UCL Université catholique de Louvain

École polytechnique de Louvain (EPL)

Application of feature subset selection on meaningful features leads to promising results for spike sorting

Dissertation presented by
Cédric TITS

for obtaining the Master's degree in
Biomedical Engineering
Option(s): Mathematical Modeling & Life Sciences Technologies

Supervisor(s)
Michel VERLEYSEN , Karl FARROW

Reader(s)
Julie DUQUÉ, Philippe LEFÈVRE

Academic year 2015-2016

Abstract

Understanding the visual system is a real interest since it contributes to one of the most important sense for human: the vision. To achieve this goal and with a focus on neuronal responses, spike sorting is generally applied on electrophysiological recordings of neural activity evoked by light stimuli. In this process, Principal Component Analysis (PCA) represents the tool commonly utilized for the feature extraction step. However, results are not perfect and knowledge on the phenomenon is not taken into account in this technique. Based on that, this work proposes a set of features directly extracted from the spike waveforms on which feature subset selection algorithms are applied. It shows moreover that these embedded methods, using the best of filter and wrapper tools, lead at least to similar results than PCA, and sometimes to better performance levels. Furthermore, a new technique for the selection of relevant features is proposed. This combination could give rise to a decrease of the time required to apply spike sorting due to the dimension reduction it allows...

ACKNOWLEDGEMENTS

First and foremost, I would like to express my gratitude towards my supervisor Karl Farrow for his follow-up, proximity, pertinent remarks and interest about the ideas I have proposed throughout this research.

I am also thankful to my supervisor Michel Verleysen for his guidance and the structure he brought me in the creation of this work.

Finally, I am highly grateful to the imec company for its welcome and the infrastructure it makes available to me to complete successfully this document

TABLE OF CONTENTS

List of Tables

Table **Page**

List of Figures

Figure **Page**

Part I

Introduction

CHAPTER 1

INTRODUCTION

One of the particularities of the human being is its ability to move in its environment. The combination of vision and proprioception informaition contributes to the realization of this action. Proprioception can be defined as the perception, conscious or not, of the relative position of the different parts of the body. Besides that, vision represents the ability to sense light rays. Together, they are indispensable to movement, because they allow the individual to determine and estimate the distance to cover between the starting and ending points of the displacement.

With a focus on the human vision, a well-structured network processing the information can be identified. The eye captures light rays. At the level of the retina, the back-end of the eye, photoreceptors, which are nothing other than nervous cells, ensure the transduction/transformation of the visual flux in an electrical flux under the form of nervous impulses, which can be seen as electrical discharges. The optical nerve conveys the nerve flux through the chiasma, a physical localization where the nerve fibers, which are trapped in the optical nerve until this point and now uncovered, cross each other. Their expansions form the optical tracts which decompose into optical radiations that project themselves on the visual cortex, the part of the brain processing the visual information (more information available in [73]).

To form this visual network, small pieces are linked together to integrate and transmit the nervous signal between the periphery, represented by the eye, and the kernel,

represented by the brain: the neuron. The activity that can be recorded from them is named action potential, if related to the internal event, or spikes, if related to the waveform captured extracellularly [30].

1.1 Motivation

It has been shown that certain neurons or neuron populations respond specifically to stimuli. In other words, these neurons discharge in presence of a certain pattern in the stimulus, while others do not show such a behavior [50]. Hence, it is normal to think that each of these neuron populations gives rise to responses with their own characteristics [71].

The study of these particularities is of main interest in the understanding of the brain functioning. Among all techniques allowing the identification and characterization of neurons, electrophysiology studies the bioelectrical activity of living tissues, recorded via intracellular or extracellular implanted electrodes [30, 69].

1.2 Work statement

Building from this, the aim of this work is to bring a contribution to the understanding of the visual system with a focus on neuronal responses.

As previously said, to achieve this goal, electrophysiology is used. To analyze the data coming from the recordings of this technique, spike sorting is applied. Briefly, this procedure tends to extract the individual responses of neurons from the recorded general neuronal activity (see section 2.1 for a detailed explanation).

In the beginning, neuroscientists performed manually spike sorting based on their knowledge and the observation of the recorded signals. However, due to the limited space in which data can be observed, which leads to human biases [36, 58] and with the increasing quantity of information that is possible to record and store, this task became rapidly time-consuming [76] and intractable for neuroscientists. That is why machine learning algorithms, algorithms improving automatically their performance

with experience, have emerged to help them in the automation of this process.

However, in most of these algorithms, the insight/knowledge of the neuroscientists is not included and strong data processing tools are often preferred. This work proposes to approach the spike sorting problem in a different way. Indeed, the ultimate goal of spike sorting is to obtain, at the end of the process, the individual spike responses. The latter ones give then rise to an analysis of their specificities/particularities. Why not combine a part of this step within the spike sorting technique with an identification of important characteristics?

With this insight and trying to think as a neuroscientist, this work proposes to extract some particular features from the spikes identified in the spike sorting procedure instead of applying strong technical tools. However, due to the lack of knowledge for the field of neurosciences, it is obvious that some of the chosen features may be uninteresting. To alleviate this problem, feature subset selection tools are applied to select the most relevant ones for trying at the end to give rise to better results than Principal Component Analysis (PCA), a machine learning tool commonly applied for spike sorting.

1.3 Overview

Chapter 2 defines the terms and provides an overview of the general process of spike sorting and feature subset selection, explaining the different steps of the procedure for each of their application. That allows to introduce the following terms: filtering, spike detection, feature extraction, clustering, filters, wrappers, etc. Its goal is to initiate the reader to the foundations of the work idea and to provide the basic knowledge required to understand the next chapters.

Chapter 3 surveys the literature for the spike sorting process, including feature subset selection, defined in chapter 2. Furthermore, it allows the reader to become aware of the main limitations of the techniques used for now and explains the necessity of a novel approach for this problem.

Chapter 4 proposes the new methodology investigated in this work to enhance the results of spike sorting. It starts with an explanation of the approach, details the database

that is used and defines the different features which are extracted from the spikes and considered as reflecting the neuroscientist's insight used during the manual spike sorting. Then, it describes the feature subset selection algorithms selected to rank the features by degree of relevance. Finally, the clustering algorithm and the evaluation criteria employed to assess the quality of this choice are defined.

Chapter 5 discusses the results got from the database and proposes a new method for selecting the relevant features.

Chapter 6 and 7 provide a discussion based on the obtained results, conclude and suggest future work.

Part II

Review and methods

CHAPTER 2

BACKGROUND

This chapter describes the general processes of spike sorting and feature subset selection. Its goal is to introduce and to describe these techniques so that the reader can understand the next chapters developing the ideas and general objectives of this work. At the end, a small section explains the interest the interactions between these techniques can bring to the problem to solve.

2.1 Spike sorting process

The brain involves a huge quantity of neurons, firing or not, depending on the context and the different stimuli presented to them [30]. To study this aptitude, experiments are performed and generally conducted as follows. For a certain time, a trial is performed and recorded. After this period, data acquisition stops and a new trial is carried out until obtaining the desired number of trials. It is then possible to build a database, called A (for acquisition) and within which the recordings are stored, each row representing a trial and noted A_i. Each column of A, noted A^j, represents an acquisition instant. Hence, if M acquisition instants and N trials are considered, the data matrix can be written as:

$$A_{[N\times M]} = \begin{bmatrix} — & A_1 & — \\ — & A_2 & — \\ & \vdots & \\ — & A_N & — \end{bmatrix} = \begin{bmatrix} | & | & & | \\ A^1 & A^2 & \cdots & A^M \\ | & | & & | \end{bmatrix} \tag{2.1}$$

Based on these recordings and hence data, the pretension of spike sorting is to identify populations of neurons with similar responses to stimuli. Spike sorting is then defined as the process of setting spikes into groups, based on the similarity of their shapes or waveforms [74, 80]. The general procedure guiding spike sorting is presented on figure 2.1.

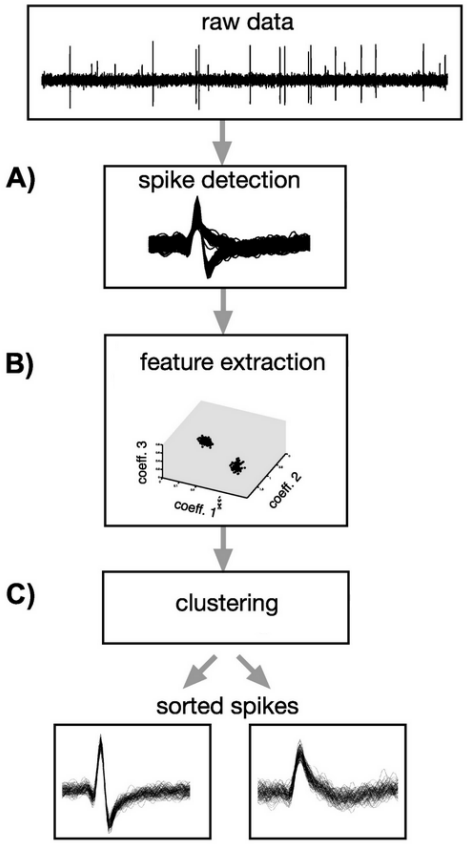

Figure 2.1: General spike sorting process. A) Extraction and superposition of the spikes coming from the raw data. B) Feature extraction. C) Clustering of the spikes from the extracted features to obtain the sorted spikes. Figure adapted from [76].

Shortly, spike sorting carries out as follows. After data acquisition, a filter is applied on the raw data to try to extract the most interesting information, the data of interest, from the recorded signal containing background noise. The second step of the process is to identify the firing instants and extract the different parts of the signal associated to them. These parts represent spikes and the technique to obtain them is called spike detection. The third step consists in the definition of features from the isolated spikes. Finally, based on the similarity of their features, the different spikes are clustered, are put into different groups. To do it, clustering algorithms are used. At the end of the procedure, the spike groups are returned and characterize the firing neurons.

For an easier understanding of the general process of spike sorting, the next subsections explain more precisely the different steps of the procedure.

2.1.1 Filtering

During electrophysiology, recorded data are most of the time noisy, due to different artifacts such as movements, outside electrical and magnetic devices, etc [30]. Hence, after data acquisition, filtering is used (not explicitly represented on figure 2.1). This first step consists in the application of a filter on the raw data to enhance the Signal to Noise Ratio (SNR), what measures of the quality of the signal. If the SNR is too low, that means that the signal is highly noisy. In the other hand, if the SNR is high, the signal is clean and not noisy.

To accomplish this step, the choice of the filter is really important, because some filters have the particularity to distort the signal [75], what is not desirable in this case since the spike shapes have to be conserved. In addition to that and since the recorded signal contains a large range of frequencies, parasitic or not, the filter must be band-passing, allowing hence some chosen frequencies [80]. At the end of this process, the matrix A, presented at equation 2.1, is modified in a matrix $\tilde{A}$.

2.1.2 Spike Detection

The actual first step of the spike sorting process is spike detection, in other words, the detection and the extraction of spikes within the filtered signal (corresponding to step A

on figure 2.1). This process bases itself on the fact that spikes have an amplitude higher than the surrounding noise. Once their position is localized, they are extracted out of the signal and the sequence of the process can continue. Hence, a new matrix X is defined, constructed from $\tilde{A}$ and for which each row contains a single spike (noted S_i):

$$X = \begin{bmatrix} — & S_1 & — \\ — & S_2 & — \\ & \cdots & \\ — & S_K & — \end{bmatrix} \tag{2.2}$$

It must be noted that the number of points representing S_i's is smaller than M. Furthermore, K, the number of spikes, is usually higher than N, since it is reasonable to think that several spikes are extracted from each recording. M and N are to refer to equation 2.1 of section 2.1.

2.1.3 Feature Extraction

Depending on the current filter and the frequency at which the original data are recorded, each extracted spike can be represented by a large number of points, each point being a measure instant, as described in section 2.1.

It can be interesting to reduce the size the database X (see equation 2.2), since working with a large amount of data is computationally expensive, can lead to overfitting (a property of algorithms that model really well data but are not able to generalize their results to other data) when models have to be built from them and can show a tendency to the curse of dimensionality (the fact that algorithm cannot utilize low dimensional metrics in high dimensional spaces) [6, 8]. To avoid these problems, feature extraction is utilized as second step of the spike sorting process [2] (corresponding to step B on figure 2.1).

The goal here is to find new features that most represent the spikes and which can retain as much information as possible from the database. That is based on the fact that a neuron may generate spikes with unique extracellular signal features [30, 71, 74]. This

step gives rise to an easier application of the clustering algorithm [19, 33].

2.1.4 Clustering

Once the features are extracted, the clustering or unsupervised classification step (corresponding to step C on figure 2.1) can be performed. It consists in the creation of groups by using a measure of the similarity between features [19]. This similarity can take several definitions and, depending on them, some algorithms are preferred or not. The ultimate goal of this process phase is to distribute the spikes into different clusters, to finally identify populations of neurons with equivalent activities or at least similar stimuli related responses.

2.2 Feature subset selection process

Practical machine learning algorithms are known to degrade in performance when the number of irrelevant features for predicting the output is too high [49].

