Unsupervised Spike Sorting Based on Discriminative Subspace Learning

Mohammad Reza Keshtkaran and Zhi Yang

***Abstract*— Spike sorting is a fundamental preprocessing step for many neuroscience studies which rely on the analysis of spike trains. In this paper, we present two unsupervised spike sorting algorithms based on discriminative subspace learning. The first algorithm simultaneously learns the discriminative feature subspace and performs clustering. It uses histogram of features in the most discriminative projection to detect the number of neurons. The second algorithm performs hierarchical divisive clustering that learns a discriminative 1-dimensional subspace for clustering in each level of the hierarchy until achieving almost unimodal distribution in the subspace. The algorithms are tested on synthetic and *in-vivo* data, and are compared against two widely used spike sorting methods. The comparative results demonstrate that our spike sorting methods can achieve substantially higher accuracy in lower dimensional feature space, and they are highly robust to noise. Moreover, they provide significantly better cluster separability in the learned subspace than in the subspace obtained by principal component analysis or wavelet transform.**

I. Introduction

Extracellular recording is a method of measuring neuronal activity, which is commonly used by neuroscientists to study brain functions. Neural signals are recorded by inserting microelectrodes into the brain tissue which picks up local field potentials, the action potentials (also called spikes) from a few surrounding neurons, and noise. For obtaining (multi-)unit activity, the signal is often filtered in 300–5000 Hz frequency band and spikes are identified through using a spike detection method. In many neuroscience studies, it is necessary to sort the spikes after detection so that the spikes generated by an individual unit fall into one cluster. The spike sorting stage is fundamental to the neuroscience studies which involve the analysis of spike rates, spike time synchrony, and inter-spike interval [1], [2].

Common spike sorting methods involve detecting neural spikes, extracting and selecting features from detected spike waveforms, detecting the number of neurons, and assigning the spikes to their originating neurons [1]. For spike detection, methods based on absolute value thresholding, nonlinear energy operator, and wavelet have been widely used [2].

Among different feature extraction methods, principal component analysis (PCA) and discrete wavelet transforms (DWT) are most commonly used [1]–[3]. PCA projects the spikes to a set of orthogonal basis vectors that represent the largest variance of the data. In wavelet-based methods the spikes are decomposed into wavelets and the decomposition coefficients are used as features. A good feature extraction method should retain the most useful information for discriminating different spike shapes in a reasonably low dimension [2]. Some efforts have been done to come up with better feature extraction for spike sorting including methods based on waveform derivatives [4], Laplacian eigenmaps [5], wavelet optimization [6], and Fourier transform [7]. Although these methods can provide more discriminative features with reasonable dimension reduction, they do not necessarily seek for the most discriminative subspace for clustering [8]. Hence, the clusters which appear inherently separable in some discriminative subspace may overlap if projected using conventional features extraction methods. Such cluster overlaps increases the misclassification, may lead to incorrect detection of number of clusters, hindering reliable clustering. Therefore, feature learning methods which seek for the most discriminative subspace is expected to provide an optimal cluster separation thus improving the clustering performance in spike sorting.

Depending on the proximity of an electrode to the surrounding neurons, the recording may contain several spike waveforms generated by different neurons. In practice, the number of the contributing neurons (i.e. clusters) is not known *a priori* and needs to be detected from the data. Incorrect selection of the number of the clusters heavily affects the performance of the clustering algorithm. One way to set the number of neurons is through visual inspection of spike waveforms by an expert observer. This method makes the spike sorting non-automatic and is prone to human error. It is generally desired that the algorithm detects the number of the clusters in an unsupervised manner so as to eliminate the user intervention [1], [2].

Automatic detection of the number of neurons often depends on the method used for clustering. Some common clustering methods used for spike sorting include k-means, mixture models, neural networks, superparamagnetic clustering (SPC), and self-organizing maps [1], [7], [9], [10]. Several approaches based on Bayesian/Akaike information criterion, gap statistics, and statistical test for Gaussian distribution have been proposed to learn the number of clusters for k-means and mixture model [11], [12]. In SPC the number of clusters are detected by sweeping the temperature and choosing the clusters which are bigger than a certain size [10]. In general, various combinations of feature extraction and clustering have been applied to spike sorting with different levels of success [2], [3].

