Neural signal clustering with noisy sparse subspace clustering

Zamar Edwin
Department of Mathematics
zjedwin@umich.edu

Charles Lu
Department of Biomedical Engineering
lucw@umich.edu

Philip Vu
Department of Biomedical Engineering
philipv@umich.edu

Abstract

Neural signals are often high dimensional, noisy, and sparse, presenting a significant challenge to conventional clustering algorithms, such as principal components analysis and Gaussian mixture models. Here, we apply a recently reported noisy subspace clustering algorithm to neural signals. We find that while the noisy subspace clustering is a powerful tool for clustering certain types of noisy data, it is not well suited for neural signals when compared to traditional methods.

1 Introduction

Modern neuroscience is capable of observing and recording increasingly large datasets of neural signal. For the purpose of interpreting signals, i.e., attributing signals to specific movements, emotions, etc., it is often necessary to determine the neural source of recorded waveforms. Individual neurons produce characteristic waveforms, called “action potentials”, when activated. The different shapes of action potentials (examples in Figure 1) can be used to differentiate between signals from different neural “units” recorded by an electrode or array of electrodes, a process known as “spike sorting”.

Spike sorting is a challenging problem because data is often sparse and noisy. Spikes can be rare events in a recording, and environmental noise and artifacts can look very similar to action potentials (see Figure 1). Most neuroscience laboratories perform spike sorting manually, as the process is still largely considered an art. Some laboratories, however, have adopted semi-automated methods, consisting of four steps that can obtain lower error rates [1]. First, the spikes are detected by threshold crossings of neural signal that been passed through a highpass filter. Second, each spike waveform is transformed to a lower dimensional subspace, typically using principle component analysis (PCA). Third, the principal components of the neural spikes are then divided into groups using a clustering algorithm. Lastly, the results are manually verified to adjust for any errors made by the previous automated algorithms. This method can work well for single-channel electrode recordings, reaching error rates of 5% or lower compared to ground truth data.

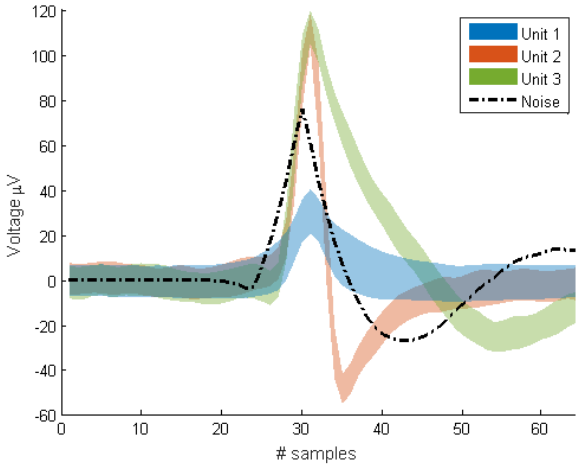

Figure 1: Distribution of action potentials from three different units in blue, orange, and green. A representative noise signal is overlaid in black.

However, these semi-automated spike-sorting methods break down for modern multi-electrode array devices. Conventional arrays can record from up to 96 channels simultaneously, resulting in very high-dimensional data. Although many different spike sorting methods have been proposed [2-5], no method has been robust enough to be adopted by the experimental community. Here, we test a novel extended clustering algorithm, sparse subspace clustering (SSC), which combines steps 2 and 3 of the semi-automated method previously described.

The SSC algorithm combines steps 2 and 3 by forming a sparse representation for the data, employing this representation to construct an affinity matrix defining similarities between data points, and finally applying the spectral clustering algorithm to the graph Laplacian of the affinity matrix to yield clusters. Therefore, this method does not require the data to first be projected into a lower dimensional subspace. Explicitly, similar to many clustering algorithms, the SSC algorithm is based on the idea of writing each data point x_i as a linear (affine) combination of neighboring data points. However, rather than defining neighbors by means of angular or Euclidean distances, SSC allows any two data points in the data set to be neighbors. This freedom induces an ill-posed problem with many possible solutions. Therefore the property of sparsity is invoked wherein every point is written as a linear (affine) combination of all other data points by minimizing the number of nonzero coefficients $c_{i,j}$ such that $x_i = \sum_{j \neq i} c_{i,j} x_j$. This problem is a combinatorial optimization problem, however, so the following simpler optimization problem is solved

