Temporal Segmentation with Bayesian Binning[*]

Dominik M. Endres[†] Martin A. Giese[‡]
Section for Computational Sensomotorics, Department of Cognitive Neurology, University Clinic Tübingen,
Hertie Institute for Clinical Brain Research and Center for Integrative Neuroscience
Frondsbergstrasse 23, D-72070 Tübingen

Abstract

Bayesian Binning (BB) is an exact inference technique which was originally developed for applications in Computational Neuroscience, e.g. modeling spike count distributions [1] or estimating peri-stimulus time histograms (PSTH) [2]. BB encodes a (conditional) probability distribution (or density) which is piecewise constant in the domain of interest. This suggests that BB might be useful for retrospective temporal segmentation tasks, too. We illustrate the potential usefulness of BB for temporal segmentation on two examples. First, we segment neural spike train data, demonstrating that BB is able to locate change points in the PSTH correctly. Second, we employ BB for (human) action sequence segmentation. We show that BB accurately identifies the transition points in the action sequence (e.g. a change from 'walking' to 'jumping').

1 Introduction

Retrospective temporal segmentation, i.e. change point detection after the complete (and possibly repeated) observation of a time sequence is a traditional way of analyzing neurophysiological spike train data. A change point of particular interest in Neuroscience is the *latency* [3], which signals the beginning of a neural response to an experimentally controlled event, such as a stimulus onset. While the definition of latency varies substantially between authors, most definitions are based on the temporal location of a rapid change of the firing rate of the neuron. Thus, to determine latency (or any other rapid change point in the neural response), it is important to obtain a reliable estimate of the firing rate, particularly given the considerable amount of noise and uncertainty usually present in neurophysiological measurements. Recently, there have been several advances in the field of rate estimation, e.g. [4, 5, 2, 6, 7]. Of these methods, Bayesian binning (BB) [2] is well suited for retrospective temporal segmentation, since it estimates a peri-stimulus time histogram (PSTH), whose bin boundaries comprise the desired change points. Moreover, because it avails its user of the full posterior over the model parameters, it can be employed to compute posterior distribution over change points, rather than just their expectations. We will demonstrate this in section 2.

BB is not limited to binning neural signals. The main prerequisites for the application of BB are: the variable to be binned must be totally ordered (e.g. points in time), and there must be a notion of constancy definable for each bin. In the case of neural signals, the constant quantity is the firing rate. If we instead tried to segment action sequences, we might be interested in finding time bins within which the type of action is constant. For example, imagine a person walking into the center of our view, then jumping about and finally waving us goodbye. We could then segment this sequence into three bins, and label the bins 'walk', 'jump' and 'wave'. We will show how to employ BB to this end in section 3.

[*] To be presented at the NIPS 2009 workshop on temporal segmentation

[†] dominik.endres@klinikum.uni-tuebingen.de

[‡] martin.giese@uni-tuebingen.de

This contribution is mainly intended to illustrate the usefulness of BB on temporal segmentation tasks via examples. For more rigorous validations of the method, see [1, 2, 8].

