of the PF compared to the uniform stimulus-selection approach (Figure 5D). We also confirmed that the posterior entropy of the inferred parameters decreased more rapidly with our adaptive stimulus-sampling algorithms in all cases (Figure 5E and 5F). This was expected because our algorithms explicitly attempt to minimize the posterior entropy by maximizing the mutual information.

For each true model, we compared the performances of four different adaptive methods (Figure 6A and 6B), defined by performing inference with MAP or MCMC and assuming the lapse rate to be fixed at zero or including nonzero lapse parameters. Each of these

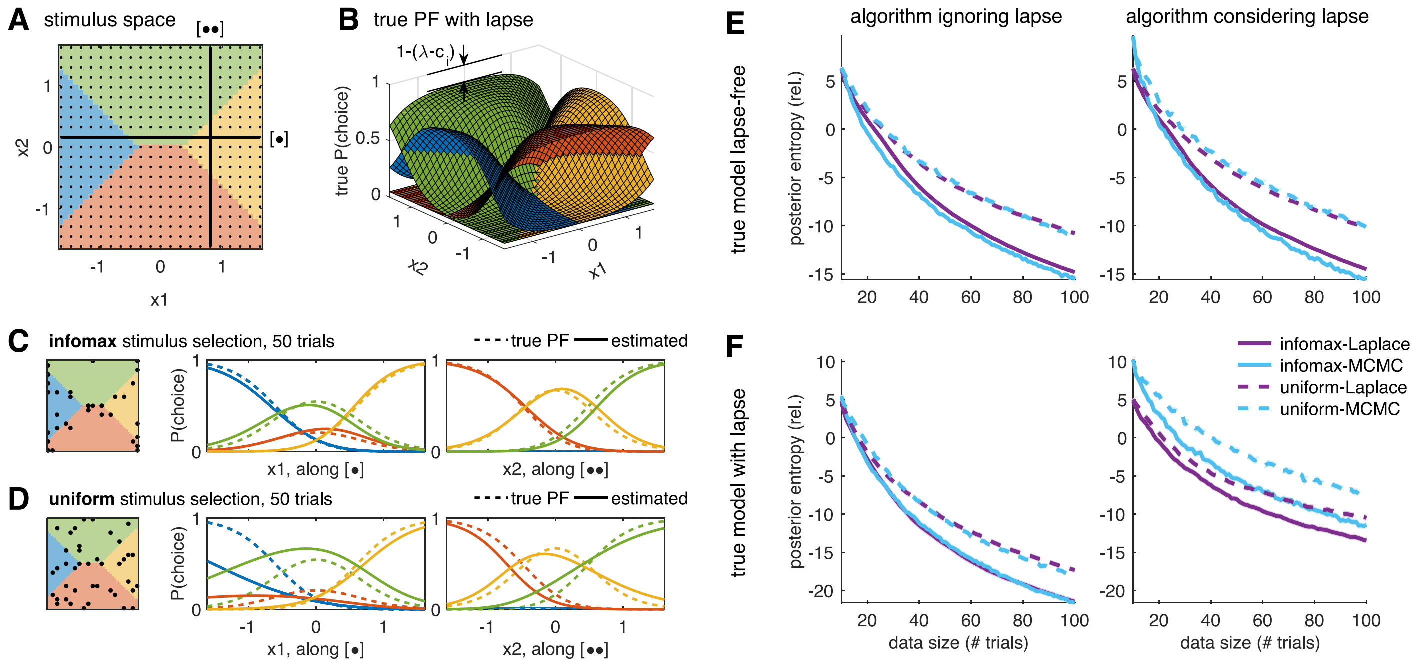

Figure 5. The simulated experiment. (A) At each trial, a stimulus was selected from a 2-D stimulus plane with a 21 × 21 grid. The two lines, running along x_1 and x_2 respectively, indicate the cross-sections used in (C–D). Colors indicate the most likely response in the respective stimulus regime, according to the true psychometric function shown in (B), with a consistent color code. (B) Given each stimulus, a simulated response was drawn from a true model with four alternatives. Shown here is the model with lapse, characterized by a nondeterministic choice (i.e., the choice probability does not approach 0 or 1) even at an easy stimulus, far from the choice boundaries. (C–D) Examples of Laplace approximation–based inference results after 50 trials, where stimuli were selected either (C) using our adaptive infomax method or (D) uniformly, as shown at left. In both cases, the true model was lapse free, and the algorithm assumed that lapse was fixed at zero. The two sets of curves show the cross-sections of the true (dotted) and estimated (solid) psychometric functions, along the two lines marked in (A), after sampling these stimuli. (E–F) Traces of posterior entropy from simulated experiments, averaged over 100 runs each. The true model for simulation was either (E) lapse free or (F) with a finite lapse rate of $\lambda = 0.2$, with a uniform lapse scenario $c_i = 1/4$ for each outcome $i = 1, 2, 3, 4$. In algorithms considering lapse (panels on the right), the shift in posterior entropy is due to the use of partial covariance (with respect to weight) in the case of Laplace approximation. The algorithm either used the classical multinomial logistic model that assumes zero lapse (left column) or our extended model that considers lapse (right column). Average performances of adaptive and uniform stimulus-selection algorithms are plotted in solid and dashed lines, respectively; algorithms based on Laplace approximation and Markov-chain Monte Carlo sampling are plotted in purple and cyan. The lighter lines show standard-error intervals over 100 runs, which are very narrow. All sampling-based algorithms used the semiadaptive Markov-chain Monte Carlo method with chain length $M = 1{,}000$.

inference methods was also applied to data selected according to a uniform stimulus-selection algorithm. We quantified performance using the mean squared error between the true response probabilities $p_{ij} = p(y = j|\mathbf{x}_i, \boldsymbol{\theta}_{\text{true}})$ and the estimated probabilities $\hat{p}_{ij}$ over the 21 × 21 grid of stimulus locations $\{\mathbf{x}_i\}$ and the four possible responses $\{j\}$. For MAP-based inference, estimated probabilities were given by $\hat{p}_{ij} = p(y = j|\mathbf{x}_i, \widehat{\boldsymbol{\theta}}_{\text{MAP}})$. For MCMC-based inference, probabilities were given by the predictive distribution, evaluated using an average over samples: $\hat{p}_{ij} = \frac{1}{M}\sum_m p(y = j|\mathbf{x}_i, \boldsymbol{\theta}_m)$, where $\{\boldsymbol{\theta}_m\}$ represent samples from the posterior.

When the true model was lapse free (Figure 6A), lapse-free and lapse-aware inference methods performed similarly, indicating that there was minimal cost to incorporating parameters governing lapse when lapses were absent. Under all inference methods, infomax stimulus selection outperformed uniform stimulus selection by a substantial margin. For example, infomax algorithms achieved in 50–60 trials the error levels that their uniform stimulus-selection counterparts required 100 trials to achieve.

By contrast, when the true model had a nonzero lapse rate (Figure 6B), adaptive stimulus-selection algorithms based on the lapse-free model failed to select optimal stimuli, performing even worse than uniform stimulus-selection algorithms. This emphasizes the impact of model mismatch in adaptive methods, and the importance of a realistic psychometric model. When lapse-aware models were used for inference, on the other hand, both Laplace-based and MCMC-based adaptive stimulus-selection algorithms achieved a significant speedup compared to uniform stimulus

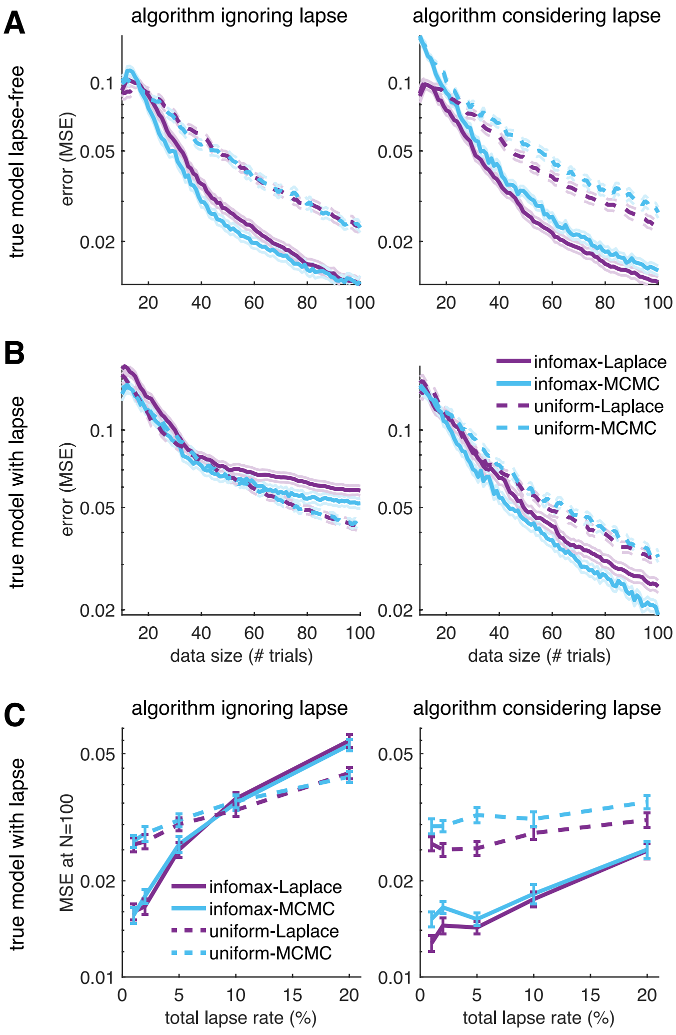

Figure 6. The simulated experiment, continued; results from the same set of simulated experiments as in Figure 5. (A–B) Traces of the mean squared error, where the true model was either (A) lapse free or (B) with a total lapse rate of $\lambda = 0.2$, uniformly distributed to each outcome. Standard-error intervals are plotted in lighter lines as in Figure 5E and 5F. (C) Effect of lapse, tested by adding varying total lapse rates λ. Shown are the mean squared error after $N = 100$ trials of each stimulus-selection algorithm, equivalent to the endpoints in (B). Error bars indicate the standard error over 100 runs, equivalent to the lighter line intervals in Figure 5E and 5F.

selection, while the MCMC-based adaptive algorithm performed better. This shows that the MCMC-based infomax stimulus-selection method can provide an efficient and robust platform for adaptive experiments with realistic models. When the true behavior had lapses, the MCMC-based adaptive stimulus-selection algorithm with the lapse-aware model automatically included easy trials, which provide maximal information about lapse probabilities. These easy trials are typically in the periphery of the stimulus space (strong-stimulus regimes, referred to as "asymptotic performance intensity" by Prins, 2012).