In this paper, we propose two unsupervised spike sorting algorithms based on discriminative subspace learning to

This work is supported by Minister of Education Tier-1 funding R-263-000-A47-112, Singapore A*STAR Public Sector funding R-263-000-699-305, and Young Investigator Award R-263-000-A29-133.

Mohammad Reza Keshtkaran and Zhi Yang are with the Department of Electrical and Computer Engineering, National University of Singapore, 117576, Singapore (e-mail: keshtkaran@nus.edu.sg)

978-1-4244-7929-0/14/$26.00 ©2014 IEEE

extract low dimensional and most discriminative features from the spike waveforms, and perform clustering. The core part of the algorithms involves iterative subspace selection using linear discriminant analysis (LDA) and clustering using k-means. The first algorithm directly uses the core part to simultaneously learn the most discriminative subspace and perform clustering. It uses histogram of features in the most discriminative projection to detect the number of neurons. The second algorithm performs hierarchical divisive clustering that learns a discriminative 1-dimensional subspace for clustering in each level of the hierarchy until achieving almost unimodal distribution in the subspace. The algorithms are tested on synthetic and *in-vivo* neural data, and their performances are compared with two commonly used spike sorting methods PCA-$k$means and `Wave_clus` [10] with regard to sorting accuracy, robustness to noise, and separation of the clusters.

The rest of this paper is organized as follows. Sec. II describes the two proposed spike sorting algorithms based on discriminative subspace learning. Sec. III gives the comparative results on synthetic and *in-vivo* data. And Sec. IV concludes the paper.

II. Method

In this work, we assume that the spikes are already detected through one of the common approaches, aligned to their peak, and stored in X. The following section describes the discriminative subspace learning method which is the core building block of the two proposed algorithms. The proposed algorithms for detecting the number of neurons and sorting the spikes are presented in Sec. II-B and Sec. II-C.

A. Discriminative Subspace Learning using LDA and k-means

Most of the feature extraction and dimensionality reduction techniques that have been used for spike sorting give a projection subspace which is not necessarily the most discriminative one. Feature extraction (followed by dimension reduction) is a crucial stage which determines the quality of the next stage clustering. Thus, feature extraction should transform the data in such a way that similar data points are close to each other while dissimilar ones are well-separated from each other. For this purpose, we utilize a discriminative subspace learning technique using iterative application of LDA and k-means.

Suppose the spikes are stored in $X_{(m\times n)}$ where m is the number of samples stored for each spike waveform and n is the total number of the detected spikes. We are interested in finding a discriminative projection matrix $W_{(m\times d)}$ to transform the spike waveforms to a d-dimensional ($d<m$) feature space $Y_{(d\times n)}$ so that the potential clusters have maximum separability:

$$Y = W^T X. \tag{1}$$

Clustering is then performed in the subspace Y. For now, it is assumed that the number of clusters is known and is indicated by K; we will discuss the selection of K in Sec. II-B and Sec. II-C. Let $L_{(n\times K)}$ be the cluster indicator matrix which assigns each data point (i.e. spike) x_i to its corresponding cluster $\mathcal{C}_\mathrm{k}$ so that $L_{i,k}=1$ if $x_i\in\mathcal{C}_\mathrm{k}$, and $L_{i,k}=0$ otherwise. The cluster density and separability can be quantized by within- and between-class scatter matrices respectively defined as

$$\begin{aligned} S_\mathrm{w} &= \sum_{k=1}^{K}\sum_{x_i\in\mathcal{C}_\mathrm{k}}(x_i-\mu_k)(x_i-\mu_k)^T \\ &= (X-ML^T)(X-ML^T)^T, \end{aligned} \tag{2}$$

and

$$S_\mathrm{b} = \sum_{k=1}^{K} n_k\mu_k\mu_k^T = ML^TLM^T, \tag{3}$$

where μ_k is the center of $\mathcal{C}_\mathrm{k}$, $M=(\mu_1\cdots\mu_K)$, and n_k is the number of points in $\mathcal{C}_\mathrm{k}$. To achieve a high cluster separability, the within-class scatter should be small and/or the between-class scatter should be large. This leads to the following optimization problem:

