On the Analysis of Multi-Channel Neural Spike Data

Authored by:

Lawrence Carin
Bo Chen
David E. Carlson

Abstract

Nonparametric Bayesian methods are developed for analysis of multi-channel spike-train data, with the feature learning and spike sorting performed jointly. The feature learning and sorting are performed simultaneously across all channels. Dictionary learning is implemented via the beta-Bernoulli process, with spike sorting performed via the dynamic hierarchical Dirichlet process (dHDP), with these two models coupled. The dHDP is augmented to eliminate refractoryperiod violations, it allows the ?appearance? and ?disappearance? of neurons over time, and it models smooth variation in the spike statistics.

1 Paper Body

The analysis of action potentials (?spikes?) from neural-recording devices is a problem of longstanding interest (see [21, 1, 16, 22, 8, 4, 6] and the references therein). In such research one is typically interested in clustering (sorting) the spikes, with the goal of linking a given cluster to a particular neuron. Such technology is of interest for brain-machine interfaces and for gaining insight into the properties of neural circuits [14]. In such research one typically (i) filters the raw sensor readings, (ii) performs thresholding to ?detect? the spikes, (iii) maps each detected spike to a feature vector, and (iv) then clusters the feature vectors [12]. Principal component analysis (PCA) is a popular choice [12] for feature mapping. After performing such sorting, one typically must (v) search for refractory-time violations [5], which occur when two or more spikes that are sufficiently proximate are improperly associated with the same cluster/neuron (which is impossible due to the refractory time delay required for the same neuron to re-emit a spike). Recent research has combined (iii) and (iv) within a single model [6], and methods have been developed recently to address (v) while performing (iv) [5]. Many of the early methods for spike sorting were based on classical clustering techniques [12] (e.g., K-means and GMMs, with a fixed num-

ber of mixtures), but recently Bayesian methods have been developed to account for more modeling sophistication. For example, in [5] the authors employed a modification to the Chinese restaurant formulation of the Dirichlet process (DP) [3] to automatically infer the number of clusters (neurons) present, allow statistical drift in the feature statistics, permit the ?appearance?/?disappearance? of neurons with time, and automatically account for refractorytime requirements within the clustering (not as a post-clustering step). However, [5] assumed that the spike features were provided via PCA in the first two or three principal components (PCs). In [6] feature learning and spike sorting were performed jointly via a mixture of factor analyzers (MFA) formulation. However, in [6] model selection was performed (for the number of features and number of neurons) and a maximum likelihood (ML) ?point? estimate was constituted for the model parameters; since a fixed number of clusters are inferred in [6], the model does not directly allow for the ?appearance?/?disappearance? of neurons, or for any temporal dependence to the spike statistics. There has been an increasing interest in developing neural devices with C ¿ 1 recording channels, each of which produces a separate electrical recording of neural activity. Recent research shows increased system performance with large C [18]. Almost all of the above research on spike sorting 1

300
Ground Truth
?300
PC?2
?100 ?300
?300 Unkown Neuron Known Neuron
(a)
100
100
?100
?500 ?500 ?700 ?700 ?500 ?300 ?100 100 300 500 PC?1
HDP?DL
GMM
100 PC?2
PC?2
100 ?100
300
300 K?means
PC?2
300
?400
?100 PC?1
200
500
?500 ?700
(b)
?100 ?300

?400
?100 PC?1
(c)
200
500
?500 ?700 ?500 ?300 ?100 100 300 500 PC?1
(d)

Figure 1: Comparison of spike sorting on real data. (a) Ground truth; (b) K-means clustering on the first 2

principal components; (c) GMM clustering with the first 2 principal components; (d) proposed method. We label using arrows examples K-means and the GMM miss, and that the proposed method properly sort.