However, the effect of model mismatch due to nonzero lapse only becomes problematic at a high enough lapse rate; in the simulation shown in Figures 5F and 6B, we used a high lapse rate of $\lambda = 0.2$, which is more typical in the case of less sophisticated animals such as rodents (see, e.g., Scott, Constantinople, Erlich, Tank, & Brody, 2015). With lapse rates more typical in well-designed human psychophysics tasks ($\lambda \lesssim 0.05$; see, e.g., Wichmann & Hill, 2001a, 2001b), infomax algorithms still tend to perform better than uniform sampling algorithms (Figure 6C).

Finally, we measured the computation time per trial required by our adaptive stimulus-selection algorithms on a personal desktop computer with an Intel i7 processor. With the Laplace-based algorithm, the major computational bottleneck is the parameter-space integration in the infomax calculation, which scales directly with the model complexity. We could easily achieve tens-of-milliseconds trials in the case of the simple two-alternative forced-choice task, and sub-second trials with 2-D stimuli and four-alternative responses, as used in the current set of simulations (Figure 7A and 7B). With the MCMC-based algorithm, the time per trial in the sampling-based method is limited by the number of samples M in each MCMC chain rather than by the model complexity. Using the standard implementation for the Metropolis–Hastings sampler in Matlab, a time per trial of approximately 0.1 s was achieved with chains shorter than $M \lesssim 200$ (Figure 7C and 7D, top panels). This length of $M \approx 200$ was good enough to represent the posterior distributions for our simulated examples (Figure 7C and 7D, bottom panels), although we note that longer chains are required to sample a more complex posterior distribution, and this particular length M should not be taken as the benchmark in general.

Optimal reordering of real data set

A second approach for testing the performance of our methods is to perform an off-line analysis of data from real psychophysical experiments. Here we take an existing data set and use our methods to reorder the trials so that the most informative stimuli are selected first (for a similar approach, see Lewi, Schneider, Woolley, & Paninski, 2011). To obtain a reordering, we iteratively apply our algorithm to the stimuli shown during the experiment. On each trial, we use our adaptive algorithm to select the optimal stimulus from the set of stimuli $\{\mathbf{x}_i\}$ not yet incorporated into the model. This selection takes place without access to the actual responses $\{\mathbf{y}_i\}$. We update the posterior using the stimulus $\mathbf{x}_i$ and the response $\mathbf{y}_i$ it actually elicited during the experiment, then proceed to the next trial. We can then ask whether adding the data according to the

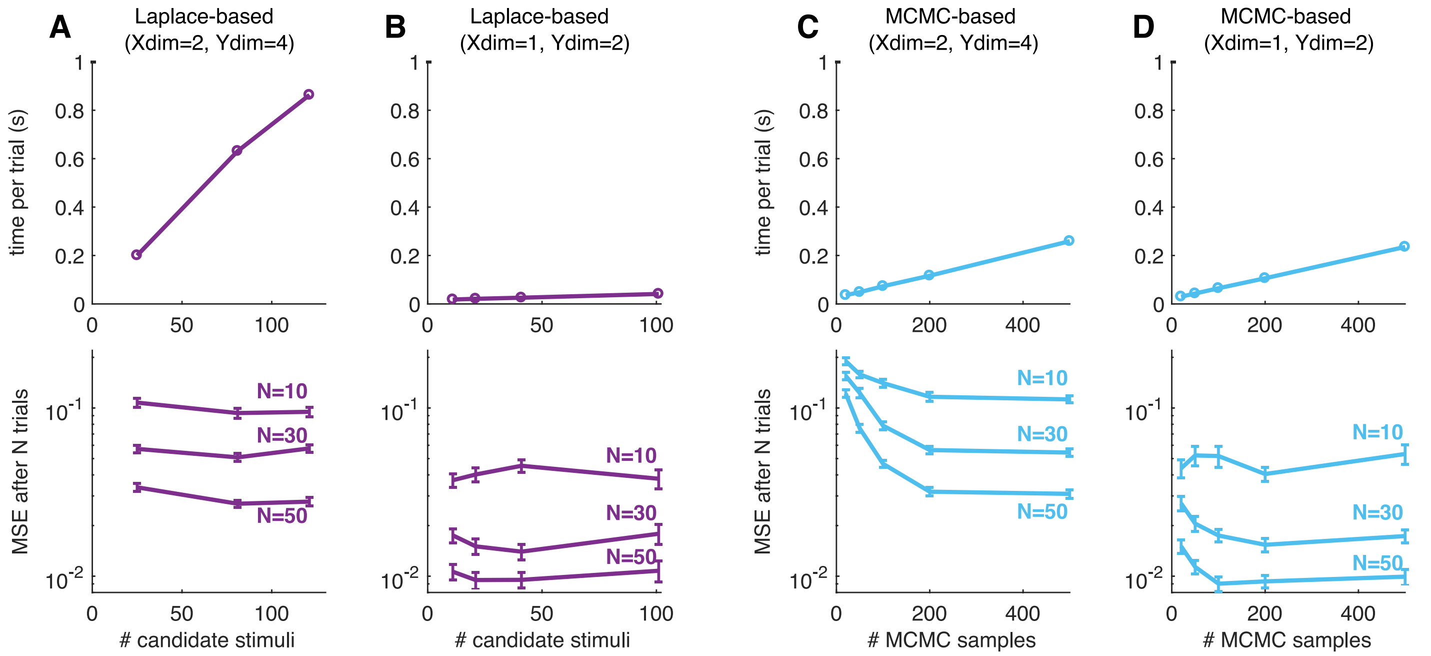

Figure 7. Computation time and accuracy. (A–B) The computation times for the Laplace-based algorithms grow linearly with the number of candidate stimulus points, as shown on the top panels, because one needs to perform a numerical integration to compute the expected utility of each stimulus. In general, there is a trade-off between cost (computation time) and accuracy (inversely related to the estimation error). The bottom panels show the mean squared error of the estimated psychometric function, calculated after completing a sequence of *N* trials, where the 10 initial trials were selected at regular intervals and the following trials were selected under our adaptive algorithm. Error estimates were averaged over 100 independent sequences. Error bars indicate the standard errors. The true model used was the same as in either (A) Figure 5, with two-dimensional stimuli and four-alternative responses, described by nine parameters; or (B) Figure 3, with one-dimensional stimuli and binary responses, with only two parameters (slope and threshold). The different rates at which the computation time increases under the two models reflect the different complexities of numerical quadrature involved. We used lapse-free algorithms in all cases in this example. (C–D) We similarly tested the algorithms based on Markov-chain Monte Carlo sampling using the two models as in (A–B). In this case, the computation times (top panels) grow linearly with the number of samples in each chain and are not sensitive to the dimensionality of the parameter space. On the other hand, the estimation-error plots (bottom panels) suggest that a high-dimensional model requires more samples for accurate inference.

proposed reordering would have led to faster narrowing of the posterior distribution than other orderings.

To perform this analysis, we used a data set from macaque monkeys performing a four-alternative motion-discrimination task (Churchland, Kiani, & Shadlen, 2008). Monkeys were trained to observe a motion stimulus with dots moving in one of the four cardinal directions and to report this direction of motion with an eye movement. The difficulty of the task was controlled by varying the fraction of coherently moving dots on each trial, with the remaining dots appearing randomly (Figure 8A). Each moving-dot stimulus in this experiment could be represented as a 2-D vector, where the direction of the vector is the direction of the mean movement of the dots, and the amplitude of the vector is given by the fraction of coherently moving dots (a number between 0 and 1). Each stimulus presented in the experiment was aligned with one of the two cardinal axes of the stimulus plane (Figure 8B). The PF for this data set consists of a set of four 2-D curves, where each curve specifies the probability of choosing a particular direction as a function of location in the 2-D stimulus plane (Figure 8C).

This monkey data set contained more than 10,000 total observations at 29 distinct stimulus conditions, accumulating more than 300 observations per stimulus. This multiplicity of observations per stimulus ensured that the posterior distribution given the full data set was narrow enough that it could be considered to provide a ground-truth PF against which the inferences based on the reordering experiment could be compared.

The first 100 stimuli selected by the infomax algorithms had noticeably different statistics from the full data set or its uniform subsampling (the first $N = 100$ trials under uniform sampling). On the other hand, the sets of stimuli selected by both MAP-based and MCMC-based infomax algorithms were similar. Figure 8D shows the histogram of stimulus components along one of the axes, $p(x_2 \mid x_1 = 0)$, from the first $N = 100$ trials, averaged over 100 independent runs under each stimulus-selection algorithm using the lapse-free model.