Feature subset selection, also called selection of variables, attribute selection or selection of a subset of variables [63], is a process of selecting a feature subset of interest in the creation of a model [48]. Coupled with unsupervised learning, characterized by the absence of class labels, it is a difficult task whose the goal is to find the feature subset that best discovers "natural" groupings from data [26]. It has the particularity to choose features from the original features, and does not construct new ones [49, 57, 58, 72]. To do this, the algorithms targeting this problem must withdraw/cancel features which are not of interest or which are redundant, on the basis of an evaluation criterion (see subsection 2.2.2 for more details) [2, 58]. Interactions between features must hence be taken into account in the process. These techniques are utilized to simplify models in eradicating inappropriate and unnecessary features, to make them easier to interpret, to reduce the learning time of the algorithms, to enhance model generalization, to increase their accuracy rate and to reduce overfitting [2, 26, 34, 55, 58, 91].

An optimal feature subset need not be unique because it may be possible to achieve the same accuracy using different sets of features (e.g., when two features are perfectly

correlated, one can be replaced by the other) - R. Kohavi & G. H. John ([49], the problem)

Feature selection technique is different from feature extraction or feature construction technique [79]. The latter aims to construct features of interest from the original features, either using visual analysis or by applying algorithms allowing the projection of the starting space to a new space [2, 9]. Definition 1 provides a mathematical frame for the comprehension of these two terms. Sometimes, in literature, these two notions are considered identical, resulting in difficulties for the understanding. In this document, they are considered as opposite, since the comparison is based on the conservation or not of a part of the original features.

Definition 1. *Feature subset selection or extraction is a process of selecting a map of the form $x' = f(x)$ by which a sample $x(x_1, x_2, \cdots, x_n$ is a n-dimensional measurement space $(\mathbb{R}^n)$ is transformed into a point $x'(x'_1, x'_2, \cdots, x'_{n'}$ in a n'-dimensional $(n' < n)$ feature space $(\mathbb{R}^{n'})$. For feature extraction, there is no restriction on the function f. In the case of feature subset selection, the function f must be so that x' is a subset of x [5].*

The general process of feature selection is exposed on figure 2.2. Subset generation is a procedure that produces subsets of variables, candidates for an evaluation, on the basis of a research strategy. Each candidate subset is assessed and compared with the other candidates via a certain evaluation criterion. If the new subset turns out to be better, it replaces the last best subset [51]. The generation and evaluation processes are repeated until a stopping criterion is verified. Afterwards, the best subset is usually validated by prior knowledge of the phenomenon or on the basis of different tests performed on synthetic or real databases. These four steps[18] are described in the following sections.

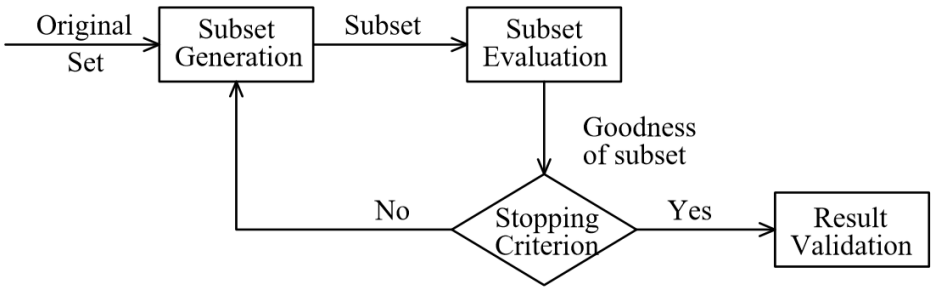

Figure 2.2: General feature selection process. Figure from [58].

2.2.1 Subset Generation

The generation of subsets is essentially a process of heuristic search [52, 54] for which each state in the research space specifies a candidate subset for evaluation. The nature of this process is determined by two basic problems: the starting point and the search strategy.

First of all, the initial point for the start of the research must be determined, what already influences the research process. The research can start with an empty set to which features are successively added, what is called forward search. A second possibility is to start with the full feature set and successively cancel some of the features, what is called backward elimination [22, 58, 63]. Finally, an alternative method can be used, for which the process starts with an empty set and a full set and continues in removing and adding simultaneously features to both sets. This last method is called bi-directional search [54]. The research can also start with a feature set randomly selected among all possible sets. The goal here is to avoid to fall in a local optimum [24].

The second problem is to decide the research strategy to follow: complete, sequential or random. Complete search guarantees to find the optimal result according to the evaluation criterion [48]. The exhaustive search is complete. However, a complete search is not necessarily an exhaustive search [58]. Different heuristic functions may be utilized to reduce search space without compromising the chances to find the optimal result. Therefore, even if the search space is $O(2^N)$, a smaller number of subsets is evaluated [24, 66].

Besides that, sequential search renounces to entirety and can hence lead to non-optimal subsets [58]. This is called greedy search. The algorithms with sequential search are simple to implement and rapid in the production of results [48].

The final possibility is the random search, which starts with a randomly selected subset and proceeds in different ways. The first way is to follow a sequential search, what is nothing different than injecting randomness within the classical sequential approaches described previously [24]. The second way is to make evolve the initial subset by random modifications (addition or cancellation of features) [10]. For all these approaches, the utilization of randomness helps to avoid local optima in the search space. The optimality of subset selection relies on the available resources.

2.2.2 Subset evaluation

As previously mentioned, each new subset needs that its quality be assessed by an evaluation criterion. If the new one is superior than the one before, the new subset then replaces the previous subset. *The main objective of evaluation is to compute the selective capability of feature or subset to make out the class label* - S. Vasalakshi & V. Radha ([91], evaluation criterion methods). It must be precised that an optimal subset, selected via one criterion, may be non-optimal, according to another. The evaluation criteria can be roughly categorized in two groups according to their type: filters and wrappers [22, 57, 72]. They are described in the following two sub-sections.

2.2.2.1 Independent criteria - filters

In this case, for a database X, the algorithm starts the research from a given subset S_0, that can be empty or not, and searches through the feature space on the base of the research strategy. An independent measure, a measure that does not take into account the clustering and using intrinsic properties of the data, is computed on each subset S. If it is found to be better than the previous one, it is conserved as the new best one. The research iterates until the predefined stopping criteria is reached. The algorithm returns the last best subset S_{best} as final result. It turns out that the results of the research are both dependent of the research strategy and of the evaluation criterion [58].

The criteria utilized to rank the subsets are called filter criteria or more simply filters [17, 35, 56, 97]. The process of subset selection via filter methods is summarized on figure 2.3. They measure the quality of a feature or the functionality of a subset by the exploitation of intrinsic characteristics of the data [97]. The clustering algorithm is ultimately applied in a next step [2, 25]. The most popular independent criteria for unsupervised learning problems are the distance and information measures.

Dependency measures are also known as correlation measures or similarity measures. They assess the ability of a first variable to predict the value of another variable. In feature selection for clustering, the association between two random features measures

the similarity between the two.

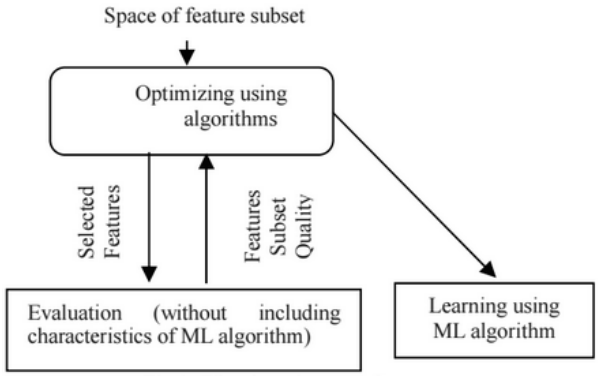

Figure 2.3: Representation of the filter method for the selection of features. Figures from [91].

Information measures typically determine the information gain from a feature I. It consists in computing the difference of information contained in the database (prior knowledge) with and without the feature I (posterior knowledge). Feature I is preferred to feature J if the information gain from I is higher than the one from J [58].

The advantages of filter methods are that they are easily scalable with high dimensional datasets, computationally simple, fast in process and independent of data mining algorithm [72]. Their main disadvantages are that they ignore the interaction between the classifier and the features and that most of them consider each feature separately than as a set, what leads in a degradation of the classification performance, in comparison with wrappers [91].

2.2.2.2 Dependent criteria - wrappers

For a database X, the algorithm starts the research from a subset S_0, that can be empty or not, and searches through the feature space on the base of the strategy search. The clustering algorithm is then applied on each subset S. A dependent measure, a measure that takes into account the clustering algorithm, is then computed on the results of the clustering algorithm applied on each subset S. If it is found to be the better that the previous one, it is conserved as the new best one. The algorithms iterates until the predefined stopping criterion is reached and returns the final best subset S_{best} as final

result. Therefore, the results of the research are dependent of the research strategy, the clustering algorithm and the evaluation criterion [58].

The criteria utilized to rank the subsets are called wrapper criteria or more simply wrappers [13, 25, 46, 49]. The process of subset selection via wrapper methods is summarized on figure 2.4. Hence, wrappers necessitate a predefined research algorithm, utilize its performance and consider the clustering algorithm as a black box [2, 33, 49] to determine the ranking of the subsets according to their predictive power [58], a measure of the cluster quality based on heuristic criteria. *The feature subset selection algorithm conducts the induction algorithm itself as part of the evaluation function* - G. H. John, R. Kohavi & K. Pfleger ([42], The wrapper model)

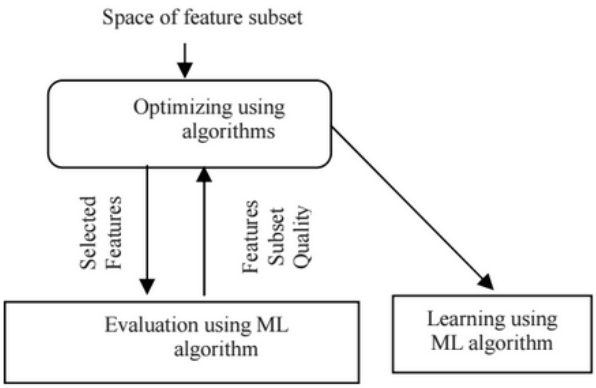

Figure 2.4: Representation of the wrapper method for the selection of features. Figures from [91].

Due to these interactions, wrappers generally provide better results than filter models as they find the features which correspond the best to the predefined research algorithm [2, 49, 52]. However, their disadvantages are the high risk of overfitting, due to the reduction of independent evaluation criteria, in comparison with filters, and their computational demand since clustering is integrated [2, 91].

2.2.3 Stopping criteria

The stopping criteria determines when the feature selection process should stop. Different possibilities exist. The easiest one is in the case of the complete search. Indeed, the feature selection process stops when the feature subset space has been completely

investigated.

Another stopping criterion can be associated to a predefined limit. The first possibility is the exceeding of a certain number of features, a certain number of iterations of the search process or even a certain level of quality (according to the evaluation criterion). A second possibility of stopping the research can occur when the last action (addition or cancellation of a feature) does not produce a better subset (according to the evaluation criterion) [58].

2.2.4 Validation criteria/criterion

Finally, it is important to validate the results obtained by the subset generation, subset evaluation and stopping criterion steps. One of the most easiest validation criterion is the prior knowledge on data [58]. Comparing them with the results provided by the process can already show the efficiency, or not, of the feature selection process. However this prior knowledge is rarely available.

To overcome this problem, data for which results are known in advance are generally used. It is often synthetic data, but that can also be real data. In this case, the performance of the process can be assessed visually or on the basis of a criterion taking into account the desired results and the one provided by the process.

2.3 Interaction between the techniques

Spike sorting is the technique that is commonly used to extract individual responses of neurons from neuronal activity recordings. After filtering, spike detection and feature extraction, clustering is performed (see section 2.1 for more details).

However, among the features generated in feature extraction, some of them may not be interesting and can be seen as noise on data. To alleviate this problem, feature subset selection is applied on the feature sets to select the most relevant one, according to a search strategy (through subset generation), some evaluation criteria/criterion as filters and wrappers and a stopping criteria, often linked to the search strategy. The

selection is then assessed by validation criteria/criterion (see section 2.2 for more details).

Because of the interest of feature subset selection, this work takes advantage of it to perform spike sorting. In this context, spike sorting is considered here, as described on figure 2.1, to which a step of feature subset selection is added between step B (feature extraction) and step C (clustering).

CHAPTER 3

STATE-OF-THE-ART

This chapter explores the literature related to spike sorting, including feature subset selection. It precises what already exists to approach these problems and the actual limitations of these procedures. Its ends by a small remark addressed to the reader and justifying the need of a novel approach.

3.1 Spike detection

To detect the localization of action potentials from the recorded and filtered signal, a threshold is usually used, if the Signal to Noise Ration (SNR) is good [71, 87]. In this case, each time the signal crosses this threshold, its is assumed that an action potential has occurred [53].

However, since the threshold level determines the trade-off between the action potentials and the background noise, missed classifications may occur. If the threshold is set too high, then it may be expected that the noise is not taken as an action potential. But, some actual action potentials may not be considered, what leads to false negative (Type I error). Conversely, if the threshold is fixed too low, then it may be expected that each action potential is detected. But, in this case, some noise artifacts may be considered as action potentials, what gives rise false positives (Type II error) [53].

