A Rate and History-Preserving Resampling Algorithm for Neural Spike Trains

Matthew T. Harrison
mtharris@cmu.edu
Department of Statistics
Carnegie Mellon University
Pittsburgh, PA 15213

Stuart Geman
stuart_geman@brown.edu
Division of Applied Mathematics
Brown University
Providence, RI 02912

September 9, 2008

Abstract

Resampling methods are popular tools for exploring the statistical structure of neural spike trains. In many applications it is desirable to have resamples that preserve certain non-Poisson properties, like refractory periods and bursting, and that are also robust to trial-to-trial variability. "Pattern jitter" is a resampling technique that accomplishes this by preserving the recent spiking history of all spikes and constraining resampled spikes to remain close to their original positions. The resampled spike times are maximally random up to these constraints. Dynamic programming is used to create an efficient resampling algorithm.

1 Introduction

Neural spike trains (conceptualized as point processes) have complicated dynamics, most notably time-varying firing rates (including trial-to-trial variability) and spike-history effects (such as refractory periods and bursting). Here we develop an algorithm, *pattern jitter*, for generating artificial spike trains that are maximally random while preserving the smoothed firing rates and the recent spike histories in a recorded spike train. Monte Carlo samples generated by the algorithm can be used to quantify how these dynamics constrain the sampling variability of spike trains, and thereby help reveal the source, and in some cases the statistical significance, of additional characteristics, such as synchronous spikes or other forms of repeated spike-time patterns in the recorded data (cf. Date et al., 1998; Oram et al., 1999; Nádasdy et al., 1999; Baker & Lemon, 2000; Gerstein, 2004; Ikegaya et al., 2004; Amarasingham, 2004; Harrison, 2005; Shmiel et al., 2005; Shmiel et al., 2006; Pazienti et al., 2008; Fujisawa et al., 2008; Geman et al., 2008). Quantifying variability is of course the essence of any statistical procedure.

Let $\tilde{x} := (\tilde{x}_1, \ldots, \tilde{x}_n)$ denote the observed spike train, a non-decreasing sequence of spike times. This can be from a single experimental trial or from a concatenation of multiple trials. Our primary purpose is to formulate a probability distribution for *resampled* spike trains based on a fixed $\tilde{x}$. Section 4 discusses connections between the resampling distribution and statistical

models for the variability in $\tilde{x}$. For convenience, we assume that spike times have been discretized into equal-length time bins and are then represented as integers, in units of bins. A continuous version of the resampling algorithm is presented in the Appendix.

The random vector $X := (X_1, \ldots, X_n)$ denotes the resampled spike train and $p(x_1, \ldots, x_n) := p(x) := \Pr(X = x)$ is its probability distribution. Again, all of this is with respect to a fixed and known $\tilde{x}$, which we suppress from the notation. In Section 2 we describe the distribution p and in Section 3 we use dynamic programming to create an efficient and exact sampling algorithm for p. Some generalizations are discussed in Section 5.

2 The resampling distribution

In order to preserve smoothed firing rates, we require that each resampled spike remain close to its corresponding original spike. This is the "jitter" principle developed in Date et al. (1998), Amarasingham (2004), Harrison (2005) and Geman et al. (2008). Since spikes stay in approximately the same place, the resamples preserve all coarse-temporal variation in firing rates, including trial-to-trial variation. Formally, for each $i = 1, \ldots, n$,

$$X_i \in \Omega_i \quad \text{where} \quad \Omega_i := \big\{\tilde{x}_i - \lceil L/2 \rceil + 1, \ldots, \tilde{x}_i - \lceil L/2 \rceil + L\big\}, \tag{1}$$

and where $\lceil \cdot \rceil$ denotes the ceiling function (round up). Hence X_i is constrained to the L bins centered around $\tilde{x}_i$. We call Ω_i the ith jitter window. (Motivated by statistical considerations, we will later modify the definition of Ω_i; see Section 4.) The parameter L controls the degree of smoothing: small L preserves rapid changes in firing rate, but introduces less variability into the resamples.

To preserve recent spike-history effects, we require that the resamples and the original recording have identical patterns of spiking and not-spiking in the R bins preceding each spike. The parameter R controls the amount of history that is preserved. Larger values of R enforce more regularity in the firing patterns across the resampled spike trains. Additionally, we require that spike ordering be preserved, which amounts to a coarse history constraint in that the total number of spikes preceding each spike is preserved. It is not hard to see that these two history constraints are equivalent to the constraint that, for each $i = 2, \ldots, n$,

$$X_i - X_{i-1} \in \Gamma_i \quad \text{where} \quad \Gamma_i := \begin{cases} \big\{\tilde{x}_i - \tilde{x}_{i-1}\big\} & \text{if } \tilde{x}_i - \tilde{x}_{i-1} \leq R, \\ \big\{R+1, R+2, \ldots\big\} & \text{if } \tilde{x}_i - \tilde{x}_{i-1} > R. \end{cases} \tag{2}$$

In words, $X_i - X_{i-1}$ is forced to match $\tilde{x}_i - \tilde{x}_{i-1}$ whenever the latter is R or less, and $X_i - X_{i-1}$ is forced to be strictly larger than R otherwise.