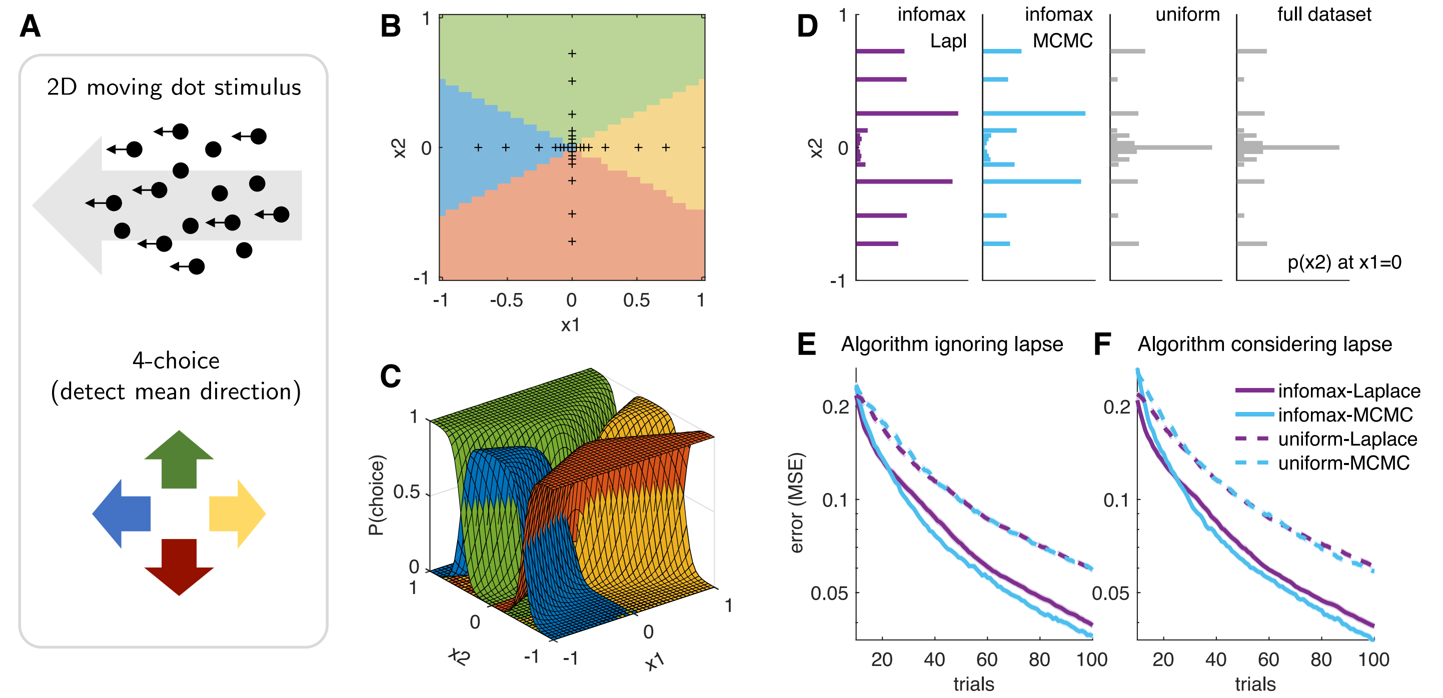

Figure 8. Optimal reordering of a real monkey data set. (A) The psychometric task consisted of a 2-D stimulus presented as moving dots, characterized by a coherence and a mean direction of movement, and a four-alternative response. The four choices are color-coded consistently in (A–C). (B) The axes-only stimulus space of the original data set, with 15 fixed stimuli along each axis. Colors indicate the most likely response in the respective stimulus regime according to the best estimate of the psychometric function. (C) The best estimate of the psychometric function of monkeys in this task, inferred from all observations in the data set. (D) Stimulus selection in the first $N = 100$ trials during the reordering experiment, under the inference method that ignores lapse. Shown are histograms of x_2 along one of the axes, $x_1 = 0$, averaged over 100 independent runs in each case. (E–F) Error traces under different algorithms, averaged over 100 runs. Algorithms based on both Laplace approximation (purple) and Markov-chain Monte Carlo sampling (cyan; $M = 1{,}000$) achieve significant speedups over uniform sampling. Because the monkeys were almost lapse free in this task, inference methods that (E) ignore and (F) consider lapse performed similarly. Standard-error intervals over 100 runs are shown in lighter lines, but are very narrow.

Because the true PF was unknown, we compared the performance of each algorithm to an estimate of the PF from the entire data set. With the MAP algorithm, the full-data-set PF was given by $p_{ij} = p(y = j|\mathbf{x}_i, \widehat{\boldsymbol{\theta}}_{\text{full}})$, evaluated at the MAP estimate of the log posterior, $\widehat{\boldsymbol{\theta}}_{\text{full}} = \text{argmax}_{\boldsymbol{\theta}} \log p(\boldsymbol{\theta}|\mathcal{D}_{\text{full}})$, given the full dataset $\mathcal{D}_{\text{full}}$. For the MCMC algorithm, the full-data-set PF was computed by $p_{ij} \approx \frac{1}{M}\sum_m p(y = j|\mathbf{x}_i, \boldsymbol{\theta}_m)$, where the MCMC chain $\{\boldsymbol{\theta}_m\} \sim \log p(\boldsymbol{\theta}|\mathcal{D}_{\text{full}})$ sampled the log posterior given the full data set. The reordering test on the monkey data set showed that our adaptive stimulus-sampling algorithms were able to infer the PF to a given accuracy in a smaller number of observations, compared to a uniform sampling algorithm (Figure 8E and 8F). In other words, data collection could have been faster with an optimal reordering of the experimental procedure.

Exploiting the full stimulus space

In the experimental data set considered in the previous section, the motion stimuli were restricted to points along the cardinal axes of the 2-D stimulus plane (Figure 8B; Churchland et al., 2008). In some experimental settings, however, the PFs of interest may lack identifiable axes of alignment or may exhibit asymmetries in shape or orientation. Here we show that in such cases, adaptive stimulus-selection methods can benefit from the ability to select points from the full space of possible stimuli.

We performed experiments with a simulated observer governed by the lapse-free PF estimated from the macaque-monkey data set (Figure 8C). This PF was either aligned to the original stimulus axes (Figure 9A and 9B) or rotated counterclockwise by 45° (Figure 9C). We tested the performance of adaptive stimulus selection using the Laplace infomax algorithm, with stimuli restricted to points along the cardinal axes (Figure 9A) or allowed to be among a grid of points in the full 2-D stimulus plane (Figure 9B and 9C).

The simulated experiment indeed closely resembled the results of our data set reordering test in terms of the statistics of adaptively selected stimuli (compare Figure 9A to the purple histogram in Figure 8D). With the full

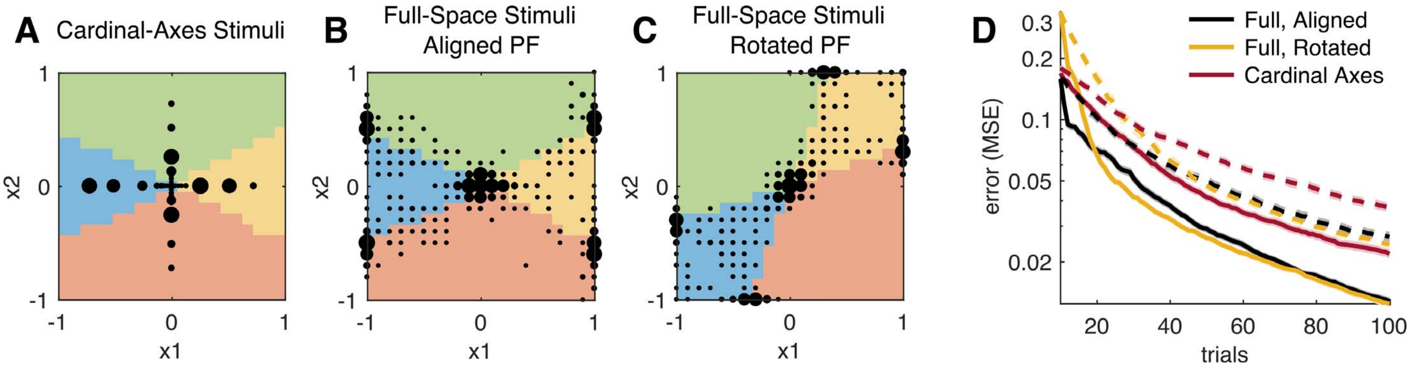

Figure 9. Design of multidimensional stimulus space. (A–C) Three different stimulus-space designs were used in a simulated psychometric experiment. Responses were simulated according to fixed lapse-free psychometric functions (PFs), matched to our best estimate of the monkey PF (Figure 8C). Stimuli were selected within the respective stimulus spaces: (A) the cardinal-axes design, as in the original experiment; (B) the full stimulus plane, with the PF aligned to the cardinal axes of the original stimulus space; and (C) the full stimulus plane, with rotated PF. The black dots in (A–C) indicate which stimuli were sampled by the Laplace-based infomax algorithm during the first $N = 100$ trials of simulation, where the dot size is proportional to the number of trials in which the stimulus was selected (averaged over 20 independent runs, and excluding the 10 fixed initial stimuli). (D) The corresponding error traces, under infomax (solid lines) or uniform (dashed lines) stimulus selection, averaged over 100 runs respectively. Colors indicate the three stimulus-space designs, as shown in (A–C). Standard-error intervals over 100 runs are shown in lighter lines.

2-D stimulus space aligned to the cardinal axes, on the other hand, our adaptive infomax algorithm detected and sampled more stimuli near the boundaries between colored regions in the stimulus plane, which were usually not on the cardinal axes (Figure 9B). Finally, we observed that this automatic exploitation of the stimulus space was not limited by the lack of alignment between the PF and the stimulus axes; our adaptive infomax algorithm was just as effective in detecting and sampling the boundaries between stimulus regions in the case of the unaligned PF (Figure 9C).

The error traces in Figure 9D show that we can infer the PF at a given accuracy in an even smaller number of observations using our adaptive algorithm on the full 2-D stimulus plane (orange curves) compared to the cardinal-axes design (black curves). This also confirms that we can infer the PF accurately and effectively with an unaligned stimulus space (red curves) as well as with an aligned stimulus space. For comparison purposes, all errors were calculated over the same 2-D stimulus grid, even when the stimulus selection was from the cardinal axes. (This had negligible effects on the resulting error values: Compare the black curves in Figure 9D and the purple curves in Figure 8E.)

Discussion

We developed effective Bayesian adaptive stimulus-selection algorithms for inferring psychometric functions, with the objective of maximizing the expected informativeness of each stimulus. The algorithms select an optimal stimulus adaptively in each trial, based on the posterior distribution of model parameters inferred from the accumulating set of past observations.

We emphasize that in psychometric experiments, especially with animals, it is crucial to use models that can account for nonideal yet common behaviors, such as omission (no response; an additional possibility for the outcome) or lapse (resulting in a random, stimulus-independent response). Specifically, we constructed a hierarchical extension of a multinomial logistic model that incorporates both omission and lapse. Although we did not apply these additional features to real data, we performed simulated experiments to investigate their impacts on the accurate inference of PFs. To ensure applicability of the extended model in real-time closed-loop adaptive stimulus-selection algorithms, we also developed efficient methods for inferring the posterior distribution of the model parameters, with approximations specifically suited for sequential experiments.