2 Bayesian Binning of neural spike trains

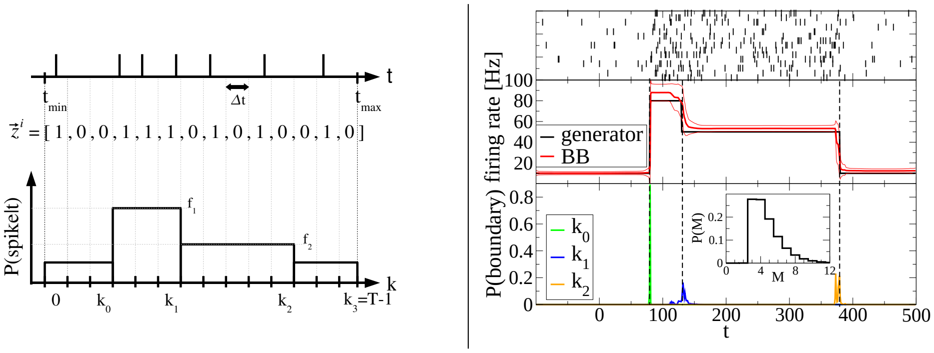

Figure 1: **Left** (adapted from [2]): *top*: A spike train, recorded between times t_{min} and t_{max} is represented by a binary vector $\vec{z}^i$. *Bottom*: The time span between t_{min} and t_{max} is discretized into T intervals of duration $\Delta t = (t_{max} - t_{min})/T$, such that interval k lasts from $k \times \Delta t + t_{min}$ to $(k+1) \times \Delta t + t_{min}$. Δt is chosen such that at most one spike is observed per Δt interval for any given spike train. Then, we model the firing probabilities $P(spike|t)$ by $M + 1 = 4$ contiguous, non-overlapping bins (M is the number of bin boundaries inside the time span $[t_{min}, t_{max}]$), having inclusive upper boundaries k_m and $P(spike|t \in (t_{min} + \Delta t(k_{m-1} + 1), t_{min} + \Delta t\ k_m]) = f_m$. **Right**: *top*: Rastergrams of excitatory responses recorded from a simulated neuron. Each tick mark represents a spike. *Middle*: PSTH from Bayesian binning (BB). The thick red line represents the predictive firing rate, the thin red lines show the predictive firing rate ±1 standard deviation. Models with $3 \leq M \leq 5$ were included, covering $\geq 96\%$ of the posterior over the number of bin boundaries M. The segmentation of the neural response into 4 pieces is clearly visible. The black line shows the firing rate from which the spike trains in the above rastergrams were generated. The inferred PSTH follows the generator very closely. *Bottom*: marginal posterior distributions of the bin boundaries k_i for a model with $M = 3$, which is the maximum of the M-posterior (shown in the inset). The dashed vertical lines represent the change points of the spike generator. The marginal bin boundary posteriors exhibit narrow peaks close to the generating change points, i.e. the spike trains have been correctly segmented.

We model a PSTH on $[t_{min}, t_{max}]$, discretized into T contiguous intervals of duration $\Delta t = (t_{max} - t_{min})/T$ (see fig.1, left). Spike train i can then be represented by a binary vector $\vec{z}^i$ of dimensionality T. The PSTH is modelled using $M + 1$ contiguous, non-overlapping bins having inclusive upper boundaries k_m. The firing probability $P(spike|t \in (t_{min} + \Delta t(k_{m-1} + 1), t_{min} + \Delta t\ k_m]) = f_m$ is constant within each bin. Note that the firing rate $= \frac{f_m}{\Delta t}$. M is the number of bin boundaries within $[t_{min}, t_{max}]$. The probability of a spike train $\vec{z}^i$ of independent spikes/gaps is then

$$P(\vec{z}^i|\{f_m\}, \{k_m\}, M) = \prod_{m=0}^{M} f_m^{s(\vec{z}^i,m)}(1 - f_m)^{g(\vec{z}^i,m)} \tag{1}$$

where $s(\vec{z}^i, m)$ is the number of spikes and $g(\vec{z}^i, m)$ is the number of non-spikes, or gaps in spiketrain $\vec{z}^i$ in bin m, i.e. between intervals $k_{m-1} + 1$ and k_m (both inclusive). This implies $s(\vec{z}^i, m) + g(\vec{z}^i, m) = T$. In other words, we model spiketrains using an inhomogeneous Bernoulli process with piecewise constant probabilities. If we observe a multiset $\{\vec{z}^i\}$, we compute the joint likelihood assuming independence between spiketrains given $\{f_m\}$ and $\{k_m\}$.

