ARTICLE Communicated by Tatyana Sharpee

Bayesian Active Learning of Neural Firing Rate Maps with Transformed Gaussian Process Priors

Mijung Park
mjpark@mail.utexas.edu
Electrical and Computer Engineering, University of Texas at Austin, Austin, TX 78712-1684, U.S.A.

J. Patrick Weller
jpweller@uw.edu
Gregory D. Horwitz
ghorwitz@uw.edu
Physiology and Biophysics, University of Washington, Seattle, WA 98195, U.S.A.

Jonathan W. Pillow
pillow@mail.utexas.edu
Center for Perceptual Systems, University of Texas at Austin, Austin, TX 78712-1043, U.S.A.

A firing rate map, also known as a tuning curve, describes the nonlinear relationship between a neuron's spike rate and a low-dimensional stimulus (e.g., orientation, head direction, contrast, color). Here we investigate Bayesian active learning methods for estimating firing rate maps in closed-loop neurophysiology experiments. These methods can accelerate the characterization of such maps through the intelligent, adaptive selection of stimuli. Specifically, we explore the manner in which the prior and utility function used in Bayesian active learning affect stimulus selection and performance. Our approach relies on a flexible model that involves a nonlinearly transformed gaussian process (GP) prior over maps and conditionally Poisson spiking. We show that infomax learning, which selects stimuli to maximize the information gain about the firing rate map, exhibits strong dependence on the seemingly innocuous choice of nonlinear transformation function. We derive an alternate utility function that selects stimuli to minimize the average posterior variance of the firing rate map and analyze the surprising relationship between prior parameterization, stimulus selection, and active learning performance in GP-Poisson models. We apply these methods to color tuning measurements of neurons in macaque primary visual cortex.

A supplemental appendix is available online at http://www.mitpressjournals.org/doi/suppl/10.1162/NECO_a_00615.

Neural Computation **26, 1519–1541** (2014)
doi:10.1162/NECO_a_00615

1 Introduction

A primary goal in systems neuroscience is to understand the functional relationship between stimuli and neural spike responses. When the parameter space is relatively low dimensional, this relationship is often described by a firing rate map or tuning curve, which is the mean spike rate as a function of the parameter vector of interest (Hubel & Wiesel, 1959; Georgopoulos, Schwartz, & Kettner, 1986; Maldonado, Gdecke, Gray, & Bonhoeffer, 1997; Carandini & Ferster, 2000; Ringach, Shapley, & Hawken, 2002; Ohki, Chung, Ch'ng, Kara, & Reid, 2005). Examples of firing rate maps include contrast-response functions, hippocampal place fields, grid fields in entorhinal cortex, and movement fields in motor cortex. Firing rate map estimation is an important component of neurophysiology experiments, often as a preliminary step for characterizing response properties before beginning a new experiment or in cases where the firing rate map changes due to some experimental manipulation.

In traditional or passive learning experiments, the experimenter collects data and then fits the model parameters offline using a fixed data set. In active learning experiments, by contrast, the experimenter fits model parameters online as data are collected and uses the model uncertainty to select subsequent stimuli in an optimal or maximally informative manner. The basic rationale is that one can often learn the model parameters much more quickly by selecting stimuli adaptively, informed by the data collected so far during the experiment. A variety of studies have developed active learning methods for closed-loop neurophysiology experiments, with applications to receptive field estimation (Lewi, Butera, & Paninski, 2009), color processing in macaque V1 (Horwitz & Hass, 2012), sound processing in grasshopper auditory receptor neurons (Machens, Gollisch, Kolesnikova, & Herz, 2005), and dynamic nonlinear stimulus integration in the retina (Bölinger & Gollisch, 2012). An overview of the active learning application to sensory physiology can be found in Benda, Gollisch, Machens, and Herz (2007).

Bayesian active learning relies on an explicit probabilistic model of the neural response and a utility function that determines the most useful stimulus given posterior uncertainty. Here, we consider Bayesian active learning under a Poisson observation model with a gaussian process (GP) prior. Gaussian processes provide a flexible, nonparametric prior over functions, allowing us to specify general beliefs about the function (e.g., degree of smoothness) without assuming a particular analytic form for the firing rate map. Because spike rates cannot be negative, we model the firing rate map as a gaussian process transformed by a fixed nonlinearity to ensure positive spike rates (Cunningham, Shenoy, & Sahani, 2008; Adams, Murray, & MacKay, 2009; Rad & Paninski, 2010; Park, Horwitz, & Pillow, 2011).

In this article, we examine how the choice of utility function and nonlinearity affects the performance of Bayesian active learning for neural firing

rate map estimation. We show that infomax learning, which selects stimuli based on an information-theoretic criterion (Lindley, 1956; Luttrell, 1985; Bernardo, 1979; MacKay, 1992; Paninski, 2005), is highly sensitive to the choice of nonlinearity in a manner that is somewhat surprising. We derive an alternate utility function based on minimizing the integrated posterior variance, which is equivalent to minimizing expected mean squared error in the firing rate map estimate. We develop an efficient algorithm for selecting stimuli according to this criterion and demonstrate its performance using simulated and real neural data sets.

The article is organized as follows. In section 2, we introduce the GP-Poisson model for firing rate map responses. In section 3, we review the basics of Bayesian active learning. In section 4, we derive information-theoretic and variance-based Bayesian active learning methods. In section 5, we examine the empirical properties of these two learning methods and their dependence on the choice of the nonlinear transformation function. In section 6, we compare these methods in simulated experiments with real and artificial neural data.

2 GP-Poisson Model for Firing Rate Maps

Following previous work on firing rate map estimation (Rad & Paninski, 2010; Park et al., 2011), we model the neural response as conditionally Poisson given a stimulus, with rate determined by the value of underlying firing rate map. We perform inference under transformed GP prior over the firing rate map, resulting in a hierarchical GP-Poisson model (see Figure 1A). Details of the model and Bayesian inference procedure are as follows.

2.1 Poisson Encoding Model. Let r_i denote the observed spike count on the ith trial of an experiment in response to a vector stimulus $\mathbf{x}_i$, and let $\lambda(\mathbf{x})$ denote the firing rate map, a function that takes on a nonnegative value at every point in stimulus space. We model the response $r_i|\mathbf{x}_i$ as Poisson:

$$p(r_i|\mathbf{x}_i) = \frac{1}{r_i!}\lambda_i^r e^{-\lambda_i}, \tag{2.1}$$

where $\lambda_i = \lambda(\mathbf{x}_i)$ is the spike rate at $\mathbf{x}_i$. Let $\mathcal{D}_t = \{(\mathbf{x}_i, r_i)\}_{i=1}^t$ denote the data collected up to time t in an experiment. Then we have the log likelihood of the data

$$\mathcal{L}(\boldsymbol{\lambda}) = \log p(\mathbf{r}|\boldsymbol{\lambda}) = \mathbf{r}^\top \log \boldsymbol{\lambda} - \mathbf{1}^\top \boldsymbol{\lambda} + c, \tag{2.2}$$

where $\mathbf{r} = (r_1, \ldots, r_t)^\top$, $\boldsymbol{\lambda} = (\lambda(\mathbf{x}_1), \ldots, \lambda(\mathbf{x}_t))^\top$, $\mathbf{1}$ is a vector of ones, and c is a constant.

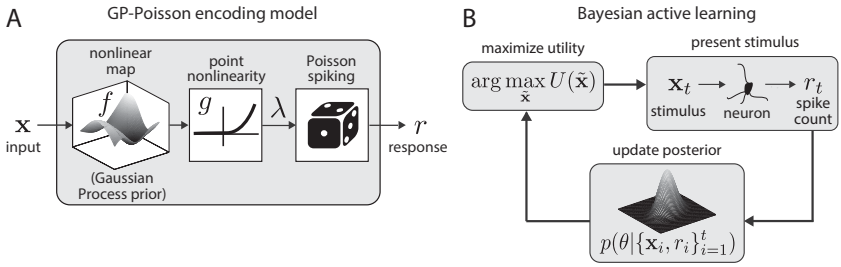

Figure 1: (A) GP-Poisson encoding model. A vector input $\mathbf{x}$ passes through a function f, whose scalar output is transformed via a nonlinearity g into a positive spike rate λ. The response r is a Poisson random variable with mean $\lambda(\mathbf{x}) = g(f(\mathbf{x}))$. (B) Schematic of Bayesian active learning for closed-loop neurophysiology experiments. After presenting a stimulus $\mathbf{x}_t$, we record the response r_t (upper right) and update the posterior over model parameters θ (bottom). Then we select a new stimulus by maximizing the expected utility $U(\tilde{\mathbf{x}})$, which quantifies the expected usefulness (e.g., information gain) of the stimulus $\tilde{\mathbf{x}}$ under the current posterior (upper left). We present this stimulus, and repeat.