In most of the applications, the threshold is related to the variance of the noise contained within the signal or at least an estimation of this variance [71]. A compromise between Type I and Type II errors, fixing the threshold at 4 to 5 times the estimator of the variance, is proposed in [76].

Other techniques are used to try to enhance the performance of this task. Among them it can be cited the application of energy operators (detection in changes in local energy signal) [7, 44, 65, 81], Continuous and Discrete Wavelets Transform (CWT or DWT) [11, 41, 45, 67], Independent Component Analysis (ICA, blind source seperation) [88] or fuzzy and probability theories [4].

3.2 Feature extraction

As a reminder, the goal of the feature extraction step is to treat each individual action potential coming from the spike detection step on the basis of its intrinsic properties or the features that can be extracted from them (see 2.1.3 for more details).

To complete this objective, the first types of algorithms bases itself on the amplitude of the peaks. However, this approach is unsuccessful since it is quite sensitive to noise as well as to intrinsic properties of the spike shapes [53].

An alternative consists in the capture of different characteristics of the signal that may be used in the clustering algorithm later. The features that are the most often employed are peak amplitude, width and energy. However, Wheeler and Heetderksl have shown that such features are not optimal for differentiating spike shapes in general in [92].

Finally, detected action potentials can also be perceived as a collection of M data. In this view, each action potential is symbolized by a point in a M-dimensional space [74]. Based on that, applications of dimensionality reduction algorithms can be considered. Principal Component Analysis (PCA) stands for being the most popular one. It provides an ordered set of orthogonal vectors depicting the directions for which the data possesses the higher variance. It assumes that each action potential can be acquired by a linear combination of principal components. Considering only the first K principal components, it is hence possible to find a K-dimensional space from the initial M-dimensional space.

This technique is applied in [1, 31, 53]. In addition to this, non-linear PCA (non linear combination) and linear discriminant analysis (best representation in the subspace) are also employed [22].

The most common alternative to PCA is wavelet transform. The wavelet functions derive from shifted and dilated versions of a predefined mother wavelet. The convolution between the signal and the functions leads to the creation of coefficients, representing a time-frequency decomposition of the signal. In this way, really small variations of the action potential are detected with few coefficients. Depending on the retained coefficients, wavelet transform can bring better results than PCA [40, 76]. However, due to the complexity of wavelet transform, PCA is often preferred.

Other techniques are also used for this process [14, 23, 29]: superparamagnetic clustering, laplacian eigenmaps, discriminant analysis.

3.3 Feature subset selection

Besides feature extraction, if the starting features mean something of interest on the signal, feature selection can be considered. Indeed, its most advantage is to conserve the meaningful features from the initial set of features for the problem of interest and that, based on their intrinsic properties (see 2.2 for more details).

It appears in literature that research strategies are mainly sequential and can be either filters, wrappers (see 2.2.2 for more details) or an hybrid solution. What is called hybrid solution is also often referred as embedded methods. They take advantage of filters and wrappers. Generally, in embedded methods, features are ranked according to a certain criterion. The sets with the best rankings and with an increasing number of features are provided to the clustering algorithm. The final results is evaluated according to a second criterion. At the end of the process, the subset with the best assessment is returned. It relies both on the prior ranking (characteristic to the filters) and the clustering algorithm (characteristic to the wrappers). That is why these algorithms are considered as hybrid solutions.

This section mainly focuses on feature subset selection for the problem of unsuper-

vised learning. In [58], the authors create a tabular containing references from various publications talking about feature subset selection for both classification (supervised learning) and clustering (unsupervised learning). An updated copy is presented on table 3.1. The following subsections describe the main idea behind certain algorithms present in this table.

Table 3.1: Synthetic table summarizing the different publications and introducing algorithms dealing with the problem of unsupervised learning (clustering). Inspired from [58].

Clustering algorithms	**Hybrid**	**Wrapper**	**Filter**	
Type	Filter + Wrapper	Predictive Accuracy or Cluster Goodness	Information	Dependency
Paper	Dash-Liu's[19]	AICC[21], FSSEM[25], Visual-FSSEM[26], ELSA[46], EM&K-ELSA[47]	Dash's[17], SBUD[20], CE[90]	Mitra's[64], PDF's[2], FD[27]

3.3.1 Filter solution

Filters possess the particularities to be easily scalable for large dimensional datasets, computationally simple, with a fast processing, but they do not take into account the clustering algorithm. Among them, filters based on dependency and information can be distinguished (see subsubsection 2.2.2.1 for more details).

Dependency

Among filter methods for unsupervised learning, M. Alibeigi, S. Hashemin and A. Hamzeh present in [2] an unsupervised feature selection technique using feature density functions. Their idea is to suggest a filter method in which an estimation of the probability density function (pdf) is performed, a dependency measure. The features are then ranking according to similarity between these statistical distributions. This proximity is

computed by the Mean Square Error (MSE), in other words, the sum of the squared difference. The algorithm returns the feature for which the MSE is lower than a value determined by the user.

Another algorithm using dependency measure is introduced by P. Mitra, C.A. Murthy and S.K. Pal in [64]. They define a new feature similarity measure between two features, called maximum information compression index, that they describe as *nothing but the eigenvalue for the direction normal to the principle component direction of the feature pair*. The features are selected via a threshold and the computation of the distances between a certain number of features, this number varying with the iterations on the process.

Information

Besides the algorithm utilizing dependency measure, M. Dash, K. Choi, P. Scheuermann and H. Liu advance a filter solution using an information measure, the entropy. The method comes from the observation that *data with clusters has very different point-to-point distance histogram than data without clusters*. To do that, a distance-histogram is computed between the features and the entropy is calculated so that its value is low if data has distinct clusters and high otherwise. The subset to return is selected by forward selection research of the best value of this entropy.

M. Dash, H. Liu and J. Yao in [20] define an entropy based on a similarity measure. The latter is computed from the distances between features so that closer features have higher similarity and far apart features have low similarity. To find the most important variables, they use a Sequential backward selection algorithm for Unsupervised Data (SUD). Finally, the algorithm returns the subset for which the entropy value is maximum.

In [90], R. Varshavsky, A. Gottlieb, M. Linial and D. Horn choose to compute a Contribution to Entropy (CE). In this paper, the entropy is calculated from the squared eigenvalues, as developed in [3]. This kind of entropy is minimal for an ultra-ordered dataset and maximum for a disordered dataset. The CE value is then determined as the difference between the entropy of the matrix and the entropy of the matrix to which one feature is canceled. The set of features to return is selected as the set of features for which the CE value is higher than the mean CE values, plus the standard deviation of

these latter ones.

3.3.2 Wrapper solution

Wrappers have the particularity to take into account the clustering algorithm and due to this interaction they often provide better results than filters (see subsubsection 2.2.2.2 for more details).

J. D. Dy and C. E. Brodley use Feature Subset Selection wrapped around Expectation-Maximization algorithm (FSSEM) with scatter separability and maximum likelihood as evaluation criteria [25]. The EM algorithm represents *the expectation-maximization algorithm applied to estimating the maximum likelihood parameters of a finite Gaussian mixture*. To generate the subsets to assess, the authors use sequential forward search.

Y. Kim, W. K. Street and F. Menczer, in [46], present ELSA, an Evolutionary Local Selection Algorithm that provides the optimal subset according to multi-objective or Pareto optimization, using the $K-means$ clustering algorithm.

3.3.3 Hybrid solution

Hybrid algorithms take advantage of the ability of filters to rank the subsets according to each other and the interaction of the wrappers with the clustering algorithm [16, 58, 68, 96]. They can be understood as a combination of filters and wrappers.

M. Dash and H. Liu propose in [19] such an hybrid solution. A similarity-based entropy measure is computed (exactly as in [20]). In their approach, the features are first ranked according to these values and a subset of important features is then selected, based on the quality, evaluated by scatter separability, of the clusters provided by the $k-means$ algorithm.

3.4 Clustering process and evaluation

In a final step of the spike sorting process, it is necessary to identify the type of clustering algorithm which is used, as well as the evaluation criteria for the representation quality of the clusters. These two elements are detailed in the next two subsections.

3.4.1 Clustering Algorithms

The final step of the spike sorting process, clustering, consists in grouping together the points of the feature space in several clusters. It exists different techniques leading to the desired results.

The first one defines manually the boundaries on a representation of the data in a subspace. That implies an analysis of the feature projections in 2 or 3-dimensional spaces.

To alleviate this problem, window discriminators have been introduced. The process is the following. A manually size-defined window slides along the signals. According to an increasing size of this window, action potentials comprised within it are assumed to belong to a same group and can then be classified [77]. The main limitations related to this technique are that it needs the human intervention and a recalibration of the window during the experiment.

Another idea that can be investigated is to attempt to realize action potential templates to superpose on the signal. Action potentials are extracted by correspondence, what is called template matching [62, 82, 85]. Each template represents a particular class or group. However, this approach necessitates manual interventions and readjustments. Furthermore, it is not proved that all possibles templates are available.

Besides that, statistical distributions of points in the feature space can be considered. This process is referred as Bayesian clustering. In this context, Expectation Maximization (EM) [94] is a first possibility. Next to that, Infinite Mixture Modeling (IMM) can integrate prior knowledge about the problem [95]. The hypothesis consists in considering that each cluster is modeled as a Gaussian distribution. Hence, the set of points can be represented as a sum of Gaussians to which a Gaussian stationary noise is added.

However, the hypothesis here is often violated in practice.

Non parametric algorithms can also be investigated. They base themselves on interactions between nearest neighbors. Among them, the most popular one is certainly $k-means$ algorithm or Lloyd method [60, 61]. In this one, from the number of clusters k, the points of the feature space are attributed to different groups according to their proximity in terms of distances. *If similarity is distance-based, then, for a pair of data points in a cluster, there exist at least a few features on which the points are close to each other* - M. Dash & H. Liu ([19], introduction). To do this, what is called centroids is defined, one for each of the cluster. In this case, since k clusters are considered, k centroids are defined. Each centroid represents then a group. Afterwards, each point is attributed to the collection of points for which the distance between the point and the centroid of the considered group is the smallest, in comparison with the centroids of the other groups [23, 86]. Hence, the sum of the square Euclidian distances of each point to its nearest cluster center is minimized [9].

3.4.2 Evaluation criteria

An important part of clustering is the evaluation of the resulting clusters [38, 43]. Evaluation criteria represent criteria that are used to assess the representation quality of the clusters at the exit of the clustering algorithm. They can be seen as tools answering the question: "How well represented are the clusters?". There exists a lot of different techniques and some of them are cited hereafter, classified on whether they are particular to spike sorting or general.

3.4.2.1 General criteria

The first assessment that can be performed is the visual evaluation of the features vectors. To do so, the clusters are plotted in different colors. The goal for the user consists in inspecting the partition of the points in the space [85]. However, the visual representation can only be realized in a 3-dimensional space whereas the number of features often exceeds this value.

Scatter separability invariant is another possibility [22]. It is often used since it is invariant under any nonsingular linear transformation [28]. This intuition behind this criterion is that a good clustering result is the one for which the points within the clusters are close to each other and the distance between the clusters is large.

Besides that, Maximum Likelihood (ML) represents a measure of how likely the data are, given the parameters and the model. In this context, a statistical distribution is associated to each cluster. Expectation Maximization (EM) algorithm is then applied and this measure is computed from the result.

3.4.2.2 Particular criteria

Another natural way to evaluate the quality of the clustering is to represent the waveform representation. The goal here is to assess whether the spikes in a cluster have similar shapes in plotting them on a single graph [85]. It has the advantage of clearly indicating any outlier event [38].

In [83], N. Schmitzer-Torbert, J. Jackson, D. Henze, K. Harris and A. Redish propose L_{ratio}, based on the L quantity, as a new quantitative cluster separation metric. The L value is computed for each cluster and represents a measure of the isolation of the cluster. The more the cluster is isolated from the other data, the higher is its value. L_{ratio} is obtained as the L quantity, normalized by the number of spikes contained in the cluster.

K. D. Harris, H. Hirase, X. Leinekugel, D. A. Henze and G. Buzsaki introduce in [37] Isolation Distance of a cluster C, which estimates how distant the cluster spikes are from the other spikes.

3.4.3 Validation criteria

Validation criteria represent tools that can be used to assess the final quality of the results provided by the process. In other words, the application of a certain criterion, different than the evaluation criteria, allows to evaluate if the outputs of the process seems correct or not. In the case where artificial data or labeled data are at disposal, the criterion gives a value of the clustering efficiency. It calculates how close are the

results provided by the process to the labeled data. In the case where neither artificial nor labeled data are available, this criterion provides an insight of the quality of the results (see 2.2.4 for more details).

One of the most used index for the comparison of the clustering results is the Rand Index (RI) defined in [78]. Its goal is to compare the clustering results of two clustering algorithms or to confront such a result to a known partitioning solution. For each pair of points, it counts how many are in a same cluster in both partitions, how many are in separate clusters in both partitions, how many are in a same cluster in the first partition and in different clusters in the second partition, and the reverse. Finally, the Rand Index is computed in the range between 0 and 1, 0 meaning that the partitions are totally different and 1 meaning the opposite.

