**A study of deep clustering in spike sorting**

Eugen-Richard Ardelean1, Raluca Laura Portase1

1Department of Computer Science, Technical University of Cluj-Napoca, Cluj-Napoca, Romania

**\* Corresponding authors:** [ardeleaneugenrichard@gmail.com](mailto:ardeleaneugenrichard@gmail.com), [placehoder-raluca]

**ORCID Author IDs:**

Eugen-Richard Ardelean: 0000-0002-0098-4228

Raluca Laura Portase: 0000-0002-8985-4728

**Abstract: [**TBWritten]

**Keywords:** clustering, [TBAdded]

# Introduction

## Spike Sorting [TBRewritten]

The individual activation events of a neuron are called action potentials or spikes. Attributing such a spike to the specific neuron that produced it based on its characteristics is referred to as spike sorting [1]. Spike sorting, by definition, handles extracellular recordings, which capture the activity of multiple neurons in the proximity of the recording electrode [2] thus, the neuron of a spike is unknown at the time of recording. The main assumption of spike sorting is that each distinct neuron tends to generate spikes of similar shapes [3], yet markedly different from the shapes of spikes of other neurons. In reality, the shape of spikes is muddled by the background noise, inducing variability, which generates a cluster in the feature space instead of a single point. Therefore, it is important to find or generate features that are able to separate the spikes and that are preferably as few as possible.

The spike sorting pipeline can be broken up into four sequential steps [1]: filtering, spike detection, feature extraction, and clustering. Importantly, the separability of clusters is driven by the feature extraction technique and not by the clustering method. Here, we investigate the impact of different feature extraction techniques on the separability of resulting clusters, nevertheless the other steps of the spike sorting are topics of ongoing research in this domain as well. We would like to draw attention to the fact that a golden feature extraction method does not exist and the performance of each depends on the characteristics of the data [1] [4]. Here, we employ …

The spike sorting pipeline can be modified depending on the approach used, offline or online. In offline spike sorting, the sorting is done only after the data acquisition, while in online it happens during. In the filtering step of the raw signal, a band-pass filter is applied in order to isolate the relevant frequency band (usually between 300 and 3000Hz [4]) where the spike’s frequency components are expressed. Next, spike detection typically involves amplitude thresholding, while compromising between missing spikes and including noise in the data. The third step, and the focus of this study, is the feature extraction step, whereby the most informative features are identified and extracted in order to reduce the dimensionality of the data and reduce the computation load of the clustering while maintaining the data separability. In the final steps, spikes are clustered in the feature space such that similar spikes are separated into groups, each group assumed to have been generated by the same neuron. Alternatively, a supervised manual approach was commonly used where the researcher could classify spikes by hand. Nonetheless, such methods are rapidly becoming impractical as new multi-array probes are developed [5] as the number of recorded neurons has seen an exponential increase since the 1950s [6]. A template matching approach that is applied on a subsampled set of data has become increasingly popular and can substitute of the spike detection and feature extraction steps, while also being computationally efficient [7].

## Deep Clustering Algorithms

## The challenges of spike sorting [TBRewritten]

The process of spike sorting is challenging due to an array of difficulties. First, because neuronal firing occurs on millisecond timescales, even relatively brief recordings generate an abundant data volume [3]. Second, rather than being stationary, the activity of neurons is regulated by brain circuits such that they can fire with markedly different firing rates [13] [14]. This results in different relative frequencies at different times, leading to clusters of different sizes and an inherent imbalance in the data. Many clustering algorithms have difficulties tackling imbalanced data especially when coupled with overlap. Finally, in practice various phenomena can alter or contaminate the estimated spike shape, such that clusters are not always distinct, but often overlap. Single unit activity is defined as the activity of a single neuron that can be separated as a single cluster, while the activity of distal neurons is represented in the signal as low amplitude spikes and most often cannot be separated due to a low signal-to-noise ratio and as such, is denominated as multiunit activity [4].

The aim is to create a representation that is unaffected by slight changes in waveform shape as a result of noise and phenomena such as the electrode drift that may modify the shape of the waveform. [ADD EXAMPLES OF ALGOS AND WHY THEY WOULD BE GOOD]

The paper is organized as follows: section 2 presents a critical view of conventional feature extraction methods used in spike sorting, provides a description of the proposed method, and presents the datasets and metrics used in the analysis. In section 3, the methods are evaluated considering multiple metrics and their performances are interpreted critically. Section 4 discusses the limits of the proposed method and the conclusions we have reached.