Advantages of adaptive stimulus selection

We observed two important advantages of using Bayesian adaptive stimulus-selection methods in psychometric experiments. First, our adaptive stimulus-selection algorithms achieved significant speedups in learning time (number of measurements), both on simulated data and in a reordering test of a real experimental data set, with and without lapse in the underlying behavior. Importantly, the success of the algorithm depends heavily on the use of the correct model family; for example, adaptive stimulus selection fails when a classical (lapse-ignorant) model is used to measure behavior with a finite lapse rate. Based on the simulation results, it seems good practice to always use the lapse-aware model unless the behavior under study is known to be completely lapse free, although it should be checked that the addition of the lapse parameters does not make the inference problem intractable, given the constraints of the specific experiments. (One way to check this is using a simulated experiment, where lapse is added to the PF inferred by the lapse-free model, similar to what we did in this article.) The computational cost for incorporating lapses amounts to having k additional parameters to sample, one per each available choice, which is independent of the dimensionality of the stimulus space.

Second, our adaptive stimulus-selection study has implications on the optimization of experimental designs more generally. Contrary to the conventional practice of accumulating repeated observations at a small set of fixed stimuli, we suggest that the (potentially high-dimensional) stimulus space can be exploited more efficiently using our Bayesian adaptive stimulus-selection algorithm. Specifically, the algorithm can automatically detect the structure of the stimulus space (with respect to the PF) as part of the process. We also showed that there are benefits to using the full stimulus space even when the PF is aligned to the cardinal axes of the stimulus space.

Comparison of the two algorithms

Our adaptive stimulus-selection algorithms were developed based on two methods for effective posterior inference: one based on local Gaussian approximation (Laplace approximation) of the posterior, and another based on MCMC sampling. The well-studied analytical method based on the Laplace approximation is fast and effective in simple cases, but becomes heavier in the case of more complicated PFs because the computational bottleneck is the numerical integration over the parameter space that needs to be performed separately for each candidate stimulus. In the case of sampling-based methods, on the other hand, the computational speed is constrained by the number of MCMC samples used to approximate the posterior distribution, but not directly by the number of parameters or the number of candidate stimuli. In general, however, accurately inferring a higher dimensional posterior distribution requires more samples, and therefore a longer computation time. We note that our semiadaptive tuning algorithm helps with the cost–accuracy trade-off by optimizing the sampling accuracy in a given number of samples, without human intervention, although it does not reduce the computation time itself.

To summarize, when the PF under study is low dimensional and well described by the MNL model, for example in a two-alternative forced-choice study with human subjects, the Laplace-based approach provides a lightweight and elegant approach. But if the PF is higher dimensional or deviates significantly from the ideal model (e.g., includes large lapse), MCMC sampling provides a flexible and affordable solution. Results suggest that our MCMC-based algorithm will be applicable to most animal psychometric experiments, as the model complexities are not expected to significantly exceed our simulated example. However, one should always make sure that the number of MCMC samples being used is sufficient to sample the posterior distribution under study.

Limitations and open problems

One potential drawback of adaptive experiments is the undesired possibility that the PF of the observer might adapt to the distribution of stimuli presented during the experiments. If this is the case, the system under measurement would no longer be stationary nor independent of the experimental design, profoundly altering the problem one should try to solve. The usual assumption in psychometric experiments is that well-trained observers exhibit stationary behavior on the timescale of an experiment; under this assumption, the order of data collection cannot bias inference (MacKay, 1992). However, the empirical validity of this claim remains a topic for future research.

One approach for mitigating nonstationarity is to add regressors to account for the history dependence of psychophysical behavior. Recent work has shown that extending a psychophysical model to incorporate past rewards (Bak et al., 2016; Busse et al., 2011; Corrado, Sugrue, Seung, & Newsome, 2005; Lau & Glimcher, 2005), past stimuli (Akrami, Kopec, Diamond, & Brody, 2018), or the full stimulus–response history (Fründ, Wichmann, & Macke, 2014) can provide a more accurate description of the factors influencing responses on a trial-by-trial basis.

Our work leaves open a variety of directions for future research. One simple idea is to reanalyze old data sets under the multinomial response model with omissions included as a separate response category; this will reveal whether omissions exhibit stimulus dependence (e.g., occurring more often on difficult trials) and will provide greater insight into the factors influencing psychophysical behavior on single trials. Another set of directions is to extend the MNL observer model to obtain a more accurate or more flexible model of psychophysical behavior; particular directions include models with nonlinear stimulus dependencies or interaction terms (Cowley, Williamson, Clemens, Smith, & Byron, 2017; DiMattina & Zhang, 2011; Hyafil & Moreno-Bote, 2017; Neri & Heeger, 2002), models with output nonlinearities other than the logistic (Kontsevich & Tyler, 1999; Schütt et al., 2016; A. B. Watson, 2017; A. B. Watson & Pelli, 1983), or models that capture overdispersion, for example due to nonstationarities of the observer, via a hierarchical prior (Schütt et al., 2016). In general, such extensions will be much easier to implement with the MCMC-based inference method, due to the fact that it does not rely on gradients or Hessians of a particular parameterization of log likelihood. Finally, it may be useful to consider the same observer model under optimality criteria other than mutual information—recent work has shown that infomax methods do not necessarily attain optimal performance according to alternate metrics (e.g., mean squared error; I. M. Park & Pillow, 2017; M. Park et al., 2014)—or using nongreedy selection criteria that optimize stimulus selection based on a time horizon longer than the next trial (Kim et al., 2017; King-Smith et al., 1994).

Keywords: adaptive stimulus selection, *sequential optimal design*, *Bayesian adaptive design*, *psychometric function*, *closed-loop experiments*

Acknowledgments

We thank Anne Churchland for sharing the monkey data. JHB was supported by the Samsung Scholarship for the study at Princeton. JWP was supported by grants from the McKnight Foundation and the Simons Collaboration on the Global Brain (SCGB AWD1004351), and by the NSF CAREER Award (IIS-1150186). Computational work was performed using resources at Princeton University and the KIAS Center for Advanced Computing. A Matlab implementation of our methods is available online at https://github.com/pillowlab/adaptivePsychophysicsToolbox.

Commercial relationships: none.
Corresponding author: Jonathan W. Pillow.
E-mail: pillow@princeton.edu.
Address: Department of Psychology and Princeton Neuroscience Institute, Princeton University, Princeton, NJ, USA.

References

Akrami, A., Kopec, C. D., Diamond, M. E., & Brody, C. D. (2018, February 15). Posterior parietal cortex represents sensory history and mediates its effects on behaviour. *Nature*, *554*(7692), 368–372, https://doi.org/10.1038/nature25510.

Bak, J. H., Choi, J. Y., Akrami, A., Witten, I. B., & Pillow, J. W. (2016). Adaptive optimal training of animal behavior. In D. D. Lee, M. Sugiyama, U. V. Luxburg, I. Guyon, & R. Garnett (Eds.), *Advances in neural information processing systems 29* (pp. 1947–1955). Red Hook, NY: Curran Associates, Inc.

Barthelmé, S., & Mamassian, P. (2008). A flexible Bayesian method for adaptive measurement in psychophysics. *arXiv:0809.0387*, 1–28.

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

Busse, L., Ayaz, A., Dhruv, N. T., Katzner, S., Saleem, A. B., Scholvinck, M. L., . . . Carandini, M. (2011). The detection of visual contrast in the behaving mouse. *The Journal of Neuroscience*, *31*(31), 11351–11361, https://doi.org/10.1523/JNEUROSCI.6689-10.2011.

Carandini, M., & Churchland, A. K. (2013). Probing perceptual decisions in rodents. *Nature Neuroscience*, *16*(7), 824–831, https://doi.org/10.1038/nn.3410.

Cavagnaro, D. R., Myung, J. I., Pitt, M. A., & Kujala, J. V. (2010). Adaptive design optimization: A mutual information-based approach to model discrimination in cognitive science. *Neural Computation*, *22*(4), 887–905, https://doi.org/10.1162/neco.2009.02-09-959.

Chaloner, K., & Larntz, K. (1989). Optimal logistic Bayesian design applied to logistic regression experiments. *Journal of Statistical Planning and Inference*, *21*, 191–208, https://doi.org/10.1016/0378-3758(89)90004-9.

Chaloner, K., & Verdinelli, I. (1995). Bayesian experimental design: A review. *Statistical Science*, *10*, 273–304.

Churchland, A. K., Kiani, R., & Shadlen, M. N. (2008). Decision-making with multiple alternatives. *Nature Neuroscience*, *11*(6), 693–702, https://doi.org/10.1038/nn.2123.

Corrado, G. S., Sugrue, L. P., Seung, H. S., & Newsome, W. T. (2005). Linear-nonlinear-Poisson models of primate choice dynamics. *Journal of the Experimental Analysis of Behavior*, *84*(3), 581–617, https://doi.org/10.1901/jeab.2005.23-05.

Cowley, B., Williamson, R., Clemens, K., Smith, M., & Byron, M. Y. (2017). Adaptive stimulus selection for optimizing neural population responses. In I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, & R. Garnett (Eds.), *Advances in neural information processing systems 30* (pp. 1395–1405). Red Hook, NY: Curran Associates, Inc.

DiMattina, C. (2015). Fast adaptive estimation of multidimensional psychometric functions. *Journal of Vision*, *15*(9):5, 1–20, https://doi.org/10.1167/15.9.5. [PubMed] [Article]

DiMattina, C., & Zhang, K. (2011). Active data collection for efficient estimation and comparison of nonlinear neural models. *Neural Computation*, *23*(9), 2242–2288, https://doi.org/10.1162/NECO_a_00167.

Fründ, I., Wichmann, F. A., & Macke, J. H. (2014). Quantifying the effect of intertrial dependence on perceptual decisions. *Journal of Vision*, *14*(7):9, 1–16, https://doi.org/10.1167/14.7.9. [PubMed] [Article]

Gardner, J. R., Song, X., Weinberger, K. Q., Barbour, D., & Cunningham, J. P. (2015). Psychophysical detection testing with Bayesian active learning. In M. Meila & T. Heskes (Eds.), *Proceedings of the thirty-first Conference on Uncertainty in Artificial Intelligence* (pp. 286–297). Arlington, VA: AUAI Press.