At the end of the spike sorting process, it is possible to generate the entire response of each neuron by placing each spike, considered as coming from this neuron, at the corresponding temporal localization. Since a neuron presents a refractory period [39], an interval histogram plotting the distribution between the spike temporal locations is realized. With a typical refractory period of 1 ms [32], if there is no centered gap in the interval histogram, it means that there exist at least one spike of another neuron that has been supposed belonging to this neuron. In practice, a small fraction of violation of the refractory period is tolerated (around 0.5%) [85], since noise artifacts are also present in clusters.

It is to remark that the validation can also be applied by using the tools related to the evaluation criteria (see 3.4.2).

3.5 Note to the reader

Throughout this chapter, available machine learning techniques have been exposed to treat spike detection, feature extraction, feature subset selection, clustering process and its evaluation. However, none of them is actually the best. Depending on the situation, one will be preferred to another, regarding to a compromise between its performance and its facility to be implemented. Challenging these problem by another side can then be envisaged and could bring surprising results. Chapter 4 details the approach investi-

gated in this work: smart, visual and geometric spike features can enhance spike sorting process or, at least, match the performance of PCA, the common technique used in this task.

However as written in section 3.2, one may think that such an approach has already been investigated in other publications. Concerning the approach of the problem via visual and intelligent spike feature, the answer is yes. However, these authors only utilized the amplitude and the width of the peaks and their energy. This vision is quite limited because the representation of a spike cannot be realized on their own basis. That is why chapter 4 tends to show that a multitude of features are needed to characterize a spike. The approach of this work resides hence in two points: the extraction of intelligent features from the spikes and the utilization of feature subset selection on them to select the ones that are able to provide the best results in the spike sorting process.

CHAPTER 4

PROPOSED METHODOLOGY

In this chapter, the protocol allowing us to realize the experiments on the data. In other terms is established, the following lines define the parameters and techniques/algorithms used in the computation, with a view of solving the basic problem. For each feature, the general idea of its choice is also presented. The structure of the chapter follows the one of the general process of spike sorting to end with the general process that is used to obtain the desired results.

4.1 Investigated approach and intuition

The idea of the approach used in this work to treat spike sorting problem is to mimic the neuroscientist's decision making during spike classification. During this task, this person is able to detect and/or interpret visual characteristics from the form of action potentials. Furthermore, if it is asked to a neuroscientist to represent an action potential or a spike, the shape of the curve he would draw would be really characteristic and could look like the representations on figure 4.1. Typically, the spike shape presents at least one positive peak and sometimes a second, negative, and a third, positive.

Since the waveforms are characteristic, it is normal to think that they have common particularities and that they differ on other aspects. Hence, based on these representations it is possible to extract some features (presented in section 4.3) that are certainly

more informative than the features build by the techniques evoked in chapter 3 (such as PCA or Wavelet Transform).

However, even if lots of them can be computed on the basis of signals, the relative relevance of one according to the others is not easy to evaluate and it is also possible that some of them are totally uninteresting. Hence, if the goal is to exploit as much as possible the information that can be extracted from the signal, it is essential to use feature subset selection on the feature set that is generated.

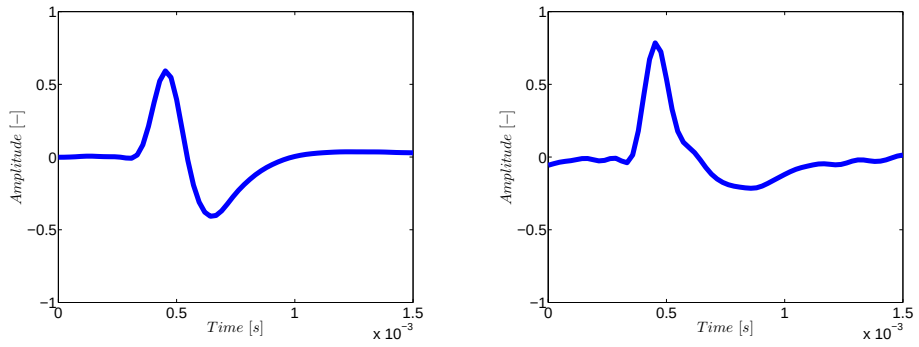

Figure 4.1: Typical spike shapes representations.

4.2 Database

Different kind of data can be used to achieve the goal of this work. The first one consists in collecting recorded data on which a manual spike sorting has be applied. The second possibility resides in the creating of an artificial database. To do so, artificial or real spikes are displayed in time.

In this work, artificial data, publicly available, are used. They represent simulating extracellular signals recorded using a single micro electrode, are generated for the purpose of [93] and totally correspond to the need of this work. They are engendered by superimposing real spikes randomly positioned onto a noise background and uniformly scaled (75 – 125%) in amplitude to mimic the different spatial distance from each neuron to the electrode. A total of 9 real spikes are used and a refractory period of 3 ms is maintained, for a total duration of 60 s and a 24000 Hz frequency.

Different noise level (η), or amount of noise in comparison with spike signal, are produced and defined as follows:

$$\eta = \frac{1}{SNR} = \left(\frac{RMS_{noise}}{RMS_{signal}}\right)^2 \tag{4.1}$$

Where SNR represents the signal-to-noise ratio, RMS_{signal} the root-mean-square (RMS) amplitude calculated from the spikes extracted by applying spike detection, and RMS_{noise} accounts for RMS computed from the rest of the signal.

The interest of such a database resides in the fact that expected outputs of the spike sorting process are known. Hence, the performance of the techniques can be evaluated by comparison with them. A sample of a signal generated with 4 different types of spikes and a 0.15 noise level is presented on figure 4.2.

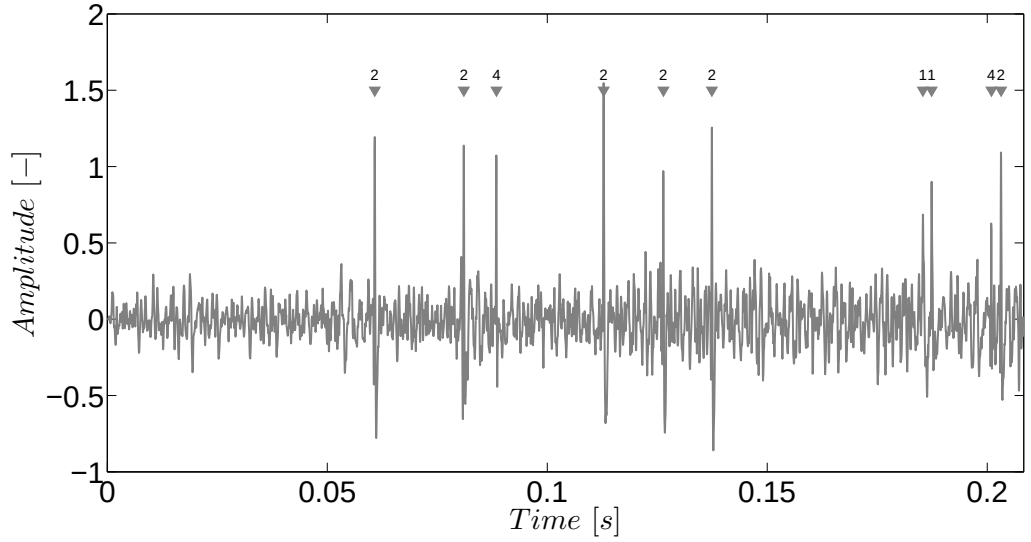

Figure 4.2: Synthetic signal sample generated from 4 different spikes and for a 0.15 noise level. Down arrows indicate localization of the spikes and numbers identify the membership of a spike to a cluster. First number 2 means that this first spike belongs to the same clusters as the second spike, also referred with a 2. Database coming from [93].

In this case, spike localization are used to extract spikes. Spike detection is then not required (see subsection 2.1.2 for more details) and the process is applied from this step.

4.3 Different available features

In the following subsections are presented the characteristics it is decided to take into account in order to mimic the decision making of a neuroscientist. These can be either specific to the studied domain or general and applicable to any signal. It is to precise that this list is empirical and not exhaustive. A summary of the selected features among those proposed is presented in the end of the section.

4.3.1 Peak-to-Peak Amplitude

The first types of algorithms solving the spike sorting problem were based on the amplitude of the peaks, as explained in section 3.2. That seems logic because one of the characteristics of a signal i, whatever it is form biological origin or not, is certainly its total amplitude, denoted by M_i and defined as follows:

$$M_i = max(X_i) - min(X_i) \tag{4.2}$$

In this expression, X_i represents the line vector characterizing the action potential, $max()$ and $min()$ respectively the maximum positive amplitude of the action potential, corresponding to the magnitude of the positive peak, and the maximum negative amplitude of the action potential, corresponding to the magnitude of the negative peak. The maximum value of the signal is also considered as being positive and the minimum value of the signal as being negative.

In neurophysiology, the response of neurons generally includes a positive peak and a negative peak. Therefore, the total amplitude of the signal is also called peak-to-peak amplitude. This must be understood as the distance on the amplitude axis between the positive and negative peaks (see figure 4.3).

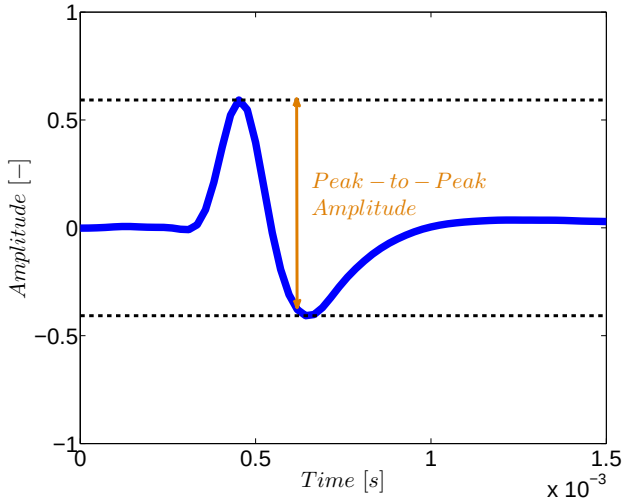

Figure 4.3: Spike (in blue) and its associated Peak-to-Peak Amplitude (double arrow in orange). Vertical black dashed lines represent respectively maximum (upper) and minimum (lower) amplitudes.

4.3.2 Relative amplitudes

Since authors demonstrate the inefficiency of the peak-to-peak amplitude to distinguish spikes (see section 3.2), it would be normal to cancel it from the set of features. However, the presence of peaks in the action potentials is a fundamental and intrinsic property of the neural response and must be taken into account within the process, that is a conviction.

Since the value of M_i defined in expression 4.2 may change during a unique recording, notably due to electrode drift (explained thereafter), this definition should be modified. To do this, two types of amplitudes are considered, one related to the positive peak and the other one related to the negative peak. They are named relative amplitudes. The relative positive amplitude Amp_i^+, associated to the positive peak of the action potential i is defined as:

$$Amp_i^+ = \frac{max(X_i)}{M_i} = \frac{max(X_i)}{max(X_i) - min(X_i)} \tag{4.3}$$

Similarly, the relative negative amplitude Amp_i^- of the action potential i, related to the negative peak, is defined as:

$$Amp_i^- = -\frac{min(X_i)}{M_i} = -\frac{min(X_i)}{max(X_i) - min(X_i)} \tag{4.4}$$

In these expressions, $max()$ and $min()$ represent respectively the maximum positive amplitude of the action potential, corresponding to the magnitude of the positive peak, and the maximum negative amplitude of the action potential, corresponding to the magnitude of the negative peak. The negative sign in expression 4.4 assures a positive value for Amp_i^-. Amp_i^+ and Amp_i^- are naturally included in the interval $[0,1]$, since they are both positive and their sum is equal to one.

The presence of a denominator representing a normalization factor may seem strange. This idea comes from a tentative of reducing the impacts of electrode drift on the spike sorting process. Suppose a neuron present at some distance from the recording electrode at instant t_0 and that provides a total amplitude response R_0. If there happens an electrode drift at instant t_1, then a total amplitude R_1 can be recorded for this same neuron in the same physiological conditions. If the drift gives rise to a greater distance between the neuron and the electrode, then it is observed that $R_0 > R_1$. The opposite situation, in which the drift engender a smaller distance between the neuron and the electrode, leads to $R_0 < R_1$. Assuming that the response of the neuron differs in total amplitude between these two moments, then normalizing all the signals by their peak-to-peak magnitude can result in signals representing identical responses. This explains the application of this denominator. In addition, this normalization is performed several times and for the same reasons for the extraction of the following features.

Let's take back the two spikes of figure 4.1 and see how these features differentiate them. The representation is proposed on figure 4.4, for which the normalization has been applied.

