"Noise-Outlier Graphical Identification of Biological Events for Seizure Detection"

Mark R. Bower

This is an R Markdown Notebook. When you execute code within the notebook, the results appear beneath the code.

Abstract

Biological signal detection algorithms are commonly based on thresholding to reduce noise and iterative optimization methods (e.g., Maximum Likelihood Estimation) to cluster similar signals. Thresholding became popular when computer storage was limited and expensive. Optimization algorithms became prominent when CPU power increased to the point where individual computers could hold all of the data points obtained in eperiments in memory, simultaneously. While subsequent improvements in technology have far exceeded these early advances, these constraints have continued to limit algorithm advancement.

Computer technology can now record and process terabytes of continuously recorded data and analyze each local peak, eliminating the need for noise reduction. In fact, bootstrapping techniques can utilize the properties of Recent advances in graph theory, however, offer new capabilities. In particular, we noted that collections of biological signal events share three properties relative to collections of noise events: hey are relatively larger in amplitude, repeatable and rare compared to noise events. Combining these ideas, we analyzed all peaks in several different *in vivo* datasets and used graph theoretic methods to separte signals from noise and identify clusters of similar signal data. Unlike traditional optimization-based algorithms, this algorithm runs in linear time.

Event processing (certainly in biology) can be seen as the result of two compromises: rapidly eliminating noise detections to save storage space, and assuming idealized data distributions to simply statistical computations. Users were both "data poor" and "computation poor" and so were required to accept reasonable, simplifying assumptions that reduced the power of their experiments. Technical advances of the past decades, however, have eliminated these constraints. Both storage space and computational power are now relatively cheap and readily available. We are now both "data rich" (terabytes of data storage can be purchased at reasonable prices) and "computation rich" (computational clusters are readily available that can utilize dozens if not hundreds of processors on a task).

We then use this algorithm on three data sets (a rat model of epilepsy; ambulatory, intracranial, human EEG; high-frequency, inctracranial human EEG) to identify both single-neuron action potentials and field-potential events (i.e., inter-ictal spikes (IIS) and sharp-wave ripples (SWR)). In addition to replicating prior results, we extend SRC theory by showing the necessity of post-seizure sleep for consolidation.

Introduction

An old saying goes "One person's signal is another's noise." Signal processing pipelines have long placed noise reduction and elimination at the start of processing. Historically, this arose from limited compuational power available to researchers, which placed a greater emphasis on reducing the amount of data to be processed early in the processing stream. These limitations are no longer so relevant and have led to the emergence of algorithms that replace fixed assumptions about data and distributions with computationally intensive approches that rely solely on the data themselves. So-called "bootstrap" algorithms are but one example [...].


The field of signal detection and clustering, however, has not seen as many applications of these techniques, relying more on classical optimization approaches, such as Maximum Likelihood Estimation, which assume underlying distributions such as a mixture-of-gaussians. Importantly, the field continues to rely on noise elimination as an early processings step, often ignoring the significant amount of infomration that could be utilized. This is particularly important in the analysis of bioligcal data, because so much of what is recorded is noise. Consider a system that is detecting action potentials from three different neurons that each have an average rate of 2 Hz. The duration of an action potential roughly 1 msec, so on average 6 msec of action potential waveforms will be recorded per second, leaving 994 msec of noise. The signal constitutes less than 1% of the recorded data, so noise elimintation invalidates over 99% of the avquired data *as a first step*! This is a significant loss of information, even if it is only noise.

In addition, clustering algorithms have routinely relied on error minimization approaches [Duda and Hart]. Other approaches have been less utilized. One such area is graphical methods that utilize connectivity to identify clusters. If the factors that govern connectivity are time-local, then clusters naturally become time-varying and thus les sensitive to time-varying factors such as drift.

The algorithm follows four steps: 1) iteratively place events into a graph, 2) find graphical communities, 3) normalize graph identities, 4) determine whether a given node should be persisted to a databse as 'signal' or 'noise'. This algorithm requires six parameters based on expected temporal properties of events: 1) characteristic duration, 2) correlation window, 3) community window, 4) minimum number of detections, 5) minimum average event rate, 6) maximum average event rate. None are based on sequential time-voltage relationships, or on waveform "shape". Instead, measurements of number, duration, and rate are used.