# Materials and Methods

## State of the art Feature Extraction [TBRewritten]

As stated above, a crucial step in spike sorting is the description of spikes with a compact set of informative features. The aim of dimensionality reduction is to transform a dataset with a dimensionality of X into a dataset with Y dimensions, where Y<<X. Another important aim is to retain as much of the data geometry as possible, such that relations in the original space are retained in the reduced space, which is especially useful for spike sorting. Dimensionality reduction techniques can be divided by several criteria, such as: convexity or linearity [19]. From the point of view of convexity, PCA is a convex algorithm, while Isomap is a non-convex approach. Among the first features used in the spike sorting were the spike amplitude and its width [20]. Afterward, methods based on probabilistic models, created through empirical analysis, that used the entire waveform were developed [21]. These could process a low number of electrodes. Shortly thereafter, transforms started being used to project the high-dimensional space of the waveform into a low-dimensional space through the use of principal components [22], the wavelet transform [23] and various combinations of them. Manual sorting of spikes is usually performed on a low dimensional space, containing features such as the amplitude, the peak-to-trough ratio, etc [24]. The peak-to-trough ratio was found to be representative of the neuron type, inhibitory neurons produce narrow spikes and thus have a small peak-to-trough ratio, while excitatory have a large ratio [25].

In [26], the authors propose M-Sorter, an automatic method for spike detection and classification based on coefficients obtained through the wavelet transform and template matching. The proposed method separates spike sorting into two steps, the spike detection by multiple correlation of wavelet coefficients on the band pass filtered waveforms of the recorded signal and template matching for the classification of spikes to the neurons that generated it. The multiple correlation of wavelet coefficients is also used in the generation of templates through the application of K-Means. Each spike is assigned to the cluster to which it has the smallest distance.

### Linear feature extraction methods

Principal Component Analysis (PCA) [27] is the most frequently used algorithm for feature extraction, including spike sorting [28]. PCA projects the spikes onto new characteristics called Principal Components that are a new set of orthogonal axes formed by linear combinations of the input features. The reduction of dimensionality of the feature space is performed by solving a problem of eigenvalues and eigenvectors. By retaining the most prominent principal components, PCA preserves the variance as much as possible while being able to reduce the number of features. It is common to keep only the first two or three principal components resulted from PCA [29] [30]. These frequently retain more than 70% of the variance from the original space. However, variance does not necessarily offer the best separation [1] [4]. To put it in another way, information required for separability may be encoded in those low-variance features that are discarded. Finally, PCA and its variations have been used in spike sorting for a long time [4] and it is still used in recently developed spike sorting pipelines [31].

Another linear method is Independent Component Analysis (ICA) [32] mainly designed for source separation. ICA is a linear unsupervised technique for dimensionality reduction that searches for independent components by relying on the statistical properties of the data. ICA has been previously applied to spike sorting with promising results [33] [34].

Linear Discriminant Analysis (LDA) [35] is a supervised linear learning technique with the goal of increasing the inter-cluster distance and decreasing intra-cluster distance. LDA assumes that the data has a Gaussian distribution. However, for our problem LDA is not a fit candidate due to several considerations. First, it is a supervised learning technique which cannot be applied to unlabelled data, as is the case in spike sorting. Second, the Gaussian distribution assumption is often violated in spike sorting due to: electrode drift, shape variation from bursts, simultaneous firing, multi-unit activity, and non-stationary background noise [1].

### Non-linear feature extraction methods

In the category of unsupervised non-linear dimensionally reduction techniques Isomap [36] uses Isometric Mapping to learn the low-dimensional projection in a manifold space while retaining the distances of the original space. It uses the geodesic distance, which can be thought of as the shortest path along the curved surface of the manifold space.

T-distributed Stochastic Neighbor Embedding (t-SNE) [37] is a non-linear dimensionality reduction method that minimizes the divergence between input features and the reduced feature space by using pairwise probability similarities. The divergence of two distributions is calculated using KL divergence, which is minimized by applying gradient descent. Due to its high time complexity, several orders of magnitude higher than PCA, and its main function being visualization, t-SNE was not considered a suitable candidate. A computation of a few seconds for PCA can become tens of minutes for t-SNE. Furthermore, from empirical observations, the separation offered by t-SNE for the datasets used here was small to non-existent.

## State of the art Clustering algorithms [TBRewritten]

## Preprocessing [TBRewritten]

Alignment has to be applied as a first step of preprocessing before the execution of the feature extraction method. We have used the following formula, for multiple types of alignment at a chosen index:

Naturally, the point of start of a portion of the samples has to be changed; this is indicated in formula (1) through the *new\_start* and *old\_start* terms. The *index* in equation (1) represents the point to which all spikes will be shifted. Thus, we can choose to align the maximum peak of all spikes to the average index of the maximum peak across all samples, as shown in Fig 4c. Another, better option, is to align the amplitudes to the middle of the sample as it provides information about the spike from the perspective of both pre- and post-amplitude. The *peak* in equation (1) represents the index at which the desired point of reference (typically, the peak) is found. For the alignment of the amplitude, it is the index of the maximum peak of each sample. The formula permits the alignment of any point of reference, such as the minimum peak [44].