has been performed on a single channel, or when multiple channels are present each is typically analyzed in isolation. In [5] C = 4 channels were considered, but it was assumed that a spike occurred at the same time (or nearly same time) across all channels, and the features from the four channels were concatenated, effectively reducing this again to a single-channel analysis. When C 1, the assumption that a given neuron is observed simultaneously on all channels is typically inappropriate, and in fact the diversity of neuron sensing across the device is desired, to enhance functionality [18]. This paper addresses the multi-channel neural-recording problem, under conditions for which concatenation may be inappropriate; the proposed model generalizes the DP formulation of [5], with a hierarchical DP (HDP) formulation [20]. In this formulation statistical strength is shared across the channels, without assuming that a given neuron is simultaneously viewed across all channels. Further, the model generalizes the HDP, via a dynamic HDP (dHDP) [17] to allow the ?appearance?/?disappearance? of neurons, while also allowing smooth changes in the statistics of the neurons. Further, we explicitly account for refractory times, as in [5]. We also perform joint feature learning and clustering, using a mixture of factor analyzers construction as in [6], but we do so in a fully Bayesian, multi-channel setting (additionally, [6] did not account for time-varying statistics). The learned factor loadings are found to be similar to wavelets, but they are matched to the properties of neuron spikes; this is in contrast to previous feature extraction on spikes [11] based on orthogonal wavelets, that are not necessarily matched to neuron properties. To give a preview of the results, providing a sense of the importance of feature learning (relative to mapping data into PCA features learned offline), in Figure 1 we show a comparison of clustering results on the first channel of d533101 data from hc-1 [7]. For all cases in Figure 1 the data are depicted in the first two PCs for visualization, but the proposed method in (d) learns the number of features and their composition, while simultaneously performing clustering. The results in (b) and (c) correspond respectively to widely employed K-means and GMM analysis, based on using two PCs (in these cases the analysis are employed in PCA space, as have been many more-advanced approaches [5]). From Figures 1 (b) and (c), we observe that both K-means and GMM work well, but due to the constrained feature space they incorrectly classify some spikes (marked by arrows). How-

ever, the proposed model, shown in Figure 1(d), which incorporates dictionary learning with spike sorting, infers an appropriate feature space (not shown) and more effectively clusters the neurons. The details of this model, including a multi-channel extension, are discussed in detail below.

2 2.1

Model Construction Dictionary learning

We initially assume that spike detection has been performed on all channels. Spike n 2 {1, . . . , Nc } (c) on channel c 2 {1, . . . , C} is a vector xn 2 RD , defined by D time samples for each spike, centered at the peak of the detected signal; there are Nc spikes on channel c. (c)

Data from spike n on channel c, xn , is represented in terms of a dictionary D 2 RD?K , where K is an upper bound on the number of needed dictionary elements (columns of D), and the model 2

(c)

infers the subset of dictionary elements needed to represent the data. Each xn is represented as (c) (c) (c) x(c) n = D? sn + ?n (c)

(c)

(1)

(c)

where ?(c) = diag(1 b1 , 2 b2 , . . . , K bK) is a diagonal matrix, with b = (b1 , . . . , bK)T 2 {0, 1}K . Defining dk as the kth column of D, and letting ID represent the D ? D identity matrix, the priors on the model parameters are dk ? N (0,

1 ID) , D

(c)

(c) k

? T N + (0,

c

1

),

1 ?(c) n ? N (0, ?c)

(2)

(c)

where ?c = diag(?1 , . . . , ?D), and T N + (?) represents the truncated (positive) normal distribution. Gamma priors (detailed when presenting results) are placed on c and on each of the ele(c) (c) ments of (?1 , . . . , ?D). For the binary vector b we impose the prior bk ? Bernoulli(?k), with ?k ? Beta(a/K, b(K 1)/K), implying that the number of non-zero components of b is drawn Binomial(K, a/(a + b(K 1))); this corresponds to Poisson(a/b) in the limit K ! 1. Parameters a and b are set to favor a sparse b. (c)

This model imposes that each xn is drawn from a linear subspace, defined by the columns of D with corresponding non-zero components in b; the same linear subspace is shared across all channels (c) c 2 {1, . . . , C}. However, the strength with which a column of D contributes toward xn depends on the channel c, as defined by ?(c) . Concerning ?(c) , rather than explicitly imposing a sparse (c) diagonal via b, we may also draw k ? T N + (0, ck1), with

shrinkage priors employed on the ck (i.e., with the ck drawn from a gamma prior that favors large ck ; which encourages many of the diagonal elements of ?(c) to be small, but typically not exactly zero). In tests, the model performed similarly when shrinkage priors were used on ?(c) relative to explicit imposition of sparseness via b; all results below are based on the latter construction. 2.2

Multi-Channel Dynamic hierarchical Dirichlet process (c)

We sort the spikes on the channels by clustering the {sn }, and in this sense feature design (learning {D?(c) }) and sorting are performed simultaneously. We first discuss how this may be performed via a hierarchical Dirichlet process (HDP) construction [20], and then extend this via a dynamic (c) HDP (dHDP) [17] considering multiple channels. In an HDP construction, the {sn } are modeled as being drawn (c) s(c) n ? f (?n) ,

