Efficient active learning with generalized linear models

Jeremy Lewi
Bioengineering
Georgia Institute of Technology
`jlewi@gatech.edu`

Robert Butera
Electrical and Computer Engineering
Georgia Institute of Technology
`rbutera@ece.gatech.edu`

Liam Paninski
Department of Statistics
Columbia University
`liam@stat.columbia.edu`

Abstract

Active learning can significantly reduce the amount of training data required to fit parametric statistical models for supervised learning tasks. Here we present an efficient algorithm for choosing the optimal (most informative) query when the output labels are related to the inputs by a generalized linear model (GLM). The algorithm is based on a Laplace approximation of the posterior distribution of the GLM's parameters. The algorithm requires only low-rank matrix manipulations and a single two-dimensional search to choose the optimal query and has complexity $O(n^2)$ (with n the dimension of the feature space), making active learning with GLMs feasible even for high-dimensional feature spaces. In certain cases the two-dimensional search may be reduced to a one-dimensional search, further improving the algorithm's efficiency. Simulation results show that the model parameters can be estimated much more efficiently using the active learning technique than by using randomly chosen queries. We compute the asymptotic posterior covariance semi-analytically and demonstrate that the algorithm empirically achieves this asymptotic convergence rate, which is generally better than the convergence rate in the random-query setting. Finally, we generalize the approach to efficiently handle both output history effects (for applications to time-series models of autoregressive type) and slow, non-systematic drifts in the model parameters.

Introduction

Active learning refers to the problem of selecting the optimal training set for a learning agent, given our prior knowledge and any previously observed examples. Active learning is particularly useful in settings 1) for which it is expensive to obtain labeled training examples (e.g. user ratings systems or geophysical applications), or 2) for which we have limited time to collect data (e.g. biological applications).

Optimal inputs are typically selected by optimizing an objective function which quantifies the expected utility of measuring the response to a particular input. One important objective function is the expected mutual information between the response and the parameters of the model being fitted to the data (Fedorov, 1972; Mackay, 1992; Chaloner and Verdinelli, 1995). Previous work has shown that constructing a training set using this criterion is in general asymptotically more efficient and never less efficient than using a randomly constructed training set (Paninski, 2005).

Active learning, however, is limited by the computational challenges of the required optimization (Cohn et al., 1996; Gilad-Bachrach et al., 2005). In particular, there are two major computational hurdles. First, the utility of each possible input depends on what has already been learned from labeled examples. Therefore, we need to be able to rapidly update the fitted model as we add new points to the training set. Second, evaluating and optimizing the mutual information over the (high-dimensional) space of unlabeled examples requires a high-dimensional integration followed by a search over the possible inputs. In general, the complexity of both the integration and optimization scales exponentially with the dimensionality. As emphasized above, many applications require that we perform these tasks in near-real time.

We present methods for solving these problems using generalized linear models (GLM) for the input-output relationship and certain Gaussian approximations of the posterior distribution of the model parameters. Our emphasis is on developing techniques that scale well in high dimensions. We solve the first problem by using efficient rank-one update methods to update the Gaussian approximation to the posterior, and the second problem by a reduction to a highly tractable two-dimensional optimization problem. Our results show that the resulting algorithm produces a training set which can be up to an order of magnitude more informative than the set produced by random sampling. Moreover, the algorithm is efficient enough for near real-time implementation.

Methods

The point of regression analysis is to learn the likelihood $p(r|\vec{x})$ which maps inputs $\vec{x}$ into outputs or class labels r. Our focus here is on parametric representations of the likelihood $p(r|\vec{x}, \vec{\theta})$ given a finite-dimensional parameter $\vec{\theta}$. The parameters are learned from a labeled training set $\{(\vec{x}_1, r_1)...(\vec{x}_t, r_t)\}$ (for convenience we will often denote the sequence of labeled training points $\underline{\vec{x}}_t$ and $\underline{r}_t$ respectively).

The GLM is a tractable and flexible parametric model which consists of a linear stage followed by a scalar nonlinearity[1]. Only the projection of the input, $\vec{x}$, on the parameter vector, $\vec{\theta}$, influences the output. We denote this linear projection as $\varepsilon = \vec{\theta}^T\vec{x}$. The linear projection is then pushed through a nonlinearity known as the "link" function, which gives the mean $\mu_r = f(\varepsilon)$: the output r follows a distribution in the exponential family with mean μ_r. It is convenient to work with the canonical parameterization, with $g()$ denoting the mapping from ε to the canonical parameter $\eta : \eta = g(\varepsilon)$. The GLM is therefore summarized by the conditional probability:

$$p(r|\vec{x}, \vec{\theta}) = p(r|\eta = g(\vec{\theta}^T\vec{x})) = h(r)\exp(r\eta - N(\eta)) \tag{1}$$

where $h(r)$ determines the distribution in the exponential family and $N(\eta) = \log \int h(r)\exp(r\eta)$ normalizes the distribution.

The GLM is quite flexible in that it can be used for either classification or regression by choosing an appropriate distribution in the exponential family. We may furthermore easily enforce various constraints on the output (e.g., r might be discrete or continuous, and/or positive, bounded, binary, etc.). Here we will assume that the exponential family that best describes the data, along with the nonlinearity g is chosen *a priori*; thus we will not consider the problem of optimizing these parameters, and our goal here will simply be to learn the coefficients of the linear stage, $\vec{\theta}$, using as few examples as possible.

[1]For simplicity, we restrict our attention here to the case of scalar responses r; the case of vector observations will be treated elsewhere.