We make a non-informative prior assumption for $p(\{f_m\}, \{k_m\})$, namely

$$p(\{f_m\}, \{k_m\}|M) = p(\{f_m\}|M)P(\{k_m\}|M). \tag{2}$$

i.e. we have no a priori preferences for the firing rates based on the bin boundary positions (we assume that the bin boundary positions are independent of the firing rates). Note that the prior of the

f_m, being continuous model parameters, is a density. Given the form of eqn.(1) and the constraint $f_m \in [0,1]$, it is natural to choose a conjugate prior

$$p(\{f_m\}|M) = \prod_{m=0}^{M} \mathrm{B}(f_m; \sigma_m, \gamma_m). \tag{3}$$

where the Beta density $\mathrm{B}(f_m; \sigma_m, \gamma_m)$ is defined in the usual way (see e.g. [9]).

There are only finitely many configurations of the k_m. Assuming we have no preferences for any of them, the prior for the bin boundaries becomes

$$P(\{k_m\}|M) = \frac{1}{\begin{pmatrix} T-1 \\ M \end{pmatrix}}. \tag{4}$$

where the denominator is just the number of possibilities in which M ordered bin boundaries can be distributed across $T-1$ places (bin boundary M always occupies position $T-1$, see fig.1 , hence there are only $T-1$ positions left).

Having thus completed the model specification, we can now proceed in the standard Bayesian fashion to compute the posterior distribution/density of the model parameters.

$$P(\{f_m\}, \{k_m\}|\{\vec{z}^i\}, M) = \frac{P(\{\vec{z}^i\}|\{f_m\}, \{k_m\}, M) p(\{f_m\}|M) P(\{k_m\}|M)}{P(\{\vec{z}^i\}|M)} \tag{5}$$

The main computational effort is determined by the evaluation of the normalization constant $P(\{\vec{z}^i\}|M)$ of this posterior, which requires us to integrate/sum over all model parameters. Integrating over the firing probabilities $\{f_m\}$ is straightforward, since we chose a conjugate prior (eqn.(3)). The reasoning behind the prior over the bin boundaries (eqn.(4)) might suggest a complexity of $\mathcal{O}(T^M)$ for the summation over all configurations. However, as demonstrated in [2], we can devise an algorithm which accomplishes this sum in $\mathcal{O}(MT^2)$, i.e. exact inference is feasible. By a similar algorithm, exact posterior expectations of certain functions of the model parameters can also be computed [1].

Fig.1, right, shows results on an artificially created set of 30 spike trains (top panel). This number of repeated trials is typically obtainable from single-cell recordings in high level visual cortex. We chose to present results on artificial data here, since the ground truth, i.e. the generator of the data is then known for comparison. The spike rate of the generator, which is comprised of 4 bins (3 bin boundaries), is shown in the middle panel (black line). Superimposed is the predictive, or expected firing rate inferred with BB (red line) from the spike trains shown in the top panel. The thin red lines represent the predictive rate ± one posterior standard deviation. The predictive rate closely follows the generator. Furthermore, BB has discovered the complexity of the generator, as measured by the number of bin boundaries M: the inset in the bottom panel shows the posterior distribution of M, which peaks at $M = 3$. It therefore comes as no surprise that the marginal posteriors of the bin boundaries k_0, k_1 and k_2, shown in the bottom panel, are closely concentrated around the bin boundaries of the generator (dashed vertical lines).

bin boundary	location in generator	BB posterior expectation ± one standard deviation
k_0	80 ms	79.9 ± 1.9 ms
k_1	130 ms	131.9 ± 9.6 ms
k_2	380 ms	376.7 ± 3.3 ms

Table 1: Bin boundary positions of the generator in fig.1, right and expected bin boundary positions ± one posterior standard deviation as inferred with Bayesian binning (BB).The expected boundaries are within one standard deviation of the generator's boundaries.

This observation is confirmed by the comparison shown in table 1: all bin boundary expectations are within one standard deviation of the generator's boundaries. In other words, those expectations can be employed to segment the spiketrain temporally.