A simple consequence of the history constraint in (2) is that the interspike interval (ISI) histogram of each resampled spike train exactly matches the ISI histogram of the observed spike train for ISIs of R bins or less. Even higher order ISI histograms (counts of unique tuples of consecutive ISIs) are preserved for ISIs of R bins or less. For ISIs of more than R bins, the jitter principle in (1) ensures that the ISI histogram of each resampled spike train is coarsely similar to that of the original. The commonly used models for neural spike trains, such as inhomogeneous renewal processes or inhomogeneous Markov interval processes (Kass & Ventura, 2001), are typically specified so that the distribution of ISIs is more structured for short ISIs and more smooth for long ISIs, as

is consistent with (1) and (2), provided that R is not excessively short and L is not excessively long. Thus, to the extent that an observed spike train conforms to such a model, the resampling distribution will preserve the essential history-dependent features of the model.

There are many distributions that preserve (1) and (2). Since our goal is to impose no additional structure, we make no additional constraints: the allowable spike configurations are distributed uniformly, meaning that

$$p(x) := \frac{1}{Z} \underbrace{\mathbb{1}\{x_1 \in \Omega_1\}}_{h_1(x_1)} \prod_{i=2}^{n} \underbrace{\mathbb{1}\{x_i \in \Omega_i\}\mathbb{1}\{x_i - x_{i-1} \in \Gamma_i\}}_{h_i(x_{i-1}, x_i)}, \tag{3}$$

where $\mathbb{1}\{A\}$ is the indicator function of the set A, and where Z is a normalization constant that depends on the Ω_i's and the Γ_i's, and hence on the parameters L and R and the original spike train, $\tilde{x}$. (We provide further motivation for our choice of the uniform distribution in Section 4.)

In principle, one could sample from p by independently and uniformly choosing all n spike times and rejecting any configuration of spikes that does not satisfy both (1) and (2). But this is impractical, and it makes more sense to exploit the evident Markov property of p.

3 Sampling from the resampling distribution

The customary Markov chain representation for p is

$$p(x) = p_1(x_1) \prod_{i=2}^{n} p_i(x_i|x_{i-1}), \tag{4}$$

where $p_1(x) := \Pr(X_1 = x_1)$ is the initial distribution and $p_i(x_i|x_{i-1}) := \Pr(X_i = x_i|X_{i-1} = x_{i-1})$ are the transition probabilities. As usual (cf. Frey, 1998), there is a dynamic-programming algorithm for computing this representation using the h_i's in (3). In this regard, it is important to note that the h_i's are trivial, both to compute and to represent, once $\tilde{x}$, L and R are specified.

In the following calculations, although we often suppress explicit mention of Ω_i, functions of x_i are implicitly restricted to Ω_i and sums over x_i are implicitly over $x_i \in \Omega_i$. For $1 < i < n$ define

$$\beta_i(x_{i-1}, x_i) := \sum_{x_{i+1}} \cdots \sum_{x_n} \prod_{j=i}^{n} h_j(x_{j-1}, x_j) = h_i(x_{i-1}, x_i) \sum_{x_{i+1}} \beta_{i+1}(x_i, x_{i+1}),$$

where we take $\beta_n(x_{n-1}, x_n) := h_n(x_{n-1}, x_n)$. Similarly, define

$$\beta_1(x_1) := \sum_{x_2} \cdots \sum_{x_n} h_1(x_1) \prod_{j=2}^{n} h_j(x_{j-1}, x_j) = h_1(x_1) \sum_{x_2} \beta_2(x_1, x_2).$$

As indicated by the formulae, the β_i's can be computed recursively (in reverse) from the h_i's. The entire computation requires $O(m^2 n)$ operations and a similar amount of storage, where

$$m := \max_{i=1,\ldots,n} |\Omega_i|$$

is the size of the largest jitter window. (For now, $m = L$, but we are anticipating generalizations in Section 5.)

The Markov representation is in terms of the β_i's:

$$p_1(x_1) = \frac{\beta_1(x_1)}{\sum_{x'_1} \beta_1(x'_1)} \quad \text{and} \quad p_i(x_i|x_{i-1}) = \frac{\beta_i(x_{i-1}, x_i)}{\sum_{x'_i} \beta_i(x_{i-1}, x'_i)}.$$

To verify this, notice that

$$\Pr(X_1 = x_1, \ldots, X_i = x_i) = \sum_{x_{i+1}} \cdots \sum_{x_n} p(x) = \beta_i(x_{i-1}, x_i) \frac{1}{Z} h_1(x_1) \prod_{j=2}^{i-1} h_j(x_{j-1}, x_j),$$

so

$$\begin{aligned}\Pr(X_i = x_i|X_1 = x_1, \ldots, X_{i-1} = x_{i-1}) &= \frac{\beta_i(x_{i-1}, x_i) \frac{1}{Z} h_1(x_1) \prod_{j=2}^{i-1} h_j(x_{j-1}, x_j)}{\sum_{x'_i} \beta_i(x_{i-1}, x'_i) \frac{1}{Z} h_1(x_1) \prod_{j=2}^{i-1} h_j(x_{j-1}, x_j)} \\ &= \frac{\beta_i(x_{i-1}, x_i)}{\sum_{x'_i} \beta_i(x_{i-1}, x'_i)}.\end{aligned}$$

The dependence on $x_1, \ldots, x_{i-2}$ disappears, so this expression must be $p_i(x_i|x_{i-1})$. (This is in fact a proof of the earlier claim that p is Markov.) The formula for p_1 is verified in a similar manner.[1]

Sampling from (4) is trivial. First sample X_1 from $p_1(\cdot)$ and then recursively sample X_i from $p_i(\cdot|X_{i-1})$. An entire sample requires $O(mn)$ operations and a similar amount of storage.

