Spike sorting the other way

Alexander G Dimitrov

Center for Computational Biology, Montana State University
Bozeman MT 59717

Abstract

The problem of identifying neural activity from extracellular recordings is still with us. In recent years there have been a lot of improvements in clustering and identification over the first simple techniques. However, the manner in which data samples are selected as putative spike candidates has remained almost unchanged. This part of data processing still remains a largely heuristic procedure. In this work we offer a more quantitative approach to identifying putative spike shapes, which then serve as inputs to clustering methods.

The usual approach of identifying putative spike events is to invent certain heuristics, which in many cases are sufficient to trigger on most of the events. These include simple amplitude triggers, triggers in alternate spaces (e.g., total Fourier power), coarse description of the shape of the spike (first go down, then go up, then wiggle just so) and others [5]. We decided to approach this problem by first identifying segments of data that do not contain spikes. The assumption behind this approach is that while regions containing spikes may have very complicated characteristics, spike-free regions are relatively simple, and can be described in a simple manner. We model these examples of "no spike" events (null hypothesis) with a maximum entropy [2], multivariate Gaussian model. We then search for events in the data which are "very different" from the null event model, using the negative log likelihood from the model as a distance function (Mahalanobis distance in this case). Peaks of the distance correlate well with positions of spike events in the data, and events extracted in this manner form a relatively clean dataset for subsequent clustering.

The following figures the important steps in this process. First, samples are selected at random from the dataset. The samples consist of data in temporal windows of length L (free parameter) across N electrodes. We model the samples as a multivariate Gaussian of dimension $N \cdot L$. To decrease the data size

Email address: alex@cns.montana.edu (Alexander G Dimitrov).

requirement, we perform probabilistic PCA, which preserves the eigenstructure for the top K principle components, and models the orthogonal space as a spherical Gaussian [1].

If the samples were truly Gaussian, the Mahalanobis distance d should be distributed according to the $\chi_{2N \cdot L}$ distribution [6]. Asymptotically in $N \cdot L$, $\hat{d} = (d - mean(d))/std(d)$ is a normal distribution with zero mean and variance of one ($\mathcal{N}(0,1)$). We assume that this is approximately correct for large window sizes L as well, and take this normal distribution as the expected one for $\hat{d}$ under the normality assumption. Under this assumption, we observe two types of outliers (Figure 1). First, there are events that are extremely improbable under this model (all dashed line events with $\hat{d} > 5$ here fall in this category). Those events are discarded from the training dataset. Another set of events (events with $3 < \hat{d} < 5$ here) are likely to occur, but there is an excessive amount of them compared to the model. We resample these events at random, with probability $p(\text{sample}) = p(\text{expected})/p(\text{observed})$ (the discrepancy between expected and observed number of events). We then re-estimate the model using the resampled dataset. This procedure is repeated several times, until expected and observed distributions are "similar". At that point, we assume that we have built a satisfactory noise model. Here as a measure of similarity we use the Kullback-Leibler divergence, $KL(p(\text{observed})||p(\text{expected}))$ [4].

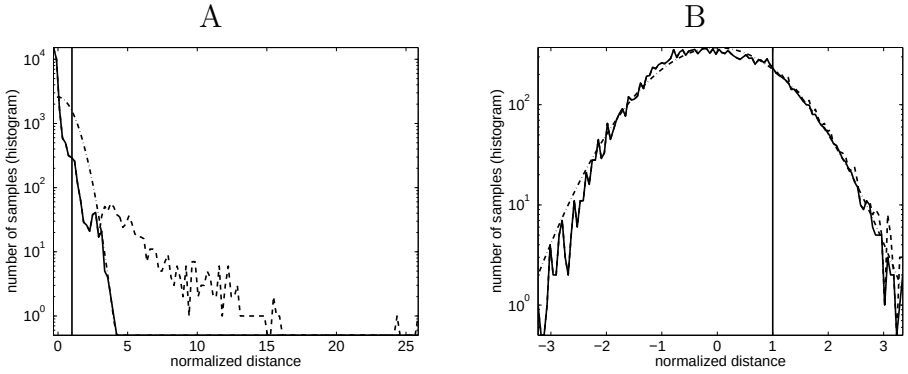

Fig. 1. *Histograms of the initial distribution of d (A) and the distribution of d after resampling (B). (- -) The histogram of d. (- .) The expected histogram under the model. Samples on the right side of the plot are far from the model. The vertical line indicates a distance, to the right of which we resample the data. (–) The histogram of d from the resampled data. In B) the same objects can be seen at point near the end of the resampling procedure. The data (–) now fits the expected distribution (- .) much better than before.*

Once we have a noise model, we pass the full dataset through it, using a running window to collect samples and calculate the distance $\hat{d}$ of these samples to the noise model. The outcome of one such application is shown in Figure 2. Peaks of the distance to the noise model (circles) seem to correlate well

with spike events. This effect can be seen in detail in Figure 3. The threshold above which we look for peaks can in principle be set automatically. Since we assume that $\hat{d} \propto \mathcal{N}(0,1)$, one can pick a threshold above a desired noise level (e.g., 3, which here indicates 3σ away from $\hat{d} = 0$). The evidence in Figure 3 suggests that such a threshold is usually too low. We interpret this to mean that the noise model is too broad and accepts certain spike events as noise (e.g., the first even near $0.2s$). The current algorithm relies on manual input of this threshold.