$$\max_{W,L}\mathrm{Tr}\frac{W^TS_\mathrm{w}W}{W^TS_\mathrm{b}W} \tag{4}$$

Some solutions to this problem have been presented in [8]. With W fixed, (4) becomes a k-means clustering in feature space Y; with L fixed, it can be solved for W by LDA. A straightforward method is to fix W (initialized by PCA) and use k-means to obtain L, and using the updated L then perform LDA to update W and iterate this procedure until convergence (i.e. L remains the same between two iterations). This method is referred to as *LDA-Km* through which, the data are simultaneously clustered while the discriminative subspaces are selected. Since k-means is used as a building block of the algorithm, the limitations inherent in k-means such as local convergence and sensitivity to initialization also exist here. However, various methods exist to mitigate these problems such as repeating k-means clustering with different initializations and picking the one which leads to the least intra-cluster distance [12].

B. Proposed Algorithm 1

The first algorithm is based on the direct application of LDA-Km to detect the number of clusters and sort the spikes. It is desired to discover in the data as many potential clusters as possible while avoiding over-clustering. In this work we propose a technique which uses the histogram of features in 1-dimensional discriminative subspace, and seeks the peaks of the histogram to infer the number of the clusters in the data. It is assumed that the clusters are denser at their centers and more spread at borders. This transforms to identifiable peaks in the histogram of the features in the most discriminative subspace. Therefore, the number of the peaks would be a reasonable indicator of the number of the clusters.

Simultaneous subspace learning and clustering with different values of K leads to different features spaces. We look for a clustering with the biggest K in which the number of the peaks in the histogram is equivalent to (or not less than) K. As so, we keep increasing K and do the subspace learning until the number of the peaks appearing in the histogram becomes less than K and remains unchanged (no well-separated cluster for bigger K). This method is summarized in Algorithm 1.

It should be noted that, the 1-dimensional discriminative subspace for calculating the histogram is obtained by performing LDA with the iterative trace ratio (ITR) solution [13], which leads to better separation of the clusters in the subspace. Furthermore, similar to other density estimation methods, the parameters used for calculating the histogram such as bin width and smoothing factor, affect the estimation and may alter the number of the observed peaks. Several approaches can be utilized for adjusting these parameters for more reliable density estimation [14].

C. Proposed Algorithm 2

Here, we propose a clustering algorithm based on divisive clustering scheme and use it for spike sorting. In this method, the data are initially partitioned into two clusters (using LDA-Km) in the most discriminative 1-dimensional subspace. The resultant clusters are marked as *child*. After that, the distribution of samples in the obtained subspace is tested for unimodality using Anderson-Darling (AD) test (or similar tests). If the test score is higher than a predefined threshold, which indicates that the distribution is multimodal, each of the *child* clusters are again partitioned into two clusters and marked as *child*, otherwise they are merged into one cluster and marked as *final*. The same process is then repeated for each *child* cluster until all the clusters are marked as *final*. In this process, clusters with sizes smaller than a predefined value, or sparse samples that lie far from cluster centers, can be neglected and assigned as outliers.

The benefit of this method compared with Algorithm 1 is that 1-dimensional subspace is used for clustering which leads to a faster convergence and better handling of clusters with different densities. Moreover, this algorithm has the ability to handle outliers. Since this is a hierarchical clustering approach, a drawback is that misclassified samples in the top levels will affect the clustering results in subsequent levels. The summary of the proposed algorithm is provided in Algorithm 2.

It should be noted that, the threshold controls how similar the distribution of the clusters should be to gaussian to be considered as *final*. In general, a larger threshold value causes the algorithm to extract more spread clusters, whereas a smaller value results in the extraction of more compact and smaller clusters which may lead to over-clustering. Based on extensive tests on synthetic and real data, threshold values of 30–50 (AD test score) seem to provide a reasonable trade-off.

