Proceedings of the 28th IEEE
EMBS Annual International Conference
New York City, USA, Aug 30-Sept 3, 2006

Efficient model-based design of neurophysiological experiments

Jeremy Lewi
School of Bioengineering
Georgia Institute of Technology
Email: jlewi@gatech.edu

Robert Butera, *Senior Member, IEEE-EMBS*
School of Electrical and Computer Engineering
Georgia Institute of Technology
Email: rbutera@ece.gatech.edu

Liam Paninski
Department of Statistics
Columbia University
Email: liam@stat.columbia.edu
http://www.stat.columbia.edu/∼liam

***Abstract*— We apply an adaptive approach to optimal experimental design in the context of estimating the unknown parameters of a model of a neuron's response. We present an algorithm to choose the optimal (most informative) stimulus on each trial; this algorithm can be implemented efficiently even for high-dimensional stimulus and parameter spaces (in particular, no high-dimensional numerical optimizations or integrations are required). Our simulation results show that model parameters can be estimated much more efficiently using this adaptive algorithm rather than random sampling. We also show that this adaptive approach leads to superior performance in the case that the model parameters are nonstationary, as would be expected in real experiments.**

I. Introduction

In neurophysiology experiments, minimizing the number of trials needed to characterize a neural system is essential for maintaining the viability of a preparation and ensuring robust results. Various approaches have been developed to optimize experiments in an online manner by choosing the "best" stimuli given prior knowledge of the system and the observed history of the cell's responses. The "best" stimulus can be defined a number of different ways depending on the experimental objectives. One reasonable choice is the stimulus which maximizes the firing rate of a neuron ([1], [2], [3]). This definition is well suited for experiments aimed at identifying a neuron's preferred stimulus. Alternatively, when investigating the coding properties of sensory cells it makes sense to define the optimal stimulus in terms of the mutual information between the stimulus and response [4], [5].

Here we take the approach that the optimal stimulus is the one which tells us the most about how a neural system responds to its inputs (see [6], [7] for further discussion and references). We are concerned with neural systems in which the probability $p(r|\vec{x})$ of the neural response r given an arbitrary stimulus, $\vec{x}$, can be described by a model of the conditional response, $p(r|\vec{x},\vec{\theta})$, which has a finite number of parameters given by the vector $\vec{\theta}$. Since we estimate these parameters from experimental trials, we want to choose our stimuli so that we constrain the model parameters as much as possible.

This work was supported by the Department of Energy Computational Science Graduate Fellowship Program of the Office of Science and National Nuclear Security Administration in the Department of Energy under contract DE-FG02-97ER25308 and by the NSF IGERT Program in Hybrid Neural Microsystems at Georgia Tech via grant number DGE-0333411.

In the simplest cases, these types of optimizations can be done using heuristics based on experience with a particular system. When mapping out the receptive fields of V1 cells, for example, it is more efficient to use stimuli which are orientated edges rather than randomly sampling the set of visual images. However, in general it is difficult to perform such optimizations manually. Therefore, there is a need to develop computational methods to perform this optimization efficiently and with minimal user intervention.

Two inconvenient facts make realizing this goal in a computationally efficient manner difficult: 1) model complexity — we typically need a large number of parameters to accurately model a system's response $p(r|\vec{x})$; and 2) stimulus complexity — we are typically interested in neural responses to stimuli $\vec{x}$ which are themselves very high-dimensional (e.g., spatiotemporal movies if we are dealing with visual neurons). In particular, it is computationally challenging to 1) update our *a posteriori* beliefs about the model parameters $p(\vec{\theta}|r,\vec{x})$ given new stimulus-response data, and 2) find the optimal stimulus quickly enough to be useful in an online experimental context.