Representing and updating the posterior. As emphasized above, our first key task is to efficiently update the posterior distribution of $\vec{\theta}$ after t trials, $p(\vec{\theta}_t|\underline{\vec{x}}_t, \underline{r}_t)$, as new input-output pairs are observed. The key assumption of this paper is that this posterior may be approximated with a single Gaussian with suitably chosen mean and covariance parameters $\vec{\mu}_t$ and C_t. We use the standard Laplace approximation, in which $\vec{\mu}_t$ is taken to be the maximum a posteriori (MAP) estimate for $\vec{\theta}$, $\mu_t \equiv \arg\max_{\vec{\theta}} p(\vec{\theta}|\underline{\vec{x}}_t, \underline{r}_t)$, and the covariance matrix C_t is the negative inverse Hessian of the log-posterior evaluated at $\vec{\mu}_t$. This approximation is justified asymptotically by the main result of (Paninski, 2005), a central limit theorem for active learning which indicates that a Gaussian approximation of the posterior will be asymptotically accurate.

How can we justify this (strong) Gaussian assumption in a non-asymptotic sense? One key property of the Gaussian is that it is a smooth, log-concave function, with no non-global local maximum that would complicate the search for the MAP solution μ_t. Thus, we will restrict our attention to the subset of GLMs with smooth, log-concave likelihood terms[2]. If we further assume that the prior distribution on $\vec{\theta}$ is log-concave (e.g., Gaussian, for simplicity), then the posterior distribution — the normalized product of the prior and likelihood term, both of which are log-concave — will also be log-concave, and therefore we can expect that the Gaussian assumption will be reasonable. (In some special cases, e.g. linear regression with Gaussian noise, this Gaussian approximation will be exact.)

We therefore develop a set of constraints to ensure the likelihood is concave. We start by considering conditions sufficient to ensure concavity with respect to ε. The second derivative of the log likelihood is

$$\frac{d^2 \log p(r|\eta)}{d\varepsilon^2} = [r - E(r|\eta)]\frac{d^2 g(\varepsilon)}{d\varepsilon^2} - Var(r|\eta)(\frac{dg(\varepsilon)}{d\varepsilon})^2;$$

thus, for distributions for which r is unbounded, the only way to ensure the log likelihood remains negative for all r is to restrict $\frac{d^2g(\varepsilon)}{d\varepsilon^2} = 0$. This implies that g is linear in ε, i.e., the GLM is in the canonical parameterization. (As is well-known, the log-likelihood in this case is always concave because the variance is non-negative.) On the other hand, if r is bounded above

[2]Note that the smoothness assumption on the likelihood function rules out the case of noiseless (deterministic) observations. In the noiseless setting, computing the posterior involves computing the intersection of half-planes corresponding to the feasible set (Freund et al., 1997; Gilad-Bachrach et al., 2005), and the Gaussian approximation is invalid.

or below, then weaker conditions are sufficient for log concavity. For example, in the Poisson case, $r \geq 0$, and here it is known that if the link function is convex and log-concave, then the likelihood is guaranteed to be log-concave (Paninski, 2004).

The discussion above is in terms of concavity with respect to ε; to translate these results into statements about concavity in $\vec{\theta}$, we need only note that

$$\frac{d^2 \log p(r|\eta)}{d\vec{\theta}^2} = \vec{x}\frac{d^2 \log p(r|\eta)}{d\varepsilon^2}\vec{x}^T. \qquad (2)$$

This rank-one matrix is negative semidefinite if the log likelihood is concave with respect to ε; thus, concavity in ε implies concavity in $\vec{\theta}$.

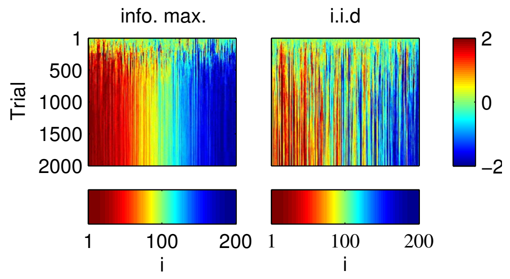
Now that we have restricted our attention to those GLMs for which we may guarantee that the log-posterior is concave, and therefore that the Gaussian approximation is reasonable, we may discuss how to efficiently compute the mean and covariance parameters $\vec{\mu}_t$ and C_t. In general, computing these terms directly requires $O(tn^2 + n^3)$ time (where $n = \dim(\vec{\theta})$); the complexity increases with t because to compute the posterior we must form a product of t likelihood terms, and the n^3 term is due to the inverse of the Hessian matrix. Unfortunately, this is too slow when t or n becomes large.

Therefore we take a recursive approximate approach, as in the derivation of the extended Kalman filter, and use our Gaussian approximation of $p(\vec{\theta}_{t-1}|\underline{\vec{x}}_{t-1}, \underline{r}_{t-1})$ to update the posterior at time t. To see how this simplifies matters, we write out the derivatives of the posterior:

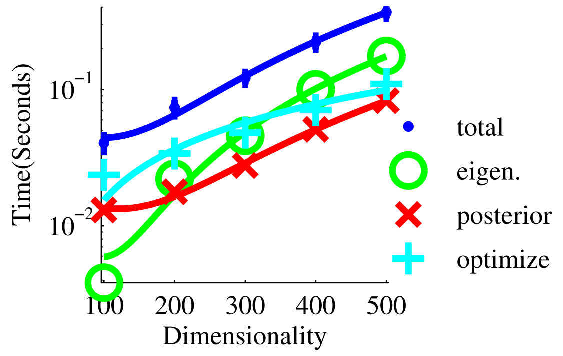
$$\frac{d \log p(\vec{\theta}|\underline{r}_t, \underline{\vec{x}}_t)}{d\vec{\theta}} = -(\vec{\theta} - \vec{\mu}_{t-1})^T C_{t-1}^{-1} + \left[r_t - \frac{dN}{d\eta}\right]\frac{dg}{d\varepsilon}\vec{x}_t^T \qquad (3)$$

$$\begin{aligned}\frac{d^2 \log p(\vec{\theta}|\underline{r}_t, \underline{\vec{x}}_t)}{d\theta_i d\theta_j} &= -C_{t-1}^{-1} + \Bigg(\left[r_t - \frac{dN}{d\eta}\right]\frac{d^2 g(\varepsilon)}{d\varepsilon^2} \\ &- \frac{d^2 N(\eta)}{d\eta^2}\left(\frac{dg(\varepsilon)}{d\varepsilon}\right)^2\Bigg)\vec{x}_t\vec{x}_t^T \qquad (4)\end{aligned}$$