2.2 Transformed Gaussian Process Prior. A popular approach for defining a prior distribution over smooth, nonnegative functions is to transform a gaussian process (GP) prior via a nonlinearity with positive range. We adopt this approach and model the rate map as $\lambda(\mathbf{x}) = g(f(\mathbf{x}))$, the composition of a real-valued function $f(\mathbf{x})$, governed by a GP prior, and a fixed scalar nonlinearity $g(\cdot)$ that maps f to the space of nonnegative spike rates (see Figure 1A).

The GP prior over f entails a multivariate normal distribution over the function values at any finite collection of points $(\mathbf{x}_1, \ldots \mathbf{x}_t)$ (Rasmussen & Williams, 2006):

$$\mathbf{f} \sim \mathcal{N}(\boldsymbol{\mu}_\mathbf{f}, K), \tag{2.3}$$

where $\mathbf{f} = (f(\mathbf{x}_1), \ldots f(\mathbf{x}_t))^\top$ denotes the function evaluated at these points, $\boldsymbol{\mu}_\mathbf{f} = \mu_f \mathbf{1}$ is a constant vector with value μ_f (which denotes the prior mean), and the covariance matrix K has the $i,\ j$th element given by a kernel function: $k(\mathbf{x}_i, \mathbf{x}_j)$. Here, we employ the popular gaussian kernel, which enforces smoothness on f,

$$K_{i,j} = k(\mathbf{x}_i, \mathbf{x}_j) = \rho \exp\big(-||\mathbf{x}_i - \mathbf{x}_j||^2/(2\tau^2)\big), \tag{2.4}$$

where the hyperparameters ρ and τ control the marginal variance and smoothness of f, respectively. The GP prior therefore has three

hyperparameters: $\phi = \{\mu_f, \rho, \tau\}$. The log GP prior at the observed points $(\mathbf{x}_1, \ldots, \mathbf{x}_t)$ is given by

$$\log p(\mathbf{f}|\phi) = -\frac{1}{2}\log|2\pi K| - \frac{1}{2}(\mathbf{f} - \boldsymbol{\mu}_\mathbf{f})^\top K^{-1}(\mathbf{f} - \boldsymbol{\mu}_\mathbf{f}). \tag{2.5}$$

2.3 Posterior Inference. Given a data set consisting of stimulus-response pairs, our goal is to infer the firing rate map $\lambda(\mathbf{x})$ at any stimulus $\mathbf{x}$. A simple, tractable approach to this problem is to maintain a gaussian process approximation to the posterior over f, which is the firing rate map transformed by g^{-1}. This approach is justified by the fact that the log likelihood is concave in f subject to certain conditions on g (Paninski, 2004). By contrast, a direct GP prior for the firing rate map λ would assign positive probability to negative spike rates. From a gaussian process approximation to the posterior over f, we can obtain the posterior over λ by transforming it via g. In this article, we consider two popular choices of g: (1) exponential and (2) soft-rectifying or *softrect*, given by $g(x) = \log(e^x + 1)$. When the g is exponential, the approximate posterior over λ is a log-normal process.

Our inference scheme involves two steps. First, we numerically find the posterior mode of f at the finite number of stimulus points $\{\mathbf{x}_1, \ldots, \mathbf{x}_t\}$ presented so far in the experiment; then we make a GP approximation to the posterior over f at its mode using Laplace's method. Details are as follows.

Given data $\mathcal{D}_t$, we first compute the maximum a posteriori estimate $\mathbf{f}_\mathbf{map}$ by numerically maximizing the log posterior, given by

$$\log p(\mathbf{f}|\mathcal{D}_t, \phi) = \mathbf{r}^\top \log(g(\mathbf{f})) - \mathbf{1}^\top g(\mathbf{f}) - \frac{1}{2}(\mathbf{f} - \boldsymbol{\mu}_\mathbf{f})^\top K^{-1}(\mathbf{f} - \boldsymbol{\mu}_\mathbf{f}) + c, \tag{2.6}$$

where, as before, $\mathbf{f} = (f(\mathbf{x}_1), \ldots f(\mathbf{x}_t))^\top$ denotes the vector of function values at the points $(\mathbf{x}_1, \ldots \mathbf{x}_t)$. The log posterior over $\mathbf{f}$ is concave so long as g is convex and log concave (Wedderburn, 1976; McCullagh & Nelder, 1989; Paninski, 2004), which is true for both choices of $g(\cdot)$ considered here. This means convex optimization methods will provably converge to a global minimum of the negative log posterior and provides justification for making a gaussian approximation to the posterior.

Second we make a gaussian approximation to the $p(\mathbf{f}|\mathcal{D}_t, \phi)$ at $\mathbf{f}_\mathbf{map}$. The Laplace approximation results from a second-order Taylor series expansion of the log posterior around its mode. The resulting approximation is

$$p(\mathbf{f}|\mathcal{D}_t, \phi) \approx \mathcal{N}(\mathbf{f}_\mathbf{map}, \Sigma), \tag{2.7}$$

where covariance Σ is the negative inverse Hessian of the log posterior, evaluated at $\mathbf{f} = \mathbf{f}_{\mathbf{map}}$ (Bishop, 2006).

Given this approximation, we obtain a gaussian marginal posterior over $\mathbf{f}^* = f(\mathbf{x}^*)$, the function evaluated at a collection of test stimuli $\mathbf{x}^* = \{\mathbf{x}_i^*\}_{i=1}^N$, by integrating out $\mathbf{f}$ from the joint distribution $p(\mathbf{f}^*, \mathbf{f}|\mathbf{x}^*, \mathcal{D}_t, \phi)$,

$$p(\mathbf{f}^*|\mathbf{x}^*, \mathcal{D}_t, \phi) = \int p(\mathbf{f}^*|\mathbf{x}^*, \mathbf{f}, \phi)p(\mathbf{f}|\mathcal{D}_t, \phi)\mathrm{d}\mathbf{f} \approx \mathcal{N}(\boldsymbol{\mu}_t, \Lambda_t), \tag{2.8}$$

with mean and covariance

$$\boldsymbol{\mu}_t = \boldsymbol{\mu}_{\mathbf{f}} + K^* K^{-1}(\mathbf{f}_{\mathbf{map}} - \boldsymbol{\mu}_{\mathbf{f}}), \tag{2.9}$$

$$\Lambda_t = K^{**} - K^*(L^{-1} + K)^{-1}K^{*\top}, \tag{2.10}$$

which depend on cross-covariance matrices $K_{ij}^* = k(\mathbf{x}_i^*, \mathbf{x}_j)$ and $K_{ij}^{**} = k(\mathbf{x}_i^*, \mathbf{x}_j^*)$. In section 4, we employ this approximate posterior distribution to evaluate the expected utility of candidate stimuli.

2.4 Marginal Likelihood for Hyperparameters. The posterior over f (see equation 2.8) is influenced by the choice of hyperparameters ϕ. We set ϕ by maximizing marginal likelihood, or "evidence," given by

$$p(\mathbf{r}|\phi) = \int p(\mathbf{r}|\mathbf{f})p(\mathbf{f}|\phi)d\mathbf{f}. \tag{2.11}$$

Although this integral is not analytically tractable for the GP-Poisson model, we can approximate it using Laplace's method. First, rearrange the terms in Bayes' rule to write the marginal likelihood,

$$p(\mathbf{r}|\phi) = \frac{p(\mathbf{r}|\mathbf{f})p(\mathbf{f}|\phi)}{p(\mathbf{f}|\mathcal{D}_t, \phi)}. \tag{2.12}$$

Then use the Laplace approximation to the posterior to evaluate the denominator and evaluate the entire expression at $\mathbf{f} = \mathbf{f}_{\mathbf{map}}$, where this approximation is most accurate (Rasmussen & Williams, 2006).

For fast optimization, we adopt a method (introduced in Park et al., 2011) that involves iterating the following three steps until convergence: (1) numerically optimize the posterior to find $\mathbf{f}_{\mathbf{map}}$ and the Laplace approximation given the data and current hyperparameters; (2) compute a gaussian "site potential" approximation to the likelihood by taking the ratio of the (gaussian approximate) posterior to the gaussian prior; and (3) maximize the marginal likelihood for ϕ, equation 2.12, while using the local gaussian approximation to the likelihood to analytically determine $\mathbf{f}_{\mathbf{map}}$. (See appendix A for details.)

3 Bayesian Active Learning

In the following, we briefly review the basics of Bayesian active learning and describe fast posterior update methods that enable us to apply active learning to neurophysiology experiments.