?n(c) ? G(c) ,

G(c) ? DP(?c G) ,

G ? DP(?0 G0) (3) P1 where a Q draw from, for example, DP(?0 G0) may be constructed [19] as G = i=1 ?i ?i? , where ?i = Vi h¡i (1 Vh), Vi ? Beta(1, ?0), ?i? ? G0 , and ?i? is a unit point measure situated at ?i? . P1 (c) P1 (c) Each of the G(c) is therefore of the form G(c) = i=1 ?i ?i? , with i=1 ?i = 1 and with the {?i? } shared across all G(c) , but with channel-dependent (c-dependent) probability of using elements of {?i? }. Gamma hyperpriors are employed for {?c } and ?0 . In the context of the model developed in Section 2.1, the density function f (?) corresponds to a Gaussian, and parameters ?i? = (??i , ?i) correspond to means and precision matrices, with G0 a normal-Wishart distribution. The proposed model may be viewed as an mixture of factor analyzers (MFA) [6] applied to each channel, with the addition of sharing of statistical strength across the C channels via the HDP. Sharing is manifested in two forms: (i) via the shared linear subspace defined by the columns of D, and (ii) via hierarchical (c) clustering via HDP of the relative weightings {sn }. In tests, the use of channel-dependent ?(c) was found critical to modeling success, as compared to employing a single ? shared across all channels. P1 (c) The above HDP construction assumes that G(c) = i=1 ?i ?i? is time-independent, implying that (c)

(c)

the probability ?i that xn is drawn from f (?i?) is time invariant. There are two ways this assumption may be violated. First, neuron refractory time implies a minimum delay between consecutive firing of the same neuron; this effect is addressed in a relatively straightforward manner discussed in Section 2.3. The second issue corresponds to the ?appearance? or ?disappearance? of neurons (c) [5]; the former would be characterized by an increase in the value of a component of ?i , while the (c) latter would be characterized by one of the components of ?i going to zero (or near zero). It is 3

desirable to augment the model to address these objectives. We achieve this by application of the dHDP construction developed in [17]. As in [5], we divide the time axis into contiguous, non-overlapping temporal blocks, where block j corresponds to spikes observed between times ?j 1 and ?j ; we consider J such blocks, indexed (c) j = 1, . . . , J. The spikes on channel c within block j

are denoted {xjn }n=1,Ncj , where Ncj represents the number of spikes within block j on channel c. In the dHDP construction we have (c)
(c)
(c)
(c)
(c)
(c)
(c)
sjn ? f (?jn) , ?jn ? wj Gj + (1 (c)
(c)
where w1
(c)
Gj ? DP(?jc G) , G ? DP(?0 G0) , wj (c)
= 1 for all c. The expression wj
(c)
Gj , while with probability 1
(c)
(c)
wj)Gj
(4)
1
? Beta(c, d)
(5) (c)
controls the probability that ?jn is drawn from (c)
(c)
wj parameter ?jn is drawn from Gj
1 . The cumulative mixture (c) (c) model + (1 wj)Gj 1 supports arbitrary levels of variation from block to block in (c) the spike-train analysis: If wj is small the probability of observing a particular type of neuron (c) doesn?t change significantly from block j 1 to j, while if wj ? 1 the mixture probabilities can (c) change quickly (e.g., due to the ?appearance?/?disappearance? of a neuron); for wj in between (c) (c) w j Gj

these extremes, the probability of observing a particular neuron changes slowly/smoothly with consecutive blocks. The model therefore allows a significant degree of flexibility and adaptivity to changes in neuron statistics. 2.3

Accounting for refractory time and drift

To demonstrate how one may explicitly account for refractory-time conditions within the model, (c) (c) assume the time difference between spikes xj? and xj? 0 is less than the refractory time, while all other spikes have temporal separations greater than the refractory time; we consider two spikes of this type for notational convenience, but the basic formulation below may be readily extended to (c) (c) more than two spikes of this type. We wish to impose that xj? and xj? 0 should not be associated (c)

with the same cluster/neuron, but otherwise the model is unchanged. Hence, for n 6= ? 0 , ?jn ? (c) ? (c) = w(c) G(c) + (1 w(c))G(c) as in (4). Assuming G ? (c) = P1 ? ? G j j j j j 1 j i=1 ?ji ?i , we have the new conditional generative construction (c)

(c)
?j? 0 —?j? ?
1 X i=1
(c)
? ?ji [1
P1
l=1
(c)
(c)
I(?j? = ?i?)]
? ?jl [1
(c)
I(?j? = ?l?)]
(6)
?i?

where I(?) is the indicator function (it is equal to one if the argument is true, and it is zero otherwise). (c) (c) This construction imposes that ?j? 0 6= ?j? , but otherwise the model preserves that the elements of ? (c) . Note that the time associated with {?i? } are drawn with a relative probability consistent with G j a given spike is assumed known after detection (i.e., it is a covariate), and therefore it is known a priori for which spikes the above adjustments must be made to the model. (c)

