Spike Train Analysis for Single Trial Data [1]

Richard Romero, Tai Sing Lee

Center for the Neural Basis of Cognition & Computer Science Department
Carnegie Mellon University, Pittsburgh, PA 15213, U.S.A.

Abstract

Traditional methods in neural data analysis are not appropriate for analyzing the spike train of a single experimental trial. We show that, by constructing a model of firing statistics, a more accurate estimate of the firing rate for a single spike train can be obtained. The model is based on the assumption that the neuron's spikes are generated by a non-homogeneous Poisson process which follows Markovian dynamics. We test the method by reconstructing the input stimulus based on the neurons' responses either on the raw spike data or the firing rate estimate. The spike data were recorded from macaque V1 neurons in response to a sinewave grating undergoing pseudo-random walk. For a large percentage of the cells studied, the reconstruction is significantly improved by using the estimated firing rate over the raw spikes, suggesting that estimated rate reflects more accurately the underlying state of the neurons.

Key words: V1 neurons; HMM model; Firing rate estimation; Stimulus Reconstruction.

1 Introduction

Most neural data analysis techniques from *in vivo* experiments use a repeated trial approach to estimate the instantaneous firing rate of a neuron. The standard practice has been to apply a linear smoothing operation across trials and time. The implicit assumption is that the mean underlying firing rate is unchanged across trials. In addition, smoothing across time removes high frequency components of a signal, which may or may not be significant for encoding or decoding the signals.

[1] This research has been supported by NSF LIS 9720350, NSF CAREER 9984706 NIH Vision Research core grant EY08098, and a NIH High Performance Computing for Biomedical Research Center grant to Pittsburgh Supercomputing Center. Romero is supported by an NSF Graduate Research Fellowship. Corresponding author: T.S. Lee tai@cnbc.cmu.edu

A peri-stimulus time histogram (PSTH), constructed across trials and using a certain bin width, is a special case of a moving average filter. Linear filtering, however, only re-scales the frequency components in the original signal. In neurophysiological research, the type of filter (i. e., Gaussian versus moving average,) and its parameters can greatly influence the resulting estimate for a firing rate. In general, there is no objective way to assess which frequency components are important and which are not. If the goal is to optimally estimate the instantaneous firing rate from a single trial, one can pose the problem as that of finding the optimal linear filter. Relevant work includes Kalman smoothing and bandwidth selection [1-2].

Suppose that instead of trying to find an optimal filter, we are looking for a probabilistic description of the neuron's response behavior. For simplicity, we assume that the neuron's spikes follow a non-homogeneous Poisson process, dependent only on an unobserved firing rate $r(t)$. Furthermore, we assume that changes in $r(t)$ follow a Markovian dynamic, so the rate $r(t)$ is stationary and dependent only on the previous rate $r(t-1)$. Our goal is to find $P(r(t) \mid S_i)$, the probability of an unobserved firing rate given the observed spike train, where $S_i(t)$ is the neural response for trial i at time t. With these assumptions, we can use a Hidden Markov Model (HMM) [3] to estimate $P(r(t) \mid S_i)$ by building a model of how the firing rate changes over time. The hidden states in the model correspond to different firing rates, and the transition probabilities between states model how the firing rates change over time. The "training" of the HMM gives us a maximum likelihood estimate for the transition probabilities between states and the probability of observing a spike given the current state. The only parameter we need to choose for the model is the number of hidden states. Based on this probabilistic description using the HMM, we can compute the maximum likelihood estimate of the instantaneous firing rate from the spikes of a single trial.

2 Methods

Let us define the number of hidden states as M, the presence of a spike in trial i at time t as $S_i(t) = 1$ and $S_i(t) = 0$ otherwise, and $J_i(t) = j$ to indicate we are in hidden state j in trial i at time t :

$$\pi(j) = P(J_i(1) = j) \tag{1}$$
$$A_{j,k} = P(J(t+1) = k \mid J(t) = j) \tag{2}$$
$$B_j = P(S(t) = 1 \mid J(t) = j) \tag{3}$$
$$\lambda = \{A, B, \pi\} \tag{4}$$

Eq. 1 is the probability that we start in a particular state j. Eq. 2 is the probability of transition from state j to state k, independently of a particular data sequence. Eq. 3 is the probability of observing a spike given we are in state j and corresponds to the firing rate associated with this state. These probabilities can be trained using the Baum-Welch learning algorithm [3].