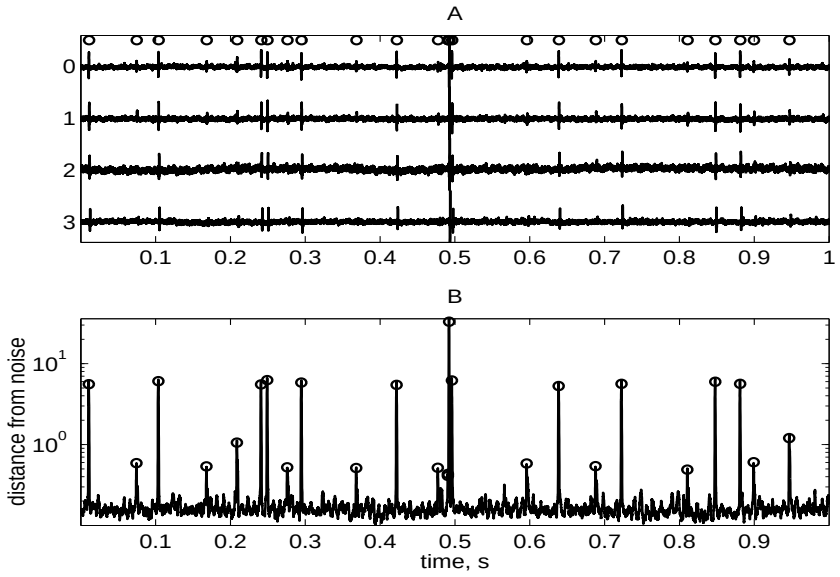

Fig. 2. *Raw data traces (A) and their distance to the noise model (B). The distance function d was used as a filter on the data to obtain the trace in B). Note that B) is on a logarithmic scale. Peaks of $\hat{d}$ are identified and marked with circles on both panel A) and panel B).*

Given the results in the above figures, we assume that peaks of $\hat{d}$ indicate the presence of spikes. The next step collects all events that are marked by peaks in the distance to the noise model (locally most unlikely according to the model). Some of the events are shown on Figure 4. As can be seen, they are mostly well aligned, and even visually one can discriminate several distinct event classes. Some of the alignment problems can actually be traced to the code we use to register peaks, so the true alignment is even better. The few apparently strongly misaligned events on both ends are actually second spikes occurring close to the main spike in the center.

The set of events shown on Figure 4 can be used as a sample set to train a model with which to discriminate these events, by any of the methods reviewed in [5], or by other clustering approaches [3,6].

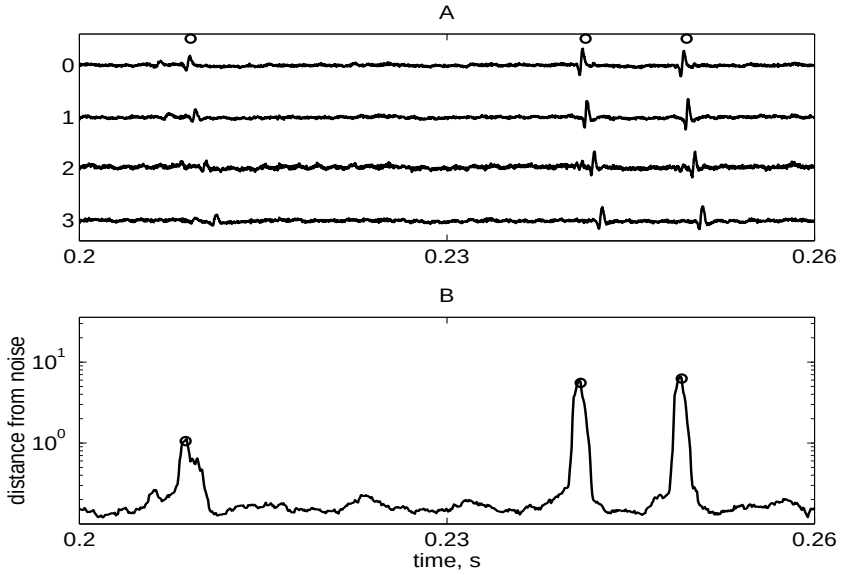

Fig. 3. *An expanded portion of 60 ms of Figure 2 shows a detailed view of the correspondence between raw data (A) and its distance to the noise model (B).*

Acknowledgments

Research supported in part by NSF grants EIA0129895 and MRI9871191. The author thanks Maneesh Sahani for pointing out that the resampling algorithm can be expressed as a mixture model which explicitly includes the second hypothesis as a uniform density component.