4 Statistical considerations

In statistical applications, we typically want to make statements about the (hypothetical) distribution of the observed data $\tilde{x}$. Is there a connection between our resampling distribution and some model for $\tilde{x}$? Certainly there is a *heuristic* connection: The constraints (1) and (2) describe a collection of spike trains that are similar to the recorded train, $\tilde{x}$, and the distribution p weights them all equally. Assuming that the *model* for $\tilde{x}$ also assigns an approximately uniform measure to members of this collection, then samples from p approximate samples from the model, and can be used to build approximate confidence intervals, hypothesis tests, and so on. The statistical intuition is therefore very similar to the intuition underlying the bootstrap (Davison & Hinkley, 1997; Ventura, 2004).

As it stands, however, it would be hard to go beyond such heuristic interpretations of p. But a minor change in the formulation endows p with a precise statistical interpretation, while preserving the essential properties of the resampling scheme. The idea is to modify p so that it can be expressed as

$$p(x) = \frac{1}{Z} \mathbb{1}\{S(x) = S(\tilde{x})\} \tag{5}$$

[1]Note that the β_i's can be scaled by arbitrary constants without affecting the resulting p_i's and that this scaling can happen during the recursive computation of the β_i's. This is important in an actual implementation to prevent underflow or overflow. A good choice is to normalize each β_i to sum to one (a double sum over both arguments) before it is used in the next step of the recursion.

for some (vector-valued) summary statistic S that is specified independently of $\tilde{x}$. If (5) holds, then we can formally interpret p as a model for the *conditional* distribution of $\tilde{x}$, given the observed value of the summary statistic $S(\tilde{x})$, and as a consequence the resampled and the original spike trains will be, as a collection, exchangeable (under the model). Notice that such a representation cannot hold under the original formulation (3), since each $\tilde{x}_i$ lies at the center of its jitter window, Ω_i, requiring that any such representation employs an S that depends on $\tilde{x}$.

In the context of jittering without history constraints, this problem was addressed in Date et al. (1998), Amarasingham (2004), Harrison (2005) and Geman et al. (2008) by simply modifying the jitter windows so that they are determined *a priori* from a fixed partitioning of time. Jitter windows *contain* the associated spikes, but are not necessarily *centered* on the associated spikes. Formally, the modified jitter windows are $\Omega_i = \big\{\lfloor \tilde{x}_i/L \rfloor L, \ldots, \lfloor \tilde{x}_i/L \rfloor L + L - 1\big\}$, where $\lfloor \cdot \rfloor$ denotes the floor function (round down), so that Ω_i (possibly identical for many indices, i) is the interval in the *a priori* partition that contains $\tilde{x}_i$. This same modification works here, but interacts with the history constraint in a way that can substantially reduce the variability of the resamples, correspondingly reducing the power of the associated tests. A better solution borrows the fixed-partition idea, but only applies it to certain spikes. To describe the idea, we introduce the notion of a *pattern*.

4.1 Modifying the jitter windows

The history constraints in (2) uniquely partition the original spike train and any valid resampled spike train into *patterns*. A pattern is a maximal sequence of consecutive spikes such that every neighboring pair of spikes has interspike interval no greater than R bins. Notice that (2) guarantees that a resampled spike train has, exactly, the same ensemble of patterns, in the same order, as the recorded spike train $\tilde{x}$, albeit with shifted positions. If we think of the position of a pattern as the position of its first spike, then the resampling distribution p developed in Section 2 can be viewed as a *pattern jitter*, wherein the position of each pattern is jittered within the L-length window centered at its first spike, but with the constraint that two consecutive patterns remain separated by at least R bins. If, on the other hand, we choose an *a priori* partitioning and apply this partitioning to the positions of patterns, then for a suitable summary statistic S the resulting resampling distribution p has the representation given in (5), and the desirable interpretation as a model for the conditional distribution of $\tilde{x}$ given $S(\tilde{x})$.