3.1 Overview of Bayesian Active Learning. Bayesian active learning (depicted in Figure 1B) requires three ingredients: (1) an observation model $p(r|\mathbf{x}, \theta)$, describing the conditional response r given a stimulus $\mathbf{x}$ and a parameter vector θ; (2) a prior distribution over the model parameters $p(\theta)$; and (3) a utility function U given by Chaloner and Verdinelli (1995),

$$U(\theta, \{\mathbf{x}, r\}|\mathcal{D}_t), \tag{3.1}$$

which quantifies the usefulness of the candidate stimulus-response pair $(\mathbf{x}, r)$ for learning about θ given the data recorded up to time t.

In Bayesian active learning, we choose the stimulus with maximal expected utility $U(\mathbf{x}|\mathcal{D}_t)$, which is obtained by averaging the utility function with respect to the joint distribution over θ and r given the data

$$U(\mathbf{x}|\mathcal{D}_t) = \mathbb{E}_{r,\theta|\mathcal{D}_t,\mathbf{x}}\left[U(\theta, \{\mathbf{x}, r\}|\mathcal{D}_t)\right] = \int U(\theta, \{\mathbf{x}, r\}|\mathcal{D}_t)\, p(r, \theta|\mathcal{D}_t, \mathbf{x}) \mathrm{d}\theta \mathrm{d}r, \tag{3.2}$$

where the joint distribution factorizes with respect to the posterior over θ and the predictive distribution for r: $p(r, \theta|\mathcal{D}_t, \mathbf{x}) = p(\theta|\mathcal{D}_t, \{\mathbf{x}, r\})p(r|\mathcal{D}_t, \mathbf{x})$. The optimal stimulus at the next time step is (by definition) the stimulus with maximal expected utility:

$$\mathbf{x}_{t+1} = \arg\max_{\mathbf{x}\in\mathcal{X}} U(\mathbf{x}|\mathcal{D}_t), \tag{3.3}$$

where $\mathcal{X}$ denotes the stimulus space. This greedy rule requires sequential posterior updates as new observations come in, which are discussed in the next section.

As we show here, the choice of utility function can critically affect the performance of an active learning algorithm. Previously proposed utility functions include prediction error on test data (Roy & Mccallum, 2001; Cohn, Ghahramani, & Jordan, 1996), misclassification error (Kapoor, Horvitz, & Basu, 2007), gain in mutual information (Lindley, 1956; Luttrell, 1985; Bernardo, 1979; MacKay, 1992; Paninski, 2005; Lewi et al., 2009), and the mutual information between function values at tested and all untested locations (Krause, Singh, & Guestrin, 2008).

3.2 Fast Posterior Updates. Bayesian learning methods require the posterior over parameters given a new observation to compute the expected utility (see equation 3.2). Here, we describe a method for fast, approximate, recursive posterior updates under the GP-Poisson model, which allow us to rapidly compute expected utilities (Lewi et al., 2009). Note that we do not use these approximate rules for updating our posterior after an actual observation; we use them only for quickly computing expected utilities, which require expectations with respect to the posterior given hypothetically observed data.

Let $\mathbf{f}^* = f(\mathbf{x}^*)$, where $\mathbf{x}^*$ denotes a grid of points that we will use for representing the posterior over f. During a single step of active learning, we have a gaussian approximation to the posterior over $\mathbf{f}^*$ given the data collected up to time t (see equation 2.8). Any hypothetical stimulus-response pair $(\mathbf{x}', r')$ yields a new likelihood term that must be added to the log posterior,

$$\log p(\mathbf{f}^*|\boldsymbol{\mu}', \Lambda') \approx \log p(\mathbf{f}^*|\boldsymbol{\mu}_t, \Lambda_t) + \log p(r'|\lambda(\mathbf{x}')), \tag{3.4}$$

where $\boldsymbol{\mu}'$ and Λ' denote the posterior mean and covariance given $(\mathcal{D}_t, \mathbf{x}', r')$, respectively. For notational simplicity, we have dropped dependence of the posterior on hyperparameters ϕ. We can update the mean and covariance by differentiating equation 3.4 and equating the two sides:

$$\begin{aligned}
\frac{\partial}{\partial \mathbf{f}^*} \log p(\mathbf{f}^*|\boldsymbol{\mu}', \Lambda') &= -\Lambda'^{-1}(\mathbf{f}^* - \boldsymbol{\mu}') \\
&\approx -\Lambda_t^{-1}(\mathbf{f}^* - \boldsymbol{\mu}_t) + \frac{\partial}{\partial \mathbf{f}^*} \log p(r'|\lambda(\mathbf{x}')), \\
\frac{\partial^2}{\partial^2 \mathbf{f}^*} \log p(\mathbf{f}^*|\boldsymbol{\mu}', \Lambda') &= -\Lambda'^{-1} \approx -\Lambda_t^{-1} + \frac{\partial^2}{\partial^2 \mathbf{f}^*} \log p(r'|\lambda(\mathbf{x}')).
\end{aligned}$$

Solving for the mean and covariance $(\boldsymbol{\mu}', \Lambda')$ yields

$$\begin{aligned}
&\text{Mean update: } \boldsymbol{\mu}'(\mathbf{x}^*) = \boldsymbol{\mu}_t(\mathbf{x}^*) + \Delta\Lambda_t(\mathbf{x}^*, \mathbf{x}'), \\
&\text{Covariance update: } \Lambda'(\mathbf{x}_i^*, \mathbf{x}_j^*) \approx \Lambda_t(\mathbf{x}_i^*, \mathbf{x}_j^*) - \frac{\mathcal{J}_{\boldsymbol{\mu}(\mathbf{x})}\Lambda_t(\mathbf{x}_i^*, \mathbf{x}')\Lambda_t(\mathbf{x}_j^*, \mathbf{x}')}{1 + \mathcal{J}_{\boldsymbol{\mu}'(\mathbf{x}')}\Lambda_t(\mathbf{x}', \mathbf{x}')},
\end{aligned} \tag{3.5}$$

where the first derivative of log likelihood is denoted by $\Delta = \frac{\partial}{\partial f} \log p(r'|\lambda(\mathbf{x}'))$; the second derivative of log likelihood, the so-called observed Fisher information, is denoted by $\mathcal{J}_{\boldsymbol{\mu}'(\mathbf{x}')} = -\frac{\partial^2}{\partial f^2} \log p(r'|\lambda(\mathbf{x}'))$; and both first and second derivatives are evaluated at $f = \boldsymbol{\mu}'(\mathbf{x}')$.

4 Active Learning of Firing Rate Maps

Equipped with the first two ingredients for Bayesian active learning, as specified in section 2, we now consider the third: the utility function. The utility function specifies the learning objective and must be efficiently maximized in each trial to rapidly select the next stimulus. Here we consider two utility functions: (1) a popular information-theoretic utility function, which leads to an infomax learning rule, and (2) a posterior-variance-based utility (or loss) function, which seeks to minimize the expected squared deviation (or posterior variance) of the firing rate map estimate.

4.1 Information-Maximization Learning. In infomax learning, the goal is to maximize information gain about the firing rate map λ from the observed data. Therefore, the expected utility function is the mutual information between λ and a future response r' to a candidate stimulus $\mathbf{x}'$, given $\mathcal{D}_t = \{(\mathbf{x}_1, r_1), \ldots, (\mathbf{x}_t, r_t)\}$, the data collected so far in the experiment. Because mutual information is not altered by invertible transformations and $\lambda = g(f)$, the mutual information between λ and r' is the same as the mutual information between f and r'. The expected utility is therefore

$$U(\mathbf{x}'|\mathcal{D}_t) = I(f; r'|\mathcal{D}_t\mathbf{x}'), \tag{4.1}$$

$$= H_{\mathcal{D}_t}(f) - H_{(\mathcal{D}_t,\mathbf{x}')}(f|r'), \tag{4.2}$$

where $H_{\mathcal{D}_t}(f)$ denotes the entropy of f given D_t and $H_{(\mathcal{D}_t,\mathbf{x}')}(f|r')$ denotes the conditional entropy of f given r', conditioned on $\mathcal{D}_t$ and x'. Because f is a function (and thus infinite-dimensional), the mutual information (see equations 4.1 and 4.2) is not actually well defined. We therefore define the mutual information using a fixed, uniform grid of points $\mathbf{f}^*$, as defined above. The firing rate map on the grid of points is denoted by $\boldsymbol{\lambda}^* = g(\mathbf{f}^*)$. Although this gridding approach will not scale well to high dimensions, it suffices here because the stimulus space is relatively low-dimensional (i.e., two to three dimensions for the examples we consider).[1]

Under a gaussian approximate posterior (see equation 2.8), the posterior entropy is proportional to the log determinant of the posterior covariance matrix (evaluated on a grid of function values), which reduces the expected utility to

$$I(f; r'|\mathcal{D}_t, \mathbf{x}') \approx \log|\Lambda_t| - \mathbb{E}_{r'|\mathcal{D}_t,\mathbf{x}'} \log|\Lambda_{t+1}|,$$

$$\approx \sigma_t^2(\mathbf{x}')\ \mathbb{E}_{r'|f,\mathbf{x}'}\mathbb{E}_{f|\boldsymbol{\mu}_t(\mathbf{x}'),\sigma_t^2(\mathbf{x}')}[\mathcal{J}_{f(\mathbf{x}')}], \tag{4.3}$$

[1] To avoid the curse of dimensionality, one can use a nonregular grid, sampled from a probability distribution that puts high resolution in the regions strongly affecting the expected utility (Hennig & Schuler, 2012).