$$\min_{c_i} \|c_i\|_1 \qquad s.t. \qquad x_i = X_{-i} c_i \tag{1.1}$$

It is known that when the subspaces are either independent or disjoint, this optimization problem is minimized when $c_{i,j} = 0$ only if x_i and x_j are in different subspaces [6-7]. Hence, the sparsest representation of the data points is achieved when each point is written as a linear (affine) combination of the other points in its own subspace. Given a sparse representation for each data point, the graph affinity matrix is defined as $W = |C| + |C|^t$. The segmentation is then obtained by applying the spectral clustering algorithm to the graph Laplacian.

The noisy subspace clustering algorithm described by Wang and Xu [8] extends the analysis of SSC to the event where data points do not lie exactly in subspace. Therefore, their method does not attempt to solve the optimization problem (1.1) exactly. Instead, a penalty in the 2-norm of the error is added to (1.1). Specifically, the following optimization problem is considered

$$\min_{c_i} \|c_i\|_1 + \frac{\lambda}{2} \|x_i - X_{-i} c_i\|^2 \tag{1.2}$$

Because the formulation of (2.2) coincides with standard LASSO, this algorithm is called LASSO-SSC. This method hinges on the LASSO Subspace Detection Property. That is, given

noisy sample points X and subspaces $\{S_l\}_{l=1}^{k}$ the subspaces satisfy the LASSO subspace detection property with λ, if and only if, for all i the optimal solution to c_i with parameter λ satisfies: (1) c_i is not the zero vector, and (2) nonzero entries of c_i correspond to only columns of X sampled from the same subspace as x_i. This property ensures that the weight matrix C and the affinity matrix W are block diagonal with each subspace cluster represented by a disjoint block. With this setup, Wang and Xu provide sufficient conditions upon which the LASSO subspace detection properties hold in fully deterministic models, deterministic data with random noise, uniformly random data with random noise, and finally fully random models. For each of these models, bounds are provided for λ regulating the magnitude of acceptable noise for full recovery of the subspaces.

This method has been shown to be robust against random noise added to unlabeled input data points [8]. To verify this, we passed simulated and real neuronal spike recordings into the SSC to test performance. Additionally, we compare SSC's performance against other clustering algorithms that require PCA as an initial step: k-means, Gaussian mixture models, and spectral clustering. We hypothesize that by combining dimensionality reduction and cluster detection, the error rates of automated spike sorting will significantly drop and a more robust and accurate algorithm may be available.

2 Methods

2.1 Data Sets

Validation of clustering algorithms was performed using three data sets: 1) a simple mixture of Gaussians with labels; 2) a set of simulated action potentials with labels; and 3) a set of real action potentials recorded from human cortex, unlabeled. The mixture of Gaussians contained four clusters, with randomly generated means and covariance matrices, and uniform background noise. The simulated action potentials were created using methods described by Martinez et al [9], and contain three units. Noise was generated using a $1/f^2$ distribution to simulate background neural activity, with false spikes defined as segments of noise with values at least 4.5 times the root mean square of the background signal. Real action potentials were provided by the University of Leicester NeuroEngineering Lab and are publically available at http://www2.le.ac.uk/departments/engineering/research/bioengineering/neuroengineering-lab/software.

2.2 Clustering Algorithms

In addition to the noisy subspace clustering algorithm described by Wang and Xu, we also evaluated a number of conventional approaches to neural signal clustering. We implemented the typical approach of combining principal components analysis (PCA), with each sample in time treated as a different parameter, and a clustering algorithm. Here, we use k-means, Gaussian mixture models, and spectral clustering to group action potentials that have been transformed using PCA. Similarity graphs used for spectral clustering are based on complete graphs, with σ determined empirically. k-means and GMM were run ten times on each data set to account for non-deterministic outputs. The best performing clusters of the ten runs were used for calculating performance metrics for comparison.