Here are the details. Let $k_1, \ldots, k_d$ denote the (increasing) indices of spikes in $\tilde{x}$ that start a pattern, that is, $\tilde{x}_{k_j} - \tilde{x}_{k_j - 1} > R$, where for notational convenience we define $\tilde{x}_0 := -\infty$ so that $k_1 = 1$, and $k_{d+1} := n + 1$. For each $i = 1, \ldots, n$, redefine

$$\Omega_i := \big\{\lfloor \tilde{x}_{k_j}/L \rfloor L + \tilde{x}_i - \tilde{x}_{k_j}, \ldots, \lfloor \tilde{x}_{k_j}/L \rfloor L + \tilde{x}_i - \tilde{x}_{k_j} + L - 1\big\} \quad \text{for} \quad k_j \le i < k_{j+1}. \tag{6}$$

The new Ω_i is the L-length jitter window for the ith spike that results from constraining the spike's pattern to be jittered in an *a priori* window containing the pattern's first spike, as it was located in the original spike train. Thus the new Ω_i still contains $\tilde{x}_i$, but it is not necessarily centered on $\tilde{x}_i$. Using these new Ω_i's in the definition of p, as given in (3), provides a resampling distribution with nearly identical properties, but with the additional benefit of obeying (5). One way to formulate the corresponding summary statistic is

$$S(x) := \big(S_{11}(x), \ldots, S_{n1}(x), S_{12}(x), \ldots, S_{n2}(x)\big),$$

where,

$$S_{i1}(x) := \begin{cases} \lfloor x_i/L \rfloor L & \text{if } x_i - x_{i-1} > R, \\ x_i - x_{i-1} & \text{otherwise,} \end{cases} \qquad \text{and} \qquad S_{i2}(x) := \begin{cases} 1 & \text{if } x_i - x_{i-1} > R, \\ 0 & \text{otherwise.} \end{cases} \tag{7}$$

S_{i1} records either the beginning of the jitter window for spike i or it records the interspike interval preceding spike i, depending on whether or not spike i starts a pattern. S_{i2} indicates how to interpret S_{i1}.

It is straightforward to verify that $\{x : S(x) = S(\tilde{x})\}$ is the set of all spike trains with the same ensemble of patterns, in the same order, as the recorded spike train, and with the initial spike of each pattern occurring in the same jitter window as the corresponding spike in the recorded spike train. The modified resampling distribution, p, given in (5) is simply the uniform distribution over this set of spike trains.

The dynamic programming algorithm to convert the modified p into its Markov representation is essentially unchanged. Indeed, we did not make use of the structure of the Ω_i's in the algorithm. Furthermore, the efficiency of the algorithm only varies with the sizes of the Ω_i's, and these are unchanged. Certain special cases do not require dynamic programming. For example, if $R = 0$ and if the discretization is sufficiently fine so that no (discretized) recorded spike times are identical, then pattern jitter reduces to the original jitter method first used in Date et al. (1998), which can be sampled by independently permuting the sequences of ones and zeros (spike or no spike) in each jitter window. Or if R is at least as large as the largest interspike interval in the recorded spike train, say $R = \infty$, so that the entire spike train is a single pattern, then pattern jitter is closely related to the resampling method in Pipa et al. (2007) which can be sampled by uniformly jittering each spike by the same amount.

Figure 1 shows some examples of an original spike train (from primate motor cortex[2]) and a collection of resamples from the corresponding p. Each subfigure shows different combinations of L and R. Applications of pattern jitter to real data can be found in Harrison (2005) and Geman et al. (2008). A specific application would be to test whether precisely synchronous spike pairs (say within ± 1 ms) between two simultaneously recorded neurons occurred at approximately the same rate in the recorded data as in the surrogate data, preserving, say, 50-ms histories ($R = 50$ ms) and pattern locations within 20 ms ($L = 20$ ms). Software for pattern jitter is available online.[3]

4.2 Implications of the modified p

Think of the recorded data, $\tilde{x}$, as a realization of a random process, $\tilde{X}$. If we make the assumption that the conditional distribution of $\tilde{X}$ given $S(\tilde{X}) = s$ is uniform for all s,[4] then the resampling distribution (p of equation (5)), is the conditional distribution of $\tilde{X}$ given $S(\tilde{X}) = S(\tilde{x})$. That is

$$\Pr\big(\tilde{X} = x \big| S(\tilde{X}) = s\big) = p(x) \quad \text{for all } x \text{ and for } s = S(\tilde{x}).$$

[2]Courtesy of N. Hatsopoulos.

[3]http://jitter.stat.cmu.edu/

[4]If we have a class of models with the common property that conditional distributions are always uniform after conditioning on S, then S is a sufficient statistic for this model class. The converse is not true. Lauritzen (1988) explores the mathematical foundations of building models by first specifying the sufficient statistics and also suggests the uniform distribution as an intuitive conditional model in the absence of further information.