Is this any better than a simple threshold detector? How to compare? One advantage is that the identification step produces clusters that could be used in subsequent, standard cluster-joining steps. *Another advantage is that an adaptive threshold could be set by comparing signal and noise distributions; i.e., compute a distribution on the noise and keep only those signals that exceed 95% confidence limits.* Things are running! Am seeing some strange-looking waveforms being called "noise" that seem of interest. Either they are "signal" that are being misclassified or my algorithm is good rejecting even signal-like noise.

Of concern is how to use the information of identification in a subsequent clustering step.

While the identification stage is running, start work on the contemporaneous-window problem. Really, it is the contemporaneous-window correlation problem. I don't really care to save the waveforms on other channels, just the CC.

Am missing some detections. Thinking about this has led me to The Iron Law of Noise: Low count. Small peak. All along, I have assumed that a 2D density was required to allow slopes or valleys to be identified as separatrices. Finally, it occurred to me that bootstrapping doesn't require any of that. Identify "noise" members using The Iron Law, then use bootstrapping to identify groups outside the 95% confidence noise interval. Simple and straight-forward. The free parameters are based only on physiology: Peak > 75 µV, Rate $> 1/300$ Hz (at least one event every 5 min), Rate < 5 Hz, and Count > 2.

What if you took log's of count and energy? Would you get more of a straight line against which you could do a regression? No: the curve is modeled by 1/log, which cannot be reverse-modeled by log. Log-scaling axes helps with presentation, but does not drastically alter the shape of any regression curve that could be fitted. A nonparametric, spatial bootstrap is needed. If a fast means for computing distance from the mean line could be found, 95% intervals could be computed by repeated Monte Carlo. In effect, this amounts to a "reverse threshold"; rather than setting a minimum value for events we would consider to be signals, we set a maximum value for events we are certain label as noise. Using a well-established bootstrap technique to establish confidence intervals (the ABC algorithm using the *abccon* function in R, Efron and Tibshirani, 1992) on identified noise events, we used the distribution of noise events to identify those events that had been labeled as signals that did not differ statistically from noise.

A regression curve is not needed. Instead, for each count value, compute the mean and standard deviation for all 'noise' events. Then, compute the mean + 2*SD. The figure shows the 'signal' points with the noise line. Notice how many points still fall into the noise range.