Algorithm 1: Sequential Infomax Learning.

Given the posterior $p(\mathbf{f}^*|\boldsymbol{\mu}_t, \Lambda_t)$ at time t,

1. Compute the expected information gain of each candidate $\mathbf{x}'$ (equation 4.3).

2. Select a stimulus $\mathbf{x}_{t+1}$ that maximizes the expected information gain about $\boldsymbol{\lambda}^*$.

3. Measure response r_{t+1} to $\mathbf{x}_{t+1}$.

4. Update the posterior $p(\mathbf{f}^*|\boldsymbol{\mu}_{t+1}, \Lambda_{t+1})$ and the hyperparameters.

repeat

where $\sigma_t^2(\mathbf{x}')$ is the posterior variance at $\mathbf{x}'$. The last line follows from equation 3.5, and the proof is given in Lewi et al. (2009) and MacKay (1992).

The observed Fisher information $\mathcal{J}_{f(\mathbf{x}')}$ under the GP-Poisson model is given by

$$\mathcal{J}_{f(\mathbf{x}')} = \frac{r'}{\lambda^2(\mathbf{x}')}\left[\frac{\partial\lambda(\mathbf{x}')}{\partial f(\mathbf{x}')}\right]^2 + \left[1 - \frac{r'}{\lambda(\mathbf{x}')}\right]\frac{\partial^2\lambda(\mathbf{x}')}{\partial f^2(\mathbf{x}')}. \tag{4.4}$$

If $g = \exp(\cdot)$, the observed Fisher information $\mathcal{J}_{f(\mathbf{x}')}$ is simply $\lambda(\mathbf{x}')$, which is independent of the response r'. In this case, the utility simplifies to

$$U(\mathbf{x}'|\mathcal{D}_t) = \sigma_t^2(\mathbf{x}')\exp(\boldsymbol{\mu}_t(\mathbf{x}') + \frac{1}{2}\sigma_t^2(\mathbf{x}')). \tag{4.5}$$

However, if $g = \log(\exp(\cdot) + 1)$, the observed Fisher information $\mathcal{J}_{f(\mathbf{x}')}$ depends on the response r'. The expectation with respect to $p(r'|f, \mathbf{x}')$ in equation 4.3 conveniently leaves the following:

$$U(\mathbf{x}'|\mathcal{D}_t) \approx \sigma_t^2(\mathbf{x}')\ \mathbb{E}_{f|\boldsymbol{\mu}_t(\mathbf{x}'),\sigma_t^2(\mathbf{x}')}\left[\frac{1}{\lambda(\mathbf{x}')}\left[\frac{\partial\lambda(\mathbf{x}')}{\partial f(\mathbf{x}')}\right]^2\right]. \tag{4.6}$$

Unfortunately, there is no closed form for this expected utility in this case. However, as Lewi et al. (2009) suggested, one can precompute the expected utility for each $(\boldsymbol{\mu}_t(\mathbf{x}'), \sigma_t^2(\mathbf{x}'))$ and store it in a $2D$ lookup table, to rapidly determine expected utility of $\mathbf{x}'$ during experiments. Alternatively, one can numerically compute the expected utility using Gauss-Hermite quadrature (see appendix B). A summary of the algorithm is given in algorithm 1.

Note that for both choices of nonlinearity g, the expected utility for a stimulus $\mathbf{x}'$ depends on only the posterior mean and variance of $f(\mathbf{x}')$, the function value at that point. It does not depend on the function value at other grid points, and the computational cost of evaluating the expected utility (at a single stimulus) is completely independent of grid size. The only

cost of using a finer grid over stimulus space is the linear cost of evaluating mutual information at each grid point. This independence of grid size will not hold for the utility function we consider next.

4.2 Variance-Minimization Learning. Next we consider a utility function that corresponds to minimizing the posterior variance over the firing rate map λ. This is equivalent to choosing stimuli to minimize mean squared error in the estimate of λ. We will refer to this approach as variance minimization, or *varmin* learning:

$$U(\mathbf{x}'|\mathcal{D}_t) = -\mathbb{E}_{r'|\mathcal{D}_t,\mathbf{x}'}\left[\int \mathbb{V}(\lambda(\mathbf{x})|\mathcal{D}_t, \mathbf{x}', r')\mathrm{d}\mathbf{x}\right], \tag{4.7}$$

where we denote the variance of $\lambda(\mathbf{x})$ given the data as

$$\mathbb{V}(\lambda(\mathbf{x})|\mathcal{D}_t, \mathbf{x}', r') = \int (\lambda(\mathbf{x}) - \hat{\lambda}(\mathbf{x}))^2 p(f(\mathbf{x})|\mathcal{D}_t, \mathbf{x}', r')\mathrm{d}f(\mathbf{x}), \tag{4.8}$$

and $\hat{\lambda}(\mathbf{x})$ denotes the posterior mean of $\lambda(\mathbf{x})$ given $(\mathcal{D}_t, \mathbf{x}', r')$, also known as the Bayesian least squares estimate.

We compute the expected utility by numerical integration on the regular grid of points $\{\mathbf{x}^*\}_{i=1}^N$:

$$U(\mathbf{x}'|\mathcal{D}_t) = -\mathbb{E}_{r'|f,\mathbf{x}'}\mathbb{E}_{f|\boldsymbol{\mu}_t(\mathbf{x}'),\sigma_t^2(\mathbf{x}')}\left[\sum_{i=1}^{N}\mathbb{V}(\lambda(\mathbf{x}_i^*)|\mathcal{D}_t, \mathbf{x}', r')\right]. \tag{4.9}$$

Optimizing this utility involves intractable joint optimization and expectations: computing the expected variance $\mathbb{V}(\lambda(\mathbf{x}_i^*)|\mathcal{D}_t, \mathbf{x}', r')$ requires updating the posterior over f given $(\mathbf{x}', r')$ and averaging the unknown response r' under the predictive distribution $p(r'|\mathbf{x}', \mathcal{D}_t)$, as well as averaging over the current posterior $p(f|\mathcal{D}_t)$. Several papers have suggested algorithms for this type of problem using Monte Carlo sampling (Kuck, de Freitas, & Doucet, 2006; Müller & Parmigiani, 1996). However, due to the time constraint in typical neurophysiology experiments ($\ll$500 ms), we need to rapidly compute the expectations in equation 4.9. Therefore, we adopt an approximate solution that involves computing the expected variance of f for each candidate $\mathbf{x}'$ and transforming it into the variance of λ. Details are as follows.

- **Expected variance of *f***: The posterior covariance update rule in equation 3.5 simplifies the expected variance of f to

$$\mathbb{E}_{r'|\mathcal{D}_t,\mathbf{x}'}\mathbb{V}[f(\mathbf{x}_i^*)|\mathcal{D}_t, \mathbf{x}', r'] = \sigma_t^2(\mathbf{x}_i^*) - \mathbb{E}_{r'|f,\mathbf{x}'}\mathbb{E}_{f|\boldsymbol{\mu}_t(\mathbf{x}'),\sigma_t^2(\mathbf{x}')}\left[\frac{\mathcal{J}_{\boldsymbol{\mu}'(\mathbf{x}')}\Lambda_t^2(\mathbf{x}_i^*, \mathbf{x}')}{1 + \mathcal{J}_{\boldsymbol{\mu}'(\mathbf{x}')}\sigma_t^2(\mathbf{x}')}\right], \tag{4.10}$$

Algorithm 2: Sequential Varmin Learning.

Given the posterior $p(\mathbf{f}^*|\boldsymbol{\mu}_t, \Lambda_t)$ at time t at grid points $\{\mathbf{x}_i^*\}_{i=1}^N$:

1. Compute the expected variance of $\mathbf{f}^*$ for each candidate $\mathbf{x}'$ (equation 4.10).

2. Transform the expected variance of $\mathbf{f}^*$ (from step **1**) into the expected total variance of $\boldsymbol{\lambda}^*$ for each $\mathbf{x}'$ (equation 4.13).