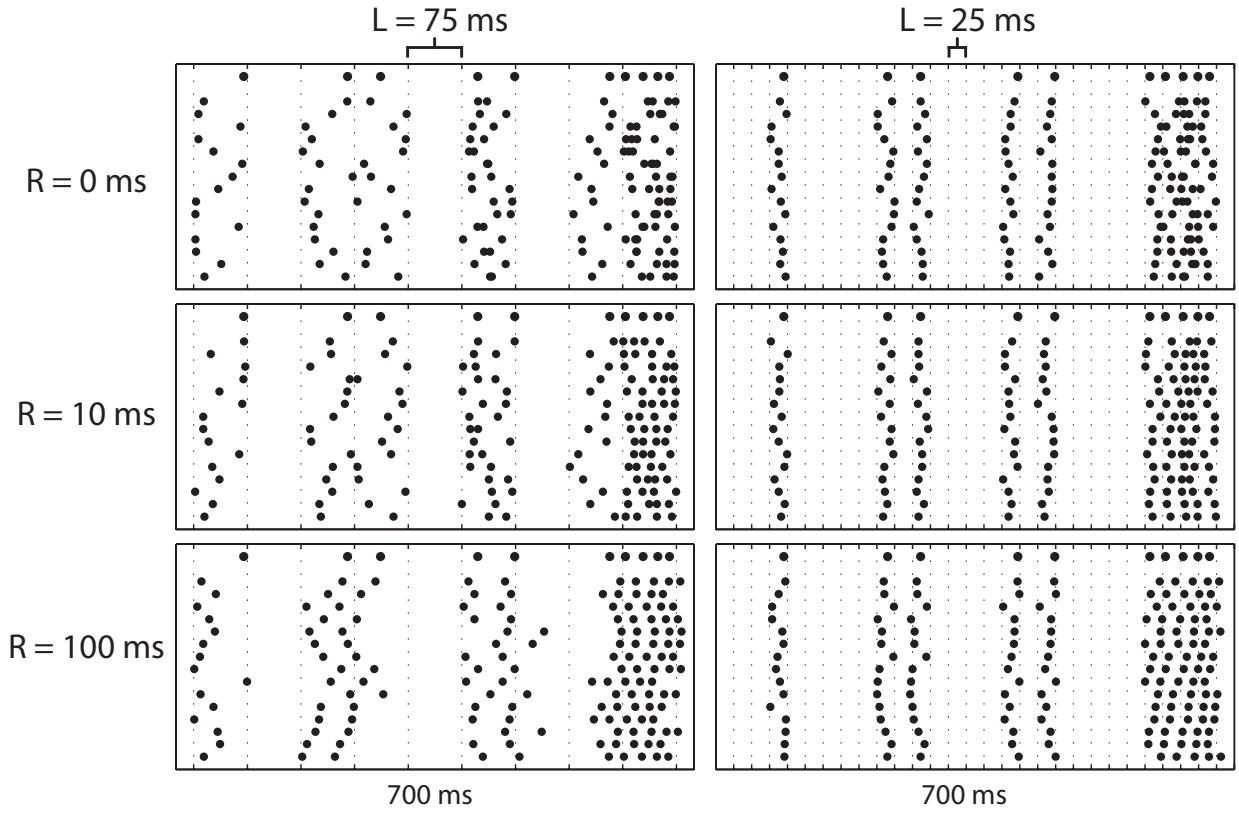

Figure 1: The top horizontal line of dots for each subfigure shows an identical sequence of observed spike times (dots) during a 700 ms interval (finely discretized to a $1/30$th ms grid). Each of the remaining 15 lines of dots shows an independent sample from the corresponding p. The subfigures on the left have $L = 2250$ bins (75 ms) and those on the right have $L = 750$ bins (25 ms). The vertical dotted lines show the boundaries of the jitter windows. Notice that coarse-temporal modulations in firing rate are preserved by the resamples, with L controlling the notion of "coarse." The top, middle and bottom rows of subfigures have $R = 0$, 300 and 3000 bins (i.e., 0, 10 and 100 ms), respectively. The only patterns in the top and middle rows are single spikes, however, since the middle row has a larger R, the resamples are more regular: spikes cannot be jittered too close to one another or new patterns will be created. The bottom row has four patterns. Notice that the patterns are preserved across resamples and that only the first spike of each pattern is constrained to remain in its original jitter window.

Monte Carlo samples from p can then be used for conditional inference, in the usual way. For example, we can compute the (conditional) variance of some estimator under the assumption of uniformity. Or, we can test the assumption of (conditional) uniformity by treating samples from p as samples under a null hypothesis, and comparing them to the original data.

Consider a search for evidence for unusual structure in a single or multiple spike trains, such as repeating sequences of interspike intervals (as in Abeles & Gerstein, 1988; Abeles et al., 1993), or precisely timed synchronies (as in Alonso et al., 1996; Dan et al., 1998). How much of the unusual structure is to be expected by chance alone? For spike trains, a proper notion of chance would necessarily account for time-varying firing rates, possibly correlated across neurons but changing from trial to trial (cf. Brody, 1998), and known spike-history effects (especially refractory periods and bursting–cf. Oram et al., 1999). The proposed resampling distribution accounts for these effects via conditioning, while introducing, by virtue of the conditional uniform distribution, no additional structure. In this sense, p quantifies what we mean by chance, and a rejection of the null hypothesis (that $\Pr\big(\tilde{X} = x \big| S(\tilde{X}) = s\big)$ is indeed uniform) is evidence for the presence of structure in the recorded data above and beyond what can be explained by time-varying firing rates (at least at time scales slower than L) and spike-history effects (of duration R or smaller).[5] For alternative approaches to similar problems, see Ventura et al. (2005a) and Ventura et al. (2005b) for model-based techniques using bootstrap resampling; see Oram et al. (1999), Nádasdy et al. (1999), Pauluis et al. (2001), Baker and Lemon (2000) and Gerstein (2004) for techniques relying on heuristic resampling schemes; and see Amarasingham (2004), Harrison (2005) and Geman et al. (2008) for formal hypothesis tests based on the jitter principle that avoid explicit resampling.