Gelman, A., Roberts, G., & Gilks, W. (1996). Efficient Metropolis jumping rules. *Bayesian Statistics*, *5*, 599–607.

Glonek, G., & McCullagh, P. (1995). Multivariate logistic models. *Journal of the Royal Statistical Society, Series B (Methodological)*, *57*(3), 533–546.

Haario, H., Saksman, E., & Tamminen, J. (2001). An adaptive Metropolis algorithm. *Bernoulli*, *7*(2), 223–242, https://doi.org/10.2307/3318737.

Heiss, F., & Winschel, V. (2008). Likelihood approximation by numerical integration on sparse grids. *Journal of Econometrics*, *144*(1), 62–80, https://doi.org/10.1016/j.jeconom.2007.12.004.

Henderson, H. V., & Searle, S. R. (1981). On deriving the inverse of a sum of matrices. *SIAM Review*, *23*(1), 53–60, https://doi.org/10.1137/1023004.

Higham, N. J. (1988). Computing a nearest symmetric positive semidefinite matrix. *Linear Algebra and Its Applications*, *103*(C), 103–118, https://doi.org/10.1016/0024-3795(88)90223-6.

Hyafil, A., & Moreno-Bote, R. (2017). Breaking down hierarchies of decision-making in primates. In J. I. Gold (Ed.). *eLife*, *6*:e16650, https://doi.org/10.7554/eLife.16650.

Kim, W., Pitt, M. A., Lu, Z., & Myung, J. I. (2017). Planning beyond the next trial in adaptive experiments: A dynamic programming approach. *Cognitive Science*, *41*(8), 2234–2252, https://doi.org/10.1111/cogs.12467.

Kim, W., Pitt, M. A., Lu, Z.-L., Steyvers, M., & Myung, J. I. (2014). A hierarchical adaptive approach to optimal experimental design paradigm of adaptive design optimization (ADO). *Neural Computation*, *26*, 2465–2492, https://doi.org/10.1162/NECO_a_00654.

King-Smith, P. E., Grigsby, S. S., Vingrys, A. J., Benes, S. C., & Supowit, A. (1994). Efficient and unbiased modifications of the QUEST threshold method: Theory, simulations, experimental evaluation and practical implementation. *Vision Research*, *34*(7), 885–912, https://doi.org/10.1016/0042-6989(94)90039-6.

Knoblauch, K., & Maloney, L. T. (2008). Estimating classification images with generalized linear and additive models. *Journal of Vision*, *8*(16):10, 1–19, https://doi.org/10.1167/8.16.10. [PubMed] [Article]

Kontsevich, L. L., & Tyler, C. W. (1999). Bayesian adaptive estimation of psychometric slope and threshold. *Vision Research*, *39*(16), 2729–2737, https://doi.org/10.1016/S0042-6989(98)00285-5.

Kujala, J. V., & Lukka, T. J. (2006). Bayesian adaptive estimation: The next dimension. *Journal of Mathematical Psychology*, *50*(4), 369–389, https://doi.org/10.1016/j.jmp.2005.12.005.

Kuss, M., Jäkel, F., & Wichmann, F. A. (2005). Bayesian inference for psychometric functions. *Journal of Vision*, *5*(5):8, 478–492, https://doi.org/10.1167/5.5.8. [PubMed] [Article]

Lau, B., & Glimcher, P. W. (2005). Dynamic response-by-response models of matching behavior in rhesus monkeys. *Journal of the Experimental Analysis of Behavior*, *84*(3), 555–579, https://doi.org/10.1901/jeab.2005.110-04.

Lesmes, L. A., Lu, Z.-L., Baek, J., & Albright, T. D. (2010). Bayesian adaptive estimation of the contrast sensitivity function: The quick CSF method. *Journal of Vision*, *10*(3):17, 1–21, https://doi.org/10.1167/10.3.17. [PubMed] [Article]

Lesmes, L. A., Lu, Z.-L., Baek, J., Tran, N., Dosher, B., & Albright, T. (2015). Developing Bayesian adaptive methods for estimating sensitivity thresholds (d') in yes-no and forced-choice tasks. *Frontiers in Psychology*, *6*, 1070, https://doi.org/10.3389/fpsyg.2015.01070.

Lewi, J., Butera, R., & Paninski, L. (2009). Sequential optimal design of neurophysiology experiments. *Neural Computation*, *21*(3), 619–687, https://doi.org/10.1162/neco.2008.08-07-594.

Lewi, J., Schneider, D. M., Woolley, S. M. N., & Paninski, L. (2011). Automating the design of informative sequences of sensory stimuli. *Journal of Computational Neuroscience*, *30*(1), 181–200, https://doi.org/10.1007/s10827-010-0248-1.

MacKay, D. J. C. (1992). Information-based objective functions for active data selection. *Neural Compu-*

tation, *4*(4), 590–604, https://doi.org/10.1162/neco.1992.4.4.590.

Metropolis, N., Rosenbluth, A. W., Rosenbluth, M. N., Teller, A. H., & Teller, E. (1953). Equation of state calculations by fast computing machines. *The Journal of Chemical Physics*, *21*(6), 1087–1092, https://doi.org/10.1063/1.1699114.

Murray, R. F. (2011). Classification images: A review. *Journal of Vision*, *11*(5):2, 1–25, https://doi.org/10.1167/11.5.2. [PubMed] [Article]

Neri, P., & Heeger, D. J. (2002). Spatiotemporal mechanisms for detecting and identifying image features in human vision. *Nature Neuroscience*, *5*(8), 812–816, https://doi.org/10.1038/nn886.

Paninski, L., Ahmadian, Y., Ferreira, D. G., Koyama, S., Rahnama Rad, K., Vidne, M., … Wu, W. (2010). A new look at state-space models for neural data. *Journal of Computational Neuroscience*, *29*(1), 107–126, https://doi.org/10.1007/s10827-009-0179-x.

Park, I. M., & Pillow, J. W. (2017). Bayesian efficient coding. *bioRxiv*, 178418, https://doi.org/10.1101/178418.

Park, M., Horwitz, G., & Pillow, J. W. (2011). Active learning of neural response functions with Gaussian processes. In J. Shawe-Taylor, R. S. Zemel, P. L. Bartlett, F. Pereira, & K. Q. Weinberger (Eds.), *Advances in neural information processing systems 24* (pp. 2043–2051). Red Hook, NY: Curran Associates, Inc.

Park, M., & Pillow, J. W. (2012). Bayesian active learning with localized priors for fast receptive field characterization. In F. Pereira, C. J. C. Burges, L. Bottou, & K. Q. Weinberger (Eds.), *Advances in neural information processing systems 25* (pp. 2357–2365). Red Hook, NY: Curran Associates, Inc.

Park, M., Weller, J. P., Horwitz, G. D., & Pillow, J. W. (2014). Bayesian active learning of neural firing rate maps with transformed Gaussian process priors. *Neural Computation*, *26*(8), 1519–1541.

Pillow, J. W., Ahmadian, Y., & Paninski, L. (2011). Model-based decoding, information estimation, and change-point detection techniques for multineuron spike trains. *Neural Computation*, *23*(1), 1–45, https://doi.org/10.1162/NECO_a_00058.

Pillow, J. W., & Park, M. (2016). Adaptive Bayesian methods for closed-loop neurophysiology. In A. E. Hady (Ed.), *Closed loop neuroscience*. San Diego, CA: Academic Press.

Prins, N. (2012). The psychometric function: The lapse rate revisited. *Journal of Vision*, *12*(6):25, 1–16, https://doi.org/10.1167/12.6.25. [PubMed] [Article]

Prins, N. (2013). The psi-marginal adaptive method: How to give nuisance parameters the attention they deserve (no more, no less). *Journal of Vision*, *13*(7): 3, 1–17, https://doi.org/10.1167/13.7.3. [PubMed] [Article]

Roberts, G. O., Gelman, A., & Gilks, W. R. (1997). Weak convergence and optimal scaling of random walk Metropolis algorithms. *Annals of Applied Probability*, *7*(1), 110–120, https://doi.org/10.1214/aoap/1034625254.

Rosenthal, J. S. (2011). Optimal proposal distributions and adaptive MCMC. In S. Brooks, A. Gelman, G. Jones, & X.-L. Meng (Eds.), *Handbook of Markov chain Monte Carlo* (pp. 93–112). Boca Raton, FL: Chapman and Hall CRC, https://doi.org/10.1201/b10905.

Schütt, H. H., Harmeling, S., Macke, J. H., & Wichmann, F. A. (2016). Painfree and accurate Bayesian estimation of psychometric functions for (potentially) overdispersed data. *Vision Research*, *122*, 105–123, https://doi.org/10.1016/j.visres.2016.02.002.

Scott, B. B., Constantinople, C. M., Erlich, J. C., Tank, D. W., & Brody, C. D. (2015). Sources of noise during accumulation of evidence in unrestrained and voluntarily head-restrained rats. *eLife*, *4*, e11308, https://doi.org/10.7554/eLife.11308.

Treutwein, B. (1995). Adaptive psychophysical procedures. *Vision Research*, *35*(17), 2503–2522, https://doi.org/https://doi.org/10.1016/0042-6989(95)00016-X.

Vul, E., Bergsma, J., & MacLeod, D. (2010). Functional adaptive sequential testing. *Seeing and Perceiving*, *23*(5), 483–515, https://doi.org/10.1163/187847510X532694.

Watson, A. B. (2017). QUEST+: A general multidimensional Bayesian adaptive psychometric method. *Journal of Vision*, *17*(3):10, 1–27, https://doi.org/10.1167/17.3.10. [PubMed] [Article]

Watson, A. B., & Pelli, D. G. (1983). QUEST: A Bayesian adaptive psychometric method. *Perception & Psychophysics*, *33*(2), 113–120, https://doi.org/10.3758/BF03202828.

Watson, C. S., Kellogg, S. C., Kawanishi, D. T., & Lucas, P. A. (1973). The uncertain response in detection-oriented psychophysics. *Journal of Experimental Psychology*, *99*(2), 180–185, https://doi.org/10.1037/h0034736.