3. Select a stimulus $\mathbf{x}_{t+1}$ that minimizes the expected total variance of $\boldsymbol{\lambda}^*$.

4. Measure response r_{t+1} to $\mathbf{x}_{t+1}$.

5. Update the posterior $p(\mathbf{f}^*|\boldsymbol{\mu}_{t+1}, \Lambda_{t+1})$ and the hyperparameters.

repeat

where directly computing the double integrals is intractable. For fast computation, we make the following point estimates: r' to its mean $\lambda(\mathbf{x}')$ and $\boldsymbol{\mu}'(\mathbf{x}')$ to the current mean $\boldsymbol{\mu}_t(\mathbf{x}')$. Under these approximations, the observed Fisher information simplifies to

$$\mathcal{J}_{\boldsymbol{\mu}'(\mathbf{x}')} \approx g(\boldsymbol{\mu}_t(\mathbf{x}')), \qquad \text{for } g = \exp(\cdot), \tag{4.11}$$

$$\approx \frac{1}{\lambda(\mathbf{x}')}\left[\frac{\partial\lambda(\mathbf{x}')}{\partial f(\mathbf{x}')}\right]^2_{f=\boldsymbol{\mu}_t(\mathbf{x}')}, \quad \text{for } g = \log(\exp(\cdot)+1). \tag{4.12}$$

- **Transformation of variance of f into variance of λ**: The expected variance of λ at $\mathbf{x}_i^*$ is computed by

$$\mathbb{E}_{r'|\mathcal{D}_t,\mathbf{x}'}\mathbb{V}(\lambda(\mathbf{x}_i^*)|\mathcal{D}_t, \mathbf{x}', r') \approx \int g^2(f_i)\mathcal{N}(\mu_i', \sigma_i'^2)\mathrm{d}f_i - \left[\int g(f_i)\mathcal{N}(\mu_i', \sigma_i'^2)\mathrm{d}f_i\right]^2, \tag{4.13}$$

where $f_i = f(\mathbf{x}_i^*)$, the mean of f_i is denoted by $\mu_i' = \boldsymbol{\mu}'(\mathbf{x}_i^*) \approx \boldsymbol{\mu}_t(\mathbf{x}_i^*)$, and the variance of f_i is denoted by $\sigma_i'^2 = \sigma'^2(\mathbf{x}_i^*)$. If $g = \exp(\cdot)$, we can compute equation 4.13 analytically due to log-normal λ; however, if $g = \log(\exp(\cdot)+1)$, we numerically compute it either via Gauss-Hermite quadrature or using a precomputed 2D lookup table.

Finally, we obtain the integrated variance of λ by summing up the expected variances of $\lambda(\mathbf{x}_i^*)$ for all the grid points $\{\mathbf{x}_i^*\}_{i=1}^N$. A summary of the algorithm is given in algorithm 2. (Pseudocode is provided in appendix C.)

4.3 Assessing Accuracy of Approximations. The active learning methods described above rely on approximate computations of expected utility. To assess the accuracy of these approximations, we performed numerical calculations of the exact utility for a single stimulus, ignoring the

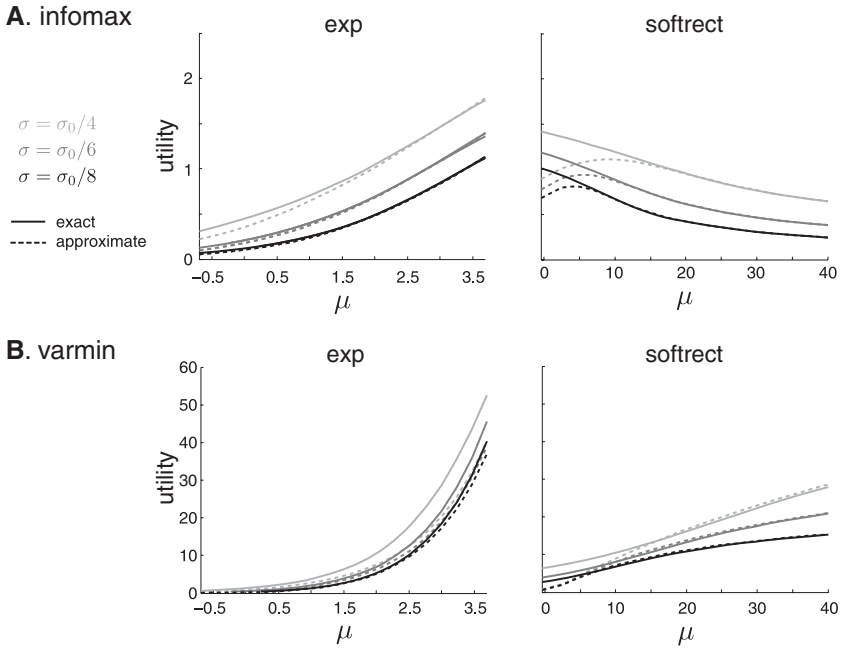

Figure 2: Comparison of exact and approximate utility function for infomax and varmin learning. (A) Infomax learning under exponential (left) and softrect nonlinearity (right). (B) Same, for varmin learning. Traces represent the exact (solid) and approximate utility (dashed) as a function of the posterior mean for three different values of the posterior standard deviation σ. Standard deviations are given in units of $\sigma_0 = g^{-1}(40)$, which is 3.7 for exponential (left) and 40 for softrect nonlinearity (right). Note that the approximate utility matches the exact utility to reasonable accuracy once the posterior becomes sufficiently concentrated.

effects of space, for different settings of a gaussian prior over the transformed rate map $f = g^{-1}(\lambda(x))$ (see Figure 2). For each setting of the prior mean μ and variance σ^2, we represented the prior using a grid of values $\{f_1, \ldots, f_n\}$ over the range $\mu \pm 4\sigma$, then computed expected utility as $U = \sum_{r=0}^{R_{\max}} u[\hat{p}(\{f_j\}|r, \mu, \sigma)]p(r|\mu, \sigma)$, where $p(r|\mu, \sigma) = \int df p(r|f)p(f|\mu, \sigma) \approx \sum_{i=1}^{n} \frac{1}{r!} g(f_i)^r e^{-g(f_i)} \frac{1}{\sqrt{2\pi\sigma^2}} \exp(-\frac{(f_i-\mu)^2}{2\sigma^2})$ is the marginal response distribution given μ and σ, and $\hat{p}(\{f_j\}|r, \mu, \sigma)$ is the gridded representation of the posterior over f given the prior and spike count r, which is proportional to $g(f_j)^r e^{-g(f_j) - \frac{(f_j-\mu)^2}{2\sigma^2}}$. We used maximal spike count $R_{\max} = 250$ to ensure we captured the support of the marginal response distribution, but larger values could be used as needed. Finally, $u(\cdot)$ is the Shannon entropy

$-\sum_{j=1}^{n} \hat{p}(f_j|r, \mu, \sigma^2) \log \hat{p}(f_j|r, \mu, \sigma^2)$ for infomax learning and the variance $-\sum_{j=1}^{n}(f_j - \bar{f})^2 \hat{p}(f_j|r, \mu, \sigma^2)$ for varmin learning.

Figure 2 shows exact and approximate utilities as a function of the prior mean μ for three different prior variances σ^2. When the prior is broad, which resembles the posterior after a small number of samples, the discrepancy between exact and approximate utilities is noticeable, especially for small μ. However, accuracy improves rapidly as the prior variance shrinks, indicating that the approximate utility calculations do not adversely affect stimulus selection once a relatively small amount of data has been collected.

We can roughly assess how many data are needed to guarantee accurate utility calculations by examining how quickly the posterior narrows as a function of the amount of data observed. For simplicity, consider the case where the prior has infinite variance, so the posterior is determined entirely by the likelihood. If we observe R total spikes during T seconds of stimulus presentation at a particular stimulus location, the MAP estimate is $\hat{f}_{map} = g^{-1}(\frac{R}{T})$, and the posterior variance under the Laplace approximation will be $\sigma^2_{post} = -(R\frac{[g(\hat{f}_{map})g''(\hat{f}_{map}) - g'^2(\hat{f}_{map})]}{g^2(\hat{f}_{map})} - Tg''(\hat{f}_{map}))^{-1}$. Under the exponential nonlinearity, this yields $\hat{f}_{map} = \log \frac{R}{T}$ and the simple result $\sigma^2_{post} = \frac{1}{R}$. The smallest-variance curves in Figure 2 (left column, with a standard deviation of $\sigma_0/8 = 0.46$) would therefore require $R \approx 4.7$ spikes.