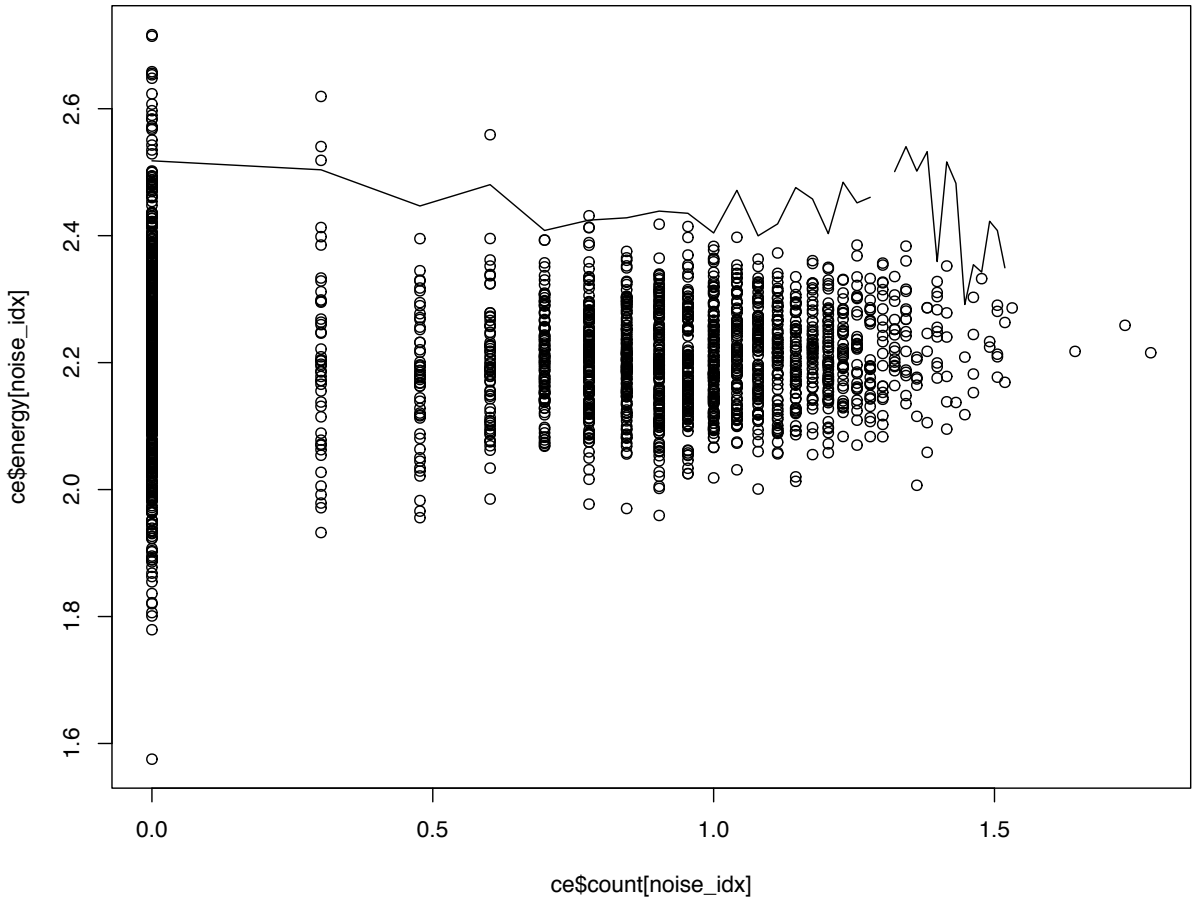


Figure 1: Points for 'noise' clusters are two-sigma line.