A HMM is trained for each neuron using the entire corpus of spike train data collected from that neuron in a particular experiment. Note that while our resulting model will vary with the statistical properties of the input signal, the estimates are not computed using the actual input signal. They are dependent only on the observed neural response. Training is done until the likelihood of the data given the model has converged. The model is then used to compute the expected firing rate at all points in time during the trial, given that trial's spike train. The Viterbi algorithm can be used to find the maximum likelihood solution [4]. To find the expected firing rate $\hat{R}(t)$, we define:

$$P(S_i(t) \mid J_i(t) = j) = \begin{cases} B_j & \text{if } S_i(t) = 1 \\ 1 - B_j & \text{if } S_i(t) = 0 \end{cases} \tag{5}$$

$$\begin{aligned} \alpha_{i,j}(t) &= P(S_i(1), S_i(2), \ldots, S_i(t), J_i(t) = j \mid \lambda) \\ &= \sum_{k=1}^{M} \alpha_k(t-1) A_{j,k} P(S_i(t) \mid J_i(t) = j) \end{aligned} \tag{6}$$

$$\begin{aligned} \beta_{i,j}(t) &= P(S_i(t+1), S_i(t+2), \ldots, S_i(T) \mid J_i(t) = j, \lambda) \\ &= \sum_{k=1}^{M} \beta_k(t+1) A_{k,j} P(S_i(t+1) \mid J_i(t+1) = k) \end{aligned} \tag{7}$$

$$\begin{aligned} \gamma_{i,j}(t) &= P(J_i(t) = j \mid S_i, \lambda) \\ &= \frac{\alpha_{i,j}(t)\, \beta_{i,j}(t)}{\sum_{k=1}^{M} \alpha_{i,k}(t)\, \beta_{i,k}(t)} \end{aligned} \tag{8}$$

$$\hat{R}_i(t) = \sum_{j=1}^{M} \gamma_{i,j}(t) B_j \tag{9}$$

We show Eqns. 6-8 in both the probabilistic and in the recursive, computational form, as given by the Baum-Welch training algorithm. Eq. 9 is our expected firing rate, computed as the expected probability of observing a spike across all hidden states. Note that we can use this equation on individual trials to estimate the firing rate at all points in time.

3 Results

To assess the utility of the method, we reconstruct the input stimulus using the data from single trial spike trains. The stimulus used in the experiment was a sinewave grating (as shown in Figure 1a), the phase of which underwent a pseudo-random walk. The step in phase was drawn from a Gaussian random noise distribution low-pass filtered to introduce a certain degree of temporal correlation in the input signals. The sinewave grating moves only along one direction, and can be fully characterized by the cosine of its phase, which corresponds to the intensity of the image at the center of the receptive field. Recordings were obtained from neurons in awake macaque V1 for approximately 400 trials, each lasting 2.2 seconds. The monkey's only task was to maintain fixation during this period.

(a) (b)