For the soft-rectifying nonlinearity, we obtain $\hat{f}_{map} = \log(e^{\frac{R}{T}} - 1)$ and posterior variance $\sigma^2_{post} = \frac{1}{R}(\frac{\omega^2}{(1-e^{-\omega})^2})$, where $\omega = \frac{R}{T}$ is the observed spike rate. Thus, posterior variance depends on not just the number of spikes but the neuron's absolute spike rate as well. For small rates ω, this resembles the case of the exponential nonlinearity, with $\sigma^2_{post} \approx \frac{1}{R}$. For larger rates, we obtain $\sigma^2_{post} \approx \frac{\omega^2}{R}$. The curves in Figure 2 (right column) indicate that the accuracy of our approximations depends on both the mean and variance of $p(f)$. In particular, for rates $\omega \approx \mu > 20$, the curve defined by with variance $\sigma^2_{post} = (\sigma_0/4)^2 = 100$ exhibits high accuracy, which can be achieved with $R = \frac{\omega^2}{100} = 4$ spikes. For a lower rate of $\omega = 10$, good accuracy is not obtained until the $\sigma^2_{post} = (\sigma_0/8)^2 = 25$ curve (dark gray), but this variance can once again be achieved with $R = 100/25 = 4$ spikes. Thus, four to five observed spikes seems sufficient to make our approximate utility calculations accurate, regardless of the choice of nonlinearity. Nevertheless, posterior variance may contract more slowly if the prior has finite variance with a low mean, suggesting we might want more accurate methods for characterizing maps with very low spike rates.

The density of spatial sampling necessary to achieve small posterior variance across the entire map can be determined by noting that due to the smoothing induced by the gaussian process prior, observing R spikes in response to a stimulus $\mathbf{x}_0$ is equivalent to observing $R\exp(-\frac{|\mathbf{x}_0 - \mathbf{x}_1|^2}{2\tau^2})$

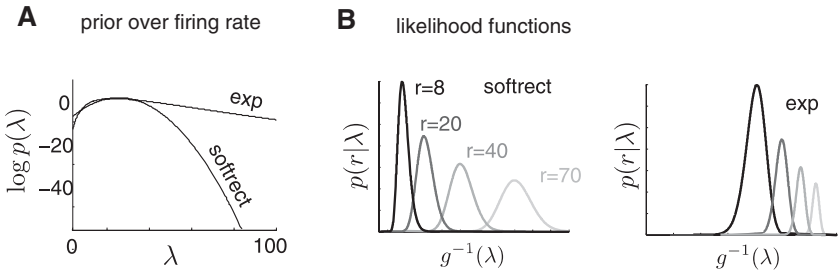

Figure 3: Effects of nonlinearity on likelihood and prior. (A) Log prior on λ under different choices of nonlinearity. With exponential nonlinearity, the log prior has a much heavier tail than that with softrect nonlinearity. (B) Poisson likelihood $p(r|\lambda)$ as a function of spike rate transformed by the inverse nonlinearity $f = g^{-1}(\lambda)$ for several fixed values of spike count r. With softrect nonlinearity, the likelihood is narrower when the spike count is low, indicating that the most informative stimuli, all else being equal, are those likely to elicit low spike counts. With exponential nonlinearity, the likelihood gets narrower with increasing r, indicating that the most informative stimuli are those likely to elicit large spike counts.

"effective spikes" at a location $\mathbf{x}_1$. Thus, to observe five effective spikes at $\mathbf{x}_1$, we would need 8.25 spikes within a radius of τ, or 13.6 spikes within a radius of $\sqrt{2}\tau$. Although length scale τ cannot in general be known a priori, we can use prior expectations about the smoothness of $\lambda(\mathbf{x})$ to determine a reasonable sampling density.

5 Effects of Nonlinearity and Utility on Learning

Empirically we observe that the stimuli selected as most useful in Bayesian active learning paradigms depends critically on both the utility function and the choice of nonlinearity g. Specifically, infomax learning selects very different stimuli when the nonlinearity is exponential instead of soft rectification. In this section, we examine how and why this phenomenon occurs.

First, we note that the nonlinearity g determines how the probability mass of the GP prior is spread out over firing rates in λ space. For the exponential nonlinearity, the marginal prior over λ is log normal, which leads to $p(\lambda)$, a very heavy right tail (see Figure 3A). By contrast, the softrect nonlinearity transforms the right tail of the gaussian prior linearly, resulting in a $p(\lambda)$ with a very light (i.e., gaussian) right tail. In practical terms, this means that under the exponential nonlinearity, the prior assigns a lot of probability mass to very high spike rates, so that (in principle) there is a lot more to learn by presenting stimuli that might probe these high firing rates.

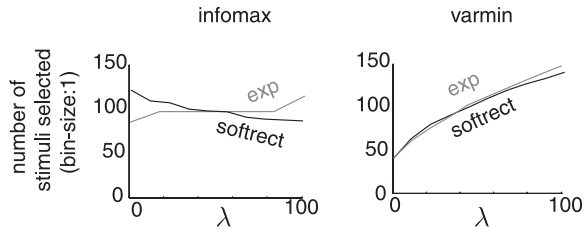

Figure 4: Effects of nonlinearity on stimulus selection in active learning. Illustration of stimuli selected during 1000 trials of active learning, using a linearly growing (simulated) tuning curve in 1D stimulus space. The stimuli selected under infomax learning depend very strongly on the choice of nonlinearity (stimuli near the peaks for exponential; stimuli near the troughs for soft-rectifying nonlinearity). Varmin learning is less sensitive to choice of nonlinearity compared to infomax learning.

This effect is compounded by the effect of g on the shape of the $p(r|x, f)$, the (transformed) Poisson likelihood (see Figure 3B). The inverse of the softrect nonlinearity leaves the shape of the likelihood relatively unchanged (see Figure 3B, left), meaning that the likelihood is broader at high firing rates, and so there is (relatively) less reduction in uncertainty when the stimulus elicits a high firing rate. (This arises due to the fact that narrowness of the likelihood determines the amount of reduction in the posterior: narrower likelihood implies greater reduction in posterior entropy). By contrast, transforming by log, the inverse of the exponential nonlinearity, reverses the ordering of likelihood width by firing rate (see Figure 3B right): the likelihood is broad when r is small and narrow when r is large.

These two factors mean that under the exponential nonlinearity, high-firing-rate regions provide the greatest gain in mutual information, while under the softrect nonlinearity, low-firing-rate regions provide the greatest gain in mutual information. Figure 4 illustrates this phenomenon with a simulated example but shows that it is somewhat less pronounced for varmin than infomax learning. That is, stimulus selection under infomax learning depends much more strongly on the choice of nonlinearity than under varmin learning. Figure 4 shows the distribution over the firing rate λ of selected stimuli, for a simulated example with true rate uniformly distributed over $[0, 100]$, under varmin and infomax learning for both choices of nonlinearity. For the softrect nonlinearity, infomax learning selects the stimuli eliciting low spike responses more often. On the contrary, with an exponential nonlinearity, infomax learning more often selects the stimulus eliciting high spike responses. Varmin learning selects the stimuli eliciting higher spike responses regardless of the nonlinearity. Thus, varmin learning is in general less sensitive to the choice of nonlinearity.

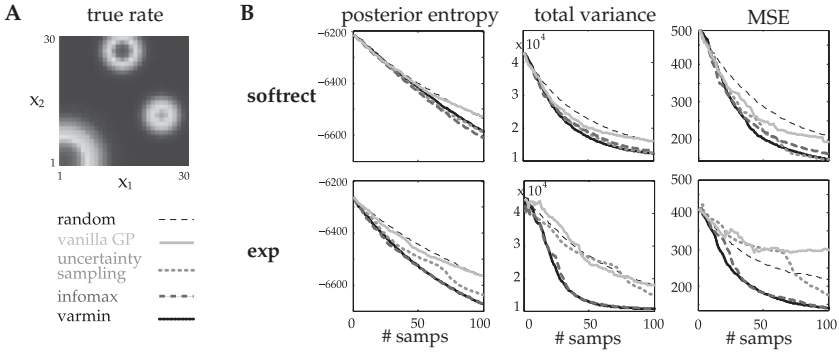

Figure 5: Estimation of 2D firing rate map (simulated data). (A) True firing rate map defined over a 2D input space. (B) Top: The posterior entropy, integrated posterior variance, and mean squared error (MSE) as a function of number of trials of active learning under the assumption of a softrect nonlinearity. Although infomax learning decreases the posterior entropy faster than varmin learning (left), varmin learning performs significantly better in terms of MSE (right). Bottom: Analogous plots under the assumption of exponential nonlinearity. Here, infomax and varmin methods perform more similarly.