References

[1] C. M. Bishop. *Neural Networks for Pattern Recognition.* Oxford University Press, New York, 1998.

[2] E. T. Jaynes. On the rationale of maximum-entropy methods. *Proc. IEEE*, 70:939–952, 1982.

[3] D. H. Johnson and D. E. Dudgeon. *Array Signal Processing.* Prentice Hall, 1993.

[4] S. Kullback. *Information Theory and Statistics.* J Wiley and Sons, New York, 1959.

[5] M. Lewicki. A review of methods for spike sorting: the detection and classification of neural action potentials. *Network: Computation in Neural Systems*, 9(4):R53–R78, 1998.

[6] L. L. Scharf. *Statistical Signal Processing.* Addusin-Wesley, 1991.

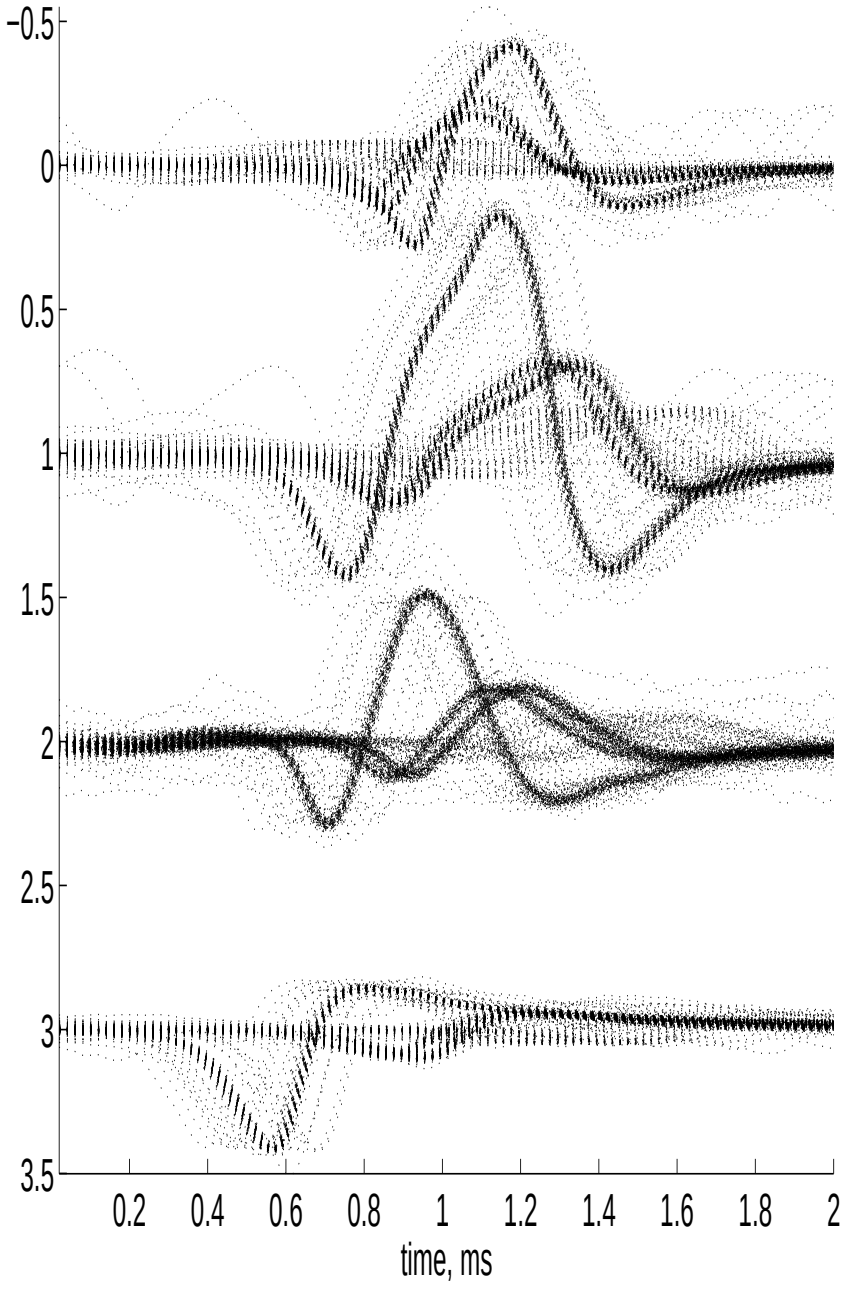

Fig. 4. *Some of the events, marked as most unlikely under the noise model.*