Now, to update $\vec{\mu}_t$ we only need to find the peak of a one-dimensional function (as opposed to a d-dimensional function); this follows by noting that the likelihood only varies along a single direction, $\vec{x}_t$, as a function of $\vec{\theta}$. At the peak of the posterior, $\vec{\mu}_t$, the first term in the gradient must be parallel to $\vec{x}_t$ because the gradient is zero. Since C_{t-1} is non-singular (by assumption), $\vec{\mu}_t - \vec{\mu}_{t-1}$ must be parallel to $C_{t-1}\vec{x}_t$. Therefore we just need to solve a one dimensional line-search to determine how much the mean changes in the direction $C_{t-1}\vec{x}_t$; this requires only $O(n^2)$ time. Moreover, from the second derivative term above it is clear



(a)



(b)

Figure 1: a) A comparison of the linear filter estimated using information maximizing input vs. i.i.d. inputs drawn uniformly from a sphere of radius e. The true filter, a sine-wave, is plotted beneath the plots of the mean. The GLM had a Poisson distribution with exponential link function. The info. max. approach requires an order of magnitude fewer training points to achieve the same accuracy as using i.i.d inputs. b) A plot of the timing of the three steps as a function of the dimensionality of $\vec{\theta}$. The timing for each step was well-fit by a polynomial of degree 2 for the eigendecomposition, posterior update and total time, and degree 1 for the line search. The times are an average over many iterations. The error-bars for the total time indicate ± 1 std. The plot confirms that the empirical running time of our algorithm is $O(n^2)$. The algorithm is also quite fast; capable of handling $n = 200$ in less than a tenth of a second. Running time was measured on a workstation with a dual core Intel Xeon 2.8GHz processor. The eigendecomposition is roughly 3x slower than in our previous results (Lewi et al., 2006) because we require 3 rank one updates as opposed to 1.

that computing C_t requires just a rank-one matrix update of C_{t-1}, which can be evaluated in $O(n^2)$ time via the Woodbury matrix lemma. Thus this Gaussian approximation of $p(\vec{\theta}_{t-1}|\underline{\vec{x}}_{t-1}, \underline{r}_{t-1})$ provides a large gain in efficiency (from $O(tn^2 + n^3)$ to $O(n^2)$); our simulations (discussed in more depth below) demonstrate that, despite this improved efficiency, the loss in accuracy due to this approximation was minimal.

Deriving an approximate objective function for the next unlabeled example. To choose the optimal unlabeled example for trial $t+1$, we want to maximize the conditional mutual information

$$I(\vec{\theta}; r_{t+1}|\underline{\vec{x}}_{t+1}, \underline{r}_t) = H(\vec{\theta}|\underline{\vec{x}}_t, \underline{r}_t) - H(\vec{\theta}|\underline{\vec{x}}_{t+1}, \underline{r}_{t+1}) \quad (5)$$

with respect to the input $\vec{x}_{t+1}$. The first term does not depend on $\vec{x}_{t+1}$, so maximizing the information requires minimizing the conditional entropy:

$$\begin{aligned} &H(\vec{\theta}|\underline{\vec{x}}_{t+1}, \underline{r}_{t+1}) = \\ &E_{r_{t+1}|\vec{x}_{t+1}} \int -p(\vec{\theta}|\underline{r}_{t+1}, \underline{\vec{x}}_{t+1}) \log p(\vec{\theta}|\underline{r}_{t+1}, \underline{\vec{x}}_{t+1}) d\vec{\theta} \\ &= (1/2) E_{r_{t+1}|\vec{x}_{t+1}} \log |C_{t+1}| + const. \end{aligned} \quad (6)$$

The equality above is due to our Gaussian approximation of $p(\vec{\theta}|\underline{\vec{x}}_{t+1}, \underline{r}_{t+1})$. Therefore, we need to minimize $E_{r_{t+1}|\vec{x}_{t+1}} \log |C_{t+1}|$ as a function of $\vec{x}_{t+1}$. Our derivation proceeds in two steps. 1) We derive an expression for $|C_{t+1}|$ which avoids any costly inverses and determinants. 2) We show how the expectation can be simplified to a 2-d integration in the general (non-canonical) case and a 1-d integration in the case of the canonical response.

From our Gaussian approximation of the posterior, C_{t+1} is just the negative inverse Hessian of the log-posterior:

$$C_{t+1} = \left(C_t^{-1} - \left. \frac{\partial^2 \log p(r_{t+1}|\vec{x}_{t+1}, \vec{\theta})}{\partial\theta_i \partial\theta_j} \right|_{\vec{\theta}=\mu_{t+1}} \right)^{-1} \quad (7)$$

Since the Hessian of the log likelihood is rank one, we can use the Woodbury lemma to evaluate the inverse and the appropriate identities to evaluate the determinant. To simplify notation we abbreviate

$$D(r_{t+1}, \varepsilon) = -\partial^2 \log p(r|\eta)/\partial\varepsilon^2,$$

with the derivative evaluated at $\varepsilon = \varepsilon_{t+1}$; thus

$$|C_{t+1}| = |C_t| \cdot (1 + D(r_{t+1}, \varepsilon)\vec{x}_{t+1}^T C_t \vec{x}_{t+1})^{-1}. \quad (8)$$

