Particle Filtering Algorithms for Neural Decoding and Adaptive Estimation of Receptive Field Plasticity

Uri T. Eden, Loren M. Frank, Riccardo Barbieri, Emery N. Brown
ueden@fas.harvard.edu

We present a set of particle filters for estimating neural receptive field dynamics from observed spike trains. We also extend a previously presented maximum a posterior (MAP) point-process filter, by using its output as the proposal density for a particle filter. As the number of particles is increased, the posterior probability distribution of encoding parameters is increasingly well expressed by the filter. We used these algorithms to construct adaptive models for real and simulated data in the CA1 region of the hippocampus of the rat, and for decoding a wind stimulus from simulated data of the cricket cercal system. We found that these algorithms could reliably track the dynamics in each of these systems. We also validate the Gaussian approximations and provide error measures on results from previously presented estimation algorithms.

Theory:

Neural receptive fields are plastic. That is, with experience the neuron alters how it responds to a relevant stimulus. In earlier work [1,2,3], we have presented a set of recursive adaptive point process estimation algorithms for identifying and tracking receptive field dynamics from spike train observations. This work deals with tracking the evolution of the probability density associated with an evolving set of receptive field parameters. We are interested in $p(\theta_k \mid N_{0:k})$, where θ_k is a parameter vector describing the receptive field at time k and $\{N_{0:k} : 0 \le t \le k\}$ is the spike train up to and including time k. We can find a recursive expression for this distribution by applying Bayes' rule and using the Chapman-Kolmogorov equation to describe the probabilistic evolution of the receptive field.

$$p(\theta_k \mid \mathbf{N}_{1:k}) = \frac{p(\mathbf{N}_k \mid \mathbf{N}_{1:k-1}, \theta_k)}{p(\mathbf{N}_k \mid \mathbf{N}_{1:k-1})} \int p(\theta_k \mid \theta_{k-1})\, p(\theta_{k-1} \mid \mathbf{N}_{1:k-1})\, d\theta_{k-1} \tag{1.1}$$

This equation is the basis for all of our recursive estimation algorithms. The integral in this expression can be quite difficult to solve without some implicit assumptions on the previous posterior density, $p(\theta_{k-1} \mid \mathbf{N}_{1:k-1})$. Our past solution has been to approximate this density with a Gaussian function, since a second order Taylor expansion of the log posterior is quadratic with no first order terms. This allows for an exact solution to the integral, which leads to the MAP adaptive estimation algorithm described in [2]. However, there is the possibility that higher order terms that are ignored by this approximation would have a significant effect on the probability density of the evolving parameter.

Sequential Monte Carlo, or particle, filters allow arbitrarily exact computation of distribution statistics as a set of underlying probability densities evolve through time [4]. The Monte Carlo approach to solving the integral in Eq. 1.1 is to average a large number of independent identically distributed samples, also called particles, drawn from this posterior distribution. The strong law of large numbers assures us that any statistic estimated by averaging this sample will approach its actual value almost surely as the number of particles is increased. It is often difficult, however, to sample effectively from an arbitrary distribution. A standard strategy for simulating this density is to use importance sampling, wherein one samples from a chosen proposal density, and weights the samples by a value proportional to the ratio between the actual posterior density and the proposal density, called the importance weight. With a judicious choice for the proposal density, this method can applied so that the importance weight at each point in time is dynamic and can be computed recursively from its previous value. To describe the evolution of the parameter vector with time, we simply generate a set of samples from the proposal distribution at each time step, recursively compute the importance weight for each particle, and compute the empirical statistics of the particle set. This technique is called sequential importance sampling (SIS), and each of the particle filters that we constructed is based on this idea. On its own, a sequential importance sampling algorithm tends to cause the variance of the importance weights to increase over time.

In order to counter this effect, we introduce a resampling step, where particles with large importance weights are broken up into numerous particles with unit weight.