2.3 Performance Metrics

For simulated data sets, ground truth labels are available. Performance of the clustering algorithms for these data were compared using classification accuracy. For the real data set, ground truth is not known, so performance is evaluated subjectively by observing cluster centroids.

3 Results

Performance metrics were obtained from PCA+k-means, PCA+GMM, spectral clustering, and SSC, when applied to a Gaussian mixture with uniform noise, a simulated neural data set with $1/f^2$ noise, and a real unlabeled neural data set.

3.1 Gaussian Mixture

Table 1: Percent accuracy of Gaussian mixture classification

	Cluster #1	Cluster #2	Cluster #3	Cluster #4	Noise
k-means	73	100	100	100	100
GMM	100	100	100	100	100
Spectral	99	0	0	0	0
SSC	100	99	100	100	100

GMM and SSC produced nearly perfect classifications of noisy Gaussian mixture classifications. k-means produces largely accurate clustering. Spectral clustering classified nearly all data into a single cluster.

3.2 Simulated Neural Data (Labeled)

Table 2: Percent accuracy of simulated neural data classification

	Neuron #1	Neuron #2	Neuron #3	Noise
k-means	92	99	100	90
GMM	91	95	100	94
Spectral	35	85	84	41
SSC	43	75	11	44

k-means and GMM produced cluster centers representative of true cluster means, and were largely correct in classification of simulated neural data and noise. Spectral clustering and SSC largely failed to identify clusters, and produced less accurate results, with most cluster assignments going to the spike with largest true representation in the dataset—Neuron #2. (Figure 2)