We therefore just need to minimize

$$\begin{aligned} &E_{r_{t+1}|\vec{x}_{t+1}} \log |C_{t+1}| \\ &= -E_\varepsilon \, E_{r_{t+1}|\varepsilon} \log\left(1 + D(r_{t+1}, \varepsilon)\sigma_\varepsilon^2\right) + const. \quad (9) \\ &\varepsilon = \vec{\theta}^T \vec{x}_{t+1} \quad \sigma_\varepsilon^2 = \vec{x}_{t+1}^T C_t \vec{x}_{t+1}. \end{aligned}$$

Thus our Gaussian approximation reduces the computation of our objective function from an intractable high-dimensional integral over $\vec{\theta}$ and r_{t+1} to a much simpler two-dimensional integral over ε and r_{t+1}. Since $p(\vec{\theta})$ is Gaussian, $p(\varepsilon)$ is a 1-dimensional Gaussian variable with mean $\mu_\varepsilon = \vec{\mu}_t^T \vec{x}_{t+1}$ and variance $\sigma_\varepsilon^2 = \vec{x}_{t+1}^T C_t \vec{x}_{t+1}$; that is, this expectation only depends on $\vec{\mu}_t$ and C_t through the two scalar variables $(\mu_\varepsilon, \sigma_\varepsilon^2)$. Thus, instead of computing this expectation

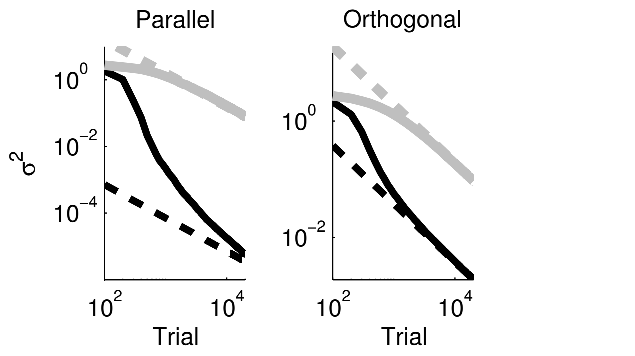


Figure 2: Comparison of the actual variance of the posterior in our simulations to the asymptotic variance predicted based on the central limit theorem. Despite our approximations, the variance of $\vec{\theta}$ estimated using information maximizing inputs converges to the predicted value. Black lines are for information maximizing inputs; grey lines are for i.i.d inputs. Dashed lines show predicted values; solid lines indicate actual values; The first axis shows the variance in the direction of the posterior mean. The second axis is the geometric mean of the variances in directions orthogonal to the mean; asymptotically the variances in these directions are equal. Fig. 1(a).

online, for each new input-output pair $\vec{x}_t, r_t$, we may precompute this expectation before training begins on a suitable 2-d region of $(\mu_\varepsilon, \sigma_\varepsilon^2)$, greatly reducing the necessary on-the-fly computation time.

We may further simplify matters if our GLM uses the canonical link function. In this case, $\frac{d^2 g(\varepsilon)}{d\varepsilon^2} = 0$, and therefore the posterior covariance is independent of the observations r_t. Thus the expectation simplifies and we just need to compute

$$\begin{aligned} &E_{r_{t+1}|\vec{x}_{t+1}} \log |C_{t+1}| \\ &= \log |C_t| - E_\varepsilon \log(1 + Var(r_{t+1}|\varepsilon)\sigma_\varepsilon^2). \end{aligned} \quad (10)$$

Finally, if $Var(r_{t+1}|\varepsilon)\sigma_\varepsilon^2$ is sufficiently small (as is the case asymptotically provided $Var(r_{t+1}|\varepsilon)$ is bounded, since $\sigma_\varepsilon^2 \to 0$ as $t \to \infty$ (Paninski, 2005)), we may use the standard linear approximation $\log(1+x) = x + o(x)$ to simplify our objective function even further. For certain models (e.g. the canonical Poisson (Lewi et al., 2007), or the standard linear Gaussian), this linear approximation allows us to compute the expectation in Eq. 10 analytically, further reducing the computational complexity.

Similar versions of our objective function Eq. 10 have appeared in other settings, e.g. the "D-optimal" strategy from the experimental design literature (Fedorov, 1972; Mackay, 1992). We may interpret the two key quantities σ_ε^2 and $Var(r_{t+1}|\varepsilon)$ as usual: the first measures our uncertainty about $\vec{\theta}$ in the direction of the input $\vec{x}$, while the second measures the system's variability given ε. Optimal design according to Eq. 10 requires a kind of balanced optimization of these two terms as a function of $\vec{x}_{t+1}$.

Computing the optimal unlabeled example. We now consider the problem of picking the optimal unlabeled training example. In many applications (e.g., document classification), the inputs must be selected from some finite set. A reasonable approach in this case is simply to compute $(\mu_\varepsilon, \sigma_\varepsilon^2)$ for the possible inputs and then perform a search over this set. More generally, we need to perform a search over a continuous set of possible inputs $\vec{x}$. For the GLM in this setting the optimal stimulus is undefined, since in general increasing the allowed stimulus power $||\vec{x}_{t+1}||_2$ increases the maximal informativeness of the stimulus. Therefore to get a well-defined optimization problem we impose a power constraint on the input:

$$\min_{\vec{x}_{t+1}} E_{r_{t+1}|\vec{x}_{t+1}} \log |C_{t+1}| \quad ||\vec{x}_{t+1}||_2 \leq e. \tag{11}$$

In addition to the power constraint, we consider the possibility that some of the components of the input are fixed. For example, we typically need to include a bias term, $\varepsilon = \vec{\theta}^T \vec{x} + m$, where the offset term m is an unknown constant which we wish to estimate in addition to $\vec{\theta}$. This situation is typically handled by simply concatenating m onto the end of $\vec{\theta}$ and the fixed term, 1, onto the end of $\vec{x}$. Another important example arises in time-series applications, when we may wish to include autoregressive terms in our GLM; here the fixed terms in $\vec{x}$ correspond to outputs which have already been observed (or in the more general graphical model setting, to outputs in the graph neighborhood of the vertex of interest). To include these fixed terms in our analysis we break $\vec{x}$ into two terms, $\vec{x} = (\vec{x}_k^T \ \vec{x}_a^T)^T$, where the subscripts k and a denote the variable and fixed components respectively.