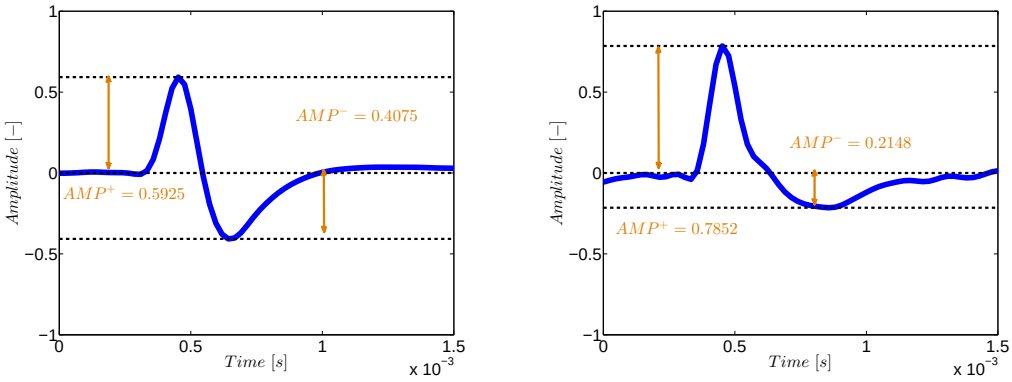

Figure 4.4: Spikes (in blue) presented on figure 4.1 and normalized by their Peak-to-Peak Amplitude (as defined in expression 4.2). On each figure, the positive (left) and negative (right) relative amplitudes are drawn in orange. Vertical black dashed lines represent respectively maximum (upper), zero (middle) and minimum (lower) amplitudes.

4.3.3 Relative Peak-to-Peak duration

Each action potential is considered as having at least two peaks, one positive and the second negative. The relative peak-to-peak temporal distance/duration D_{peak} for the action potential i can be defined as follows:

$$D_{peak_i} = \frac{|index_{min(X_i)} - index_{max(X_i)}|}{F_e \cdot D_s} \tag{4.5}$$

where $max()$ and $min()$ represent respectively the maximum positive amplitude of the action potential and the maximum negative amplitude of the action potential, F_e is the sampling rate or the frequency of data acquisition during the experimental recordings, $index(x)$ represents a function that returns the position of the points of the signal at which x is observed, within the set of points forming the action potentials. Typically, $index_{min(X_i)}$ returns the point at which the minimum amplitude is registered in the series of points. Finally, D_s describes the total duration of the spike (1.5 ms).

Physiologically speaking, D_{peak_i} represents the percentage that corresponds to the time gap between the positive and negative peaks versus the spike duration (see figure 4.5). In a general way, the positive peak is expected to be first and then the negative

peak, whence the order (maximum after minimum). However, in some cases, the peaks can be reversed. To counter this problem, the absolute value is added.

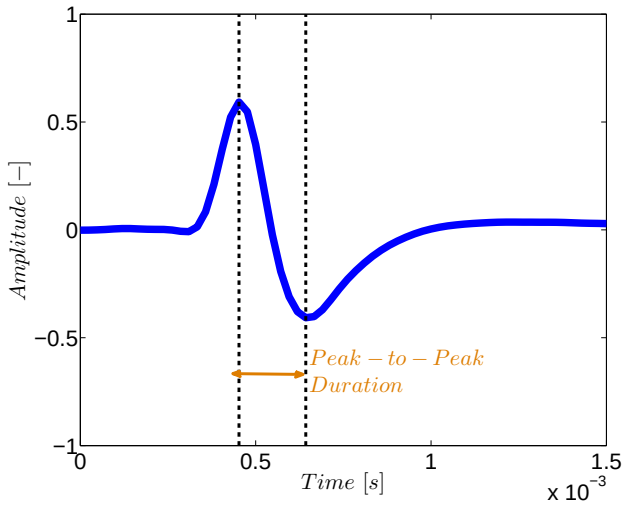

Figure 4.5: Spike (in blue), whose amplitude is normalized as explained in subsection 4.3.2. Peak-to-Peak duration is represented by the double arrow in orange. Vertical dashed lines represent respectively the indexes for the maximum (left) and minimum (right) amplitudes of the spike.

4.3.4 Relative integral/area and relative average values

One of the particularities of the responses of neurons to stimuli is the material form it takes. To characterize it, the area under the curve can be taken into account. This may represent a form of information contained in the signal. In the case of discrete time signals, the relative area A_i of the action potential i defined as:

$$A_i = \frac{1}{M_i} \cdot \sum_{j=1}^{H} X_{ij} = \frac{1}{max(X_i) - min(X_i)} \cdot \sum_{j=1}^{H} X_{ij} \tag{4.6}$$

In this expression, M represents the number of points in the set of points making up the action potential i. Again, the normalization factor is present so that different areas of the action potential are not different because of their magnitude, but well due to the

shape of the response to stimuli. From this one, the relative area A^+ can be deduced as the ratio of the area between the curve and the 0-amplitude axis and the total area.

It is to note that there would also have been able to define the average value of the normalized response AVG_i of the action potential i as:

$$AVG_i = \frac{1}{H} \cdot \sum_{j=1}^{H} \frac{X_{ij}}{M_i} = \frac{1}{H} \cdot \sum_{j=1}^{H} \frac{X_{ij}}{max(X_i) - min(X_i)} \tag{4.7}$$

This can be easily deduced from the previous expression. The relation between 4.6 and 4.7 is then:

$$AVG_i = \frac{1}{H} \cdot A_i \tag{4.8}$$

A visualization of the area under the curve and relative average value, respectively defined in expressions 4.6 and 4.7 are presented on figure 4.6.

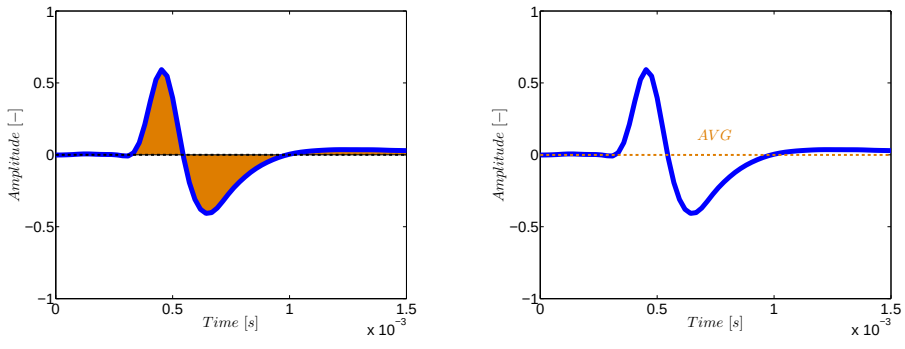

Figure 4.6: Spikes (in blue) presented on figure 4.1 normalized by their Peak-to-Peak Amplitude (as defined in expression 4.2) and with, on the left figure, the area under the curve of the spike is exposed in orange and, on the right figure, the average value (as defined in expression 4.7) is presented as a orange continuous line.

4.3.5 Relative average energy

In the same general orientation of the analysis of signals, the energy of the neuronal response can be studied, in the discrete format. Again, in order to not suffer from variations due to amplitude fluctuations, the average energy of the normalized response E_i (including H points) of the neuron i can be defined as:

$$E_i = \frac{1}{H}\sum_{j=1}^{H}\left(\frac{X_{ij}}{M_i}\right)^2 = \frac{1}{H}\sum_{j=1}^{H}\left(\frac{X_{ij}}{max(X_i) - min(X_i)}\right)^2 \tag{4.9}$$

A visualization of the energy of the signal is presented on figure 4.7.

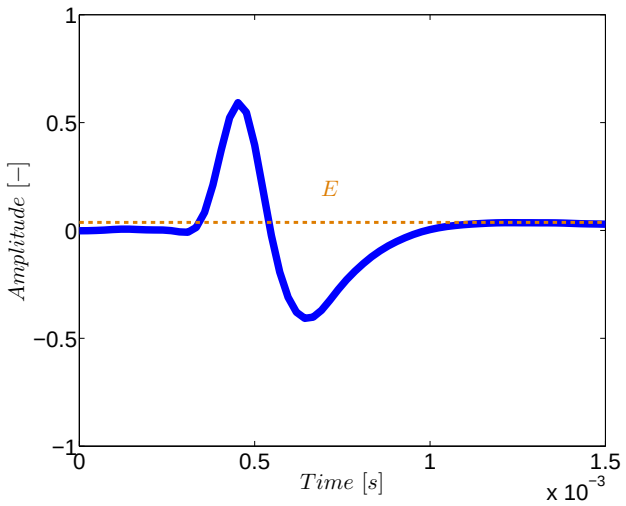

Figure 4.7: Spike (in blue) and its associated variance measure (as defined in expression 4.9) in orange continuous line.

4.3.6 Retained Features

Among the variables defined in the preceding subsections, only those that seem of interest are kept during this pre-selection process. Between the three amplitude values, only the positive relative amplitude is retained to avoid to encounter problems due to electrode drift and the relevance problem due to the linear relation between Amp^+ and

Amp^-, as explained in subsection 4.3.2. Relative peak-to-peak duration is also conserved, because it is the only time information established and, because it is strongly related to the characterization of the action potential. Instead of keeping the relative area, the average negative value is retained, because this second feature is commonly defined when signals are discussed. The relative energy and positive area values are finally kept, they strongly define spikes.

Furthermore, during the study of a signal with peaks, it may be interesting to study the first and second derivatives of the signal. In this case, this is of particular significance. Indeed, both the gradient and the concavity of the responses allow the neuroscientist to characterize the action potentials (see figure 4.8 for a representation of them). To include this in the set of features, the conserved characteristics of the above signals are applied on the normalized first and second derivatives.

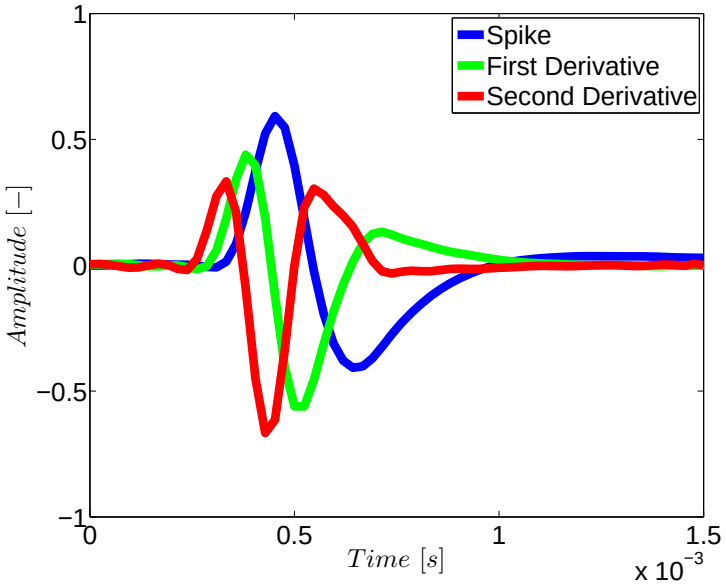

Figure 4.8: Representation of a spike (blue) with its associated first (green) and second derivatives (red). Each signal is normalized.

Moreover, the majority of the algorithms involved in spike sorting utilizes PCA that can create new variables based on linear combinations of the original variables (see section 3.2). To try to demonstrate the superiority of the newly created set of variables, another set is build with the principal components proposed by PCA.

The set of the retained features F can then be defined as:

$$F = \begin{bmatrix} — & F_1 & — \\ — & F_2 & — \\ & \vdots & \\ — & F_D & — \end{bmatrix} \tag{4.10}$$

for which:

$$F_i = \begin{bmatrix} SF_i & \dot{SF}_i & \ddot{SF}_i \end{bmatrix} \tag{4.11}$$

$$SF_i = \begin{bmatrix} AMP_i^+ & A^+ & D_{peak_i} & AVG_i & E_i \end{bmatrix} \tag{4.12}$$

In expression 4.11, the single and double dots over the names of the features represent respectively the first and second derivatives of the detected action potential related to the selected features. It must be understood here that the set of the defined features is applied on the first and second derivatives of the signal.

Hence, the matrix F, defined in 4.10 possesses a dimension $D \times 15$, D representing the number of action potentials detected in the database and 15 representing the number of features included in this stage of extraction and preselection.

Finally, the matrix $SPCA_i$ containing the different components provided by PCA for the action potential i can be defined as:

$$SPCA_i = \begin{bmatrix} PCA_{1i} & \cdots & PCA_{Mi} \end{bmatrix} \tag{4.13}$$

where PCA_j is the j^{th}component provided by the PCA. It is to precise that the PCA is not applied on the first and second derivatives of the action potentials.

4.4 Selected feature selection algorithms

In this section, the different selected algorithms used to make the feature selection are explained/detailed. The goal is to select multiple techniques and evaluate the results they provide on test sets at the end of spike sorting. To do this, filter and wrapper techniques are utilized. It is to specify that it is not possible to investigate all of the feature space. Indeed, 2^{15} opportunities, the number of combinations that is possible to create with 15 features, should be considered which is not tractable from a computation time standpoint.

Different techniques are proposed in the following subsections. The presented algorithms are all hybrids, i.e. they combine the filter and wrapper approaches. Taking into account the clustering algorithm, the wrapper approaches provide generally better results than the filter approaches. However, because of the time required in the exploration of the feature space, filter techniques are most often used. Willing to achieve a maximum of efficiency, hybrid techniques are preferred. They assist in generating a ranking of the features according to their relative importance. The choice of the subset is then determined by applying the clustering algorithm on feature sets of increasing size depending on the importance of the features.