Wichmann, F. A., & Hill, N. J. (2001a). The psychometric function: I. Fitting, sampling, and goodness of fit. *Perception & Psychophysics*, *63*(8), 1293–1313, https://doi.org/10.3758/BF03194544.

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

Zocchi, S. S., & Atkinson, A. C. (1999). Optimum experimental designs for multinomial logistic models. *Biometrics*, *55*(2), 437–444, https://doi.org/10.1111/j.0006-341X.1999.00437.x.

Appendix A

Log likelihood for the classical MNL model

Here we provide more details about the log likelihood $L = \mathbf{y}^\top \log \mathbf{p}$ under the MNL model (Equation 6), first in the lapse-free case.

A convenient property of the MNL model (a property common to all generalized linear models) is that the parameter vector p_i governing y depends only on a 1-D projection of the input, $V_i = \boldsymbol{\phi}^\top \mathbf{w}_i$, which is known as the *linear predictor*. Recall that $\boldsymbol{\phi} = \boldsymbol{\phi}(\mathbf{x})$ is the input feature vector. In the multinomial case, it is useful to consider the column vector of linear predictors for a single trial, $\mathbf{V} = [V_1, \cdots, V_k]^\top$, and the concatenated weight vector $\mathbf{w} = [\mathbf{w}_1^\top, \cdots, \mathbf{w}_k^\top]^\top$, consisting of all weights stacked into a single vector. We can summarize their linear relationship as $\mathbf{V} = X\mathbf{w}$, where X is a block diagonal matrix containing k blocks of $\boldsymbol{\phi}^\top$ along the diagonal. In other words,

$$X = \begin{bmatrix} \boldsymbol{\phi}^\top & \mathbf{0}^\top & \cdots & \mathbf{0}^\top \\ \mathbf{0}^\top & \boldsymbol{\phi}^\top & \cdots & \mathbf{0}^\top \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{0}^\top & \mathbf{0}^\top & \cdots & \boldsymbol{\phi}^\top \end{bmatrix}, \quad \mathbf{w} = \begin{bmatrix} \mathbf{w}_1 \\ \mathbf{w}_2 \\ \vdots \\ \mathbf{w}_k \end{bmatrix}. \quad (26)$$

Derivatives

It is convenient to work in terms of the linear predictor $\mathbf{V} = \{V_i\}$ first. If $N_y \equiv \sum_i y_i = 1$ is the total number of responses per trial, the first and second derivatives of L with respect to $\mathbf{V}$ are $\partial L/\partial V_j = y_j - N_y p_j$ and $\partial^2 L/\partial V_i \partial V_j = N_y p_i(\delta_{ij} - p_j)$, respectively. Rewriting in vector forms, we have

$$\frac{\partial L}{\partial \mathbf{V}} = (\mathbf{y} - N_y \mathbf{p})^\top, \quad (27)$$

$$\frac{\partial^2 L}{\partial \mathbf{V}^2} = -N_y\left(\mathrm{diag}(\mathbf{p}) - \mathbf{p}\mathbf{p}^\top\right) \equiv -N_y \Gamma(\mathbf{p}), \quad (28)$$

where $\mathrm{diag}(\mathbf{p}) = [p_i \delta_{ij}]$ is a square matrix with the elements of $\mathbf{p}$ on the diagonal and zeros otherwise.

Putting back in terms of the weight vector $\mathbf{w}$ is easy, thanks to the linear relationship $\mathbf{V} = X\mathbf{w}$:

$$\frac{\partial L}{\partial \mathbf{w}} = \frac{\partial L}{\partial \mathbf{V}} X = (\mathbf{y} - \mathbf{p})^\top X \equiv \Delta^\top, \quad (29)$$

$$\frac{\partial^2 L}{\partial \mathbf{w}^2} = X^\top \frac{\partial^2 L}{\partial \mathbf{V}^2} X = -X^\top \Gamma X \equiv -\Lambda. \quad (30)$$

Concavity

Importantly, L is concave with respect to $\mathbf{V}$ (and therefore with respect to $\mathbf{w}$). To prove the concavity of L, we show that the Hessian $H = -\mathrm{diag}(\mathbf{p}) + \mathbf{p}\mathbf{p}^\top \equiv -\Gamma$ is negative semidefinite, which is equivalent to showing that $\mathbf{z}^\top \Gamma \mathbf{z} \geq 0$:

$$\begin{aligned} \mathbf{z}^\top \Gamma \mathbf{z} &= \mathbf{z}^\top \mathrm{diag}(\mathbf{p})\mathbf{z} - (\mathbf{z}^\top \mathbf{p})^2 \\ &= \sum\nolimits_i z_i^2 p_i - \left(\sum\nolimits_j z_j p_j\right)^2 \\ &= \sum\nolimits_i p_i \left[\left(z_i - \sum\nolimits_j z_j p_j\right)^2\right] \geq 0 \quad (31) \end{aligned}$$

for an arbitrary vector $\mathbf{z}$.

Log likelihood with lapse

With a finite lapse rate λ (to recap), the MNL model is modified as $p_i = (1 - \lambda) q_i + \lambda c_i$, where

$$q_i = \frac{\exp(V_i)}{\sum\limits_j \exp(V_j)}, \quad \lambda c_i = \frac{\exp(u_i)}{1 + \sum\limits_j \exp(u_j)}. \quad (32)$$

Let us introduce the following abbreviations:

$$r_i \equiv \frac{\lambda c_i}{p_i}, \quad t_i \equiv y_i(1 - r_i), \quad s_i \equiv y_i r_i (1 - r_i), \quad (33)$$

where the dimensionless ratio $r \in [0, 1]$ can be considered as the order parameter for the effect of lapse.

Derivatives with respect to the weights

Differentiating with the linear predictor $\mathbf{V}$, we get

$$\frac{\partial q_i}{\partial V_l} = (\delta_{il} - q_l) q_i,$$

$$\frac{\partial^2 q_i}{\partial V_j \partial V_l} = \left[(\delta_{ij} - q_j)(\delta_{il} - q_l) - (\delta_{jl} q_l - q_j q_l)\right] q_i.$$

This leads to

$$\frac{\partial p_i}{\partial V_l} = (1 - \lambda) \frac{\partial q_i}{\partial V_l}, \quad \frac{\partial^2 p_i}{\partial V_j \partial V_l} = (1 - \lambda) \frac{\partial^2 q_i}{\partial V_j \partial V_l}.$$

We are interested in the derivatives of the log likelihood $L = \mathbf{y}^\top \log \mathbf{p}$ with respect to $\mathbf{V}$. The partial gradient is

$$\begin{aligned}\frac{\partial L}{\partial V_l} &= \sum_i y_i \frac{1}{p_i}\frac{\partial p_i}{\partial V_l} = (1-\lambda)\sum_i y_i \frac{q_i}{p_i}(\delta_{il} - q_l) \\ &= t_l - q_l \sum_i t_i.\end{aligned}$$

Similarly, the partial Hessian is written as

$$\begin{aligned}\frac{\partial^2 L}{\partial V_j \partial V_l} &= \sum_i y_i \left(\frac{1}{p_i}\frac{\partial^2 p_i}{\partial V_j \partial V_l} - \frac{1}{p_i^2}\frac{\partial p_i}{\partial V_j}\frac{\partial p_i}{\partial V_l}\right) \\ &= \delta_{jl}\left(s_l - q_l \sum\nolimits_i t_i\right) - \left(q_j s_l + q_l s_j\right) \\ &\quad + q_j q_l \left(\sum\nolimits_i s_i + \sum\nolimits_i t_i\right).\end{aligned}$$

In vector forms, and with $\tau \equiv \sum_i t_i$ and $\sigma \equiv \sum_i s_i$,

$$\frac{\partial L}{\partial \mathbf{V}} = (\mathbf{t} - \tau \mathbf{q})^\top; \quad (34)$$

$$\begin{aligned}\frac{\partial^2 L}{\partial \mathbf{V}^2} &= \mathrm{diag}(\mathbf{s} - \tau\mathbf{q}) - (\mathbf{q}\mathbf{s}^\top + \mathbf{s}\mathbf{q}^\top) + (\tau + \sigma)\mathbf{q}\mathbf{q}^\top \\ &= -\tau\left[\mathrm{diag}(\mathbf{q}) - \mathbf{q}\mathbf{q}^\top\right] \\ &\quad + \left[\mathrm{diag}(\mathbf{s}) - (\mathbf{q}\mathbf{s}^\top + \mathbf{s}\mathbf{q}^\top) + \sigma\mathbf{q}\mathbf{q}^\top\right]. \quad (35)\end{aligned}$$

Note that we recover $t_i \to y_i$ and $s_i \to 0$ in the lapse-free limit $\lambda \to 0$. Hence the first square bracket in Equation 35 reduces back to the lapse-free Hessian, while the second square bracket vanishes as $\lambda \to 0$.

In the presence of lapse, one might still be interested in the partial Hessian with respect to the weight parameters, $H \equiv \partial^2 L / \partial \mathbf{V}^2$, which should be evaluated as in Equation 35. To test the negative semidefiniteness of this partial Hessian, again for an arbitrary vector $\mathbf{z}$, we end up with

$$\begin{aligned}\mathbf{z}^\top H \mathbf{z} &= -\sum_j t_j \left\langle \left(z - \langle z \rangle_q\right)^2 \right\rangle_q \\ &\quad + \sum_j s_j \left(z_j - \langle z \rangle_q\right)^2, \quad (36)\end{aligned}$$

where $\langle x \rangle_q = \sum_j x_j q_j$. The partial Hessian is asymptotically negative semidefinite (which is equivalent to the log likelihood being concave) in the lapse-free limit, where $t_j \to y_j$ and $s_j \to 0$.