In addition, we have applied two other preprocessing steps: scaling and shuffling.

Chart, scatter chart

Description automatically generated

**Fig 4. Impact of alignment.** PCA applied on Sim14 with and without alignment. The white cluster is kept together but the overlap with the blue cluster remains.

## Data [TBRewritten]

### Synthetic datasets

The validation of deep clustering methods was made by comparing the different methods with [\*\*\*PCA, ICA and Isomap\*\*\*]. The chosen datasets, 95 in number and denominated as simulations, originate from the Department of Engineering, University of Leicester UK and are publicly available. Each simulation is a dataset. The creation of these simulations was based on recordings from the neocortex of a monkey. They were generated using 594 different spike shapes [45]. The original study that introduces the simulations [45] also reviews different clustering algorithms and their results. Out of 20 different units, these algorithms were able to detect 10 in the best case.

The datasets were generated based on a real dataset recorded “in vivo”. The waveform contains 316 points originally sampled at 96 KHz; afterwards this frequency was reduced to 24KHz, therefore 79 samples describe a spike. Being synthetic datasets, each of these spikes has a label, which allows for the use of external metrics to evaluate performance. Each simulation contains a multi-unit cluster, which is the noise, and a number of clusters that varies between 2 and 20. Each unique number of clusters has 5 simulations. Thus, there are 5 simulations with 2 clusters, 5 simulations with 3 clusters, and so on.

All but one of the clusters are single-units between 0 and 50μm away from the electrode. The firing rate follows a Poisson distribution with a mean between 0.1 and 2Hz. The amplitudes follow a normal distribution and have been scaled to values between 0.9 and 2 to simulate real data. No spikes with temporal overlapping are present in the data, such that spikes have at least 0.3ms between them.

The generated multi-unit cluster was added in order to increase the complexity of clustering for the tested algorithms. The simulated multi-unit contains 20 spike shapes, each of the 20 neurons firing being between 50-140μm away from the electrode. The amplitude of the spikes was fixed to 0.5, with an overall composite firing rate of 5Hz, with each of the 20 individual composing neurons having a firing rate mean of 0.25Hz following an independent Poisson distribution. Here, in order to increase clarity, the multi-unit cluster is always color-coded in white in all figures.

To evaluate the proposed approach in comparison with other state-of-the-art methods we have chosen the following 4 simulations out of the 95 available as they are representative of the issues that are present in feature extraction methods and allow for the evaluation of the methods on varying numbers of clusters covering a wide range and enabling a comprehensive evaluation of performance:

* Simulation 1 (Sim1 - Fig 3a), containing 16 single-unit clusters and a multi-unit cluster (in total 17) with 12012 samples.
* Simulation 4 (Sim4 - Fig 3b), containing 4 single-unit clusters and a multi-unit cluster (in total 5) with 5127 samples.
* Simulation 16 (Sim16 - Fig 3c), containing 8 single-unit clusters and a multi-unit cluster (in total 9) with 7556 samples.
* Simulation 35 (Sim35 - Fig 3d), containing 12 single-unit clusters and a multi-unit cluster (in total 13) with 9481 samples.
* Simulation 14, containing 3 single-unit clusters and a multi-unit cluster (in total 4) with 4507 samples. This dataset was used for the visualization of the impact of alignment on feature extraction in section 3.1.

These simulations can also be viewed in Fig 3 through the use of PCA to reduce the dimensionality from 79 to 2. The overlapping clusters produced by PCA can be clearly seen in Fig 3, in none of the datasets is it able to perfectly separate all clusters.

Scatter chart

Description automatically generated

**Fig 3. PCA projection of the synthetic datasets.** PCA projection of 4 different simulations with distinct numbers of clusters; the colors represent the cluster assignment given in the ground truth.