6 Applications

6.1 Simulated Data. To compare the performance of infomax and varmin learning, we first generated Poisson responses from a 2*D* firing map (see Figure 5A) and estimated it using each learning method. We tested random sampling to contrast the relative gain of adaptive methods, as well as uncertainty sampling, which simply picks the stimulus for which the posterior variance of firing rate map is maximal. (For the softrect nonlinearity, we approximated the posterior variance of firing rate map by the delta method (Park et al., 2011).)

For comparison, we also tested using a model with gaussian (instead of Poisson) noise, resulting in "vanilla GP" active learning procedure that offers an analytic expression for utility. For this model, the utility does not depend on the observations, meaning that the entire sequence of stimuli can be planned out before the experiment. To assess the performance of this method, we used infomax learning to select stimuli and then fit the resulting responses with a GP-Poisson model.

Figure 5B shows the performance of each criterion in terms of posterior entropy, total variance of $\boldsymbol{\lambda}$, and mean squared error (average over 100 independent repetitions). For the softrect nonlinearity, varmin learning performs noticeably better than infomax learning in terms of MSE, although infomax learning effectively decreases the posterior entropy faster than varmin learning. For the exponential nonlinearity, both varmin and

infomax methods perform similarly. Uncertainty sampling in this case focuses on picking stimuli only around the high peak areas of the true map, which results in a slower decrease in MSE than random sampling. The active learning method under vanilla GP performs as poorly as random sampling, since stimulus selection is independent of the spike responses.

6.2 Real Data. We recorded from V1 neurons in awake rhesus monkeys as visual stimulus patterns drifted through the receptive field. The orientations and spatial frequencies of the stimuli were roughly matched to the neurons' preferred values. Each neuron was stimulated with either a soft-edged bar or a Gabor pattern. For the neuron stimulated with the soft-edged bar, stimuli were chosen from a 2D stimulus space in which L- and M-cone contrast varied and S-cone contrast was fixed. The neuron was stimulated with 620 pseudo-randomly selected soft-edged bar stimuli that effectively tile the LM plane. Each stimulus was presented three times, resulting in 1860 trials. For the neurons stimulated with the Gabor pattern, stimuli were chosen from a 3D cone contrast space with the purpose of finding a collection of stimuli that all evoked the same response (e.g., 29 spikes/sec) (Horwitz & Hass, 2012).

Using the recorded data set from each neuron, we conducted simulated closed-loop experiments using the active learning methods. In the simulated experiments, we fixed g to softrect due to its superior prediction performance on test data compared to exponential g: the average 10-fold cross-validation score (log likelihood of test data) across six cells for exponential g was -3.6051 and -3.3690 for softrect. As a performance measure, we computed the mean squared difference between the estimate using all the data in each original experiment and the estimate using the data selected by each active learning method in the simulated experiment.

In Figure 6, we show that varmin learning performed significantly better than infomax learning, especially at the beginning of the experiment, in terms of the mean squared difference. We show the firing rate map estimate of the neuron using all of the collected 2D data. We also show the estimated rate map after 30 trials. The data points selected by infomax learning were mostly clustered at the low-rate region, while the data points selected by varmin learning were well spread throughout the LM plane.

In Figure 7, we show the 2D (L, M) slice of the estimated 3D firing map of each neuron while fixing the S cone contrast to the mean of the values that the S cone had during the simulated experiments, using all trials with 3D inputs (approximately 700). For these neurons, unlike the previous neuron, two learning methods performed similarly at the beginning, since the actually presented data selected by the staircase (isoresponse) design do not cover the entire 3D stimulus space. Nevertheless, as we observe more data, varmin learning decreased MSE more effectively than infomax learning did (see Figure 7). We show the 2D slices of the estimates after 300 trials

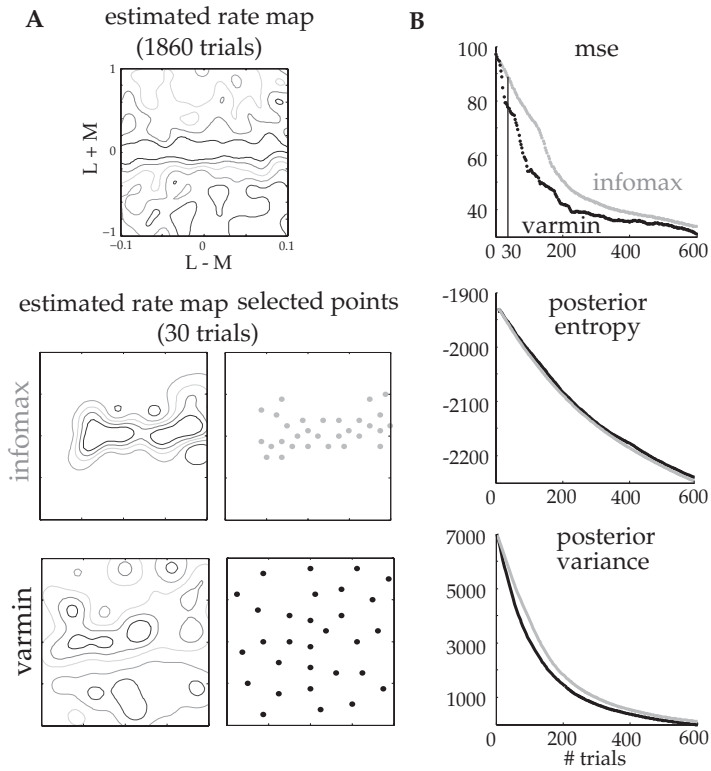

Figure 6: Active learning with neural data (2D cone contrast space). (A) Firing rate map estimate (top left) using all 1860 stimuli presented during an actual neurophysiology experiment in macaque V1. The firing rate map estimate after 30 trials under infomax learning (middle left) and varmin learning (bottom left). Plots at right show the first 30 stimuli selected with each method. (B) Mean squared error (top), posterior entropy (middle), and total posterior variance (bottom) as a function of the number of trials.

in Figure 7, where the estimates obtained by our method look closer to the estimates using all data than those obtained by infomax learning.

Finally, we conducted population analysis using the data collected from six V1 neurons. We made 10 differently shuffled data sets (as one does 10-fold cross-validation), where we used 10% of each data set as a validation set and picked the stimuli by active learning from the remaining 90% of the data set, and computed the average log likelihood of the test data for six different neurons. The gray dots in Figure 8 are the log likelihood when the number of trials is 40% of the entire data; black dots are when the number

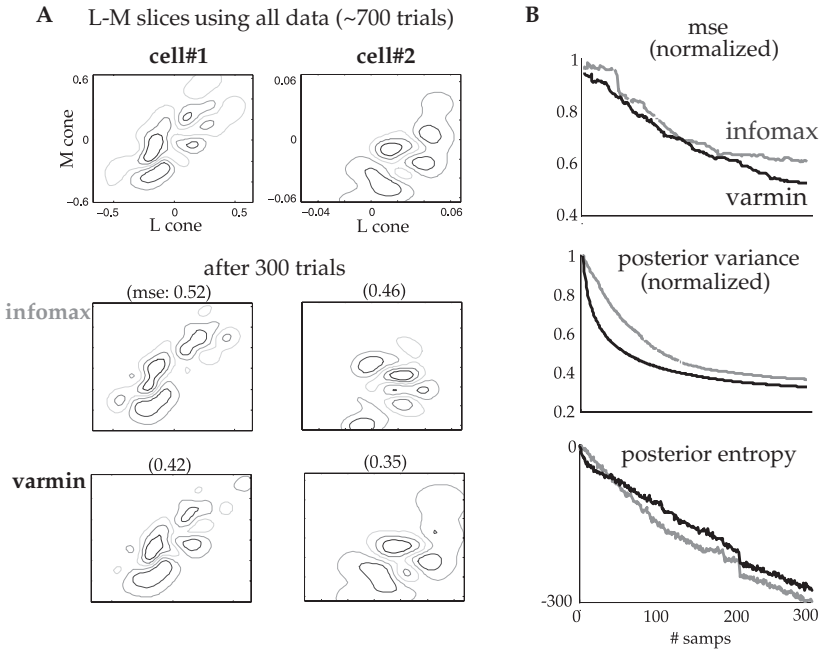

Figure 7: Active learning with neural data (3D cone contrast space) (A) 2D slices of the estimated 3D firing maps using all 700 trials (top) and after only 300 trials with infomax (middle) or varmin learning (bottom). (B) Mean-squared error, total variance, and posterior entropy as a function of the number of trials, averaged over results from four different neurons.

of trials is 80% of the entire data. On average, varmin learning achieved 1.4 times higher likelihood than infomax learning. We also conducted a paired-sample t-test and obtained the p-value 0.0130 using 80% of the entire data (black dots). Thus, we reject the null hypothesis that infomax and varmin learning methods perform similarly at the 0.05 significance level.