Derivatives with respect to lapse parameters

From Equations 2 and 3, we have $p_i = (1-\lambda)q_i + \lambda c_i$, where

$$c_i = \frac{\exp(u_i)}{\sum_j \exp(u_j)}; \qquad \lambda = \frac{\sum_j \exp(u_j)}{1 + \sum_j \exp(u_j)}. \quad (37)$$

Differentiating with respect to the auxiliary lapse parameter u_i, we have

$$\frac{\partial c_i}{\partial u_j} = (\delta_{ij} - c_i)c_j; \qquad \frac{\partial \lambda}{\partial u_j} = (1-\lambda)\lambda c_j. \quad (38)$$

The gradient is then

$$\frac{\partial p_i}{\partial u_j} = \left(\delta_{ij} - p_i\right)\lambda c_j; \quad (39)$$

using the abbreviations in Equation 33, the gradient of the log likelihood is

$$\frac{\partial L}{\partial u_j} = \sum_i y_i \frac{1}{p_i}\frac{\partial p_i}{\partial u_j} = r_j\left(y_j - N_y \cdot p_j\right). \quad (40)$$

The second derivative with respect to lapse is therefore

$$\frac{\partial^2 p_i}{\partial u_j \partial u_l} = \delta_{jl}\frac{\partial p_i}{\partial u_l} - (\delta_{ij} + \delta_{il} - 2p_i)\lambda c_l \lambda c_j; \quad (41)$$

it is useful to notice that

$$\frac{\partial p_i}{\partial u_j}\frac{\partial p_i}{\partial u_l} = \delta_{jl}\frac{\partial p_i}{\partial u_l}\lambda c_l - p_i(\delta_{ij} + \delta_{il} - 2p_i)\lambda c_l \lambda c_j. \quad (42)$$

The corresponding part of the Hessian is

$$\begin{aligned}\frac{\partial^2 L}{\partial u_j \partial u_l} &= \sum_i y_i \left(\frac{1}{p_i}\frac{\partial^2 p_i}{\partial u_j \partial u_l} - \frac{1}{p_i^2}\frac{\partial p_i}{\partial u_j}\frac{\partial p_i}{\partial u_l}\right) \\ &= \delta_{jl}\sum_i y_i \frac{1}{p_i}\left(1 - \frac{\lambda c_l}{p_i}\right)\frac{\partial p_i}{\partial u_l} \\ &= \delta_{jl}\left(s_l - r_l p_l N_y + r_l^2 p_l^2 \sum_i \frac{y_i}{p_i}\right). \quad (43)\end{aligned}$$

Finally, the mixed derivative is

$$\frac{\partial^2 p_i}{\partial u_j \partial V_l} = -(1-\lambda)\lambda c_j \cdot (\delta_{il} - q_l)q_l. \quad (44)$$

Again, it is useful to notice that

$$\frac{\partial p_i}{\partial u_j}\frac{\partial p_i}{\partial V_l} = -(\delta_{ij} - p_i)\frac{\partial^2 p_i}{\partial u_j \partial V_l}. \quad (45)$$

Hence

$$\begin{aligned}\frac{\partial^2 L}{\partial u_j \partial V_l} &= \sum_i y_i \left(\frac{1}{p_i}\frac{\partial^2 p_i}{\partial u_j \partial V_l} - \frac{1}{p_i^2}\frac{\partial p_i}{\partial u_j}\frac{\partial p_i}{\partial V_l}\right) \\ &= -s_j\left(\delta_{jl} + \frac{q_l^2}{q_j}\right). \quad (46)\end{aligned}$$

From Equations 40, 43, and 46, we see that all derivatives involving the lapse parameter scale with at least one order of r, therefore vanishing in the lapse-free limit $\lambda \to 0$.

Appendix B

The Metropolis–Hastings algorithm

The Metropolis–Hastings algorithm (Metropolis et al., 1953) generates a chain of samples, using a proposal density and a method to accept or reject the proposed moves.

A proposal is made at each iteration, where the algorithm randomly chooses a candidate for the next sample value $\mathbf{x}'$ based on the current sample value $\mathbf{x}_t$. The choice follows the proposal density function $\mathbf{x}' \sim Q(\mathbf{x}' \mid \mathbf{x}_t)$. When the proposal density Q is symmetric, for example a Gaussian, the sequence of samples is a random walk. In general the width of Q should match with the statistics of the distribution being sampled, and individual dimensions in the sampling space may behave differently in the multivariate case; finding the appropriate Q can be difficult.

The proposed move is either accepted or rejected with some probability; if it is rejected, the current sample value is reused in the next iteration, $\mathbf{x}' = \mathbf{x}_t$. The probability of acceptance is determined by comparing the values of $P(\mathbf{x}_t)$ and $P(\mathbf{x}')$, where $P(\mathbf{x})$ is the distribution being sampled. Because the algorithm only considers the acceptance ratio $\rho = P(\mathbf{x}')/P(\mathbf{x}_t) = f(\mathbf{x}')/f(\mathbf{x}_t)$, where $f(\mathbf{x})$ can be any function proportional to the desired distribution $P(\mathbf{x})$, there is no need to worry about the proper normalization of the probability distribution. If $\rho \geq 1$, the move is always accepted; if $\rho < 1$, it is accepted with a probability ρ. Consequently, the samples tend to stay in the high-density regions, visiting the low-density regions only occasionally.

Optimizing the sampler

One of the major difficulties in using the MCMC method is making an appropriate choice of the proposal distribution, which may significantly affect the performance of the sampler. If the proposal distribution is too narrow, it will take a long time for the chain to diffuse away from the starting point, producing a chain with highly correlated samples and requiring a long time to achieve independent samples. On the other hand, if the proposal distribution is too wide, most of the proposed moves will be rejected, once again resulting in the chain stuck at the initial point. In either case the chain would mix poorly (Rosenthal, 2011). In this article we restrict our consideration to the Metropolis–Hastings algorithm (Metropolis et al., 1953), although the issue of proposal-distribution optimization is universal in most variants of MCMC algorithms, with only implementation-level differences.

The basic idea is that the optimal width of the proposal distribution would be determined in proportion to the typical length scale of the distribution being sampled. This idea was made precise in the case of a stationary random-walk Metropolis algorithm with Gaussian proposal distributions, by comparing the covariance matrix Σ_p of the proposal distribution to the covariance matrix Σ of the sampled chain. Once a linear scaling relation $\Sigma_p = s_d\Sigma$ is fixed, it has been observed that it is optimal to have $s_d = (2.38)^2/d$, where d is the dimensionality of the sampling space (Gelman et al., 1996; Roberts et al., 1997). An adaptive Metropolis algorithm (Haario et al., 2001) follows this observation, where the Gaussian proposal distribution adapts continuously as the sampling progresses. That adaptive algorithm uses the same scaling rule $\Sigma_p = s_d\Sigma$ but updates Σ_p at each proposal, where Σ is the covariance of the samples accumulated so far. Additionally, a small diagonal component is added for stability, as $\Sigma_p = s_d(\Sigma + \epsilon I)$. We used $\epsilon = 0.0001$ in this work.

Here we propose and use the semiadaptive Metropolis–Hastings algorithm, which is a coarse-grained version of the original adaptive algorithm by Haario et al. (2001). The major difference in our algorithm is that the adjustment of the proposal distribution is made only at the end of each (sequential) chain, rather than at each proposal within the chain. This coarse-graining is a reasonable approximation because we will be sampling the posterior distribution many times as it refines over the course of data collection, once after each trial. Assuming that the change in posterior distribution after each new observation is small enough, we can justify our use of the statistics of the previous chain to adjust the properties of the current chain. Unlike in the fully adaptive algorithm, where the proposal distribution needs to stabilize quickly within a single chain, we can allow multiple chains until stabilization, usually a few initial observations—leaving some room for the coarse-grained approximation. This is because for our purpose, it is not imperative that we have a good sampling of the distribution at the very early stages of the learning sequence where the accuracy is already limited by the smallness of the data set.

When applied to the sequential learning algorithm, our semiadaptive Metropolis sampler shows a consistent well-mixed property after a few initial adjustments, with the standard deviation of each sampling dimension decreasing stably as data accumulate (Figure 10). Although Kujala and Lukka (2006) also had the idea of adjusting the proposal density between trials, their scaling factor was fixed and independent of the sampling dimension. Building on more precise statistical observations, our method generalizes well to high-dimensional parameter spaces, typical for multiple-alternative models. Our semiadaptive sampler provides an efficient and robust alternative to particle-filter

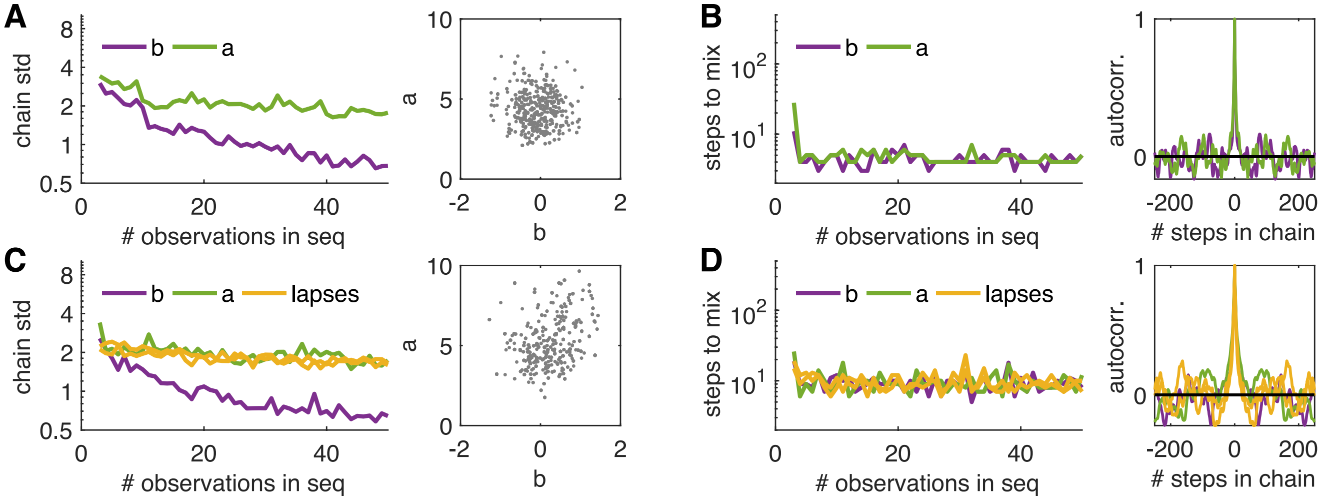

Figure 10. Statistics of the semiadaptive Markov-chain Monte Carlo algorithm in a simulated experiment, with $M = 1{,}000$ samples per chain. We used the same binomial model as in Figure 3, and the uniform stimulus-selection algorithm. (A–B) Lapse-free model. (A) The standard deviation of the samples, along each dimension of the parameter space, decreases as the learning progresses, as expected because the posterior distribution should narrow down as more observations are collected. Also shown is the scatterplot of all 1,000 samples at the last trial $N = 50$, where the true parameter values are $(a, b) = (5, 0)$. (B) The mixing time of the chain (number of steps before the autocorrelation falls to $1/e$) quickly converges to some small value, meaning that the sampler is quickly optimized. Autocorrelation function at the last trial $N = 50$ is shown. (C–D) Same information as (A–B), but with a lapse rate of $\lambda = 0.1$, with uniform lapse ($c_1 = c_2 = 1/2$).

implementations (Kujala & Lukka, 2006), which have the known problem of weight degeneration (DiMattina, 2015) as the posterior distribution narrows down with the accumulation of data.