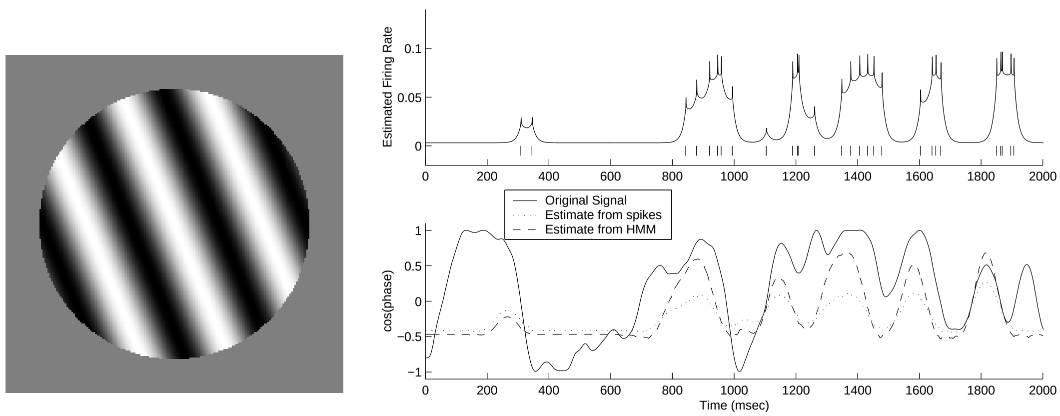

Fig. 1. (a) Stimulus display in the actual experiment. The sinewave grating moved back and forth in a pseudo-random walk. The orientation and spatial frequency of the grating were chosen according to the optimal tuning of the cell. (b) Results of the HMM estimated firing rate of a neuron (upper panel) and stimulus reconstruction (lower panel). The original signal and its estimated reconstruction using both the HMM estimated rate and raw spike data are shown.

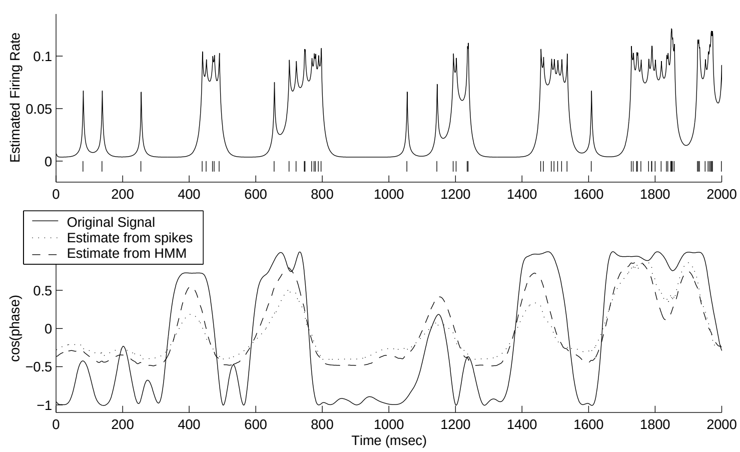

Fig. 2. Another example of the estimated firing rate and resulting reconstruction from the HMM model, this time for a cell with a higher mean firing rate.

We used $M = 10$ hidden states in training the hidden Markov models. The initial conditions were set to:

$$A_{k,j} = \begin{cases} \frac{1}{2M-2} & \text{if } k \neq j \\ 0.5 & \text{if } k = j \end{cases} \tag{10}$$

$$\pi_j = \frac{1}{M} \tag{11}$$

$$B_j = \frac{j}{M+1} \tag{12}$$

The HMM was trained until the likelihood of the data had converged, defined as $P(S_i \mid \lambda) = \sum_{j=1}^{M} \alpha_{i,j}(T)$. The model was then used to estimate the firing rate as in Eq. 9.

To reconstruct the input signal from the spike train of the neuron or the firing rate estimate from the HMM, we compute the optimal linear reconstruction kernel H [5] as follows, where $\phi_i(t)$ is the phase of the grating for trial i at time t:

$$\mathbf{X_i} = \begin{bmatrix} \hat{R}_i(1) & \hat{R}_i(2) & \dots & \hat{R}_i(\tau) \\ \hat{R}_i(2) & \hat{R}_i(3) & \dots & \hat{R}_i(\tau+1) \\ \vdots & & & \vdots \\ \hat{R}_i(T-\tau+1) & \hat{R}_i(T-\tau+2) & \dots & \hat{R}_i(T) \end{bmatrix} \tag{13}$$

$$\mathbf{Y_i} = \begin{bmatrix} \cos\phi_i(\tau) \\ \cos\phi_i(\tau+1) \\ \vdots \\ \cos\phi_i(T) \end{bmatrix} \tag{14}$$

$$H = \left(\sum_{i=1}^{N} X_i' X_i\right)^{-1} \left(\sum_{i=1}^{N} X_i' Y_i\right) \tag{15}$$

H is the optimal least-squares solution to $\mathbf{X_i} H = \mathbf{Y_i}$ across all trials i. An SVD method is used to compute the matrix inverse to alleviate problems with the temporal correlations in the data. To use H in estimating the input signal, we convolve it with the estimated firing rates $\hat{R}$. Two examples of the HMM output and the reconstruction are given in Figures 1 and 2. Also shown in these figures are the performance of a linear reconstruction obtained based on the raw spike train, computed as above but replacing $\hat{R}_i(t)$ with $S_i(t)$.