III. Results

Evaluation of spike sorting algorithms is often challenging due to the lack of ground truth [1], [2]. A common approach, however, is to use synthetic datasets, which would provide the ground truth as well as the flexibility to evaluate different characteristics of the spike sorting algorithm. Synthetic neural data is usually prepared by replicating spikes from a few spike waveform templates and adding them to a background noise mimicking neural noise, with the spike arrival time distributions similar to those of real spike trains. In this work, we evaluate our algorithms using synthetic data and compare the results to two other spike sorting methods. Furthermore, we present qualitative clustering results using real *in-vivo* data, in which case the performance of the algorithms can be qualitatively assessed by looking into cluster separation and inter-spike interval (ISI) histogram.

Algorithm 1: Spike Sorting using Discriminative Subspace Learning

```
Input: $X_{m,n}$: $n$ spikes, $m$ samples for each spike
Initialize: $K \leftarrow 1$, $P_1 \leftarrow 0$;
repeat
    $K \leftarrow K + 1$;
    do LDA-Km on $X$ to extract $K$ clusters $L^K$, and
    features $Y^K$;
    perform ITR [13] using $L^K$ to obtain 1-dimensional
    discriminative subspace of $X$;
    estimate the histogram in the 1-dimensional subspace;
    $P_K \leftarrow$ number of peaks in the histogram;
until $P_K = P_{K-1} < K$;
Cluster $X$ using $L^{K-1}$;
```

Algorithm 2: Spike Sorting using Discriminative Divisive Clustering

```
Input: $X_{m,n}$: $n$ spikes, $m$ samples for each spike
Define: $\mathcal{S} = \{S_1, S_2, \cdots, S_I\}$;
Initialize: $K \leftarrow 2$, $\mathcal{S} \leftarrow \{X\}$;
repeat
    foreach $S_i \in \mathcal{S}$ do
        do LDA-Km on $S_i$ to extract $K$ child clusters ($S_i^j$);
        if AD test score on $S_i$ < Threshold then
            Create a final cluster from $S_i$
        else
            foreach $S_i^j$ do
                if Size($S_i^j$) > Minimum Cluster Size then
                    Add $\{S_i^j\}$ to $\mathcal{S}$
                else
                    Assign $S_i^j$ to the Outliers Cluster
            Remove $S_i$ from $\mathcal{S}$;
until All the spikes are assigned to clusters;
```

We compared the two proposed algorithm with PCA-*k*means and `Wave_clus` based on criteria such as accuracy of spike sorting, dimensionality of feature space for effective clustering, and sensitivity to noise.

In PCA-*k*means, PCA is used for feature extraction and dimensionality reduction, and k-means is used for clustering. This method is supervised meaning that the number of clusters (k in k-means) must be known, and parametric assuming the clusters are spherical in the PCA space. `Wave_clus` is an spike sorting method which uses DWT for feature extraction and SPC for clustering. It has been shown very practical for spike sorting since it can automatically choose the number of clusters (unsupervised), and does not impose any assumption on the shape of the clusters (nonparametric). In this method, Haar wavelet coefficients are used as features where the