7 Discussion

We have developed an algorithm for active learning of neural firing rate maps in real-time, closed-loop neurophysiology experiments. We have shown that the widely used infomax learning behaves suboptimally in terms of mean squared error, in a fashion that depends on the choice of nonlinear transformation in the GP-Poisson model. Specifically, varmin learning selects stimuli that elicit both low and high spike rates, whereas infomax learning tends to select stimuli that elicit either all low (for softrect

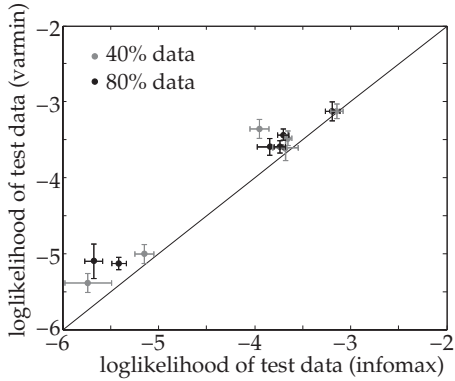

Figure 8: Log likeliood of test data across six cells. We used 10 differently shuffled data sets. Each active learning method picked the stimuli from 90% of each data set, and the remaining 10% was used as a validation set. On average, varmin learning achieved 1.4 times higher test likelihood than infomax learning.

nonlinearity) or high spike rates (for exponential nonlinearity). This provides justification for varmin learning, which selects the stimuli that minimize the expected total variance of the tuning curve estimate. When tested on simulated and real neural data, we found that varmin learning increased estimation accuracy faster than infomax learning did.

One potential criticism of varmin learning is that it requires several joint expectations, which we approximated somewhat crudely to achieve fast implementation. A natural future direction is to develop more efficient sampling-based algorithms in order to take into account the uncertainties in the firing rate map estimate when computing the expected reduction in posterior variance. This should improve the accuracy of the current varmin algorithm, assuming computational challenges can be overcome.

There are several promising applications of active learning methods for firing map estimation. First, we feel it will be useful to apply such methods to characterize response properties in higher-level visual areas (e.g., V2, V4, IT), where cells generally have highly nonlinear selectivity in relatively high-dimensional stimulus spaces. Second, we hope to extend active learning methods to multineuron recordings in order to select stimuli that are most useful for characterizing the response properties of a small population of neurons recorded in parallel. Third, we are currently working to develop more flexible response models for use in active learning protocols, including models with simple forms of nonlinear stimulus adaptation and non-Poisson noise in the neural response. Taken together, we believe these methods will greatly improve the speed and accuracy of neural characterizations in high-throughput neurophysiology experiments.

References

Adams, R. P., Murray, I., & MacKay, D. J. C. (2009). Tractable nonparametric Bayesian inference in Poisson processes with gaussian process intensities. In *Proceedings of the 26th Annual International Conference on Machine Learning*. New York: ACM.

Benda, J., Gollisch, T., Machens, C. K., & Herz, A. V. (2007). From response to stimulus: Adaptive sampling in sensory physiology. *Curr. Opin. Neurobiol.*, *17*(4), 430–436.

Bernardo, J. (1979). Expected information as expected utility. *Annals of Statistics*, *7*(3), 686–690.

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

Bölinger, D., & Gollisch, T. (2012). Closed-loop measurements of iso-response stimuli reveal dynamic nonlinear stimulus integration in the retina. *Neuron*, *73*(2), 333–346.

Carandini, M., & Ferster, D. (2000). Membrane potential and firing rate in cat primary visual cortex. *Journal of Neuroscience*, *20*(1), 470–484.

Chaloner, K., & Verdinelli, I. (1995). Bayesian experimental design: A review. *Statistical Science*, *10*(3), 273–304.

Cohn, D. A., Ghahramani, Z., & Jordan, M. I. (1996). Active learning with statistical models. *J. Artif. Intell. Res.*, *4*, 129–145.

Cunningham, J. P., Shenoy, K. V., & Sahani, M. (2008). Fast gaussian process methods for point process intensity estimation. In *Proceedings of the 25th International Conference on Machine Learning* (pp. 192–199). New York: ACM.

Georgopoulos, A., Schwartz, A., & Kettner, R. (1986). Neuronal population coding of movement direction. *Science*, *233*(4771), 1416–1419.

Hennig, P., & Schuler, C. J. (2012). Entropy search for information-efficient global optimization. *Journal of Machine Learning Research*, *13*, 1809–1837.

Horwitz, G. D., & Hass, C. A. (2012). Nonlinear analysis of macaque V1 color tuning reveals cardinal directions for cortical color processing. *Nat. Neurosci.*, *15*(6), 913–919.

Hubel, D. H., & Wiesel, T. N. (1959). Receptive fields of single neurones in the cat's striate cortex. *Journal of Physiology*, *148*(3), 574–591.

Kapoor, A., Horvitz, E., & Basu, S. (2007). Selective supervision: Guiding supervised learning with decision-theoretic active learning. In *International Joint Conference on Artificial Intelligence*. San Francisco: Morgan Kaufmann.

Krause, A., Singh, A., & Guestrin, C. (2008). Near-optimal sensor placements in gaussian processes: Theory, efficient algorithms and empirical studies. *J. Mach. Learn. Res.*, *9*, 235–284.

Kuck, H., de Freitas, N., & Doucet, A. (2006). SMC samplers for Bayesian optimal nonlinear design. In *Proc. IEEE Nonlinear Statistical Signal Processing Workshop* (pp. 99–102). Piscataway, NJ: IEEE.

Lewi, J., Butera, R., & Paninski, L. (2009). Sequential optimal design of neurophysiology experiments. *Neural Computation*, *21*(3), 619–687.

Lindley, D. (1956). On a measure of the information provided an experiment. *Ann. Math. Statist.*, *27*, 986–1005.

Luttrell, S. P. (1985). The use of transinformation in the design of data sampling scheme for inverse problems. *Inverse Prob.*, *1*, 199–218.

Machens, C. K., Gollisch, T., Kolesnikova, O., & Herz, A. V. M. (2005). Testing the efficiency of sensory coding with optimal stimulus ensembles. *Neuron*, *47*(3), 447–456.

MacKay, D. J. C. (1992). Information-based objective functions for active data selection. *Neural Computation*, *4*(4), 590–604.

Maldonado, P. E., Gdecke, I., Gray, C. M., & Bonhoeffer, T. (1997). Orientation selectivity in pinwheel centers in cat striate cortex. *Science*, *276*(5318), 1551–1555.

McCullagh, P., & Nelder, J. A. (1989). *Generalized linear models* (2nd ed.). London: Chapman & Hall.

Müller, P., & Parmigiani, G. (1996). *Bayesian analysis in statistics and econometrics: Essays in honor of Arnold Zellner*. New York: Wiley.

Ohki, K., Chung, S., Ch'ng, Y. H., Kara, P., & Reid, R. C. (2005). Functional imaging with cellular resolution reveals precise micro-architecture in visual cortex. *Nature*, *433*(7026), 597–603.

Paninski, L. (2004). Maximum likelihood estimation of cascade point-process neural encoding models. *Network: Comput. Neural Syst.*, *15*(4), 243–262.

Paninski, L. (2005). Asymptotic theory of information-theoretic experimental design. *Neural Computation*, *17*(7), 1480–1507.

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

Rad, K. R., & Paninski, L. (2010). Efficient, adaptive estimation of two-dimensional firing rate surfaces via gaussian process methods. *Network: Computation in Neural Systems*, *21*(3–4), 142–168.

Rasmussen, C., & Williams, C. (2006). *Gaussian processes for machine learning*. Cambridge, MA: MIT Press.

Ringach, D. L., Shapley, R. M., & Hawken, M. J. (2002). Orientation selectivity in macaque V1: Diversity and laminar dependence. *Journal of Neuroscience*, *22*(13), 5639–5651.

Roy, N., & Mccallum, A. (2001). Toward optimal active learning through sampling estimation of error Reduction. In *Proc. 18th International Conf. on Machine Learning* (pp. 441–448). San Francisco: Morgan Kaufmann.

Wedderburn, R.W.M. (1976). On the existence and uniqueness of the maximum likelihood estimates for certain generalized linear models. *Biometrika*, *63*(1), 27–32.

Received May 24, 2013; accepted February 10, 2014.