The representation in (6) constitutes a proper generative construction for {?jn } in the presence of spikes that co-occur within the refractory time, but it complicates inference. Specifically, recall that P1 (c) (c) (c) (c) Q (c) (c) Gj = i=1 ?ji ?i? , with ?ji = Uji Ujh), with Uji ? Beta(1, ?jc). In the original h¡i (1 construction, (4) and (5), in which refractory-time violations are not account for, the Gibbs update (c) (c) equations for {Uji } are analytic, due to model conjugacy. However, conjugacy for {Uji } is lost with (6), and therefore a Metropolis-Hastings (MH) step is required to draw these random variables with an Markov Chain Monte Carlo (MCMC) analysis. This added complexity is often unnecessary, since the number of refractory-time events is typically very small relative to the total number of spikes that must be sorted. Hence, we have successfully implemented the following approximation (c) (c) to the above construction. While the ?j? 0 is drawn as in (6), assigning ?j? 0 to one of the members of (c)

{?i? } while avoiding a refractory-time violation, the update equations for {Uji } are executed as they 4

would be in (4) and (5), without an MH step. In other words, a construction like (6) is used to assign (c) elements of {?i? } to spikes, but after this step the update equations for {Uji } are implemented as in the original (conjugate) model. This is essentially the same approach employed in [5], but now in terms of a ?stick-breaking? rather than CRP construction of the DP (here an dHDP), and like in [5] we have found this to yield encouraging results (e.g., no refractory-time violations, and sorting in good agreement with ?truth? when available). Finally, in [5] the authors considered a ?drift? in the atoms associated with the

DP, which here would correspond to a drift in the atoms associated with our dHDP. In this construction, rather that drawing the ?i? ? G0 once as in (5), one may draw ?i? ? G0 for the first block of time, and then a simple Gaussian auto-regressive model is employed to allow the {?i? } drift a small amount ? ? between consecutive blocks. Specifically, if {?ji } represents the atoms for block j, then ?j+1,i ? 1 ? N (?ji , 0), where it is imposed that 0 is large. We examined this within the context of the model proposed here, and for the data considered in Section 4 this added modeling complexity did not change the results significantly, and therefore we did not consider this added complexity when ? presenting results. This observed un-importance in imposing drift in {?ji } is likely due to the fact (c)

(c)

? that we draw sjn ? f (?jn) with a Gaussian f (?), and therefore even if the {?ji } do not change across data blocks, the model allows drift via variations in the draws from the Gaussian (effecting the inferred variance thereof).

3

Inference and Computations

For online sorting of spikes, a Chinese restaurant process (CRP) formulation like that in [5] is desirable. The proposed model may be implemented as a generalization of the CRP, as the general form of the model in Section 2.2 is independent of the specific way inference is performed. In a CRP construction, the Chinese restaurant franchise (CRF) model [20] is invoked, and the model in Section 2.2 yields a dynamic CRF (dCRF), where each franchise is associated with a particular channel. The hierarchical form of the dCRF, including the dictionary-learning component of Section 2.1, is fully conjugate, and may therefore be implemented via a Gibbs sampler. As hinted by the construction in (6), we here employ a stick-breaking construction of the model, analogous to the form of inference employed in [17]. We employ a retrospective stick-breaking (c) (c) construction [15] for Gj and G [10], such that the number of terms used to construct G and Gj is unbounded and adapts to the data. Using this construction the model is able to adapt to the number of neurons present, adding and deleting clusters as needed. In this sense the stick-breaking construction may also be considered for online implementations. Further, in this model the parameter Gibbs sampling follows an online-style inference, since the data blocks come in sequentially and the parameters for each block only depend on the previous one or a new component. Therefore, while online implementation is not our principal focus here, it may be executed with the proposed model. We also implemented a CRF implementation, for which there is no truncation. Both inference methods (stick-breaking and CRF implementations) gave very similar results. Although this paper is not principally focused on online implementations, in the context of such, one may also consider online and evolving learning of the dictionary D [13]. There is recent research on online dictionary learning, which may be adapted here, using recent extensions via Bayesian formalisms [9]; this would, for example, allow the linear subspace in which the spike shapes reside to adapt/change with data block.