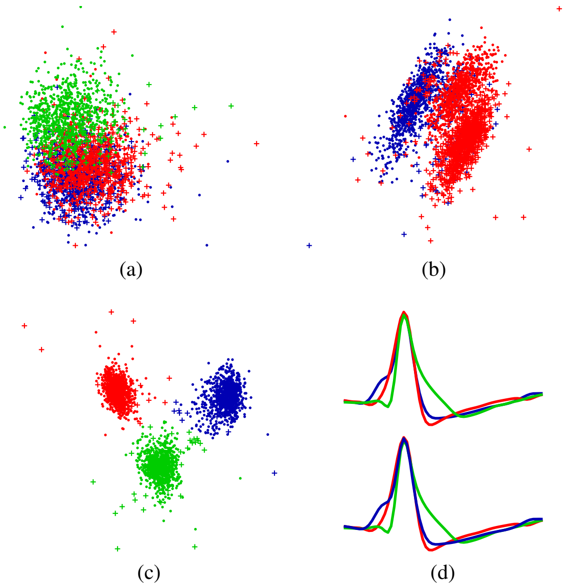

Fig. 1. Results on synthetic data (C_Difficult2_noise02). (a) PCA-$k$means where feature dimension is 10 (projection on the first two principal components is shown), and K is manually set to 3. Accuracy = 60.6%. (b) Wave_clus where feature dimension is 10 (best 2-D DWT projection is shown); SPC fails to identify the third cluster for any temperature value. Accuracy = 58.9%. (c) Proposed Algorithm 1 where feature dimension is 2 (most discriminative projections). The 3 clusters are automatically identified. Accuracy = 98.2%. Misclassified spikes are shown as '+'. (d) Upper plot: Average spike waveforms of the clustering in (c). Lower plot: Average spike waveform of the sorted spikes using Algorithm 2. Accuracy = 98.3%.

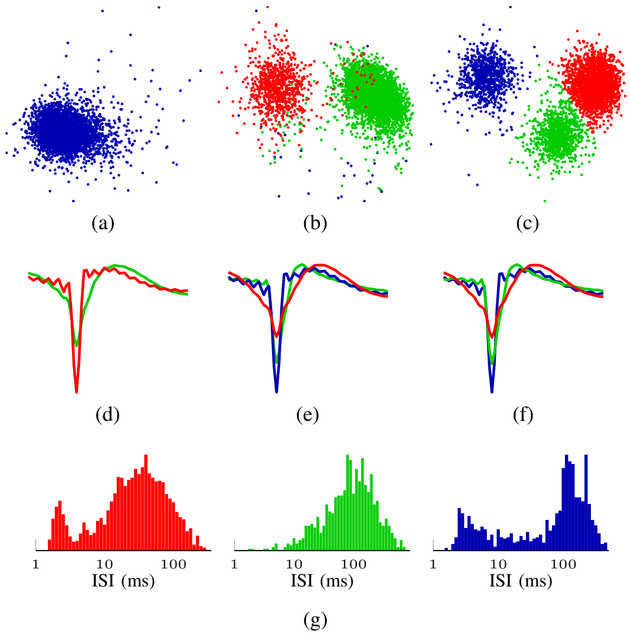

Fig. 2. Comparative results on real data recorded from the rat hippocampus (d1122101:1). (a) Projection on the first two principal components. The clusters cannot be identified through PCA. (b) Best 2-D projection of DWT features used in Wave_clus where feature dimension is 10. Only two clusters are identified (points in blue are assigned as outliers). (c) Discriminative features learned by Algorithm 1, the three clusters are automatically identified. (d) and (e) show the average spike waveforms of the clustering in (b) and (c), respectively. (f) Average spike waveform of the sorted spiked using Algorithm 2. The three clusters are automatically identified and the results are highly similar to those of Algorithm 1. The histogram of the log inter-spike interval of spikes sorted by Algorithm 2, indicating the validity of the extracted clusters. Colors correspond to the spikes in (f).

10 coefficients with largest deviation from normality were selected as the input to the SPC algorithm. In many situations, SPC clusters a subset of the spikes and the remaining spikes are forced to be classified using a KNN classifier.

For simulation with synthetic data, we have used eight challenging datasets provided by [10], which have been widely used in the literature to benchmark spike sorting algorithms. The datasets C_difficult1$\star$ and C_difficult2$\star$ are referred as Set 1 and Set 2, respectively. Each dataset contains spikes from three different neurons. Simulations are carried out for different noise levels and feature dimension of 2–10 for PCA-$k$means and Wave_clus; feature dimension was 2 for Algorithm 1, and 1 for Algorithm 2. The number of clusters is manually set to 3 for PCA-$k$means, while Wave_clus and the proposed algorithms detect it automatically. Throughout our simulations, we used the default parameters of Wave_clus and chose the best results on multiple runs of the algorithm. Wave_clus correctly detected the number of clusters for all cases except for Set $2\backslash\sigma_{\mathrm{n}}{=}0.2$ where it only detected 2 clusters. The proposed algorithms detected the true number of clusters in all the cases. The results are shown in Table I, where the first column is the dimension of the feature space, and the first row is the standard deviation of the noise (σ_n). Overlapping spikes are excluded in calculating the accuracies. It can be seen that the accuracy of PCA-$k$means and Wave_clus degrades as the dimension of the features reduces. Wave_clus (with forced clustering) performs better than PCA-$k$means especially when the noise level is high; however, the performance of both method degrade significantly as the noise level increases. Interestingly, the proposed methods give significantly higher sorting accuracies compared with the other methods and they are highly robust to noise.