4.4.1 Contribution to entropy on a leave-one-out-basis

The first proposed technique is presented in [90]. The authors of this paper define both filter and wrapper algorithms. It is based on the contribution brought by a feature to the entropy of the database. They initially consider singular value decomposition of the feature matrix F constructed in Section 4.3. To do this, the eigenvalues s_j^2 of the FF^t are computed. The normalized relative values are then defined as:

$$V_j = \frac{s_j^2}{\sum_k s_k^2} \tag{4.14}$$

It is then possible to generate an entropy based in the database [3] :

$$E^* = \frac{1}{log(N)} \cdot \sum_{j=1}^{N} V_j log(V_j) \tag{4.15}$$

This value E^* stands between 0 and 1. A value of zero means that the database is ultra-ordered and can be explained by a single eigenvalue, while a unit value means that the spectrum is evenly distributed. The value E^* does not have to be confused with the true entropy E defined by Shannon and that is based on probabilities. It is also called entropy as it measures a certain information based on the eigenvalues of the database.

Based on that, the authors then define the contribution of the ith feature to the entropy CE_i with comparison with the leave-one-out type:

$$CE_i = E\left(F_{[nXm]}\right) - E\left(F_{[nX(m-1)]}\right) \tag{4.16}$$

where, in the last matrix, the variable i is withdrawn. Features are then classified in descending order of importance. The wrapper approach is included by using the clustering algorithm.

The authors also define a filter. To do this, they determine c, the average of all CE_i, previously computed, and d the associated standard deviation. They then establish the following groups:

- All features for which $CE > c + d$ are considered as high contribution variables and must then be retained.

- All features for which $c + d > CE > c - d$ are considered as average contribution variables. These features are not kept, but only the ones with high contribution are of real interest.

- All features for which $CE < c - d$ are considered as low contribution variables and are hence not retained.

On this basis, a filter-type algorithm can be defined, because there is no intervention of the clustering algorithm in the feature selection process. The variables are chosen before applying this algorithm.

Instead of keeping this filter technique, the wrapper method can also be considered and is selected in this case. After having chosen the selection criterion, the authors propose three types of research methods:

1. Simple ranking (SR) : The ranking is realized on the basis of the provided CE values.

2. Forward Selection (FS) with two ways of implementation:

   a) FS1 : It chooses the first feature with the highest value of CE. Then, among the remaining features, the one that provides the highest CE value with the set of selected features is selected as the next most important feature. The process runs until the initial set of features is empty .

   b) FS2 : It chooses the first feature with the highest value of CE. Then, the second best feature is selected with the same criterion but based on the set of features without the first one. The CE values are then computed on a decreasing set of features. The process runs until the initial set of features is empty.

3. Backward Elimination (BE) : In this case, the process is reversed. From the original set, the feature with the lowest CE value is canceled from this set and defined as the less important feature. The CE values are then recomputed on the remaining set and the selection of the second less important feature is realized on the basis of the lowest CE value. The process runs until the initial set of features is empty.

4.4.2 Features Density Similarity - Kernel Density Estimation

The second technique is based on an estimate of the statistical density of the available variables. It is referred to [2]. In a general way, this method removes redundant features using their probability density functions.

To do this, distributions are modeled as a non-parametric estimation:

$$p(x) \simeq \frac{k}{N \cdot V} \tag{4.17}$$

where $p(x)$ is the estimated probability density function (pdf) of the input x, V is the volume surrounding x, N is the total number of instances and k is the number of instances in V.

To set the value of k and to find the volume V corresponding to data, two approaches can be considered. The first is the K Nearest Neighbor (KNN) method, where k is fixed and V is to determine. On an other side, when V is fixed and k is to find, the problem of Kernel Density Estimation (KDE) is investigated. In this case, the probability distribution functions are estimated with the KDE method with Gaussian kernels. So that the pdf's can be compared, the variables are scaled in the interval $[0, 1]$.

Once the pdf's are evaluated, the similarity between each pair of features is computed by the Mean Square Error (MSE) computed from their pdf's. If this value is under a threshold value determined by the user, the features are considered similar. Thus, one of the two features can be suppressed. Among the similar features, features that are similar to several other within the feature space are canceled. A summary of the process is presented in figure 4.9.

In this filter model, two items are to define. The first is the value of the similarity threshold that is left free to the user. Then there is also the number of features to remove that must be provided. In order to not choose empirically these values, the conversion of this filter algorithm to an hybrid algorithm is carried out. Hence, the process presented in figure 4.9 is followed until the definition of the threshold is required. At this time,

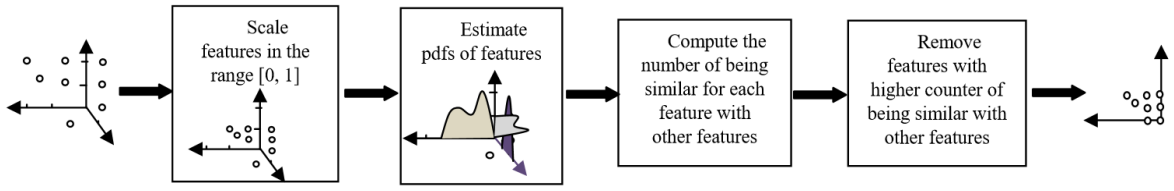

Figure 4.9: Representation of the general process of feature selection on the base of their pdf's and similarities. Figure from [2].

MSE measurements allow us to make a ranking among the features. Scores can then be attributed. The set of features is then build on the base of the computed scores of similarity.

4.4.3 Feature Dispersion

The idea of the authors of the paper [27] is that a feature has an importance/relevance proportional to its dispersion, as argued in [59]. Instead of attempting to compute the statistical dispersion as in subsection 4.4.1, they focus on Arithmetic Mean (AM) and Geometric Mean (GM), defined as follows for each feature F^j:

$$AM^j = \frac{1}{n}\sum_{i=1}^{n} F_{ij} \tag{4.18}$$

$$GM^j = \left(\prod_{i=1}^{n} F_{ij}\right)^{\frac{1}{n}} \tag{4.19}$$

The notations are modified in comparison to the original paper so that it corresponds to the convention followed in this work. They define then the relevance criterion as $R^j = AM^j/GM^j$, with $R^j \in [1, +\infty]$. However, the case where a single value in a feature vector is zero leads to a zero-value for the GM value, making this criterion useless. To alleviate this problem, the authors propose to apply the exponential function on each feature, given rise to the relevance criterion:

$$R^j = \underbrace{\frac{1}{n}\sum_{i=1}^{n}\exp\left(F_{ij}\right)}_{a^j} \Big/ \underbrace{\left(\exp\left(\sum_{i=1}^{n}F_{ij}\right)^{\frac{1}{n}}\right)}_{g^j} \tag{4.20}$$

Applying the natural logarithm of expression 4.20 does not modify the ranking since $0 < g \leq a$. After some algebraic manipulation and dropping the constant terms (bringing nothing more to the ranking objective), the Feature Dispersion (FD) criterion is obtained:

$$FD^j = \log(a^j/g^j) = \log\left(\sum_{i=1}^{n}\exp\left(F_{ij}\right)\right) - \frac{1}{n}\sum_{i=1}^{n}F_{ij} \tag{4.21}$$

If the features are normalized, their values are in the range $[-1;1]$. In this case, the Taylor first degree polynomial approximation round 0 $\exp(x) \simeq 1 + x$ can be applied on expression 4.21. Using that and writing $S^j = \sum_{i=1}^{n} F_{ij}$, this previous expression leads to:

$$FD^j \simeq \log\left(n + S^j\right) - \frac{1}{n}S^j \tag{4.22}$$

This expression is quiet efficient to compute and bring an information on how spread are the data, as claimed by the authors. The features can then be ranked according to this criterion. Features with the smaller values of FD are supposed to be more relevant than the one with higher values for this same criterion, meaning that the feature is compactly distributed.

4.5 Clustering algorithm, evaluation and validation criteria

In this section, the type of clustering algorithm used in the process of spike sorting is detailed. Subsequently, an assessment criterion of the quality representation of the clusters is explained. It allows to determine the number of features to keep in the feature selection step. Finally, validation criteria are presented.

4.5.1 Clustering algorithm - $K-means$

K-means algorithm is a clustering algorithm whose the goal is to partition n points into k clusters according to centroids (see subsection 3.4.1 for more details). A mathematical definition of the k-means problem is proposed at definition 2, coming from [9] and equivalent to definition in [70].

Definition 2. *Given a matrix $A \in \mathbb{R}^{n \times d}$ (representing n points - rows - described with respect to d features - columns) and a positive integer k denoting the number of clusters, find the $n \times k$ indicator matrix X_{opt} such that:*

$$X_{opt} = \underset{X \in \mathcal{X}}{arg\,min} ||A - XX^T A||_F^2 \tag{4.23}$$

The optimal value of the k-means clustering objective is:

$$F = \underset{X \in \mathcal{X}}{min} ||A - XX^T A||_F^2 = ||A - X_{opt} X_{opt}^T A||_F^2 \tag{4.24}$$

In the above $\mathcal{X}$ denotes the set of all $n \times k$ indicator matrices X.

This algorithm runs as follows. In a first step, the initialization of the process is realized by the selection of K points that are called centroids. The second step consists in attributing each data points to the closest centroids by a distance measure. In a third step, new centroids are then defined as being the data point which represents the mean of the clusters the previous centroids represent. The process continues by applying successively the second and third step until a stopping criterion is met.

4.5.2 Evaluation criteria

Once the feature selection algorithm has succeeded in forming the different set of features, the clustering algorithm is applied on each of them. Hence, the clustering algorithm is used on the sets containing an increasing number of features, their relevance being assessed by the evaluation criteria. To assess the quality of the clusters of this process, L_{ratio} and $R_{correlation}$ criteria are chosen.

4.5.2.1 L_{ratio}

L_{ratio} is introduced in [84]. The goal of this measure is to evaluate if one cluster is well separated from other spikes. A spike is represented by a point in a high-dimensional space, the dimension corresponding to the number of features characterizing it. This separation distance is computed from the Mahalanobis distance (noted D^2) to measure how far apart are the center of the clusters and the data points outside the cluster. This Mahalanobis distance for spikes in a cluster is distributed as χ^2 with degrees of freedom (df) equal to the number of features [15] if the assumption is made that each spikes can be represented by a multivariate normal distribution.

A L value, representing a measure of the number of data points that are outside of the cluster and that should be inside. It is obtained by the inverse cumulative distribution function for the χ^2 distribution, with the number of specified degrees of freedom. For each cluster C:

$$L(C) = \sum_{i \notin C} \left(1 - CDF_{\chi^2_{df}}(D^2_{i,C})\right) \tag{4.25}$$

In this expression, $i \notin C$ is the set of spikes that are not members of the cluster C, $CDF_{\chi^2_{df}}$ is the cumulative distribution function of the χ^2 distribution with df degrees of freedom, and $D^2_{i,C}$ is the squared Mahalanobis distance of spike i from the center of cluster C.

From 4.25, spikes close to the cluster have naturally high probabilities (approaching 1) to belong to the cluster. Spikes far from the center of the cluster C have low probabilities to belong to the cluster. Hence, low values of L are researched.

The cluster quality measure L_{ratio} is defined as L divided by the total number of spikes in the cluster (n_C):

$$L_{ratio}(C) = \frac{L(C)}{n_C} \tag{4.26}$$

This normalization allows to compare L values for clusters containing different number of spikes.

4.5.2.2 $R_{correlation}$

Linear correlation between two variables, also called Pearson's correlation coefficient and noted ρ, is a measure of the linear relationship between the variables. In other words, its measures how well a first variable can be obtained via a linear relation including the other variable. It is obtained by dividing the covariance between the two variables by the product of their standard deviations:

$$\rho_{X,Y} = corr(X,Y) = \frac{cov(X,Y)}{\sigma_X \sigma_Y} = \frac{E\left[(X-\mu_X)(Y-\mu_Y)\right]}{\sigma_X \sigma_Y} \tag{4.27}$$

where $cov(X,Y)$ represents the covariance between X and Y, μ the mean, σ the standard deviation and $E[\bullet]$ the mathematical expectation. Here, X and Y are considered as random variables which are variables for which variations in their realization can occur when the general process is repeated.

Let's come back to the spike sorting process. At its output, clusters gathering spikes are formed. It is expected that each group represents the response of a neuron, what means that the spikes must be relatively similar. Let's take again spikes represented at figure 4.1 and plot on figure 4.10 and consider each spike as a time-dependent random variable ($X(t)$ for spike on the left and $Y(t)$ for spike on the right.

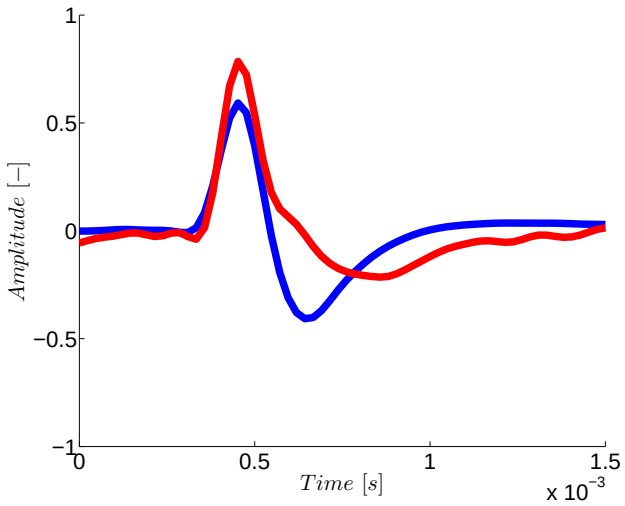

Figure 4.10: Spikes superimposed to see the proximity between shapes.