4

Example Results

For these experiments we used a truncation level of K = 60 dictionary elements. In dictionary (c) learning, the hyperparameters in the gamma priors of c and ?p were set as a c = 10 6 and 6 5 b c = 10 , a?(c) = 0.1 and b?(c) = 10 . In the HDP, we set Ga(1,1) for ?0 and ?c . In dHDP, p p we set Ga(1,1) for ?0 and ?jc . Meanwhile, in order to encourage the groups to be shared, we set QC QJ 1 (c) the prior c=1 j=1 Beta(wj ; aw , bw) with aw = 0.1 and bw = 1. These parameters have not been optimized, and many analogous settings yield similar results. We used 5000 burn-in samples and 5000 collection samples in the Gibbs sampler, and we choose the collection sample with the 5

Table 1: Summary of results on simulated data. Methods K-means GMM K-means with 2 PCs GMM with 2 PCs DP-DL HDP-DL

Channel 1 96.00% 84.33% 96.8% 96.83% 97.00% 97.39%

Channel 2 96.02% 94.25% 96.9% 96.98% 96.92% 97.08%

Channel 3 95.77% 91.75% 96.50% 96.92% 97.08% 97.08%

Average 95.93% 90.11% 96.81% 96.91% 97.00% 97.18%

maximum likelihood when presenting below example clusterings. For the K-means and GMM, we set the cluster level to 3 in the simulated data and to 2 clusters in the real data (see below).

4.1

Simulated Data

In neural spike trains it is very difficult to get ground truth information, so for testing and verification we initially consider simulated data with known ground truth. To generate data we draw from the (c) (c) model xn ? N (D(diag((c)))sn , 0.01ID). We define D 2 RD?K and (c) 2 RK , which constructs our data from K = 2 primary dictionary elements of length D = 40 in C = 3 channels. These dictionary elements are randomly drawn. We vary (c) from channel to channel, and for each P3 (c) (c) (c) spike, we generate the feature strength according to p(sn) = i=1 ?i N (sn —?i , 0.5IK) with ? = [1/3 1/3 1/3], which means that there are three neurons across all the channels. We defined (c) ?i 2 RK as the mean in the feature space for each neuron and shift the neuron mean from channel to channel. For results we associate each cluster with a neuron and determine the percentage of spikes in their correct cluster. The results are shown in Table 1. The combined Dirichlet process and dictionary learning (DP-DL) give similar results to the GMM with 2 principal components (PCs). Because the DP-DL learns the appropriate number of clusters (three) and dictionary elements (two), these models are expected to perform similarly, except that the DP-DL does not require knowledge of the number of dictionary elements and clusters a priori. The HDP-DL is allowed to share global clusters and dictionary elements between channels, which improves results as well. In Figure 2 the sample posteriors show that we peak at the true values of 3 used ?global? clusters (at the top layer of the HDP) and 2 used dictionary elements. Additionally, the HDP shares cluster information between channels, which helps the cluster accuracy. In fact, the spikes at the same time will typically be drawn from the same global cluster despite having independent local clusters as seen in the global cluster from each channel in Figure 2(b). Thus, we can determine

a global spike at each time point as well as on each channel.