Appendix C

Fast sequential update of the posterior, with Laplace approximation

Use of Laplace approximation has been shown to be particularly useful in a sequential experiment (Lewi et al., 2009), where it can be assumed that the posterior distribution after the next trial in sequence, $\mathcal{P}_{t+1}$, would not be very different from the current posterior $\mathcal{P}_t$. Let us consider the lapse-free case $\boldsymbol{\theta} = \mathbf{w}$ for the moment, where the use of Laplace approximation is valid. Rearranging from Equations 7 and 9, the sequential update for the posterior distribution is

$$\log \mathcal{P}_{t+1}(\mathbf{w}) = \log \mathcal{P}_t(\mathbf{w}) + L_{t+1}(\mathbf{w}); \quad (47)$$

or with Laplace approximation,

$$\log \mathcal{N}(\mathbf{w}|\boldsymbol{\theta}_{t+1}, C_{t+1}) \approx \log \mathcal{N}(\mathbf{w}|\boldsymbol{\theta}_t, C_t) + L_{t+1}(\mathbf{w}), \quad (48)$$

where $L_i(\mathbf{w}) = \log p(\mathbf{y}_i|\mathbf{x}_i, \mathbf{w})$ is a shorthand for the log likelihood of the ith observation.

With this, we can achieve a fast sequential update of the posterior without performing the full numerical optimization each time. Because the new posterior mode $\boldsymbol{\theta}_{t+1}$ is where the gradient vanishes, it can be approximated from the previous mode $\boldsymbol{\theta}_t$ by taking the first derivative of Equation 48. The posterior covariance C_{t+1} is similarly approximated by taking the second derivative:

$$\boldsymbol{\theta}_{t+1} = \boldsymbol{\theta}_t + C_t\Delta_{t+1}, \qquad \Delta_{t+1} = \left.\frac{\partial L_{t+1}}{\partial \mathbf{w}}\right|_{\mathbf{w}=\boldsymbol{\theta}_t} \quad (49)$$

$$C_{t+1} = \left(C_t^{-1} + \Lambda_{t+1}\right)^{-1}, \quad \Lambda_{t+1} = -\left.\frac{\partial^2 L_{t+1}}{\partial \mathbf{w}^2}\right|_{\mathbf{w}=\boldsymbol{\theta}_{t+1}}. \quad (50)$$

Using the matrix inversion lemma (Henderson & Searle, 1981), we can rewrite the posterior covariance update as

$$C_{t+1} = C_t\left[I - (I + \Lambda_{t+1}C_t)^{-1}\Lambda_{t+1}C_t\right]. \quad (51)$$

Unlike in the earlier application of this trick (Lewi et al., 2009), the covariance matrix update (Equation 50) is not a rank-1 update, because of the multinomial nature of our model (our linear predictor $\mathbf{y}$ is a vector, not a scalar as in a binary model).

Integration over the parameter space: Reducing the integration space

The evaluation of the expected utility function usually involves a potentially high-dimensional integral over the parameter space. With the Gaussian approximation of the posterior, we can reduce and standardize the integration space. The process consists of three steps: diagonalization, marginalization, and standardization. First we choose a new coordinate system of the (say q-dimensional) weight space, such that the first k elements of the extended weight vector $\mathbf{w}$ are coupled one-to-one to the elements of k-vector $\mathbf{y}$. Then we marginalize to integrate out the remaining $q - k$ dimensions, effectively changing the integration variable from $\mathbf{w}$ to $\mathbf{y}$. Finally, we use Cholesky decomposition to standardize the normal distribution, which is the posterior on $\mathbf{y}$. The resulting integral is still multidimensional, due to the multinomial nature of our model. But once the distribution is standardized, there are a number of efficient numerical integration methods that can be applied. For example, in this work, we use the sparse-grid method (Heiss & Winschel, 2008) based on Gauss–Hermite quadrature.

Diagonalization

It is clear from Equations 19, 20, 29, and 30 that all parameter dependence in our integrand is in terms of the linear predictor $\mathbf{y} = X\mathbf{w}$. That is, we are dealing with the integral of the form

$$F = \int d\mathbf{w}' \mathcal{N}(\mathbf{w}'|\widehat{\mathbf{w}}', C) \cdot f(X\mathbf{w}'), \quad (52)$$

where C is the covariance matrix and $X = \oplus_{j=1}^{k} \mathbf{g}'^{\top}_{j}$ is a fixed matrix constructed from a direct sum of k vectors. It helps to work in a diagonalized coordinate system, so that we can separate out the relevant dimensions of $\mathbf{w}$. We use the singular-value decomposition of the design matrix ($X = UGV^{\top}$ with $U = I$ and $V = Q^{\top}$). Because of the direct-sum construction, $XX^{\top}$ is already diagonal, and the left singular matrix is always I in this case. Then

$$G = XQ^{\top} = [\,G_k \quad G_q\,], \quad (53)$$

where G_k is a $k \times k$ diagonal matrix and G_q is a $k \times (q - k)$ matrix of zeros. We can now denote $\mathbf{w}_k = (w_1, \cdots, w_k)$ and $\mathbf{w}_q = (w_{k+1}, \cdots, w_q)$ in the diagonalized variable $\mathbf{w} = Q\mathbf{w}'$, such that

$$\mathbf{w} = [\mathbf{w}_k, \mathbf{w}_q]^{\top},$$
$$G\mathbf{w} = G_k \mathbf{w}_k = (g_1 w_1, g_2 w_2, \cdots g_k w_k).$$

Marginalization

Now we have

$$F = \int d\mathbf{w} \mathcal{N}(\mathbf{w}|\widehat{\mathbf{w}}, B^{-1}) \cdot f(G\mathbf{w}),$$
$$B^{-1} = QCQ^{\top}, \quad (54)$$

where B is the inverse of the new covariance matrix after diagonalization. If we block-decompose this matrix,

$$B = \begin{bmatrix} B_{kk} & B_{kq} \\ B_{qk} & B_{qq} \end{bmatrix}, \qquad B_{kq} = (B_{qk})^{\top}, \quad (55)$$

the Gaussian distribution is also decomposed as

$$\mathcal{N}(\mathbf{w}|\widehat{\mathbf{w}}, B^{-1}) = \mathcal{N}(\mathbf{w}_k|\widehat{\mathbf{w}}_k, B_*^{-1}) \cdot \mathcal{N}(\mathbf{w}_q|(\widehat{\mathbf{w}}_q - \mathbf{b}), B_{qq}^{-1}),$$

where $\mathbf{b} = B_{qq}^{-1} B_{qk} \mathbf{w}_k$ and $B_* = B_{kk} - B_{kq} B_{qq}^{-1} B_{qk}$. As the nonparallel part $\mathbf{w}_q$ is integrated out, we have marginalized the integral. It is useful to recall that if a variable $\mathbf{w} \sim \mathcal{N}(\widehat{\mathbf{w}}, C)$ is Gaussian distributed, its linear transform $\mathbf{y} = X\mathbf{w}$ is also Gaussian distributed as $\mathbf{y} \sim \mathcal{N}(\widehat{\mathbf{y}}, \Sigma)$, with $\widehat{\mathbf{y}} = X\widehat{\mathbf{w}}$ and $\Sigma = XCX^{\top}$. Changing the integration variable to $\mathbf{y} = G_k \mathbf{w}_k$ is then straightforward:

$$F = \int d\mathbf{w}_k \mathcal{N}(\mathbf{w}_k|\widehat{\mathbf{w}}_k, B_*^{-1}) \cdot f(G_k \mathbf{w}_k)$$
$$= \int d\mathbf{y} \mathcal{N}(\mathbf{y}|\widehat{\mathbf{y}}, \Sigma) \cdot f(\mathbf{y}),$$
$$\Sigma = G_k B_*^{-1} G_k^{\top}. \quad (56)$$

Standardization

Finally, in order to deal with the numerical integration, it is convenient to have the normal distribution standardized. We can use the Cholesky decomposition for the covariance matrix,

$$LL^{\top} = \Sigma_{t+1}, \quad (57)$$

such that the new variable $\boldsymbol{\theta} = L^{-1}(\mathbf{y} - \widehat{\mathbf{y}}_{t+1})$ is standard normal distributed. We can then write L directly in terms of the Cholesky decomposition of B_*:

$$L = G_k R^{-1} \quad \text{where} \quad R^{\top} R = B_*. \quad (58)$$

Importantly, with this transformation each dimension of $\boldsymbol{\theta}$ is independently and identically distributed. The objective function to be evaluated is now

$$F(\mathbf{x}) = \int d\mathbf{y} \cdot \mathcal{N}(\mathbf{y}|\widehat{\mathbf{y}}_{t+1}, \Sigma_{t+1}) \cdot f(\mathbf{y}, \mathbf{x})$$
$$= \int d\boldsymbol{\theta} \cdot \mathcal{N}(\boldsymbol{\theta}|\mathbf{0}, I) \cdot f(\boldsymbol{\phi}(\boldsymbol{\theta}), \mathbf{x}), \quad (59)$$

where $\boldsymbol{\phi}(\boldsymbol{\theta}) = \widehat{\mathbf{y}}_{t+1} + L\boldsymbol{\theta}$. Once the integration is standardized this way, there are a number of efficient numerical methods that can be applied.