In this case, the Pearson's correlation coefficient can be measured and it is observed that:

(4.28) $$\rho_{X,X} = 1$$

(4.29) $$\rho_{Y,Y} = 1$$

(4.30) $$\rho_{X,Y} = 0.7685$$

The fact that $\rho_{X,X} = \rho_{Y,Y} = 1$ is perfectly normal since X can be exactly obtained from X, and the same reasoning can be made of the relation between Y and Y. What is more interesting is the relation between X and Y. $\rho_{X,Y}$ value means there is a tendency to have a linear relation between X and Y, but without being perfect. This is the idea that is going to be used with the Pearson's correlation coefficient.

To understand the intuition, let's consider two situations with two clusters. In the first situation, cluster 1 involves spikes with waveforms close to the left one and cluster 2 involves spikes with waveforms close to the right one (see figure 4.1 for the two spikes). If pairs of spikes are taken separately in each cluster, then the values obtained are close to 1, since spikes are similar. In the second situation, consider that some spikes of the cluster 2 are included in cluster 1. In this case, depending on the situation, values vary between 1 and a lower value.

To distinguish these situations, the average ($R_{correlation}$) of all the correlation measures of pair of points is computed and that for each cluster C and defined as follows:

$$R_{correlation}(C) = \frac{1}{N} \sum_{i,j \in C, i<j} corr(S_i, S_j) \tag{4.31}$$

where S_i the spikes considered as time-dependent random variables, and N the normalization factor. This latter is equal to $(n_C * n_C - n_C)/2$, where n_C is the number of spikes in cluster C. $n_C * n_C$ represent all the possible pairs of spikes. n_C is subtracted since the $corr(S_i, S_i)$ are not of interest. Finally the factor 2 is applied since $corr(S_i, S_j) = corr(S_j, S_i)$.

Based on that, good clusters have $R_{correlation}$ value close to 1 and bad clusters have $R_{correlation}$ value close to 0.

4.5.3 Validation criteria

The purpose of the validation criterion is to check if the results provided at the end of the various stages of the process are valid and that, based on a reference database for

which the desired results are known (see subsection 2.2.4 for more details). Among the validation criteria in literature, Rand Index and Waverform reprentation are chosen and detailed in the next subsubsections.

4.5.3.1 Rand Index

In our case, the Rand Index is considered, introduced for the first time in [78]. Again, it is a dependent criterion for measuring similarity between two different clusters of the same database.

The Rand Index is based on three observations. The first is that the clustering algorithm provides a discreet result. In fact, an integer value is assigned to each point, representing the membership of a group. Secondly, clusters are defined as much by the points that are contained within them than by the points present in the other clusters. Finally, the third observation is that all the points are of equal importance in the determination the clusters.

Let's consider the database X and the results Y and Y' of two clustering process applied on it. The measure of similarity between the two clusterings can be determined as $c(Y,Y')$ and represents the number of point-pairs with identical behaviors, normalized by the total number of point-pairs that can be generated. Identical behavior are characterized by the two following situations:

- If two points belong to a same cluster in Y, then they also belong to the same cluster in Y'
- If two points are located in different clusters in Y, then they are also in two different clusters within Y'

.

Rand Index is hence defined as:

$$c(Y,Y') = \frac{a+b}{a+b+c+d} \tag{4.32}$$

where a is the number of point-pairs of X that belong to a same cluster both in Y and Y', b is the number of point-pairs of X that belong to different clusters both in Y and in Y', c is the number of point-pairs of X that belong to a same cluster in Y and to the different clusters in Y', and d is the number of point-pairs of X that belong to different clusters in Y and to a same cluster in Y'.

It can be noticed that $c \in [0,1]$, $c = 0$ meaning the two clusterings have no similarity and $c = 1$ meaning the two clusterings are identical.

4.5.3.2 Waverform representation

Waveform representation is really simple in its application. Indeed, it consists in plotting the waveforms belonging to the same cluster on a unique figure in addition to their mean. If waveforms are almost the same, the mean of the waverforms is confounded with the superposition of the waveforms. In this case, the clustering result is assumed to be correct (see subsubsection 3.4.2.2 for more details).

4.6 General Procedure

The performance of the general process of spike sorting applied on those two sets will be assessed to achieve the desired goal. The different levels of performance will be compared and a analysis of the features to retain will be performed. That will be able to if the defined features are helpful in the discrimination of neural clusters. The general process followed in this work is presented on figure 4.11 and can be divided in four different phases.

In the first one (see phase 1 on figure 4.11) and from the recording neural activity stored in the databases, for which explanations are proposed in section 4.2, spike detection is realized. It consists in the localization of the peaks (for this database, the localization are given) and in the extraction of the spikes as a 1.5 ms frame centered on them (0.5 ms before the maximum peak and 1 ms after). The process continues with the extraction of the 15 features of interest (see section 4.3) from the scaled spikes and first and second derivatives of the spikes, which are immediately rescaled in the $[-1;1]$ interval. Since the relative importance of each feature is not known in advance and that

their amplitude ranges are different, this rescale tends to attribute no prior weight to a feature in comparison to others.

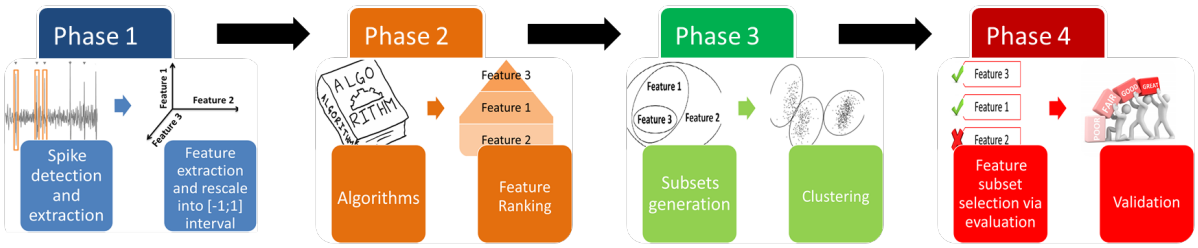

Figure 4.11: Synthetic general process. Illustrations adapted from [12, 89].

The second phase (see phase 2 on figure 4.11) consists in the ranking of the features, computed via the algorithms presented in section 4.4, what gives rise to the feature sets ranked according to their relevance. This relevance should be different for each of the considered algorithm, since each of them utilize different search technique and relevance definition. This step represents subset generation (see 2.2.1 for more details).

Once this is done, the third phase (see phase 3 on figure 4.11) can start. It consists in the application of the $k-means$ clustering algorithm (with 500 iterations to reach the local optimum and 10 replicates of the process) for the subsets generated by increasing the number of features. To better understand, let's consider that the output of the previous phase is the following ranking R: $R = \{X, Y, Z\}$, where X, Y and Z are the features. In this phase, the $k-means$ algorithm is applied on the following subsets R_i: $R_1 = \{X\}$, $R_2 = \{X, Y\}$ and $R_3 = \{X, Y, Z\}$. It is a classification task since the number of clusters is known in advance for each recordings.

In a final fourth phase (see phase 4 on figure 4.11), the partitioning result provided by the clustering algorithm is assessed by the two evaluation criteria (L_{ratio} and $R_{correlation}$, see subsection 4.5.2 for more details) to select the relevant features. The process ends by computing the performance of the selected number of features via Rand Index and by representing waveforms to validate the results (see subsection 4.5.3 for more details).

Part III

Results and reflection

CHAPTER 5

ANALYSIS OF THE RESULTS

As a reminder, the general process that has been followed to obtain results is presented on figure 5.1. It starts with the detection and extraction of spikes from which features are determined. On this basis, algorithms are applied to provide, from their own definition of relevance, a ranking between them. In a third step, subsets generation is performed. It consists in the creation of feature sets containing an increasing number of relevant features, from which the $k-means$ clustering algorithm is applied to generate the spike groups. The process ends by the selection of the most discriminatory subset and the validation of this selection is realized.

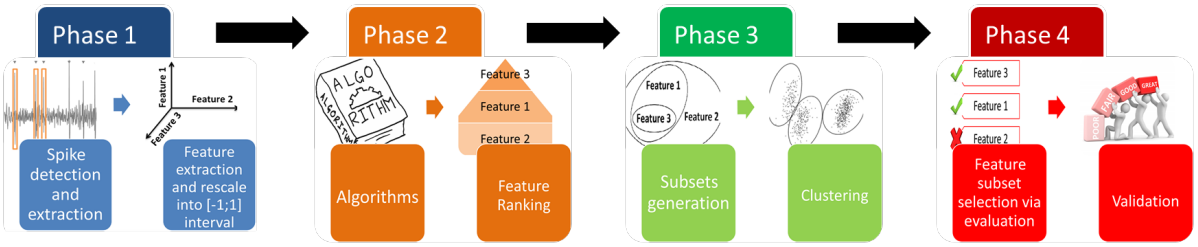

Figure 5.1: Synthetic general process. Illustrations adapted from [12, 89].

This chapter of the work is dedicated to the results. It proposes firstly a review of the starting points of the work (spike extraction and clustering task), directly followed

by a first investigation of the features extracted from the spikes. After, the way the different feature selection algorithms rank the features according to their measure of relevance is discussed. This is followed by the technique proposed for the selection of the most relevant subsets for this task. A description of their individual performance achievements and by a confrontation to the PCA results is then proposed. Finally, the validation of the process is executed.

5.1 Starting points and difficulty of the task

This section describes the starting point of the process by illustrations of the extracted waveforms and try to make understand the reader how difficult the clustering task can be.

5.1.1 Spikes

The first step of the procedure is to detect and extract spikes from the synthetic database, what can be realized via the indexes localizing them and the creation of a single frame of 1.5 ms around the positive peak (0.5 ms before and 1 ms after).

On figure 5.2, a sample of a recording (left) containing 4 different spikes with annotations concerning the localization of spikes is represented. A plot of the corresponding extracted spikes (right) is proposed. From this last representation, the goal is to recover the different types of spikes.

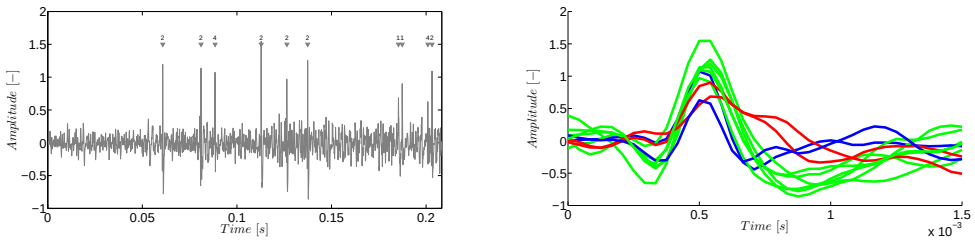

Figure 5.2: Sample of a recording (left) containing 4 different spikes with annotations concerning the localization of spikes and the plot of the corresponding extracted spikes (right), for a 0.15 noise level. The correspondences are the following: green = 2, blue = 4 and red = 1. The third type of spike is not present in this sample.

In this small representation and with few experience in the domain, the task is not so complicated and can be carried out quiet easily, and manually if needed. Now, consider the entire signal, still containing these 4 spike types and locations. On figure 5.3, the spikes present in this database are represented on a single graph.

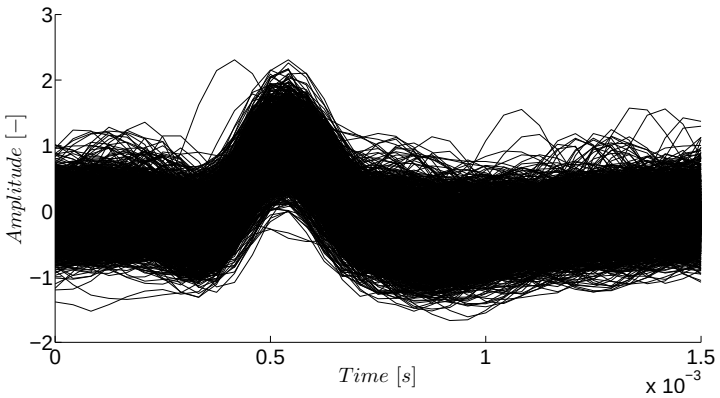

Figure 5.3: Superposition of the 2849 spikes extracted for a database with 4 different types of spikes and 0.15 noise level.

The visual inspection makes immediately appear the difference. Whereas this grouping task was easy before, it becomes rapidly more complicated/intractable in this situation, as well as time consuming since there are 2849 spikes to classify in this representation. Some artifacts, due to noise certainly, are present, making this task even more difficult.

To perform it automatically, machine learning tools are used. The most common one is certainly PCA. This algorithm considers each spike as a point in a hyper-space where coordinates/variables are the time points of the spike. Based on that, it creates new coordinates for this point by linear combinations of the previous ones. The first variable is build so that it be really different for each spike. The successive other coordinates are created with the same process but without representing the information already contained in the previous coordinates. The goal here is to be able to differentiate the spikes only from a small number of first coordinates.

However, this technique does not take into account the knowledge and typical shapes of spikes. The approach developed in this work tends to palliate this lack. It is similar to PCA since it also creates new coordinates for the spikes, but the coordinates represent

physical characteristics of the spikes, such as the percentage the positive amplitude represents among the peak-to-peak amplitude, the energy contained in the signal, etc.