In this work we present methods for solving these problems using generalized linear models (GLM) for the input-output relationship $p(r|\vec{x},\vec{\theta})$ and certain Gaussian approximations of the posterior distribution of the model parameters. Our emphasis is on finding solutions which scale well in high dimensions. We solve problem (1) by using Newton's method to update the Gaussian approximation to the posterior, and we solve problem (2) by a reduction to a highly tractable one-dimensional optimization problem. Simulation results show that our proposed algorithm produces a set of stimulus-response pairs which is more informative than the set produced by random sampling. The running time of our algorithm is also small enough that it could be run in real-time. Finally, we demonstrate how our methods may be extended to the important case that the underlying model parameters are changing with time (due, for example, to changes in the health, arousal, or attentive state of the preparation).

II. Methods

A. The Model

We model a neuron as a point process whose conditional intensity function is given as the output of a generalized linear model (GLM) [8], [9]. This class of models implicitly

assumes that neurons are only sensitive to a linear subspace of their input. Physiologists have identified a number of sensory neurons whose receptive fields can be efficiently modeled with these assumptions, for example the center surround receptive fields of retinal ganglion cells and the orientation selective receptive fields of simple cells in V1 [10]. Furthermore, the structure of this class of models allows the necessary computations to be implemented in an efficient manner. Under suitable constraints on the linear filtering and nonlinear function, the likelihood function is guaranteed to have a single global maximum [9]. The most important of these constraints is that the system can only respond to a one dimensional subspace of its input. Despite this constraint, the model is still very flexible and can easily incorporate effects due to spike history and the spiking of other neurons.

The model is summarized by the equation below for the intensity function λ:

$$\lambda(t) = E(r(t)) \quad (1)$$

$$= f\left(\sum_i \theta_i x_i(t)\right) \quad (2)$$

In the above summation the θ_i capture the dependency of the neuron's firing on the stimulus. The experimental objective is the estimation of the unknown filter coefficients, from knowledge of the stimulus, $\vec{x}(t)$, and the resulting firing rate $r(t)$. We chose the nonlinear stage of the GLM, the link function $f()$, to be the exponential function for simplicity and to ensure the likelihood function has a single global maximum.

B. Updating the Posterior

We approximate the posterior after t trials, $p(\vec{\theta}_t|\underline{\vec{x}}_t, \underline{r}_t)$, as a Gaussian with a mean equal to the location of the peak of the log posterior. We take the covariance matrix to be the negative inverse of the Hessian of the log posterior at its peak. We find the peak by using Newton's method to find the root of the log posterior. Each iteration of Newton's method requires computing the inverse of the Hessian of the log posterior. We can significantly speedup this step by using our Gaussian approximation of $p(\vec{\theta}_{t-1}|\underline{\vec{x}}_{t-1}, \underline{r}_{t-1})$ instead of computing the exact value on each Newton step. Using this approximation, computing the Hessian of $p(\vec{\theta}_t|\underline{\vec{x}}_t, \underline{r}_t)$ only takes $O(d^2)$ whereas computing the Hessian of the true posterior is $O(Nd^2)$, where N is the number of observations that have been made and d is the dimensionality of $\vec{\theta}$. Furthermore, if we use the Gaussian approximation of $p(\vec{\theta}_{t-1}|\underline{\vec{x}}_{t-1}, \underline{r}_{t-1})$, then computing the inverse Hessian of $p(\vec{\theta}_{t-1}|\underline{\vec{x}}_t, \underline{r}_t)$ is just a rank one update of the inverse Hessian of $p(\vec{\theta}_t|\underline{\vec{x}}_{t-1}, \underline{r}_{t-1})$. This rank one update can be implemented efficently using the Woodbury identity in $O(d^2)$ time whereas inversion of the Hessian of the true log posterior requires $O(d^3)$ operations. Our simulations showed that the loss in accuracy due to the approximation of $p(\vec{\theta}_{t-1}|\underline{\vec{x}}_{t-1}, \underline{r}_{t-1})$ as Gaussian was minimal but the speedup was significant. Therefore, we used this faster but less accurate implementation to update the posterior.

The Gaussian approximation of the posterior —which is the maximum entropy distribution with this mean and covariance— is justified because the posterior is the product of two log-concave terms, the likelihood function and the prior (which we assume to be Gaussian, for simplicity). Furthermore, the main theorem of [7] indicates that a Gaussian approximation of the posterior will be asymptotically accurate.