To evaluate the performance, we compare $\bar{\epsilon}^2$, the mean squared error (MSE) of the stimulus reconstructions for the HMM and raw spike estimates. Figure 3 shows the MSE's plotted against each other. Also shown is a histogram for an improvement index, computed as

$$\delta = \frac{\bar{\epsilon}^2_{\text{Raw}} - \bar{\epsilon}^2_{\text{HMM}}}{\bar{\epsilon}^2_{\text{Raw}} + \bar{\epsilon}^2_{\text{HMM}}}$$

We examine the properties of the cells for which our estimate of firing rate results in an improved reconstruction. Figure 3C show that for a moderate or high firing rate, the reconstruction is improved. We have used a time resolution

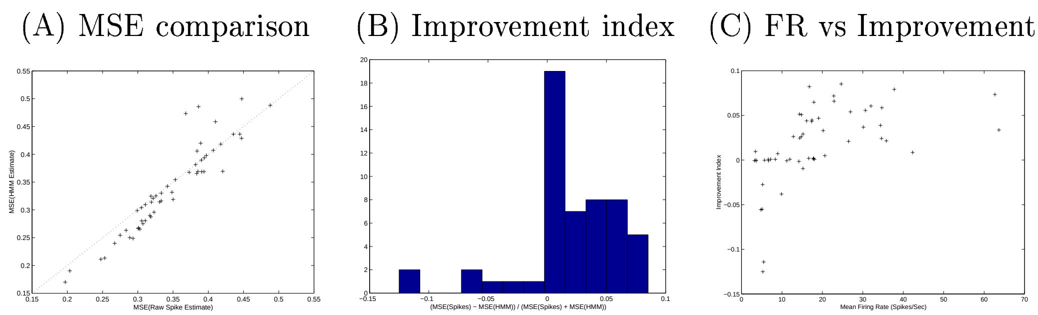

Fig. 3. Error measure comparisons: (A) A comparison of errors for the reconstruction based only on the raw spike information and the errors for the reconstruction using the HMM estimate of the firing rate. (B) A histogram of the improvement index over the set of neurons, showing a small but significant improvement of the reconstruction based on HMM estimate. (C) Firing Rate versus Improvement: Plot of the cell's mean firing rate versus the improvement index. An index of 0 corresponds to no improvement and < 0 a decrease in performance. We see that the HMM method improves the reconstruction more for neurons with higher firing rates.

of 1 msec. At this time scale, a neuron with a low firing rate may not behave in a Markov fashion at all. This particular problem hints at related issues that arise from using a discrete representation. There are proposals for training HMM's using continuous probability distributions [6]. In addition, work in the statistical literature on particle filters may also provide extensions into the continuous domain [7]. For future work, we will extend our modeling to the continuous domain.

References

[1] R. E. Kalman & R. S. Bucy, New results in linear filtering and prediction, *Journal of Basic Engineering*, 83D (1961) 95-108.

[2] S. J. Sheather & M. C. Jones, A reliable data-based bandwidth selection method for kernel density estimation. *J.R. Statistical Soc. B*, 53(3) (1991) 683-690.

[3] L. R. Rabiner & B. Juang, An introduction to hidden Markov models, *IEEE ASSP Magazine*, 3(1) (1986) 4-16.

[4] G. D. Forney, The Viterbi algorithm, *Proceedings of the IEEE*, 61(3) (1973) 268-278.

[5] F. Rieke, D. Warland, R.D. Ruyter van Steveninck, & W. Bialek, *Spikes: Exploring the Neural Code*, Cambridge, MA: MIT Press, 1997.

[6] S. Thrun, J. C. Langford & D. Fox Monte Carlo hidden Markov models: learning non-parametric models of partially observable stochastic processes *Proceedings of ICML-99, Sixteenth International Conference on Machine Learning* Ed. I. Bratko and S. Dzeroski, vol 16, 415-424, Morgan Kaufmann, San Francisco, 1999.

[7] M. K. Pitt & N. Shephard, Filtering via Simulation: Auxiliary Particle Filters, *Journal of the American Statistical Association*, 94 (446) 1999.