We have appealed to intuitive arguments to justify the choice of a conditionally uniform resampling distribution. We will briefly mention two other approaches that lead to the same choice. One is through the "Maximum Entropy Principle" (Jaynes, 1982), a common heuristic for probability models. Keeping in mind that the range of $\tilde{X}$ is finite and discrete, the maximum entropy distribution on the spike train $\tilde{X}$, given *any* distribution on $S(\tilde{X})$, places the uniform distribution on the conditional probabilities $\Pr\big(\tilde{X} = x \big| S(\tilde{X}) = s\big)$, for every s. In that entropy is a measure of *lack* of structure, the conditionally uniform distribution can be interpreted as the one that imposes the least additional structure, beyond any probabilistic factors already governing rate changes and spike-history effects.

A second approach, also through conditioning on $S(\tilde{X}) = s$, is to model the conditional distribution of $\tilde{X}$ as the conditional distribution of an appropriately discretized homogeneous Poisson process. The homogeneous Poisson process is the canonical structureless point process, and being conditionally Poisson is a reasonable notion of being conditionally structureless. It is easy to show that conditioning a homogeneous Poisson process on $S(\tilde{X}) = s$ again leads to the uniform resampling distribution.

[5] As with any hypothesis test, we are cautioned to look at both a p-value *and* a measure of the magnitude of the departure from the null. This protects against rejecting the specifics of the null, when, in fact, we really only want to reject the spirit of the null, e.g., rejecting precise uniformity (which is not particularly interesting) instead of rejecting approximate uniformity (which still captures the notion of "structureless").

5 Generalizations

With reference to equation (3), any non-negative and finite functions can substitute for the indicator functions currently defining $\{h_i\}_{i=1}^n$, without changing the basic plan of re-factorization through dynamic programming and ease of sampling through the Markov-chain representation. The modification introduced in Section 4 was of this type. In general, of course, such modifications will invalidate the particular interpretations offered in Section 4. Some simple modifications that might be desirable include holding certain spike times constant, changing the windows in which patterns are jittered (the Ω_i's), using a nonuniform jitter distribution, or introducing specific modeling assumptions like absolute and relative refractory periods regardless of whether they appear in the data. Only the first of these (holding certain *a priori* spike times constant) is likely to retain the same interpretations as our final version of pattern jitter.[6] Various generalizations along these lines are explored in Harrison (2005).

Heuristic modifications like these can be used to more adaptively control the smoothing of firing rates as dictated by the jitter window length L. For example, the length of the jitter window could be reduced in time intervals with (potentially) rapid firing rate changes, such as intervals around stimulus onset or (for data sets with repeated trials) intervals with large changes in the trial-averaged peri-stimulus time histogram (PSTH). However, jitter-induced smoothing is often desirable. The slower (smoothed) dynamics in the jitter surrogates can serve to test for faster dynamics in the recorded spike train. Thus, for example, the number of observed millisecond synchronies can be compared to the ensemble of synchrony counts produced by jitter, to give evidence for dynamics operating on a time scale faster than the jitter interval.

A more complicated endeavor is to craft an alternative function S that preserves different aspects of the recorded data, and then repeat the development of pattern jitter. Even assuming that the uniform conditional distribution still makes good modeling sense, it may be difficult to sample from. Pattern jitter avoids this problem because S is structured so that $\mathbb{1}\{S(x) = s\}$ factors into a product of simple functions of spike times, each depending only on a single pair of two neighboring spikes. Such products define first-order Markov chains, and are therefore amenable to dynamic programming. It is well known that the dynamic programming principle applies much more generally, and remains computationally feasible, provided that the graph defined by the factorization (wherein two spikes are neighbors if they appear together in one or more factors) admits to a site-visitation schedule with small maximum boundary (cf. Geman & Kochanek, 2001) or, equivalently, small tree width (cf. Lucena, 2002). Of course in situations where no such factorization exists, it may still be possible to sample (perhaps approximately) using Monte Carlo methods (cf. Chen et al., 2005 and the references therein for resampling contingency tables with marginal constraints).

Finally, one can attempt to formally relax the assumption of uniformity by developing a family of resampling distributions that are approximately uniform and investigating various properties of this family (usually worst-case bounds for a specific test statistic, analogous to the classical method of defining p-values for a composite null hypothesis). See Amarasingham (2004), Harrison (2005) and Geman et al. (2008) for further details.

[6] In practice, we actually prefer to hold the first and last spike times constant because this ensures that the resampled spike trains are valid, in terms of spike-history, even in the context of (perhaps unobserved) spikes occurring before or after the experimental interval.