C. Choosing the Optimal Stimuls

Choosing the optimal stimulus for the next trial, time $t+1$, requires maximizing the mutual information between the parameter and the response conditioned on the stimulus. The conditional mutual information is just the difference in entropies of the posterior entropies after the $t+1$ and the t trials.

$$I(\vec{\theta}; r_{t+1}|\underline{\vec{x}}_{t+1}, \underline{r}_t) = H(\vec{\theta}|\underline{\vec{x}}_t, \underline{r}_t) - H(\vec{\theta}|\underline{\vec{x}}_{t+1}, \underline{r}_{t+1}) \quad (3)$$

To simplify notation we use the symbols, $\underline{\vec{x}}_t$ and $\underline{r}_t$, to denote the sequence of all stimuli and responses up to time t, that is the sequences $\{\vec{x}_1, .., \vec{x}_t\}$ and $\{r_1, ...r_t\}$ respectively. Having observed $\underline{r}_t$, $H(\vec{\theta}|\underline{\vec{x}}_t, \underline{r}_t)$ is constant. Therefore choosing $\vec{x}_{t+1}$ to maximize the mutual information requires minimizing $H(\vec{\theta}|\underline{\vec{x}}_{t+1}, \underline{r}_{t+1})$. Since we take the posterior to be Gaussian, its entropy depends only on its covariance matrix. Using the equations for the Newtonian update of the posterior, we can express the posterior covariance at time $t+1$ as:

$$C_{t+1} = (C_t^{-1} + J_{obs}(r_{t+1}, \vec{x}_{t+1}))^{-1} \quad (4)$$

In the above equation C_t is the covariance of our Gaussian posterior at time t and J_{obs} is the observed Fisher information:

$$J_{obs}(r_{t+1}, \vec{x}_{t+1}) = -\frac{\partial^2 \log p(r_{t+1}|\vec{x}_{t+1}, \vec{\theta})}{\partial^2 \vec{\theta}} \quad (5)$$

Using matrix perturbation theory,

$$C_{t+1} = C_t - C_t J_{obs} C_t + o(C_t J_{obs}) \quad (6)$$

Plugging the above into the expression for the entropy of a Gaussian and making some additional approximations gives,

$$E_r \log\det[C_t - C_t J_{obs} C_t] \sim E_r \log\det[I - J_{obs} C_t] \quad (7)$$

$$= -E_r \mathrm{tr}[J_{obs} C_t] + o(J_{obs} C_t) \quad (8)$$

Therefore neglecting the higher order terms, we need to maximize $E_r \mathrm{tr}[J_{obs} C_t]$, which is an average over the observations of the expected Fisher information. In our case, that is with $f(\vec{\theta}^t \vec{x}) = \exp(\vec{\theta}^t \vec{x})$, we can use the moment-generating function of the multivariate Gaussian to obtain the result:

$$E_r J_{obs} = \vec{x}\vec{x}^t \exp\left(\vec{x}^t \mu_t + \frac{1}{2}\vec{x}^t C_t \vec{x}\right) \quad (9)$$

where μ_t is the mean of the Gaussian posterior at time t. Therefore to maximize $E_r \mathrm{tr}[J_{obs} C_t]$, we need to maximize:

$$F(\vec{x}) = \exp(\vec{x}^t \vec{\mu}_t) \exp(\frac{1}{2}\vec{x}^t C_t \vec{x}) \vec{x}^t C_t \vec{x} \quad (10)$$

where $\vec{\mu}_t$ and C_t are the mean and covariance of our current Gaussian estimate of the posterior of $\vec{\theta}$.

For this model the optimal stimulus is undefined, since increasing the stimulus power $||\vec{x}||_2$ increases the informativeness of any putatively "optimal" stimulus. To obtain a well-defined optimization problem, we optimize the stimulus under the usual power constraint $||\vec{x}||_2 \leq e$ for some constant $e < \infty$.