The optimization now proceeds in two steps: A) We perform a 2-d search over $(\mu_\varepsilon, \sigma_\varepsilon^2)$ to find the values $(\mu_\varepsilon, \sigma_\varepsilon^2)^*$ which maximize $E_{r_{t+1}|\vec{x}_{t+1}} \log |C_{t+1}|$, and then B) we choose $\vec{x}_{t+1}$ so that $(\mu_\varepsilon, \sigma_\varepsilon^2)^* = (\vec{\mu}_t^T \vec{x}_{t+1}, \vec{x}_{t+1}^T C_t \vec{x}_{t+1})$. Step (B) turns out to be easy once step (A) is complete.

To compute the feasible region in $(\mu_\varepsilon, \sigma_\varepsilon^2)$ space we need only compute the maximum and minimum values of σ_ε^2 as a function of μ_ε. This is sufficient because for fixed μ_ε, σ_ε^2 is continuous on the interval between its maximum and minimum values. The optimization problems we need to solve are

$$\max \sigma_\varepsilon^2 = \max_{\vec{x}_k} \vec{x}_k^T C_k \vec{x}_k + 2\vec{x}_a^T C_{ak} \vec{x}_k + \vec{x}_a^T C_a \vec{x}_a \tag{12}$$

$$\min \sigma_\varepsilon^2 = \min_{\vec{x}_k} \vec{x}_k^T C_k \vec{x}_k + 2\vec{x}_a^T C_{ak} \vec{x}_k + \vec{x}_a^T C_a \vec{x}_a \tag{13}$$

$$\text{s.t} \quad \frac{\vec{\mu}_{k,t}^T \vec{x}_k}{||\vec{\mu}_{k,t}||} = \alpha \quad ||\vec{x}_k|| \leq e^2 - e_a^2 - \alpha^2, \tag{14}$$

where $e_a = ||\vec{x}_a||_2$, and μ_ε and α are related by $\mu_\varepsilon = \alpha ||\vec{\mu}_{k,t}|| + \vec{\mu}_{a,t}^T \vec{x}_a$, and we have written C_t in block matrix form:

$$C_t = \left[\begin{array}{c|c} C_k & C_{ka} \\ \hline C_{ak} & C_a \end{array} \right].$$

This allows us to perform the optimization only over the components of $\vec{x}_{t+1}$ which we can control. To trace out the borders of the feasible region, we solve these optimization problems for $\alpha \in [-e_k, e_k]$, with $e_k^2 = e^2 - e_a^2$.

It is possible to solve these two optimization problems directly, by introducing two Lagrange multipliers. However, this direct approach leads to a somewhat difficult nonlinear search (details omitted), and a slightly less direct approach turns out to be more efficient. We begin by reformulating the optimizations slightly, to work within the linear subspace of $\vec{x}_k$ corresponding to the projection constraint in Eq. 14. For a suitable matrix A, vector $\vec{b}$, and scalar d (see Appendix), optimizing Eqs. 12 and 13 is equivalent to finding the maximum and minimum of:

$$\sigma_\varepsilon^2 = \vec{x}_k^T A \vec{x}_k + \vec{b}^T \vec{x}_k + d \quad \text{s.t} \quad ||\vec{x}_k||^2 \leq e_k^2 - \alpha^2. \tag{15}$$

Here A is defined as the rank 2 perturbation of C_k which removes the projection along the mean $\vec{\mu}_k$. As a result, we can drop the linear constraint in Eq. 14 because the value of this expression is independent of the projection of $\vec{x}_{t+1}$ on $\vec{\mu}_k$. The power constraint, however, now depends on α.

We therefore need to optimize a quadratic expression with a single quadratic constraint, Eq. 15. This is a well studied optimization problem known as the Trust Region Subproblem (TRS) (Fortin, 2000). For the TRS, it can be proved that the Karush-Kuhn-Tucker (K.K.T) conditions are both necessary and sufficient (Fortin, 2000). Therefore, we can find the minima and maxima by solving the K.K.T. conditions.

Solving the K.K.T conditions is simplified if we diagonalize A. We therefore define the following terms using the eigenbasis of A:

$$A = S \Lambda S^T \qquad \vec{y} = S^T \vec{x}_{t+1} \qquad \vec{w} = S^T \vec{b} \tag{16}$$

Using these terms we can rewrite the optimization problem in Eq. 15 as

$$\max_{\vec{y}} \sigma_\varepsilon^2 = \max_{\vec{y}} \sum_i c_i (y_i + \frac{w_i}{2c_i})^2 - \frac{w_i^2}{4c_i} \tag{17}$$

$$\min_{\vec{y}} \sigma_\varepsilon^2 = \min_{\vec{y}} \sum_i c_i (y_i + \frac{w_i}{2c_i})^2 - \frac{w_i^2}{4c_i} \tag{18}$$

$$\text{s.t} \sum_i y_i^2 \leq e_k^2 - \alpha^2. \tag{19}$$

To obtain the first order K.K.T conditions we introduce the Lagrange multiplier λ which enforces the power constraint:

$$\sigma_\varepsilon^2 = \sum_i c_i(y_i + \frac{w_i}{2c_i})^2 - \frac{w_i^2}{4c_i} + d - \lambda y_i^2 \qquad (20)$$