6 Summary

It is unlikely that any time soon the neurophysiologist will be able to fully control the myriad of inputs that influence a recorded cortical cell's activities in awake and behaving animals. Thus the notion of repeated trials, invaluable for the experimental exploration of neuronal dynamics, must be approached with some care in the ensuing statistical analysis. We have proposed some methodologies for addressing this challenge (cf. Amarasingham et al., 2006; Geman et al., 2008), including the pattern-jitter algorithm developed here.

Pattern jitter is a spike-train resampling algorithm that preserves spiking patterns and coarsely measured (i.e., smoothed) firing rates, while avoiding an explicit estimate of firing rate or any assumption of stationary dynamics. It is controlled by a parameter L, determining the degree of smoothing, and a parameter R, defining the length of preserved history and thereby the notion of a pattern. Both parameters are easy to interpret and the method can be concisely stated as resampling from the uniform distribution conditioned on the sequence of patterns and the coarsely measured pattern starting times. An efficient dynamic-programming preprocessing step transforms the original data into a Markov-chain representation that permits fast resampling. Both the preprocessing and the resampling are linear in the number of spikes. Implementation is straightforward. In that the resampled spike train preserves coarsely measured firing rates as well as spike-history effects, an ensemble of resampled spike trains can be used to gauge the significance of additional structures residing in the recorded data.

Appendix

The discretized pattern jitter preprocessing algorithm scales with the square of the number of bins in a jitter interval. For very finely discretized spike times this can sometimes be improved by passing to the limit and using a continuous version of pattern jitter. We briefly sketch the algorithm here. Complete details and software are available online.[7] Here the parameters L and R are no longer in units of bins, but are actual temporal units.

To specify the limiting distribution, we need only specify the (limiting) joint distribution of the pattern starting times. The remaining spikes are then uniquely determined by the sequence of patterns encoded in S. We could have taken a similar approach for discrete pattern jitter, but then the generalizations are less transparent. Fixing the observation $\tilde{x}$, let $k_1 < k_2 < \ldots < k_d$ denote $\{k : S_{k2} = 1\}$, i.e., those spike indices that start patterns. Let $W := (W_1, \ldots, W_d) := (X_{k_1}, \ldots, X_{k_d})$ denote the sequence of pattern start times for a generic resampled spike train. The limiting joint distribution of W is given by the density

$$f(w) := \frac{1}{Z} \underbrace{\mathbb{1}\{a_1 \leq w_1 < b_1\}}_{h_1(w_1)} \prod_{j=2}^{n} \underbrace{\mathbb{1}\{a_j \leq w_j < b_j\}\mathbb{1}\{w_j - w_{j-1} > r_j\}}_{h_j(w_{j-1}, w_j)},$$

where a_j and b_j are chosen so that $\lfloor w_j/L \rfloor = \lfloor \tilde{x}_{k_j}/L \rfloor$ and where $r_j := R + \tilde{x}_{k_j - 1} - \tilde{x}_{k_j - 1}$ is the minimum separation between W_{j-1} and W_j that keeps the $(j-1)$st pattern distinct from the jth pattern. $Z = Z(\tilde{x}, L, R)$ is a normalizing constant.

[7] http://jitter.stat.cmu.edu/

The product form of the density shows that W is a continuous state-space Markov chain. Formally, the same recursion that is used to compute the transition probabilities in the discrete case can be used to compute the transition densities in the continuous case. Simply repeat the development in Section 3 replacing sums with integrals throughout.

The specific form of the h_j's for pattern jitter, namely, indicator functions, ensures that the β_j's and the resulting transition densities will be piecewise polynomial. So they can be easily represented as lists of polynomial coefficients for each piece, and the integrals can be computed analytically. Sampling from piecewise polynomial densities is also relatively straightforward. The main burdens in an actual implementation are careful bookkeeping of all the coefficients and a few tricks to prevent numerical instabilities.

Acknowledgments

The authors are grateful to A. Amarasingham, R. Kass and C. Shalizi for many helpful discussions and comments, and also to B. Anderson, N. Hatsopoulos and W. Truccolo for testing the method on their respective datasets during its conception. This work was supported in part by the National Science Foundation under Grants DMS-0240019 and IIS-0423031, by the National Institutes of Health under Grant NIMH-2RO1MH064537 and by the Office of Naval Research under Contract ONR-N000140610749.

References

Abeles, M., Bergman, H., Margalit, E., & Vaadia, E. (1993). Spatiotemporal firing patterns in the frontal cortex of behaving monkeys. *Journal of Neurophysiology*, *70*(4), 1629–1638.

Abeles, M., & Gerstein, G. (1988). Detecting spatiotemporal firing patterns among simultaneously recorded single neurons. *Journal of Neurophysiology*, *60*(3), 909–924.

Alonso, J., Usrey, W., & Reid, R. (1996). Precisely correlated firing in cells of the lateral geniculate nucleus. *Nature*, *383*, 815–819.

Amarasingham, A. (2004). *Statistical methods for the assessment of temporal structure in the activity of the nervous system*. Unpublished doctoral dissertation, Division of Applied Mathematics, Brown University.

Amarasingham, A., Chen, T.-L., Geman, S., Harrison, M., & Sheinberg, D. (2006). Spike count reliability and the poisson hypothesis. *Journal of Neuroscience*, *26*, 801–809.