We maximize Eqn. 10 by using an eigendecomposition combined with the Lagrange method for constrained optimization to reduce our k-dimensional optimization problem to a 1-dimensional problem. Expressing Eqn. 10 in terms of the eigenvectors of C_t yields:

$$F(\vec{x}) = \exp(\sum_i u_i y_i + \frac{1}{2}\sum_i c_i y_i^2)\sum_i c_i y_i^2 \quad (11)$$

$$= g(\sum_i u_i y_i) h(\sum_i c_i y_i^2) \quad (12)$$

where u_i and y_i represent the projection of the mean and the stimulus respectively onto the i^{th} eigenvector and c_i is the corresponding eigenvalue. To simplify notation we also introduce the functions $g()$ and $h()$ which are monotonically strictly increasing functions implicitly defined by Eqn. 11. We maximize $F(\vec{x})$ by breaking the problem into an inner and outer problem by fixing the value of $\sum_i u_i y_i$ and maximizing $h()$ subject to that constraint. A single line search over all possible values of $\sum_i u_i y_i$ finds the global maximum. This approach is summarized by the equation:

$$\max_{\vec{y}:||\vec{y}||_2=e} F(\vec{y}) = max_{b:||\vec{y}||_2=e}\Big[g(b)\cdot \Big[max_{\vec{y}:||\vec{y}||_2=e,\vec{y}^t\vec{u}=b} \quad h(\sum_{i=0}^{k} c_i y_i^2)\Big]\Big] \quad (13)$$

Since h() is increasing, to maximize h() we only need to maximize:

$$max_{\vec{y}:||\vec{y}||_2=e,\vec{y}^t\vec{u}=b} \sum_{i=0}^{k} c_i y_i^2 \quad (14)$$

This last expression is a quadratic function with quadratic and linear constraints and we can solve it using the Lagrange method for constrained optimization. The result is a system of equations for y_i as a function of the Lagrange multiplier λ_1.

$$y_i = \frac{e}{||\vec{y}||_2}\frac{u_i}{2(c_i - \lambda_1)} \quad (15)$$

To find the global optimum we vary λ_1, which is equivalent to performing a search over b, and compute y_i. For each λ_1 we compute $F(\vec{y}(\lambda_1))$ and choose the stimulus, $\vec{y}(\lambda_1)$ which maximizes $F()$. It is possible to show (details omitted) that the maximum of $F()$ must occur with $\lambda_1 \geq c_0$ where c_0 is the largest eigenvalue. This restriction on the optimal λ_1 provides a significant speedup of the optimization step and simplifies the implementation of the search.

To summarize, updating the posterior and finding the optimal stimulus requires three steps: 1) a Newton update, 2) an eigendecomposition and 3) a 1-dimensional search over $\lambda_1 \geq c_0$. The complexity of these computations is

$$O(d^2 + d^3 + rd) \quad (16)$$

where d is the dimensionality of $\vec{\theta}$, and r is the number of steps in our search over λ_1.

A slight variation of the optimization strategy is to choose the stimulus which minimizes the mean squared error of our estimated $\vec{\theta}$. This leads to a similar objective function but we have omitted the details due to space.

D. Changing $\vec{\theta}$

We can account for changes in an animal preparation during an experiment by assuming these changes cause $\vec{\theta}$ to change randomly and slowly with each trial. These assumptions provide a reasonable model of changes in $\vec{\theta}$ over time due to a change in the animal's alertness or a change in its health. We model this effect by letting $\vec{\theta}$ experience mean reverting diffusion [11].

$$\vec{\theta}_{t+1} = m \cdot \vec{\theta}_t + w_t \quad (17)$$

In the above equation m is a positive scalar less than or equal to 1 and w_t is a normally distributed random variable with mean zero and covariance matrix Q. We take m and Q to be known although they could potentially be learned from the data as well. To incorporate the effects of diffusion, we approximate $p(\vec{\theta}_{t+1}|\underline{\vec{x}}_t, \underline{r}_t)$ as Gaussian with mean $m\cdot\vec{\mu}_t$ and covariance $m^2 \cdot C_t + Q$ where $\vec{\mu}_t$ and C_t are the mean and covariance of our Gaussian approximation of $p(\vec{\theta}_t|\underline{\vec{x}}_t, \underline{r}_t)$. To update the posterior and choose the optimal stimulus, we use the same procedure as in the case of constant $\vec{\theta}$.