Using the transition prior, $p(\theta_k \mid \theta_{k-1})$, in Eq. 1.1 as the proposal density leads to one of the most computationally simple particle estimation algorithms. This proposal density is easy to sample from, but does not contain any information about the current spiking activity. It may therefore not be a good approximation to the target density, especially immediately following a spike, and might require a large number of particles to converge. We constructed another filter using a Gaussian proposal density whose moments were determined by the MAP adaptive estimator. To accomplish this, we evolve each particle from the previous posterior set according to the MAP estimation algorithm and the instantaneous observation of the spike train. Then each particle is resampled according to the Gaussian posterior approximation for that particle. We then proceed to assign importance weights and resample as described above. Though more computationally intensive, this proposal density has the advantage that it includes information about the current spiking activity and more closely approximates the optimal importance distribution [5]. We present two simulated and one real data examples.

Dynamics of Cricket Cercal System: The filiform hairs on a cricket's cerci transduce wind currents into neural signals that are used to reconstruct a directional map of local air flow [7]. We modeled the response of a cercal cell to hair displacement due to a naturalistic wind stimulus. The output of this simulated cell was fed back into both of the particle filters described above. For the purpose of comparison, we also fed the spike trains back into the MAP filter [2] and compared the posterior probability distributions of the estimated wind stimulus in each case. The comparison was made by plotting the quantiles of the MAP predicted Gaussian distribution against the empirical distribution. We found that the distribution of the particle clouds only minimally deviated from the Gaussian approximation. As expected, the mean squared error between the particle filter wind estimate and it actual value, tended to decrease to a nonzero minimal value as the number of particles was increased. However, even at 100,000 particles, the particle filter only marginally outperformed the MAP estimate in this problem.

Dynamics of Hippocampal Place Fields in the Rat: We apply both MAP and the SIS filters to the problem of tracking the evolution of place fields in rat hippocampus and entorhinal cortex. Place fields are modeled as skewed Gaussians with adapting mean firing rates and adapting place-field center, variance, and skewness components. Using simulated data, we compare the tracking of encoding parameters using the particle filters and a steepest descent based point process filter described in [1]. We predict that resulting distribution will provide insight into difficulty of directly tracking skewness and higher order moments described in that paper. We tracked place field changes in real data examples of a rat moving back and forth along a linear track. We can also, for the first time, put error bounds on the estimates generated by these previous adaptive estimation algorithms.

Summary and Conclusions:

We have developed a pair of computationally simple particle filters in order to examine accuracy of the Gaussian approximation used in the in our previously reported MAP and steepest descent filters and to track receptive field dynamics in situations where these Gaussian assumptions break down. Our results indicate that for simple neural systems the Gaussian approximation is very reasonable. Although we can accurately track the stimulus using either of the particle filter methods described here, this shows that computationally simpler methods, such as the MAP filter can track comparably well. These methods also allow us to compute error bounds on other point-process estimation procedures, even without a system evolution model. The potential power of this method is that it can compute the posterior distributions to arbitrary precision, thereby extracting information optimally from the observed spiking processes.

(1) Brown, EN, Nguyen, DP, Frank, LM, Tang, MC, Wilson, MA, Solo, V (2001) An Analysis of Neural Receptive Field Plasticity By Point Process Adaptive Filtering *PNAS* 98, 12261-12266

(2) Brown, EN, Frank, LM, Tang, D, Quirk, MC, Wilson, MA (1998) A Statistical Paradigm for Neural Spike Train Decoding Applied to Position Prediction from Ensemble Firing Patterns of Rat Hippocampal Place Cells *J. Neuroscience* 18, 7411-7425

(3) Frank, LM, Eden, UT, Solo, V, Wilson, MA, Brown, EN (2001) Contrasting Paterns of receptive field plasticity in the hippocampus and the entorhinal cortex: an adaptive filtering approach. *J. Neurosci.* In press

(4) Doucet, A, de Freitas, N, Gordon, N, (2001) *Sequential Monte Carlo Methods in Practice* Springer, New York, NY

(5) van der Merwe, R, Doucet, A, de Freitas, N, Wan, E (2000) *The Unscented Particle Filter.* Technical Report

(6) Eden, UT, Frank LM, Barbieri R, Solo V, Brown, EN, (2001) *Adaptive Filtering Algorithms for Neural Spike Trains*, Proceedings from Computational Neuroscience 2001 meeting

(7) Rieke F, Warland D, Steveninick, R, Bialek, W (1997) *Spikes: Exploring the Neural Code* MIT Press, Cambridge, MA

(8) Shoham, S, (2001) *Advances Toward an Implantable Motor Cortical Interface* PhD Dissertation, University of Utah