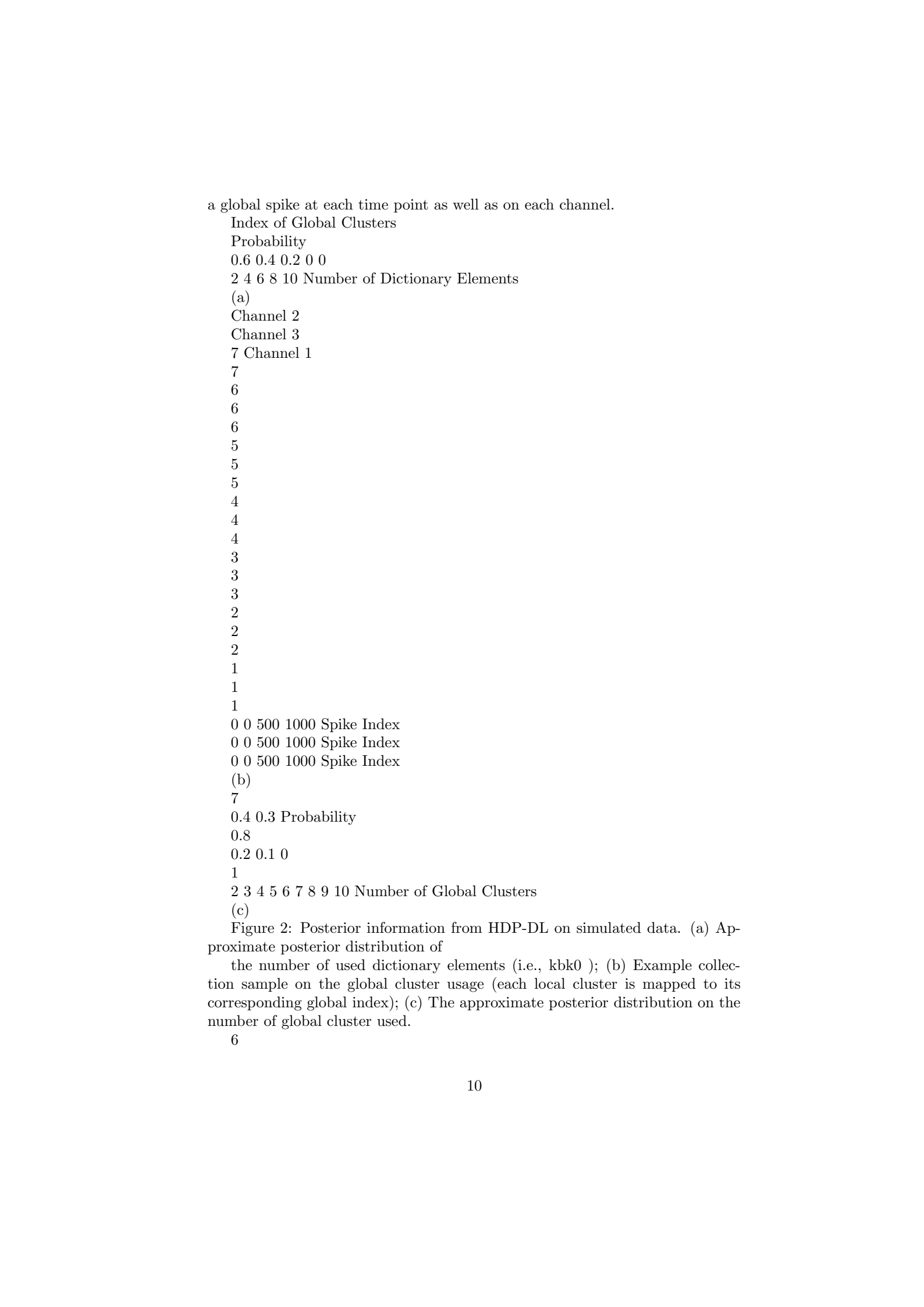

Figure 2: Posterior information from HDP-DL on simulated data. (a) Approximate posterior distribution of

the number of used dictionary elements (i.e., kbk0); (b) Example collection sample on the global cluster usage (each local cluster is mapped to its corresponding global index); (c) The approximate posterior distribution on the number of global cluster used.

Table 2: Results from testing on d533101 data [7]. KFM represent Kalman Filter Mixture method [2]. Methods K-means GMM K-means with 2 PCs GMM with 2 PCs KFM with 2 PCs DP with 2 PCs HDP with 2 PCs DP-DL HDP-DL 4.2

Channel 1 86.67% 87.43% 87.47% 89.00% 91.00% 89.04% 90.36% 92.29% 93.38%

Channel 2 88.04% 90.06% 88.16% 89.04% 89.2% 89.00% 90.00% 92.38% 93.18%

Channel 3 89.20% 86.75% 89.40% 87.43% 86.35% 87.43% 90.00% 89.52% 93.05%

Channel 4 88.4% 85.43% 88.72% 90.7% 86.87% 86.79% 87.79% 92.45% 92.61%

Average 88.08% 87.42% 88.44% 89.04% 88.36% 88.07% 89.54% 91.89% 93.05%

Real Data with Partial Ground Truth

We use the publicly available dataset1 hc-1. These data consist of both extracellular recordings and an intracellular recording from a nearby neuron in the hippocampus of a anesthetized rat [7]. Intracellular recordings give clean signals on a spike train from a specific neuron, giving accurate spike times for that neuron. Thus, if we detect a spike in a nearby extracellular recording within a close time period (¡.5ms) to an intracellular spike, we assume that the spike detected in the extracellular recording corresponds to the known neuron?s spikes. This allows us to know partial ground truth, and allows us to test on methods compared to the known information. For the accuracy analysis, we determine one cluster that corresponds to the known neuron. Then we consider a spike to be correctly sorted if it is a known spike and is in the known cluster or if it is an unknown spike in the unknown cluster. In order to give a fair comparison of methods, we first considered the widely used data d533101 and used the same preprocessing from [2]. This data consists of a 4-channel extracellular recordings and 1-channel intracellular recording. We used 2491 detected spikes and 786 of those spikes came from the known neuron. The results are shown in Figure 2. The results show that learning the feature space instead of using the top 2 PCA components increases sorting accuracy. This phenomenon can be seen in Figure 1, where it is impossible to accurately resolve the clusters in the space based on the 2 principal components, through either K-means or GMM. Thus, by jointly learning the suitable feature space and clustering, we are able to separate the unknown and known neurons clusters more accurately. In the HDP model the advantage is clear in the global accuracy as we achieve 89.54% when using 2 PCs and 93.05% when using dictionary learning. In addition to learning the appropriate feature space, HDP-DL and DP-DL can infer the appropriate number of clusters, allowing the data to define the number of neurons. The posterior distribution on the number of global clusters and number of factors (dictionary elements) used is shown in Figure 3(a) and 3(b), along with the most used elements of the learned dictionary in Figure 3(c). The dictionary elements show shapes similar to both neuron spikes in Figure 3(d) and wavelets. The spiky nature of the learned dictionary can give factors similar to those use in the discrete wavelet transform cluster in [11], which choose to use the Daubechies wavelet for its spiky nature (but here,