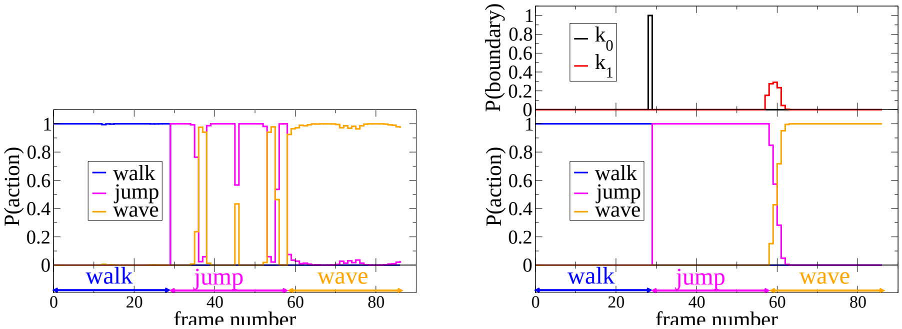

Figure 2: Action segmentation results. *Bottom, left and right*: the sequence to be segmented contained 3 actions: walk, jump, wave. In the sequence, each action was presented for 28 frames before switching to the next action. *Middle, left*: frame-based action classification results. 'Jump' and 'wave' are hard to discriminate. We computed the expected action identity (e.g. walk,jump or wave) at every point in the sequence. *Middle, right*: action classification with BB. The correct action is identified with near certainty. *Top right*: marginal posterior distribution of the bin boundaries k_0 and k_1 for a model with $M = 2$ bin boundaries. $M = 2$ was the maximum of the M-posterior for the data on which this graph is based.

3 Segmenting action sequences

In our second example, we show how to find change points between different types of actions in video sequences. By 'types of action' we mean e.g. walking, jumping, waving etc.. To employ BB to this end, we assume that the type of action is constant within contiguous, non-overlapping time intervals. As a consequence, we model the video data by a sequence of generative classifiers. There is one classifier per temporal bin. Each classifier is trained on the action types of interest. Since the classifiers are generative, their marginal probability of the data (i.e. the video sequence within their respective time bin) can serve as a replacement for the likelihood (eqn.(1)). A particularly simple example of such a generative classifier is a naive Bayes classifier, which we will use in the following. Furthermore, we substitute the prior over the classifier's parameters for the firing rate prior (eqn.(3)). The prior over the bin boundaries (eqn.(4)) doesn't need to be altered, since we still have no *a priori* preference for the location of the bin boundaries, other than requiring the bins to be non-overlapping.

We obtained video sequences of different actors performing various actions through [10]. We preprocessed the videos by computing the optic flow with the neurodynamical model for action segregation described in [11, 12], and down-sampled each frame by a factor 4 in both x and y direction. To generate sequences of different actions, we concatenated parts of the optic flow sequences for: walking, jumping in place and waving with both arms. Thus, the final sequences contained 3 actions each.

Fig.2, right top, shows the marginal posterior distributions of a bin model with $M = 2$, which is the location of the maximum of the posterior over the number of bin boundaries. The bin boundary posteriors are concentrated close to the true change points, indicating that BB is able to identify the transitions between different actions correctly. Moreover, we can also compute the expected probability of action identity $P(\text{action})$ (here: walk,jump or wave, fig.2, left middle) as a function of time. BB identifies the actions (in this example) with near certainty. For comparison, the graph in the right half of fig.2 shows results obtained with purely frame-based classification: 'jump' and 'wave' are hard to discriminate, which highlights the usefulness of BB in this scenario. While the results of BB on action segmentation are not always as crisp as those depicted in fig.2, we find that BB gives good results on this task in general.