where c_i denotes the i^{th} eigenvalue of A. We then set the derivatives with respect to y_i to zero. This leads to a system of n equations which are satisfied by any local minima or maximum:

$$2y_i(c_i - \lambda) = -w_i \quad \forall i. \qquad (21)$$

The second order K.K.T. conditions restrict the range of λ where the max and minimum can occur. For $\sigma^2_{\varepsilon,\max}$, the second order conditions are

$$c_i - \lambda \leq 0 \quad \forall i; \qquad (22)$$

Therefore, $\sigma^2_{\varepsilon,\max}$ must occur with $\lambda \geq c_{\max}$, where $c_{\max}$ is the maximum eigenvalue. The corresponding conditions for the minimum are

$$c_i - \lambda \geq 0 \quad \forall i, \qquad (23)$$

i.e., $\sigma^2_{\varepsilon,\min}$ must occur for $\lambda \leq c_{\min}$. Thus, for any value of α, we can find a value of λ to solve problem (15); this requires a nonlinear search over λ. However, recall that our goal is actually to trace out the values of $\sigma^2_{\varepsilon,\max}$ and $\sigma^2_{\varepsilon,\min}$ for *all* possible values of α, not just a single value. Therefore, instead of solving the above problems for each value of α (which requires a nonlinear search over λ), we may simply find the solutions $\vec{y}(\lambda)$ of the first order K.K.T. on the relevant ranges $\lambda \leq c_{\min}$ and $\lambda \geq c_{\max}$; the key point is that we may compute $\vec{y}(\lambda)$ directly, without any additional nonlinear searching. Now, since the K.K.T. conditions are necessary and sufficient, this procedure is guaranteed to find every local maximum and minimum of σ_ε^2 for every value of α. Thus, by tracing out the curves $\sigma^2_{\varepsilon,\max}$ and $\sigma^2_{\varepsilon,\min}$ as a function of λ, we can easily determine the feasible region in $(\mu_\varepsilon, \sigma_\varepsilon^2)$ space, completing step (A) in our program.

Having computed the feasible $(\mu_\varepsilon, \sigma_\varepsilon^2)$ region, we can perform a 2-d search to find the optimal pair $(\mu_\varepsilon, \sigma_\varepsilon^2)^*$. Now, to finish, we need only find an input $\vec{x}_{t+1}$ such that $\vec{\mu}_t^T \vec{x}_{t+1} = \mu_\varepsilon^*$ and $\vec{x}_{t+1}^T C_t \vec{x}_{t+1} = \sigma_\varepsilon^{2*}$; this requires only that we solve a one-dimensional quadratic equation. Let $\vec{x}_{\min}$ and $\vec{x}_{\max}$ denote the solutions to our optimization problem corresponding to the optimal value $\mu_\varepsilon = {\mu_\varepsilon}^*$; of course we have already computed these vectors in the previous step. Now it is sufficient to find a linear combination of these two vectors which yields σ_ε^{2*}; that is, we search $s \in [0,1]$ to find a value which solves:

$$\vec{x}(s)^T A \vec{x}(s) + \vec{b}^T \vec{x}(s) + d = \sigma_\varepsilon^{2*} \qquad (24)$$

$$\vec{x}(s) = (1-s)\vec{x}_{\min} + s\vec{x}_{\max} \qquad (25)$$

where A , $\vec{b}$, d , are as in Eq. 15. Note that this is just a quadratic equation in s. All such $\vec{x}_{t+1}(s)$ necessarily satisfy the magnitude constraint and the projection constraint, and therefore finding the solution to this quadratic equation completes our optimization task.

The bottleneck of our algorithm is the eigendecomposition of A required in the optimization step; an eigendecomposition generally requires $O(n^3)$ time. However, we may speed this up by recalling that A is a symmetric rank-two modification of $C_{k,t}$, and $C_{k,t}$ is a symmetric rank-one modification of $C_{k,t-1}$, since

$$C_{k,t} = C_{k,t-1} - \frac{D(r_t, \varepsilon)}{1 + D(r_t, \varepsilon)\sigma_\varepsilon^2} \vec{z}\vec{z}^T,$$

with $\vec{z} = C_{k,t-1}\vec{x}_{t-1,k} + C_{a,t-1}\vec{x}_{t-1,a}$. It is known (Gu and Eisenstat, 1994) that a symmetric rank-one perturbation of an eigendecomposition may be computed much more efficiently than computing the eigendecomposition from scratch; here we need only apply Gu and Eisenstat's rank-one symmetric eigenupdate code three times (once to obtain the eigendecomposition of $C_{k,t}$ from that of $C_{k,t-1}$, and twice to obtain that of A from that of $C_{k,t}$). The Gu-Eisenstat algorithm achieves an average running time here of just $O(n^2)$ because of deflation which reduces the cost of the matrix multiplications associated with finding the eigenvectors for repeated eigenvalues. Asymptotically $n-1$ of our eigenvalues are nearly equal, see discussion of asymptotic efficiency below, allowing us to speedup the eigendecomposition through deflation (Fig. 1(b)).

Finally, in certain cases, we can reduce the two-dimensional search over $(\mu_\varepsilon, \sigma_\varepsilon^2)$ to a much simpler one-dimensional search. For example if our objective function is monotonically increasing in σ_ε^2 then we only need to consider $\sigma^2_{\varepsilon,max}(\mu_\varepsilon)$ for each possible value of μ_ε; thus a one-dimensional search over $\sigma^2_{\varepsilon,max}(\mu_\varepsilon)$ is sufficient for finding the optimal input. Simple sufficient conditions may be derived fairly easily from Eqs. (9) and (10) (details omitted); for example, in the asymptotic regime, where σ_ε^2 is small, then we may use a linear expansion for $\log(1+x)$ to conclude that if $Var(r_{t+1}|\varepsilon)$ is convex in ε in the canonical case — or more generally $D(r_{t+1}, \varepsilon)$ is convex in ε for all r_{t+1} — then the objective function is increasing in σ_ε^2, and a one-dimensional search suffices. This convexity condition is satisfied, for example, in the canonical Poisson case (Lewi et al., 2007).