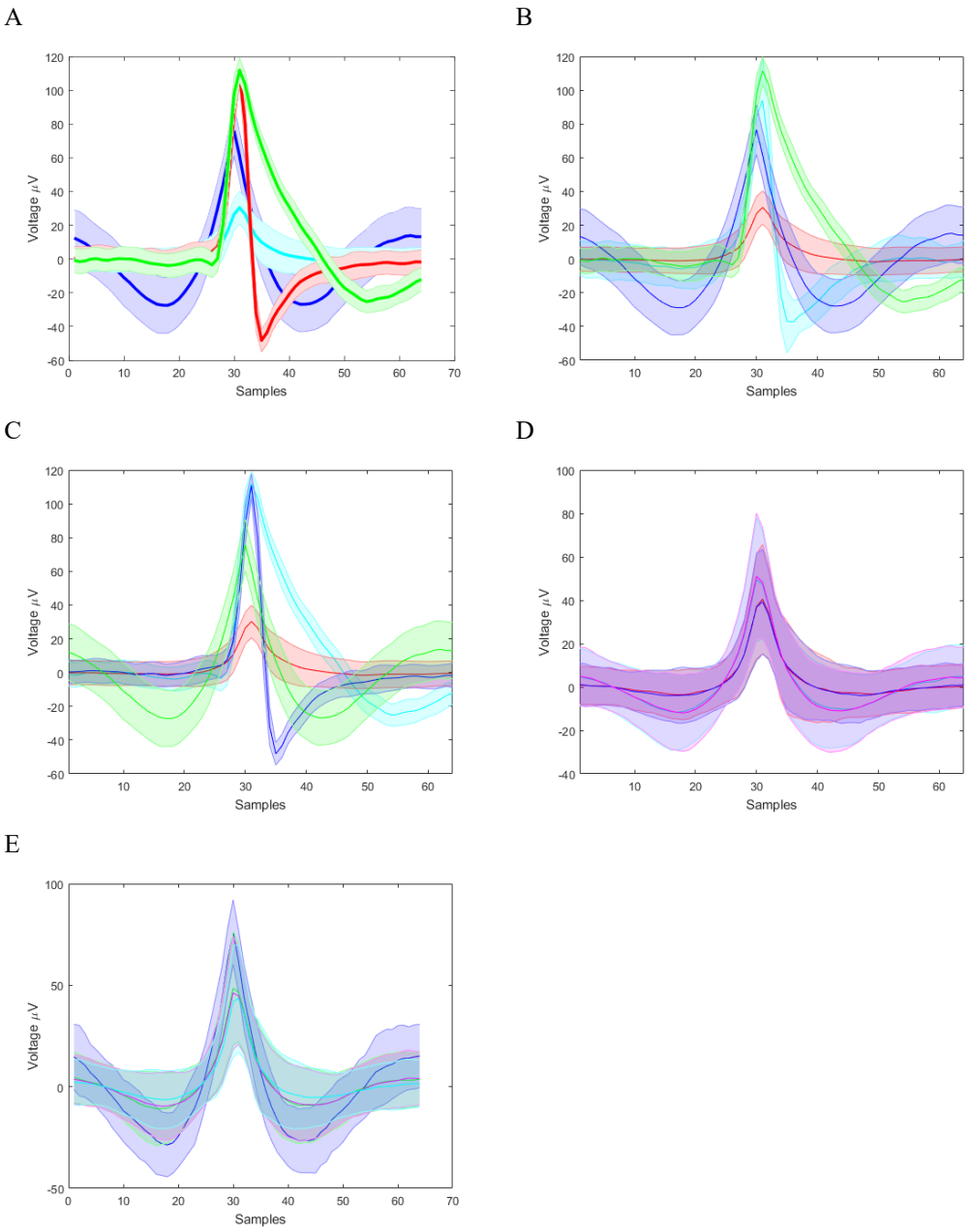

Figure 2: From simulated neural data—mean waveforms and standard deviations of each cluster from true labels (A), k-means (B), GMM (C), spectral clustering (D), and SSC (E).

3.3 Real Neural Data (Unlabeled)

k-means and GMM produced the most distinct clusters centers, with k-means exhibiting the least variance within clusters. Cluster center waveforms closely resemble action potentials. Spectral clustering and SSC produced cluster centers that are largely indistinguishable and exhibit high overlap.

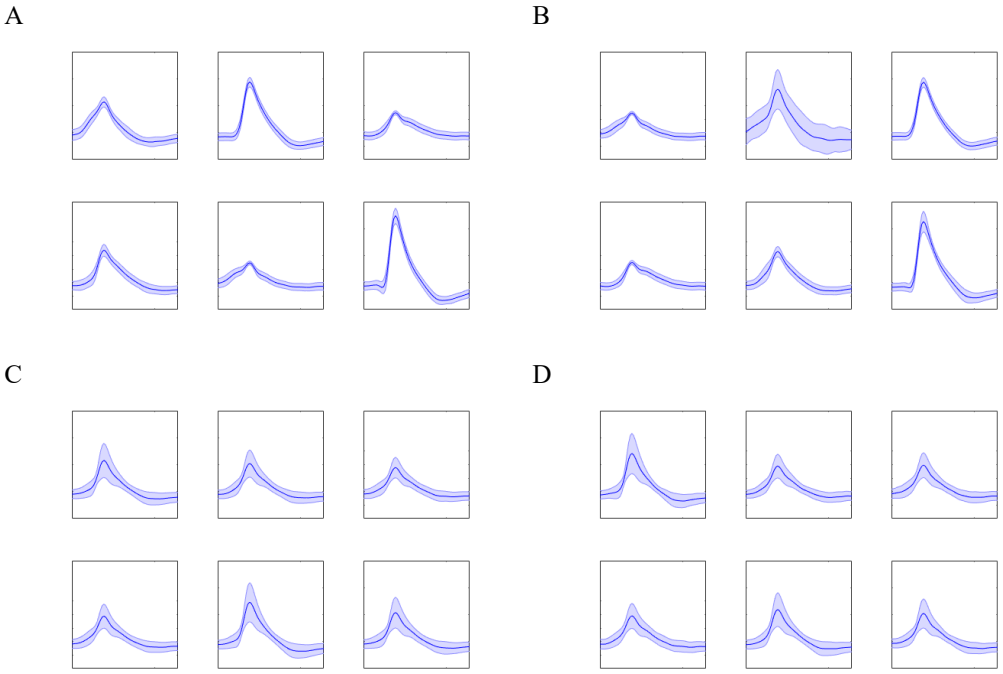

Figure 3: From real, unlabeled neural data—mean waveforms and standard deviations of each cluster from k-means (A), GMM (B), spectral clustering (C), and SSC (D)