rather than a priori selecting an orthogonal wavelet basis, we learn a dictionary that is typically not orthogonal, but is wavelet-like). Next we used the d561102 data from hc-1, which consists of 4 extracellular recording and 1 intracellular recording. To do spike detection we high-pass filtered the data from 300 Hz and detected spikes when the voltage level passed a positive or negative threshold, as in [2]. We choose this data the known neuron displays dynamic properties by showing periods of activity and inactivity. The intracellular recording in Figure 4(a) shows the known neuron is active for only a brief section of the recorded signal, and is then inactive for the rest of the signal. The nonstationarity passes along to the extracellular spike train and the detect spikes. We used the first 930 detected spikes, which included 202 spikes from the known cluster. In order to model the dynamic properties, we binned the data into 31 subgroups of 30 spikes to use with our multichannel dynamic HDP. The results are shown in 1

available from http://crcns.org/data-sets/hc/hc-1

7

0.4

0.5 0.4

Probability

Probability

0.3 0.2 0.1 0 0

0.3 0.2 0.1

1

2 3 4 5 6 7 8 9 10 Number of Global Clusters

0 20

25 30 35 40 45 Number of Dictionary Elements

(a)

(b)

(c)

(d)

Figure 3: Results from HDP-DL on d533101 data. (a) approximate posterior probability on the number of global clusters (across all channels); (b) approximate posterior distribution on the number of dictionary elements; (c) six most used dictionary elements; (d) examples of typical spikes from the data.

Methods K-means GMM K-means with 2 PCs GMM with 2 PCs DP-DL HDP-DL MdHDP-DL

Table 3: Results for d566102 data [7]. Channel 1 Channel 2 Channel 3 61.82% 78.77% 83.59% 73.85% 78.66% 74.18% 61.82% 78.77% 84.79% 75.82% 78.77% 75.71% 68.49% 81.73% 84.57% 74.40% 82.49% 85.34% 76.04% 84.79% 87.53%

Channel 4 89.39% 76.59% 89.39% 88.73% 88.73% 88.40% 90.48%

Average 78.39% 75.82% 78.69% 79.76% 80.88% 82.66% 84.71%

Recorded Signal

2000

1500

1000

500 10

20 Time, s
30
40
(a)
30
1 0.8
25
3
11
10
5
0.6
20
0.4 0.2
15
0 0 0.8
10
10 0
5
10 0
10 0
21
18
13
5
5
10
30
0.6
5 0 0
5
0.4 0.2
200
400 600 Spike Index
800 950
0 0
5
10 0
(b)
5
10 0
(c)
5
10 0
5
10

Probability of Changing

Index of Mixture Distribution

Table 3. The model adapts to the nonstationary spike dynamics by learning the parameters to model (c) dynamic properties at block 11 (w11 ? 1, indicating that the dHDP has detected a change in the characteristics of the spikes), where the known neuron goes inactive. Thus, the model is more likely to draw new local clusters at this point, reflecting the nonstationary data. Additionally, in Figure 4(c) the global cluster usage shows a dramatic change at time block 11, where a cluster in the model goes inactive at the same time the known neuron is inactive. Because the dynamic model can map these dynamic properties, the results improve while using this model. Additionally, we obtain a global accuracy (across all channels) of 82.66% using the HDP-DL and an global accuracy of 84.71% using the multichannel dynamic HDP-DL (MdHDP-DL). We also tried the KFM on these data, but we were unable to get satisfactory results with it. Additionally, we also calculated the true positive and false positive number to evaluate each method, but due to the limited space, those results were put in Supplementary Material. 1 The probability of introducing a new component for the 11th block

0.8 0.6 0.4 0.2 0

10

20 Block Index

30

(d)

Figure 4: Results of the multichannel dHDP on d561102. (a) first 40 seconds of the intracellular recording

of d561102; (b) local cluster usage by each spike in the d561102 data in channel 4; (c) global cluster usage at (c) different time blocks for the data d561102; (d) sharing weight wj at each time blocks in the fourth channel. The spike in 11 occurs when the known neuron goes inactive.