To summarize, updating the posterior and choosing the optimal input requires three steps: 1) a rank-one matrix update and one-dimensional search to compute $\vec{\mu}_t$ and C_t; 2) three rank-one modifications of our eigendecomposition of C_k ; and 3) a two-dimensional search over $(\mu_\varepsilon, \sigma_\varepsilon^2)$ (which in certain cases may be restricted to a simpler one-dimensional search). The time-complexity is dominated by the low-rank eigendecomposition updates, which in the worst case can require $O(n^3)$ but on average takes only $O(n^2)$ operations. Thus a simple Gaussian approximation of the posterior distribution reduces our apparently exponentially-difficult information-maximization problem to a quite tractable $O(n^2)$ problem.

Non-stationary $\vec{\theta}$. It is worth noting that the proposed algorithm can easily be extended to handle parameters $\vec{\theta}$ which change slowly and non-systematically with time, e.g., in applications to ratings data (where the users' preferences may change slowly over time) or biological applications (where the health of the preparation might drift with time). To handle this non-stationarity an extended Kalman-based formulation is natural. We model slow changes in $\vec{\theta}$ by letting $\vec{\theta}$ experience diffusion :

$$\vec{\theta}_{t+1} = \vec{\theta}_t + w_t \tag{26}$$

Here w_t is a normally distributed random variable with mean zero and covariance matrix Q. (We take Q to be known although it could potentially be learned from the data as well.) This implies that $p(\vec{\theta}_{t+1}|\underline{\vec{x}}_t, \underline{r}_t)$ is Gaussian with mean $\vec{\mu}_t$ and covariance $C_t + Q$, where $\vec{\mu}_t$ and C_t are the mean and covariance of the posterior, recursively computed after t trials. To update the posterior and choose the optimal stimulus, we use the same procedure described above[3].

Asymptotic efficiency We can evaluate the efficiency and accuracy of our implementation by comparing the covariance of our estimated $\vec{\theta}$ to the asymptotic value predicted by the central limit theorem in (Paninski, 2005). Asymptotically, the covariance matrix converges to an average of expected Fisher information matrices:

$$(1/t)C_t^{-1} \rightarrow E_{\vec{x}}(J_{exp}(\vec{\theta}_{true}, \vec{x})), \ t \rightarrow \infty. \tag{27}$$

J_{exp} is the expected Fisher information (evaluated at the true underlying parameter value, $\vec{\theta}_{true}$) and the expectation is over the sampling distribution of the inputs $p(\vec{x})$. For i.i.d. data in our simulations, we take $p_{iid}(\vec{x})$ to be a uniform distribution on the sphere $||\vec{x}||_2 = e$, for simplicity, while in the information-maximizing case, the expectation is taken over the distribution $p_{opt}(\vec{x})$ which minimizes the determinant of the asymptotic covariance matrix, evaluated at $\vec{\theta}_{true}$. That is,

$$p_{opt}(\vec{x}) = \arg\max_{p(\vec{x})} |E_{\vec{x}}(J_{exp}(\vec{\theta}_{true}, \vec{x}))|, \tag{28}$$

where we are taking the maximum of the log-concave determinant function over the convex set of all distributions $p(\vec{x})$ with support on $||\vec{x}||_2 \leq e$. As discussed in (Paninski, 2005), the info. max. approach will in general therefore be asymptotically superior to the i.i.d. sampling approach (in the sense that the posterior covariance of $\vec{\theta}$ will have a smaller log-determinant, i.e., that the posterior Gaussian entropy will be smaller) whenever the optimizer $p_{opt}(\vec{x})$ is non-constant as a function of $\vec{\theta}_{true}$, i.e., there is not a single distribution which simultaneously maximizes the efficiency for all (a priori unknown) values of $\vec{\theta}_{true}$.

In order to evaluate the asymptotic Fisher information, we need to solve the above optimization problem; this problem turns out to be surprisingly tractable. Without loss of generality, we choose a coordinate system in which $\vec{x}$ is aligned with $\vec{\theta}$: $\theta_i = 0 \ \forall \ i \neq 1$. Using this parameterization,

$$\begin{aligned} E_{\vec{x}} J_{exp}(\vec{\theta}_{true}, \vec{x}) = \int_{x_1} & E_{r|x_1} D(r, x_1\theta_1) p(x_1) \\ & \times \int_{x_2 \dots x_n} \vec{x}\vec{x}^T p(x_2, \dots, x_n | x_1) \end{aligned} \tag{29}$$

The second integral above is just the correlation matrix of $\vec{x}$ taken over the conditional distribution given x_1; a simple symmetry argument, along with the log-concavity of the determinant, establishes that the optimal distribution $p_{opt}(x_2 \dots x_n | x_1)$ given the power constraint is just a uniform distribution on the sphere of radius $e^2 - x_1^2$, which is the largest sphere given x_1 and the power constraint.