### Real datasets [TBRewritten]

The electrophysiological *“in vivo”* data was recorded from the brain of anaesthetized adult mice of the C57/B16 strain with A32-tet probes (NeuroNexus Technologies, Inc) at 32 kSamples /s (Multi Channel Systems MCS GmbH) during a visual stimulation. The stimuli were presented monocularly on a Beetronics 12VG3 12-inch monitor with a resolution of 1440x900, at 60fps and consisted of full-field drifting gratings (0.11 cycles/deg; 1.75 cycles/s; variable contrast 25–100%; 8 directions in steps of 45°). The animals, on which the extracellular activity was recorded, were placed in the stereotaxic holder (Stoelting Co, Illinois, United States) and anaesthetized. Anesthesia was induced and maintained with isoflurane (ISO) in oxygen (5% for induction, 1-3% for maintenance). The heart rate, respiration rate, core body temperature, and pedal reflex were constantly monitored. A circular craniotomy (1x1 mm) was performed over the left visual cortex of the animal centred on 0-0.5 mm anterior to lambda, 2-2.5 mm lateral to midline. To obtain multiunit activity (MUA) containing signals, the extracellular data was digitally filtered using a band-pass filter with a range of 300Hz-7000Hz using a bidirectional Butterworth IIR filter of order 3. An amplitude threshold, most commonly chosen between 3 and 5 [1] standard deviations of the recorded signal, was used to detect spike, which were then fed into the feature extraction algorithms. Spikes were identified as threshold crossings and subsequently used as input for the feature extraction algorithm.

Multiple datasets were accumulated from each animal over a period of 4 to 6h in order to minimise animal use. All experiments were performed in accordance with the European Communities Council Directive of 22 September 2010 (2010/63/EU) and approved by the Local Ethics Committee (3/CE/02.11.2018) and the National Veterinary Authority (147/04.12.2018).

## Performance metrics [TBRewritten]

Six metrics were used for the validation of results: Adjusted Rand Index (ARI), Adjusted Mutual Information (AMI), V-Measure (VM), Calinski-Harabasz Score (CHS), Davies-Bouldin Score (DBS), and Silhouette Score (SS). The first three metrics are external metrics, while the last three are internal [46]. These are clustering performance metrics and are a suitable evaluation of the feature extraction due to the fact that spike sorting does not stop at this step but is followed by clustering. The external metrics provide information about the ability of the clustering algorithm to correctly identify the clusters based on a ground truth, which is heavily influenced by the separability offered through the feature extraction. This is due to the fact that with perfect separation, most clustering algorithms will be able to have a high performance. On the other hand, the internal metrics characterize the clustering based on the separability and shape of clusters, thus they are adequate for the evaluation of the feature extraction through the use of the ground truth labels for synthetic datasets. In fewer words, the internal metrics outline the properties of the clusters, while the external metrics evaluate the matching between the clustering and the ground truth. In Table 1, we present a short intuitive description and the range for each of the metrics.

We chose a multitude of evaluation metrics rather than an all-encompassing one, as they will appraise the performance from multiple considerations and perspectives. Thus, a method that provides greater performance across these numerous metrics is indicative of a balanced performance with an increased likelihood of an unbiased evaluation.

**Table 1.** An intuitive description for each metric, its type and its range.

External metrics require the labels of the clustering algorithm, and the ground truth labels. Therefore, a clustering algorithm has to be applied after the feature extraction and we have chosen K-Means [47]. K-means has a long history of use as a clustering algorithm and many variations have been developed. It was introduced in spike sorting in 1988 and remained the de facto clustering algorithm for a long time [48] [49]. Furthermore, newly developed spike sorting techniques and pipelines are based on it or use it [7] [50] and in recent evaluations K-Means has been shown to still be a highly performant option, as it placed third in the evaluation of 25 clustering algorithms [48].

K-Means is a partition-based clustering algorithm. It partitions the space into k partitions, where each sample is appointed to the closest centroid based on the Euclidean distance. K-Means has several disadvantages. First, it requires the number of clusters as an input which is hard to provide for real data. Second, in its most basic form it is not deterministic, such that each execution may result in a different clustering. Through recent optimizations it has been improved and has increased stability. Third, K-Means has difficulties in separating overlapping clusters. In our case, this is an advantage: If the performance of K-Means is higher for a certain feature extraction method it denotes that the method provides better separation.

ARI [51] [52] [53] (3) is an adjustment of the Rand Index (RI) metric in order to handle chances. ARI is an external clustering metric; therefore, it requires a ground truth for the dataset. RI [54] (2) makes comparisons between pairs of points to determine if it is an agreement, when the two points are in the same cluster for both the predicted and the true labels, or a disagreement, when they belong to different clusters. The formulas used to calculate the metric are the following:

where *MaxRI* is the upper bound and *ExpectedRI* is theexpected placement of pairs in the same class using the permutation model and calculated based on the contingency table [51].

AMI [52] [55] (5) is an adjustment of the Mutual Information (MI) metric through the use of entropy, denoted as H. Moreover, AMI also contains the normalization [52] [56] [57] of Normalized Mutual Information. MI (4) is calculated between two clusters U and V, where N is the size of the dataset and |X| is the number of points in subset X.