5

Conclusions

We have presented a new method for performing multi-channel spike sorting, in which the underlying features (dictionary elements) and sorting are performed jointly, while also allowing timeevolving variation in the spike statistics. The model adaptively learns dictionary elements of a wavelet-like nature (but not orthogonal), with characteristics like the shape of the spikes. Encouraging results have been presented on simulated and real data sets.The authors would like to thank A. Calabrese for providing the KFM codes and processed d533101 data. 8

Acknowledgement The research reported here was supported under the DARPA HIST program.

2 References

[1] A. Bar-Hillel, A. Spiro, and E. Stark. Spike sorting: Bayesian clustering of non-stationary data. J. Neuroscience Methods, 2006. [2] A. Calabrese and L. Paniski. Kalman filter mixture model for spike sorting of non-stationary data. J. Neuroscience Methods, 2010. [3] T. S. Ferguson. A Bayesian analysis of some nonparametric problems. The Annals of Statistics, 1973. [4] Y. Gao, M. J. Black, E. Bienenstock, S. Shoham, and J. P. Donoghue. Probabilistic inference of arm motion from neural activity in motor cortex. Proc. Advances in NIPS, 2002. [5] J. Gasthaus, F. Wood, D. Gorur, and Y.W. Teh. Dependent Dirichlet process spike sorting. In Advances in Neural Information Processing Systems, 2009. [6] D. Gorur, C. Rasmussen, A. Tolias, F. Sinz, and N. Logothetis. Modelling spikes with mixtures of factor analysers. Pattern Recognition, 2004. [7] D. A. Henze, Z. Borhegyi, J. Csicsvari, A. Mamiya, K. D. Harris, and G. Buzsaki. Intracellular feautures predicted by extracellular recordings in the hippocampus in vivo. J. Neurophysiology, 2010. [8] J.A. Herbst, S. Gammeter, D. Ferrero, and R.H.R. Hahnloser. Spike sorting with hidden Markov models. J. Neuroscience Methods, 2008. [9] M.D. Hoffman, D.M. Blei, and F. Bach. Online learning for latent Dirichlet allocation. Proc. NIPS, 2010. [10] H. Ishwaran and L.F. James. Gibbs sampling methods for stick-breaking priors. J. Am. Stat. Ass., 2001. [11] J. C. Letelier and P. P. Weber. Spike sorting based on discrete wavelet transform coefficients. J. Neuroscience Methods, 2000. [12] M. S. Lewicki. A review of methods for spike sorting: the detection and classification of neural action potentials. Network: Computation in Neural Systems, 1998. [13] J. Mairal, F. Bach, J. Ponce, and G. Sapiro. Online learning for matrix factorization and sparse coding. J. Machine Learning Research, 2010. [14] M.A. Nicolelis. Brain-machine interfaces to restore motor function and probe neural circuits. Nature reviews: Neuroscience, 2003. [15] O. Papaspiliopoulos and G. O. Roberts. Retrospective Markov Chain Monte Carlo methods for Dirichlet process hierarchiacal models. Biometrika, 2008. [16] C. Pouzat, M. Delescluse, P. Viot, and J. Diebolt. Improved spike-sorting by modeling firing statistics and burst-dependent spike amplitude attenuation: A Markov Chain Monte Carlo approach. J. Neurophysiology, 2004. [17] L. Ren, D. B. Dunson, and L. Carin. The dynamic hierarchical dirichlet process. International Conference on Machine Learning, 2008. [18] G. Santhanam, S.I. Ryu, B.M. Yu, A. Afshar, and K.V. Shenoy. A high-performance braincomputer interface. Nature, 2006. [19] J. Sethuraman. A constructive definition of dirichlet priors. Statistica Sinica, 4:639?650, 1994. [20] Y. W. Teh, M. I. Jordan, M. J. Beal, and D. M. Blei. Hierarchical dirichlet processes. J. Am. Stat. Ass., 2005. [21] F. Wood, S. Roth, and M. J. Black. Modeling neural population spiking activity with Gibbs distributions. Proc. Advances in Neural Information Processing Systems, 2005. [22] W. Wu, M. J. Black, Y. Gao, E. Bienenstock, M. Serruya, A. Shaikhouni, and J. P. Donoghue. Neural decoding of cursor motion using a Kalman filter. Proc. Advances in NIPS, 2003. 9