Fig. 1 shows the feature extraction and clustering results on the most challenging dataset Set $2\backslash\sigma_{\mathrm{n}}{=}0.2$ (C_Difficult2_noise02). As can be seen in Fig. 1(a)–(c), the 2-D subspace learned by Algorithm 1 provides significantly better separability of clusters compared with PCA and DWT. Fig. 1(d) shows the average waveforms of spikes sorted by Algorithm 1 (upper) and Algorithm 2 (lower).

The results of simulation using *in-vivo* data is shown in Fig. 2. The data is recorded from the rat hippocampus and is publicly available [15]. We chose the recording d1122101:1 which we found challenging for spike sorting. The spikes were extracted by negative amplitude thresholding at 3 RMS, and aligned to their peak values. In total, 4541 spikes were detected and 64 samples were stored for each spike. Fig. 2(a) shows the PCA features on the first two principal components, where only 1 cluster can be identified. Fig. 2(b) shows the best 2-D projection of wavelet features used in Wave_clus, where 2 clusters are identified by the algorithm (average waveforms in Fig. 2(d)). Fig. 2(c) shows

TABLE I

COMPARATIVE RESULTS ON SYNTHETIC DATA

Dim.	Set	PCA-$k$means (%)				`Wave_clus` (%)				Algorithm 1 (%)			
σ_n:		0.05	0.1	0.15	0.2	0.05	0.1	0.15	0.2	0.05	0.1	0.15	0.2
2	1	93.2	75.2	50.8	40.9	97.7	89.0	63.1	62.2	99.6	99.4	99.1	99.2
	2	81.1	74.0	65.8	56.7	66.0	52.6	50.7	49.6	98.7	98.9	98.7	98.2
4	1	97.9	84.7	64.7	53.5	98.7	96.9	93.0	86.5				
	2	91.8	80.8	72.7	63.3	93.3	98.0	91.0	58.0	Algorithm 2† (%)			
10	1	98.9	95.1	85.5	76.8	98.4	98.95	95.4	86.5	98.1	99.2	99.1	98.7
	2	97.8	91.3	83.0	60.6	97.3	98.4	91.7	58.9	98.6	98.7	98.8	98.3

†Feature dimension is 1.

the results of clustering using Algorithm 1, where 3 clusters are detected (average waveforms in Fig. 2(e)). Algorithm 2 also detected 3 clusters whose average spike waveforms are shown in Fig. 2(f) with the corresponding inter-spike interval histograms in Fig. 2(g) which indicate the validity of the extracted clusters.

Regardless of the clustering algorithm used, the methods that rely on PCA for feature extraction usually fail to identify clusters with highly overlapping features in the PCA space, thus assigning spikes generated from different neurons to the same cluster [3], [10]. We have observed this in several *in-vivo* recordings when the spike waveforms of different neurons are very similar. DWT has been shown more effective for spike feature extraction which provides better separability of clusters. However, similar to PCA, it does not necessarily provide the most discriminative feature subspace for clustering, and may fail to correctly discriminate spikes with similar spike waveform when the noise level is high.

IV. CONCLUSION

In this paper, we have proposed two unsupervised spike sorting algorithms based on simultaneous discriminative subspace learning and clustering. Both algorithms can detect the number of clusters automatically. The first algorithm keeps increasing the number of clusters and learns the discriminative subspace until no extra separate and significant cluster can be identified. The second algorithm utilizes a divisive clustering scheme which starts by dividing the data samples into two clusters in the most discriminative 1-dimensional projection, and keeps dividing the resultant clusters in two until achieving an almost unimodal distribution in the subspace for each final cluster. This provides better convergence behavior and improved handling of uneven cluster sizes and outliers. We evaluated our algorithms against two commonly used spike sorting methods PCA-$k$means and `Wave_clus`, using both synthetic and *in-vivo* data. When tested on synthetic data, the proposed algorithms achieved significantly higher accuracies in all the cases. Furthermore, the results indicate that the algorithms are highly robust to noise. When tested on *in-vivo* data, our algorithms provided better cluster separability compared with PCA and DWT, and can find clusters which may not be identifiable in PCA and wavelet space. Through utilizing adaptive subspace learning we have combined to a single stage the three important parts of conventional spike sorting methods including feature extraction, dimensionality reduction and clustering, which resulted in a significant improvement in spike sorting performance.