V-Measure [17] (6) is the harmonic mean of Homogeneity and Completeness. A cluster is considered to be homogeneous (7) when all the points of that cluster are part of the same class. By switching the predicted and true labels, completeness is obtained. Completeness (8) is achieved when all the points of a class are part of the same cluster. We have chosen beta equal to 1 as given by the original formula [17].

where H(C|K) is the conditional entropy of the true cluster given the predicted cluster, H(C) is the entropy of the true cluster, while H(K|C) is the conditional entropy of the predicted cluster given the true cluster and H(K) is the entropy of the predicted cluster.

All the metrics presented until this point are external metrics and require a ground truth to compare with the predicted labels. Furthermore, all these metrics have bounded scores in the [0, 1] interval with higher values being more desirable.

The following three metrics are internal and therefore do not require a ground truth to be used. The internal metrics were used with the ground truth labels for the evaluation of the synthetic datasets. These metrics evaluate the intra-cluster and inter-cluster distances and the morphology of the clusters producing an adequate evaluation of the feature extraction capabilities.

DBS [58] [59] [60] (10) finds the mean similarity between clusters, where similarity, denoted as *R* (9), is defined by the distance between clusters and their sizes. The minimum value of this index is 0. The closer the result is to 0, the better separation exists between clusters. This may come as counterintuitive as it is the only metric where lower values represent a higher performance. The DBS metric is given by the following equations:

where *si* is the mean of all distances between the points of cluster *i* and its centroid, *di,j* is the distance between clusters *i* and *j* given by their centroids, and *max(Ri,j)* is the maximum similarity of clusters *i* and *j*.

CHS [46] [17] (11), also known as Variance Ratio Criterion, calculates the ratio between the intra-cluster and inter-cluster dispersion. Where *tr(X)* denotes the trace of between cluster *Bk* or within-cluster *Wk* dispersion matrix, *n* denotes the size of the dataset and *k* the number of clusters. The dispersion is defined as the sum of squared distances. For this metric, a higher value indicates a better result.

SS [17] [18] (12) is calculated by measuring the mean distance between a point and the rest of the points of that cluster and the mean distance between the point and all the points of the nearest cluster. The score is bound between [-1, 1] where -1 represents an incorrect clustering, 0 overlapping clusters, and 1 a dense clustering. SS aims for the standard concept of a cluster, dense and well separated, therefore such cases will give a higher score. The equation of SS is the following:

where *b* denotes the average of all distances between a point in cluster *i* and all points of the nearest cluster *j*, and *a* the average of all distances between a point in cluster *i* and all other points in the same cluster.

It is important to mention that although used in evaluation of spike sorting techniques [15] [16], accuracy is not a suitable performance metric. First, because spike sorting is unsupervised and accuracy requires labels. Second, neuronal data is imbalanced because of the various firing rates of individual neurons, and it is has been extensively shown that accuracy is not appropriate for evaluating tasks on imbalanced data [61] [62] [63] [64]. Nevertheless, through the use of the chosen metrics, we are able to evaluate the separation and shapes of cluster using the internal metrics and the correctness of clustering using the created features using the external metrics.

## Clustering validation scores [TBRewritten]

Internal vs external – discussion from edging distance

Table1

|  |  |  |  |
| --- | --- | --- | --- |
| Name | Type | Description | Range [worst, best] |
| ARI | External | Pair-by-pair comparison whether the points in the predicted cluster belong in the same true cluster | [-1, 1] |
| AMI | External | Mutual information based on entropy is used to calculate the agreement of true and predicted labels | [0, 1] |
| Purity | External | Cluster homogeneity as the majority class assignment. | [0, 1] |
| DBS | Internal | Ratio of the inter-cluster and intra-cluster sum of squared distances | (Inf, 0] |
| CHS | Internal | The average of a function that evaluates inter-cluster distances and the size of the cluster | [0, Inf) |
| SS | Internal | Cluster quality is evaluated as the balance between a cluster’s tightness and separation | [-1, 1] |

# Results [TBRewritten]

## Performance evaluation

### Performance evaluation of synthetic data

The 95 synthetic datasets [45] contain varying numbers of clusters and spike shapes providing the complexity required for a comprehensive evaluation of the methods. In Fig 5, we present the results obtained for each metric across all 95 datasets for each method presented. A statistical analysis using t-tests with a Bonferroni correction can be examined in the S4 Fig and a ranking of the methods based on their performance for each metric using Borda rank aggregation [65] in S2 Table.

### Performance evaluation of real data

# Discussion