Baker, S., & Lemon, R. (2000). Precise spatiotemporal repeating patterns in monkey primary and supplementary motor areas occur at chance levels. *Journal of Neurophysiology*, *84*, 1770–1780.

Brody, C. (1998). Slow variations in neuronal resting potentials can lead to artefactually fast cross-correlations in the spike trains. *Journal of Neurophysiology*, *80*, 3345–3351.

Chen, Y., Diaconis, P., Holmes, S., & Liu, J. (2005). Sequential monte carlo methods for statistical analysis of tables. *Journal of the American Statistical Association*, *100*(469), 109–120.

Dan, Y., Alonso, J., Usrey, W., & Reid, R. (1998). Coding of visual information by precisely correlated spikes in the lateral geniculate nucleus. *Nature Neuroscience*, *1*, 501–507.

Date, A., Bienenstock, E., & Geman, S. (1998, May). *On the temporal resolution of neural activity* (Tech. Rep.). Providence, RI: Division of Applied Mathematics, Brown University.

Davison, A., & Hinkley, D. (1997). *Bootstrap methods and their application*. Cambridge University Press.

Frey, B. (1998). *Graphical models for machine learning and digital communication*. Cambridge, MA: MIT Press.

Fujisawa, S., Amarasingham, A., Harrison, M., & Buzsáki, G. (2008). Behavior-dependent short-term assembly dynamics in the medial prefrontal cortex. *Nature Neuroscience*, (in press).

Geman, S., Amarasingham, A., Harrison, M., & Hatsopoulos, N. (2008). The statistical analysis of temporal resolution in the nervous system. (in preparation).

Geman, S., & Kochanek, K. (2001). Dynamic programming and the graphical representation of error-correcting codes. *IEEE Transaction on Information Theory*, *47*(2), 549–568.

Gerstein, G. (2004). Searching for significance in spatio-temporal firing patterns. *Acta Neurobiologiae Experimentalis*, *64*, 203–207.

Harrison, M. (2005). *Discovering compositional structures*. Unpublished doctoral dissertation, Brown University.

Ikegaya, Y., Aaron, G., Cossart, R., Aronov, D., Lampl, I., Ferster, D., et al. (2004). Synfire chains and cortical songs: Temporal modules of cortical activity. *Science*, *304*(5670), 559–564.

Jaynes, E. (1982). On the rationale of maximum-entropy methods. *Proceedings of the IEEE*, *70*(9), 939–952.

Kass, R., & Ventura, V. (2001). A Spike-Train Probability Model. *Neural Computation*, *13*, 1713–1720.

Lauritzen, S. (1988). *Extremal families and systems of sufficient statistics*. Spinger.

Lucena, B. (2002). *Dynamic programming, tree-width and computation on graphical models*. Unpublished doctoral dissertation, Division of Applied Mathematics, Brown University.

Nádasdy, Z., Hirase, H., Czurkó, A., Csicsvari, J., & Buzsáki, G. (1999). Replay and time compression of recurring spike sequences in the hippocampus. *Journal of Neuroscience*, *19*(21), 9497–9507.

Oram, M., Wiener, M., Lestienne, R., & Richmond, B. (1999). Stochastic nature of precisely timed spike patterns in visual system neuronal responses. *Journal of Neurophysiology*, *81*(6), 3021–3033.

Pauluis, Q., Baker, S., & Olivier, E. (2001). Precise burst synchrony in the superior colliculus of the awake cat during moving stimulus presentation. *Journal of Neuroscience*, *21*(2), 615–627.

Pazienti, A., Maldonado, P., Diesmann, M., & Grün, S. (2008). Effectiveness of systematic spike dithering depends on the precision of cortical synchronization. *Brain Research*, *1225*, 39–46.

Pipa, G., Riehle, A., & Grün, S. (2007). Validation of task-related excess of spike coincidences based on NeuroXidence. *Neurocomputing*, *70*, 2064–2068.

Shmiel, T., Drori, R., Shmiel, O., Ben-Shaul, Y., Nadasdy, Z., Shemesh, M., et al. (2005). Neurons of the cerebral cortex exhibit precise interspike timing in correspondence to behavior. *Proceedings of the National Academy of Sciences*, *102*, 18655–18657.

Shmiel, T., Drori, R., Shmiel, O., Ben-Shaul, Y., Nadasdy, Z., Shemesh, M., et al. (2006). Temporally Precise Cortical Firing Patterns Are Associated With Distinct Action Segments. *Journal of Neurophysiology*, *96*, 2645–2652.

Ventura, V. (2004). Testing for and estimating latency effects for poisson and non-poisson spike trains. *Neural Computation*, *16*, 2323–2349.

Ventura, V., Cai, C., & Kass, R. (2005a). Statistical assessment of time-varying dependency between two neurons. *Journal of Neurophysiology*, *94*, 2940–2947.

Ventura, V., Cai, C., & Kass, R. (2005b). Trial-to-trial variability and its effect on time-varying dependence between two neurons. *Journal of Neurophysiology*, *94*, 2928–2939.