III. Results

Our first set of simulation results show the efficiency of choosing information maximizing stimuli in the case where the input space is a 100 dimensions, Fig. 1. We compare our optimization strategy to results produced by random inputs which are uniformly sampled from a sphere with radius equal to the magnitude constraint imposed on the optimal stimuli. We compare the methods by looking at the posterior entropy of $\vec{\theta}$. Our results show that using optimal stimuli leads to more accurate estimates of $\vec{\theta}$ using fewer trials. Our results are supported by the conclusion of [7] that the information maximization strategy is asymptotically never worse than using random stimuli and is in general more efficient.

The running time for each step of the algorithm as a function of the dimensionality of $\vec{\theta}$ is plotted in Fig. 2. The fitted polynomials in the figure illustrate that the running times of the various steps grow as: $O(d^3)$ for diagonalization, $O(d^2)$ for the posterior update, and $O(d)$ for the 1d linesearch. When $\vec{\theta}$ was less than 200 dimensions, the total running time of our algorithm was less than a quarter second which is well within the range of tolerable latencies for many experiments.

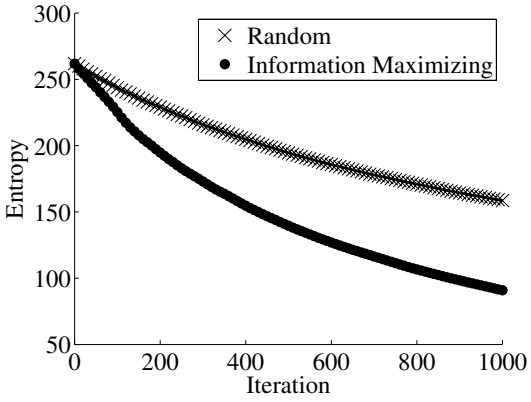

Fig. 1. Simulation results comparing the entropy of the posterior for $\vec{\theta}$ when using information maximizing stimuli compared to using random stimuli uniformly sampled from a sphere. $\vec{\theta}$ was a constant vector of length 100.

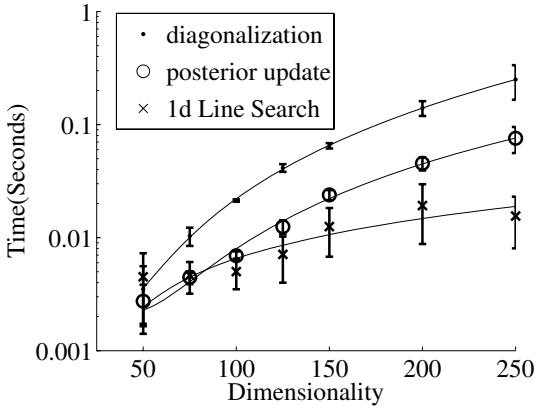

Fig. 2. A plot of the timing of the three steps performed on each iteration as a function of the dimensionality of $\vec{\theta}$. The three required steps are: updating the posterior, diagonalizing the covariance matrix of the posterior, and conducting the one dimensional line search to find the optimal stimulus. The timing for each step was fitted to a polynomial of degree 3 for the diagonalization, degree 2 for the posterior update, and degree 1 for the line search. The results shown are an average over many iterations. The errorbars indicate ± 2 std.

Our second set of simulation results deals with the case where $\vec{\theta}$ is not constant but undergoes mean reverting diffusion, Fig. 3. The results show that despite a changing $\vec{\theta}$ we are still able to choose stimuli which on average provide more information than randomly drawn stimuli.

IV. Conclusion

While designing efficient experiments based on information theory presents several computational challenges, reasonable approximations can lead to feasible implementations for experimental settings. Despite these assumptions it is still possible to see significant improvements over random sampling. In particular, we have presented one solution for dealing with problems due to large dimensional input spaces and to a changing response function over time.

An important direction for future work is reducing the running time. The bottle neck is the $O(d^3)$ eigendecomposition.