Thus to compute $p_{opt}(\vec{x})$ we need only solve the easier problem of computing the optimal distributions on x_1:

$$p_{opt}(x_1) = \arg\max_{p(x_1)} \left[\log\phi + (n-1)\log(\frac{e^2\omega - \phi}{n-1}) \right],$$

with the abbreviations $\phi = E_{x_1}(E_{r|x_1} D(r, x_1\theta_1) x_1{}^2)$ and $\omega = E_{x_1}(E_{r|x_1} D(r, x_1\theta_1))$. The objective function on the right-hand side here depends only on the two linear projections ϕ and ω of $p(x_1)$; from this fact it is not difficult to establish that an optimal $p_{opt}(x_1)$ may be chosen which is supported on just two values of x_1. The infinite-dimensional optimization problem therefore reduces to a much more tractable three-dimensional problem. Having computed the optimal

[3]The one difference is that the covariance matrix of $p(\vec{\theta}_{t+1}|\underline{\vec{x}}_{t+1}, \underline{r}_{t+1})$ is in general no longer just a rank-one modification of the covariance matrix of $p(\vec{\theta}_t|\underline{\vec{x}}_t, \underline{r}_t)$; thus, we cannot use the rank-one update to compute the eigendecomposition. However, it is usually reasonable to take the diffusion matrix Q to be of full-rank; in this case, we may apply a single whitening change of basis ($Q \rightarrow I$), implying that the eigenvectors of $C_t + Q$ expressed in the new basis are unchanged, and the eigenvalues simply $c_i + 1$, with c_i denoting the i^{th} eigenvalue of C_t. Thus in this case, our methods may be applied with only slight modifications.

$p_{opt}(x_1)$, it is straightforward to evaluate the asymptotic covariance matrix and compare it to the asymptotic covariance obtained in the i.i.d. setting (Fig. 2). In each case, the symmetry of $p(x_2 \dots x_n | x_1)$ implies that this covariance has a simple structure: one eigenvector is parallel to $\vec{\theta}_{true}$, and the eigenvalues corresponding to all of the other eigenvectors (which are orthogonal to $\vec{\theta}_{true}$) are equal.

Results

We tested our algorithm by simulating its application to system identification in experimental neuroscience. By modeling a neuron as a GLM with canonical Poisson distribution (McCullagh and Nelder, 1989; Paninski, 2004), we can use our algorithm to design optimal experiments for identifying a neuron's response function. We generated synthetic data by simulating a neuron as a GLM with exponential link function (see (Lewi et al., 2007) for details). A comparison of the posterior means, $\vec{\mu}_t$, estimated using information maximizing inputs vs. i.i.d. inputs, Fig. 1(a), shows that the information maximizing strategy converges more rapidly to the true $\vec{\theta}$. These results are supported by the conclusion of (Paninski, 2005) that the information maximization strategy is asymptotically never worse than using random stimuli and is generically strictly more efficient. Fig. 1(b) confirms that the running time of our algorithm is $O(n^2)$. Finally, the results of analyzing the asymptotic efficiency of our algorithm are shown in Fig. 2. Despite the Gaussian approximation made here, asymptotically our algorithm performs at the limits predicted by the central limit theorem of (Paninski, 2005).

Appendix

The quantities in Eq. 15 are as follows (details of the derivation will be provided elsewhere):

$$A = C_k - \frac{1}{2}\vec{v}\vec{v}^T + \frac{1}{2}\vec{w}\vec{w} \tag{30}$$

$$\vec{v} = \frac{-\vec{\mu}_k^T C_k \vec{\mu}_k + 2}{2}\vec{\mu}_k + C_k\vec{\mu}_k \tag{31}$$

$$\vec{w} = \frac{-\vec{\mu}_k^T C_k \vec{\mu}_k - 2}{2}\vec{\mu}_k + C_k\vec{\mu}_k \tag{32}$$

$$\begin{aligned}\vec{b} &= 2\alpha C_k\vec{\mu}_k - 2\alpha(\vec{\mu}_k^T C_k \vec{\mu}_k)\vec{\mu}_k \\ &\quad + 2(C_{ka}\vec{x}_a) - 2(\vec{\mu}_k^T C_{ka}\vec{x}_a)\vec{\mu}_k \end{aligned} \tag{33}$$

$$d = \alpha^2(\vec{\mu}_k^T C_k \vec{\mu}_k) + 2\alpha\vec{\mu}_k^T C_{ka}\vec{x}_a + \vec{x}_a^T C_a \vec{x}_a. \tag{34}$$

Acknowledgments

JL is supported by the Computational Science Graduate Fellowship Program of the Office of Science and National Nuclear Security Administration in the DOE under contract DE-FG02-97ER25308 and by the NSF IGERT Program in Hybrid Neural Microsystems at Georgia Tech via grant number DGE-0333411. LP is supported by grant EY018003 from the NEI and by a Gatsby Foundation Pilot Grant. We thank P. Latham for helpful conversations.

References

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

Cohn DA, Ghahramani Z, Jordan MI (1996) Active learning with statistical models. *Journal of Artificial Intelligence Research* 4:129–145.

Fedorov VV (1972) *Theory of Optimal Experiments* Academic Press.

Fortin C (2000) A Survey of the Trust Region Subproblem within a Semidefinite Framework Ph.D. diss., University of Waterloo.

Freund Y, Seung HS, Shamir E, Tishby N (1997) Selective sampling using the query by committee algorithm. *Machine Learning* 28:133–168.

Gilad-Bachrach R, Navot A, Tishby N (2005) Query by committe made real In *Advances in Neural Information Processing Systems 18*, pp. 443–450. MIT Press.

Gu M, Eisenstat SC (1994) A stable and efficient algorithm for the rank-one modification of the symmetrical eigenproblem. *SIAM Journal on Matrix Analysis and Applications* 15:1266–1276.

Lewi J, Butera R, Paninski L (2007) Real-time adaptive information-theoretic optimization of neurophysiology experiments In *Advances in Neural Information Processing Systems 19.* MIT Press.

Mackay DJC (1992) Information-based objective functions for active data selection. *Neural Computation* 4:590–604.

McCullagh P, Nelder J (1989) *Generalized linear models* Chapman and Hall, London.

Paninski L (2004) Maximum likelihood estimation of cascade point-process neural encoding models. *Network: Computation in Neural Systems* 15:243–262.

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