4 Conclusion

We demonstrated the usefulness of Bayesian binning for retrospective temporal segmentation on two examples, spike trains and action sequences. While BB is relatively well established in a Computational Neuroscience context, its application within the domain of Computer Vision is novel and needs further investigation. For example, the naive Bayes classifiers used here could be replaced with something more sophisticated, e.g. hidden Markov models (HMM). HMMs would have the advantage of being able to learn significantly more complex action types than those that can be discriminated with a naive Bayes classifier trained on optic flow. BB would then model the switching between HMMs. While HMMs typically need a lot of data to train, BB has the potential of regularizing the learning problem through binning. This allows for training on smaller data sets.

The core iteration of BB is an instance of the sum-product algorithm [13], which allows for the evaluation of arbitrary posterior averages subject to the restrictions described in [1]. We used it here to obtain the marginal distribution of each bin boundary. One could alternatively use a suitably formulated version of the max-product algorithm [13] to find the joint posterior maximum of the bin boundaries $\{k_m\}$, and use these $\{k_m\}$ as segmentation points.

5 Acknowledgements

The neural spike train segmentation work was partially supported by UK MRC fellowship G0501319. Action sequence segmentation was supported by EU FP7 project no. 215866, project name SEARISE.

References

[1] D. Endres and P. Földiák. Bayesian bin distribution inference and mutual information. *IEEE Transactions on Information Theory*, 51(11), 2005.

[2] D. Endres, M. Oram, J. Schindelin, and P. Földiák. Bayesian binning beats approximate alternatives: estimating peri-stimulus time histograms. In J.C. Platt, D. Koller, Y Singer, and S. Roweis, editors, *Advances in Neural Information Processing Systems 20*. MIT Press, Cambridge, MA, 2008.

[3] M. W. Oram, D. Xiao, B. Dritschel, and K.R. Payne. The temporal precision of neural signals: A unique role for response latency? *Philosophical Transactions of the Royal Society, Series B*, 357:987–1001, 2002.

[4] H. Shimazaki and S. Shinomoto. A recipe for optimizing a time-histogram. In B. Schölkopf, J. Platt, and T. Hoffman, editors, *Advances in Neural Information Processing Systems 19*, pages 1289–1296. MIT Press, Cambridge, MA, 2007.

[5] H. Shimazaki and S. Shinomoto. A method for selecting the bin size of a time histogram. *Neural Computation*, 19(6):1503–1527, 2007.

[6] H. Shimazaki and S. Shinomoto. Kernel bandwidth optimization in spike rate estimation. *Journal of Computational Neuroscience*, 2009. DOI 10.1007/s10827-009-0180-4.

[7] J.P. Cunningham, B.M. Yu, K.V. Shenoy, and M. Sahani. Inferring neural firing rates from spike trains using Gaussian processes. In J.C. Platt, D. Koller, Y Singer, and S. Roweis, editors, *Advances in Neural Information Processing Systems 20*. MIT Press, Cambridge, MA, 2008.

[8] D. Endres and M. Oram. Feature extraction from spike trains with bayesian binning: latency is where the signal starts. *Journal of Computational Neuroscience*, 2009. DOI 10.1007/s10827-009-0157-3.

[9] J.O. Berger. *Statistical Decision Theory and Bayesian Analysis*. Springer, New York, 1985.

[10] Lena Gorelick, Moshe Blank, Eli Shechtman, Michal Irani, and Ronen Basri. Actions as space-time shapes. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 29(12):2247–2253, December 2007.

[11] P. Bayerl and H. Neumann. A fast biologically inspired algorithm for recurrent motion estimation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 29(2):246–260, 2007.

[12] C. Beck, T. Gottbehuet, and H. Neumann. Integration of multiple temporal and spatial scales for robust optic flow estimation in a biologically inspired algorithm. In *CAIP*, pages 53–60, 2007.

[13] F.R. Kschischang, B.J. Frey, and H.-A. Loeliger. Factor graphs and the sum-product algorithm. *IEEE Transactions on Information Theory*, 47(2):498–519, 2001.