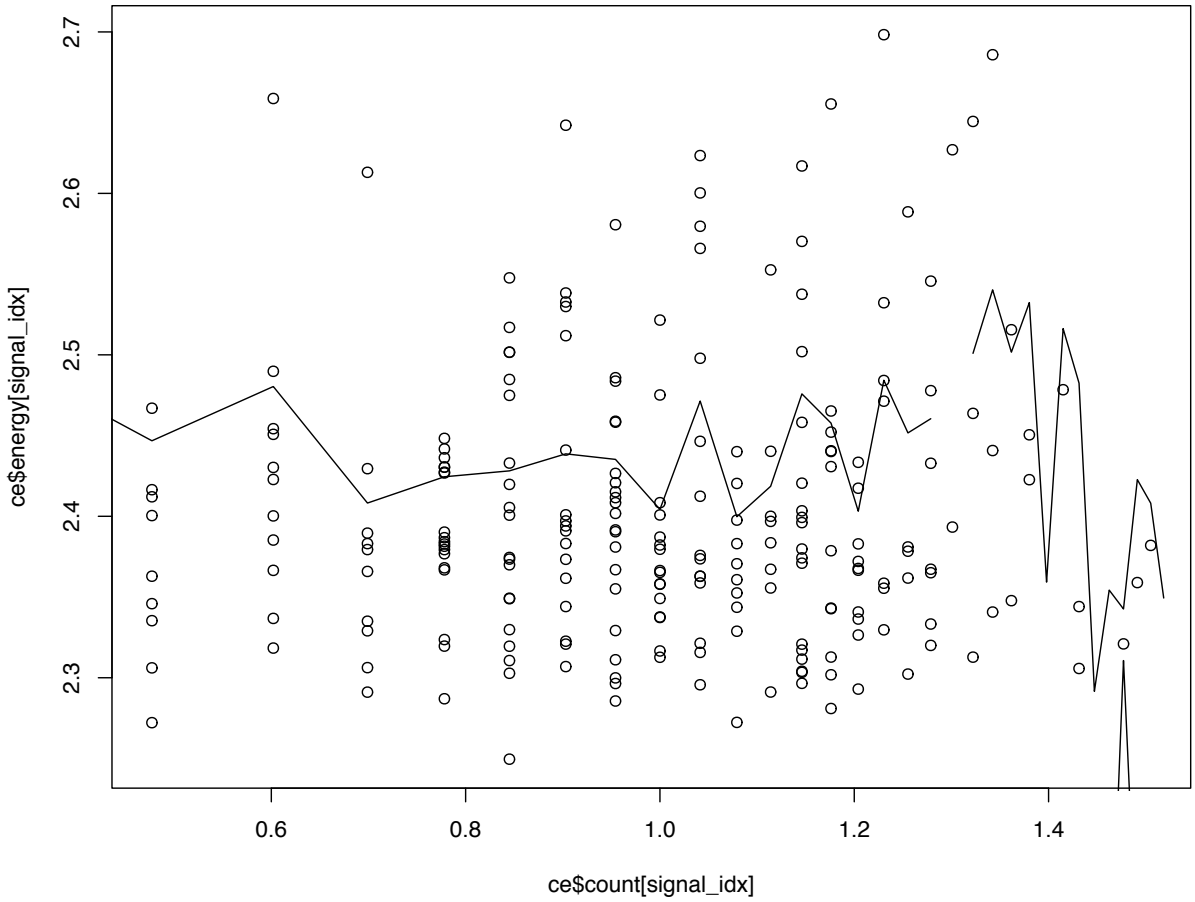


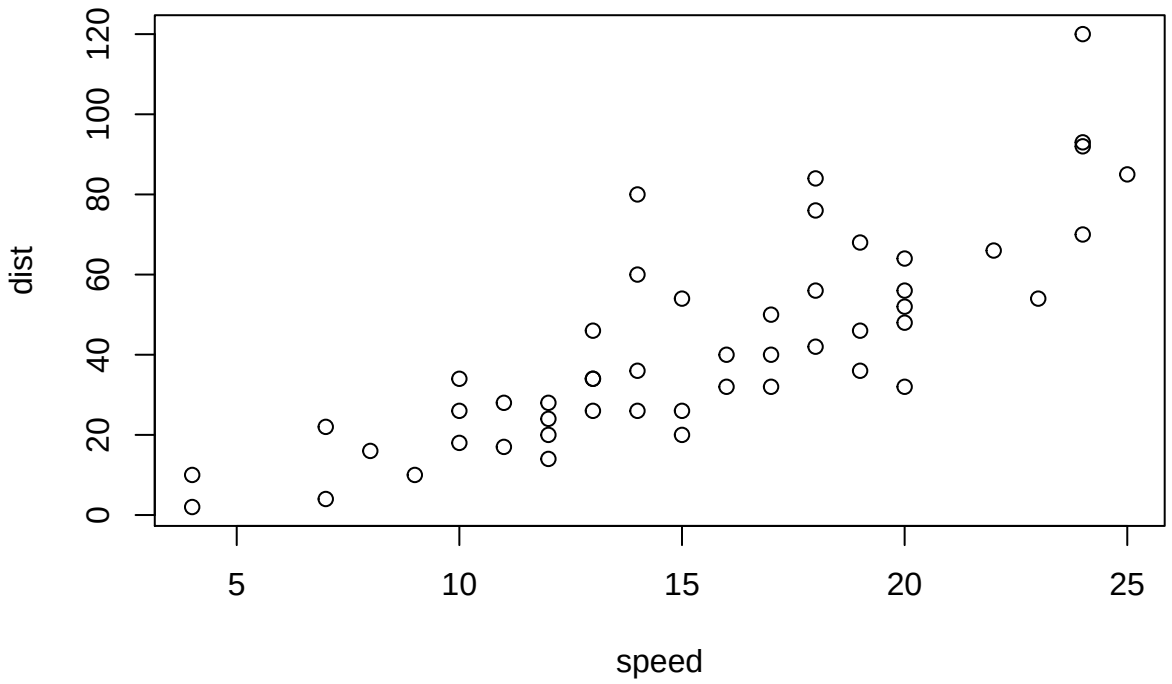
Figure 2: Points for some 'signal' clusters are outliers for 'noise' distribution.

Outline of Action

Make the Rmd document "live"; i.e., update and over-write it, continuously. It is not a journal or lab book. It is a living document that needs to be burned down, bulldozed and re-built on a daily basis. Posterity is for paper and pen. Get used to rewriting sections until they solidify over days. When a section achieves some stability, copy it into the Scrivener document.

Try executing this chunk by clicking the *Run* button within the chunk or by placing your cursor inside it and pressing *Cmd+Shift+Enter*.

```
plot(cars)
```



Add a new chunk by clicking the *Insert Chunk* button on the toolbar or by pressing *Cmd+Option+I*.

When you save the notebook, an HTML file containing the code and output will be saved alongside it (click the *Preview* button or press *Cmd+Shift+K* to preview the HTML file).