4 Discussion

In this study, we evaluated the performance of a novel noisy subspace clustering algorithm in classifying neural data, and compared its performance to conventional methods used by the field of neuroscience. To test performance, we applied each algorithm to three noisy data sets: two simulated, labeled sets and one real, unlabeled set.

We found that GMM, k-means, and SSC demonstrate greater accuracy when clustering simulated Gaussian mixture data. With simulated spikes, GMM and k-means exhibited better accuracy and performance, while SSC had a lower performance. This may suggest that SSC can be robust against different levels of noise but fail to distinguish between similarly shaped waveforms, such as neural spikes, which are often identical over many parameters. This may be attributed to a violation of the subspace detection property, in which nonzero entries of c_i correspond to columns of X sampled from difference subspaces of x_i, i.e., there may be an overlap of subspaces between data points, producing an ambiguous similarity graph on which the algorithm relies. For real spike data, GMM and k-means produced the most distinguishable clusters. Two distinct spikes appear in both GMM and K-means with tight variances around the shape of the waveform. Some clustered waveforms in SSC may be considered a spike, but further human verification would be needed to confirm. Although the SSC was initially used to address noise challenges faced when recording neural data, it appears that highly overlapping neural waveforms pose another a challenge to SSC. Overall, the standard semi-supervised spike sorting algorithms outperformed SSC. For future work, exploration of different combinations of algorithms may be needed to produce a higher-performing automated spike sorting system.

Contributions

All authors of this study contributed to each section of the report. Zamar Edwin played a significant role in interpreting the SSC algorithm and creating a pipeline for implementation of the k-means, GMM, and spectral clustering algorithms. Charles Lu prepared the SSC algorithm, created the simulated data sets used in this study, and evaluated each algorithm. Philip Vu provided much of the background knowledge for motivation of this study, led interpretation of results, and created figures.

References

[1] KD Harris, DA Henze, J Csicsvari, H Hirase, G Buzsaki (2000) Accuracy of Tetrode Spike Separation as Determined by Simultaneous Intracellular and Extracellular Measurements. J Neurophysiol 84:401-414.

[2] Ekanadham C, Tranchina D, Simoncelli EP (2014) A unified framework and method for automatic neural spike identification. J. Neurosci. Methods 222, 47–55.

[3] Carlson DE, Vogelstein JT, Wu Q, Lian W, Zhou M, Stoetzner CR, Kipke D, Weber D, Dunson DB, Carin L (2014) Multichannel electrophysiological spike sorting via joint dictionary learning and mixture modeling. IEEE Trans. Biomed. Eng. 61, 41–54.

[4] Calabrese A, Paninski L (2011) Kalman filter mixture model for spike sorting of non-stationary data. J. Neurosci. Methods 196, 159–169.

[5] Quian Quiroga R, Nadasdy Z, Ben-Shaul Y (2004) Unsupervised Spike Detection and Sorting with Wavelets and Superparamagnetic Clustering. Neural Comp 16:1661-1687.

[6] ElhamifarE , Vidal R (2009) Sparse subspace clustering. IEEE Conference on Computer Vision and Pattern Recognition.

[7] Elhamifar E, Vidal R (2010) Clustering disjoint subspaces via sparse representation. IEEE Conference on Acoustics, Speech and Signal Processing.

[8] Wang YX, Xu H (2016) Noisy sparse subspace clustering. Journal of Machine Learning Research 17:320-360.

[9] J Martinez, C Pedreira, Ison MJ, Quian Quiroga R (2009) Realistic simulation of extracellular recordings. J Neurosci Methods 184(2):285-93.