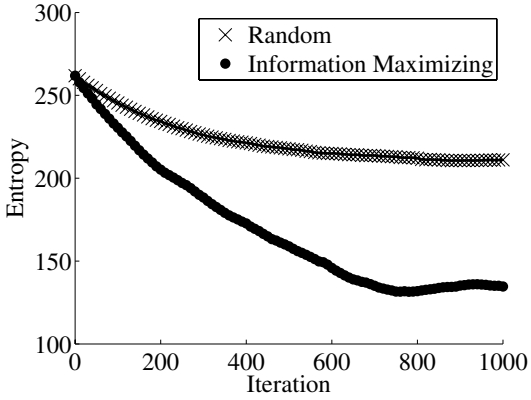

Fig. 3. A comparison of the posterior entropies using information maximizing stimuli vs. randomly chosen stimuli when $\vec{\theta}$ experiences mean reverting diffusion with $m = .999$. The covariance matrix, Q, was a non-white matrix with eigenvalues $\{.0011, .0012, ..., .0110\}$. $\vec{\theta}$ had 100 dimensions.

If we use our Gaussian approximation of $p(\vec{\theta}_{t-1}|\underline{\vec{x}}_{t-1}, \underline{r}_{t-1})$, then computing the updated eigendecomposition is just a rank one modification of the eigendecomposition from the previous time step. This is a well known linear algebra problem for which efficent algorithms exist [12]. Furthermore, in order to implement our algorithm efficently we made several approximations to simplify the computations. The efficency of the information maximization algorithm compared to random sampling can be computed if we assume we can perfectly maximize the mutual information [7]. Therefore, we can evaluate the merit of our approximations by comparing the efficency of our implementation to the theoretical limit.

References

[1] I. Nelken, Y. Prut, E. Vaadia, and M. Abeles, "In search of the best stimulus: an optimization procedure for finding efficient stimuli in the cat auditory cortex," *Hearing Research*, vol. 72, pp. 237–253, 1994.

[2] P. Foldiak, "Stimulus optimisation in primary visual cortex," *Neurocomputing*, vol. 38–40, pp. 1217–1222, 2001.

[3] K. Zhang, M. Anderson, and E. Young, "Saddle-point property of nonlinear sensory response," in *Proceedings*. Computational and Systems Neuroscience (COSYNE) Meeting, 2004.

[4] C. Machens, "Adaptive sampling by information maximization," *Physical Review Letters*, vol. 88, pp. 228 104–228 107, 2002.

[5] C. Machens, T. Gollisch, O. Kolesnikova, and A. Herz, "Testing the efficiency of sensory coding with optimal stimulus ensembles," *NEURON*, vol. 47, no. 3, pp. 447–456, 2005.

[6] A. Watson and D. Pelli, "QUEST: a Bayesian adaptive psychophysical method," *Perception and Psychophysics*, vol. 33, pp. 113–120, 1983.

[7] L. Paninski, "Asymptotic theory of information-theoretic experimental design," *Neural Computation*, vol. 17, pp. 1480–1507, 2005.

[8] P. McCullagh and J. Nelder, *Generalized linear models*. London: Chapman and Hall, 1989.

[9] L. Paninski, "Maximum likelihood estimation of cascade point-process neural encoding models," *Network: Computation in Neural Systems*, vol. 15, pp. 243–262, 2004.

[10] D. H. Hubel and T. Wiesel, "Receptive fields, binocular interaction and functional architecture in cats visual cortex," *Journal of Physiology-London*, vol. 160, no. 1, pp. 106–&, 1962.

[11] G. E. Uhlenbeck and L. S. Ornstein, "On the theory of the brownian motion," *PHYSICAL REVIEW*, vol. 36, no. 5, pp. 0823–0841, 1930.

[12] M. Gu and S. C. Eisenstat, "A stable and efficient algorithm for the rank-one modification of the symmetrical eigenproblem," *SIAM Journal on Matrix Analysis and Applications*, vol. 15, no. 4, pp. 1266–1276, Oct. 1994.