A few aspects of our proposed methods require further investigation. First, since the methods are based on k-means, they are prone to local optima, and may have problem handling very small sized clusters. Second, the methods require batch processing of spike waveforms and are only suitable for offline setting. It is, however, of practical interest to develop the online versions of the algorithms for real-time spike sorting.

REFERENCES

[1] M. S. Lewicki, "A review of methods for spike sorting: the detection and classification of neural action potentials," *Network*, vol. 9, no. 4, pp. R53–78, Nov. 1998.

[2] S. Gibson, J. Judy, and D. Markovic, "Spike sorting: The first step in decoding the brain: The first step in decoding the brain," *IEEE Signal Processing Magazine*, vol. 29, no. 1, pp. 124–143, 2012.

[3] D. A. Adamos, E. K. Kosmidis, and G. Theophilidis, "Performance evaluation of PCA-based spike sorting algorithms," *Computer Methods and Programs in Biomedicine*, vol. 91, no. 3, pp. 232–244, Sep. 2008.

[4] Z. Yang, Q. Zhao, and W. Liu, "Improving spike separation using waveform derivatives," *Journal of Neural Engineering*, vol. 6, no. 4, p. 046006, Aug. 2009.

[5] E. Chah, V. Hok, A. Della-Chiesa, J. J. H. Miller, S. M. O'Mara, and R. B. Reilly, "Automated spike sorting algorithm based on laplacian eigenmaps and k-means clustering," *Journal of Neural Engineering*, vol. 8, no. 1, p. 016006, Feb. 2011.

[6] V. Shalchyan, W. Jensen, and D. Farina, "Spike detection and clustering with unsupervised wavelet optimization in extracellular neural recordings," *IEEE Transactions on Biomedical Engineering*, vol. 59, no. 9, pp. 2576–2585, 2012.

[7] C. Yang, Y. Yuan, and J. Si, "Robust spike classification based on frequency domain neural waveform features," *Journal of Neural Engineering*, vol. 10, no. 6, p. 066015, Dec. 2013.

[8] C. Ding and T. Li, "Adaptive dimension reduction using discriminant analysis and k-means clustering," in *Proceedings of the 24th International Conference on Machine Learning*, 2007.

[9] K. H. Kim and S. J. Kim, "Neural spike sorting under nearly 0-dB signal-to-noise ratio using nonlinear energy operator and artificial neural-network classifier," *IEEE Transactions on Biomedical Engineering*, vol. 47, no. 10, pp. 1406–1411, 2000.

[10] R. Q. Quiroga, Z. Nadasdy, and Y. Ben-Shaul, "Unsupervised spike detection and sorting with wavelets and superparamagnetic clustering," *Neural Computation*, vol. 16, no. 8, pp. 1661–1687, Aug. 2004.

[11] D. Novak, J. Wild, T. Sieger, and R. Jech, "Identifying number of neurons in extracellular recording," in *4th International IEEE/EMBS Conference on Neural Engineering*, Apr. 2009.

[12] A. K. Jain, "Data clustering: 50 years beyond k-means," *Pattern Recognition Letters*, vol. 31, no. 8, pp. 651–666, Jun. 2010.

[13] H. Wang, S. Yan, D. Xu, X. Tang, and T. Huang, "Trace ratio vs. ratio trace for dimensionality reduction," in *IEEE Conference on Computer Vision and Pattern Recognition*, Jun. 2007.

[14] B. W. Silverman, *Density Estimation for Statistics and Data Analysis*. Boca Raton: Chapman and Hall/CRC, 1986.

[15] [Online]. Available: http://crcns.org/data-sets/hc/hc-1