5.1.2 Clustering task

Once these new coordinates are created, the clustering task can be operated. It consists in grouping together the spikes with similar coordinates. However, some coordinates are not of interest in this context and their cancellation can either facilitate the process or does not change anything.

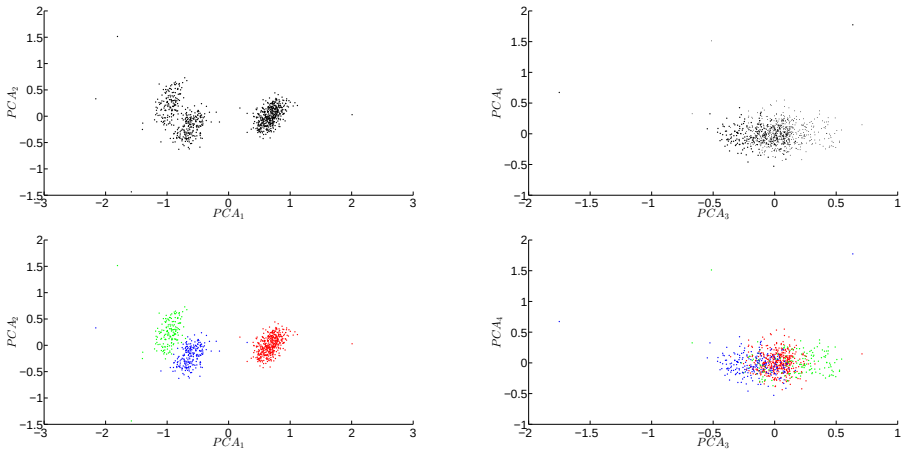

Figure 5.4: Representation of the same spike points, first versus second PCA components (left) and third versus fourth PCA components (right) for 777 spikes extracted from a database containing 3 types of spikes and a 0.05 noise level. Each color represents one type of spike. The clustering process performs good results for the representation on the left and poor results for the representation on the right.

To understand that, on figure 5.4, let's consider the representation of 3 spike types according to their first and second PCA components (left) and third and fourth PCA components (right). With a quick look at the black and white graphs (upper ones) and knowing that there exist 3 populations in this representation, it is obvious than the separation of data in 3 homogeneous sets is extremely easy in the left representation and difficult in the right one, whereas the spikes are represented according to different pairs of coordinates. Furthermore, the discrimination that is made here leads to good results for the left representation and to poor results for the right representation. Indeed, with the colors on the lower graphs that depict the belonging of each spike to a class

(all points in red make a group, all points in blue form another one and all points in red give rise to a third one), it is clear that the left representation is preferred in this task.

Consequently, this simple example demonstrates the importance of the selection of the coordinates/variables from which spikes can be represented. From the four coordinates of the above representations, it is clear that the third and fourth PCA components are not of interest in this clustering task. Indeed, first and second PCA components are sufficient to allow an easy visualization of the spike groups and lead to good results.

5.2 First investigation of the feature space

First PCA components often represent good discriminators for the detection of neuron types. However, it is not easy to figure out what represents these components in terms of physical characteristics of spikes. It is then interesting to build coordinates/variables that are understandable at first. And, with a smart choice of some them, it could be possible to have as clear representation like what PCA provides.

Targeting this objective, 15 meaningful features are created and rescaled in the $[-1;1]$ interval so that each of them have the same weight to avoid they impact the process due to their range of values:

A^+ *Positive relative area*. Fraction of the area above the 0 amplitude.

Amp^+ *Positive relative amplitude*. Percentage the amplitude of the positive peak represents against the total amplitude of the signal.

AVG *Average relative value*. Average of the signal.

D_{peak} *Relative Peak-to-Peak Duration*. Fraction of the peak-to-peak amplitude against the spike duration.

E *Relative energy*. Energy of the signal.

$\dot{A}^+$ *Positive relative area of the first derivate of the signal*.

$\dot{Amp}^+$ *Positive relative amplitude of the first derivate of the signal*.

$\dot{AVG}$ *Average relative value of the first derivate of the signal.*

$\dot{D}_{peak}$ *Relative Peak-to-Peak Duration of the first derivate of the signal.*

$\dot{E}$ *Relative energy of the first derivate of the signal.*

$\ddot{A}^+$ *Positive relative area of the second derivate of the signal.*

$\ddot{Amp}^+$ *Positive relative amplitude of the second derivate of the signal.*

$\ddot{AVG}$ *Average relative value of the second derivate of the signal.*

$\ddot{D}_{peak}$ *Relative Peak-to-Peak Duration of the second derivate of the signal.*

$\ddot{E}$ *Relative energy of the second derivate of the signal.*

A representation of their distribution under the form of histograms are presented on figure 5.5. It can be observed that each feature behaves differently. Some of them (A^+ and AVG for example) seem to make already appear types of neurons (with peaks that can be clearly identified).

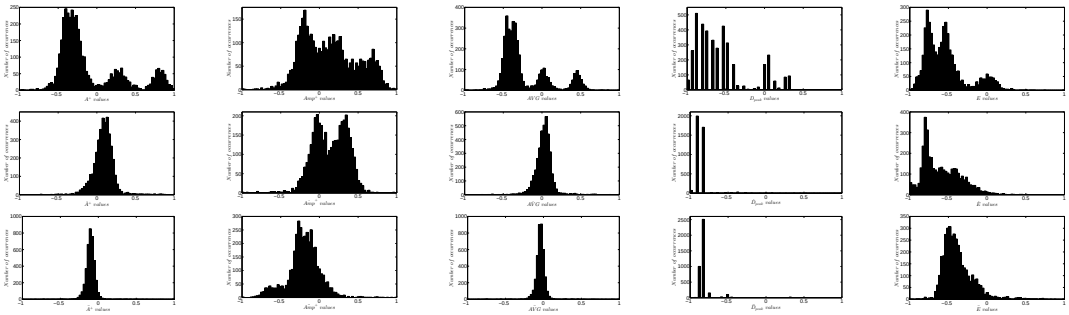

Figure 5.5: Histograms (75 bins) of the 15 meaningful features. From left to right and in descending order: A^+ (first left), Amp^+ (second left), AVG (middle), D_{peak} (first right) and E (second right) applied on spike (first row), first (second row) and second (third row) derivative of the spike.

However, to distinguish at best the different neurons, combinations of features must be considered. A representation of the 2-dimensional spaces engender by some of these pairs is proposed on figure 5.6.

It can be already observed that, for some pairs of features, the discrimination is possible, but that it is not the case for all of them and not specially in a 2-dimensional

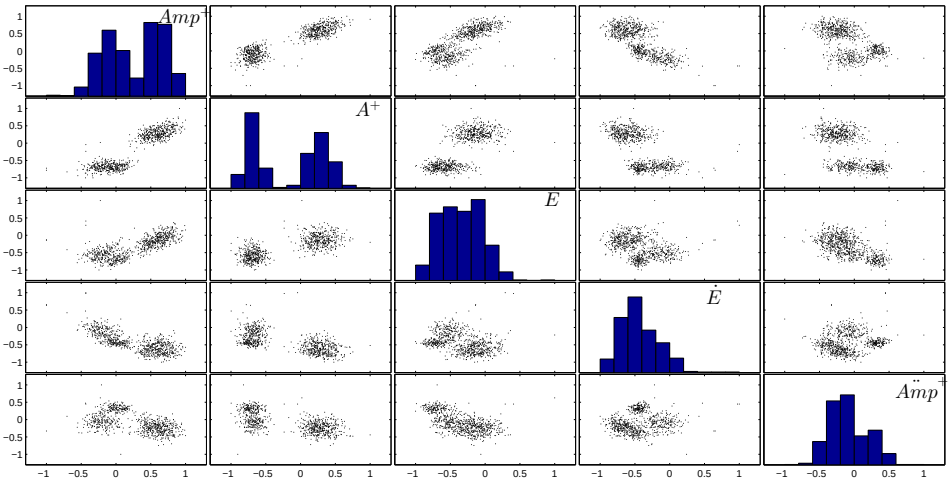

Figure 5.6: 2-dimensional space engendered by the pairs of the Amp^+, A^+, E, $\dot{E}$ and $\ddot{Amp}^+$ features for a database containing 3 types of spikes and a 0.05 noise level.

space. Indeed, some features may be irrelevant for this task when they are coupled with another feature, but can reveal their interest when they are combined with multiple ones, what is not possible to illustrate in 2 or 3-dimensional spaces.

5.3 Feature Ranking

With PCA, the relevance ranking is already known since the first component contains the most of information, the second one a bit less, etc. The relevance of the new features, extracted from the spike shapes, must then be assessed. This order among variables is called feature ranking, what is essentially characteristic to filters, algorithms that do not utilize the clustering process to achieve this task (see subsubsection 2.2.2.1 for more details).

Let's consider that the input of the feature subset selection algorithm is the following: $\{X;Y;Z\}$, where X, Y and Z are features. If the output of the feature ranking process is $\{Y;Z;X\}$, then its means that Y is more relevant than Z, which is itself more relevant than X. This step establishes the relevance order of the variables.

As results of this step, two types of rankings can be envisaged and summarized on figure 5.7.

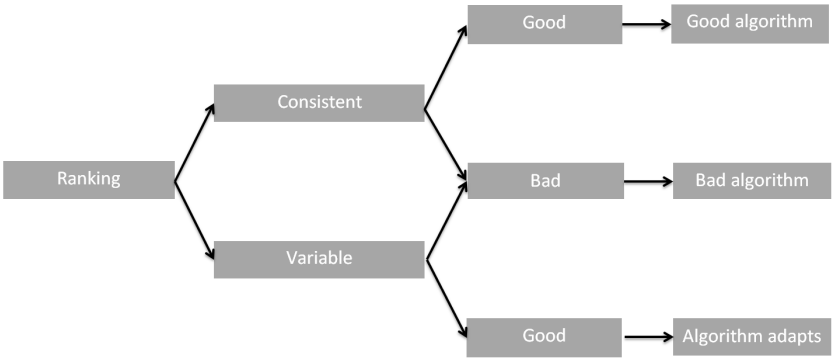

Figure 5.7: Different cases of ranking situations with their associated causes.

The first one is the one for which whatever the number of clusters in the database, the features are always ranked in the same order ("consistent" path on figure). In this case, the conclusion that some variables are really more relevant than others comes naturally. Hence, according to the feature selection algorithm, the ranking and the process of feature selection can lead to good results in the case the ranking is well done ("good algorithm" on figure) and to poor results if the ranking is actually not correct ("bad algorithm" on figure). More specifically, if the ranking is conserved whatever the number of clusters in the database, it can be envisaged that the retained features are almost always the same, what means that these variables represent features that must be taken into account in the spike sorting process.

The second one is the one for which the feature rankings vary with the number of clusters in the database ("variable" path on figure). It means that the relevant order of the features depends on the given number of clusters. If poor performance level is reached at the end of the process, the feature ranking algorithm is assumed to be bad ("bad algorithm" on figure). In the other hand, if a high performance level is realized, then the algorithm possesses the ability to adapt to the database ("algorithm adapts" on figure). In this case, the best performance results can be obtained if the number of retained features also vary with the situation. More precisely, in such a situation, the selected features are not the best for the spike sorting process.

However, since there exist different feature algorithms and that the definition of relevance and/or similarity between features are different for each of them, the information contained in the extracted features may be understood differently. This leads to the fact that the two previous types of rankings are not separated by strict and rigid boundaries and must be considered with some flexibility.

This last remark is especially relevant in this case. In appendix A, tables containing the ranking of the features according to the number of clusters in the data for a noise level equal to 0.15 and for the Backward Elimination (table A.2), Feature Dispersion (table A.5), Forward Selection 1 (table A.8), Forward Selection 2 (table A.11) Kernel Density Estimation (table A.14) and Simple Ranking (table A.17) algorithms are provided.

Based on them, it can be observed that the ranking changes with the database, what means that the ranking is sensitive to the number of clusters in the database, with some relative constancy however (see Feature Dispersion, Forward Selection 2 and Simple Ranking algorithms). In addition to that, it appears that each algorithm ranks the features differently, with some similarities however for several of them (Backward Elimination, Forward Selection 1, Forward Selection 2 and Simple Ranking). Furthermore, it seems that all the algorithm comes to the fact that the relative positive and negative amplitudes of the first and second derivatives of the signals show a low relevance according to the feature set.

5.4 Selection of the relevant features

Let's consider the output of subset generation step is the following set of ranked features: $\{F_1;F_2;\cdots;F_{14};F_{15}\}$. Than, for the sets of increasing number of features ($\{F_1\}$, $\{F_1;F_2\}$, $\{F_1;F_2;F_3\}$, ...), the $k-means$ algorithm is applied, with the number of clusters contained in the database that is known. Among all subsets, the one that provides the best discrimination between the clusters must be selected. This step is called feature subset selection. To perform that, L_{ratio} and $R_{correlation}$ are used. In this section, it is shown that the utilization of these two measures together contributes to a better choice of the most discriminatory subset than them considered separately. Based on them, a technique for this selection is proposed and allow the detection of high levels of performance, without being especially the best ones.

5.4.1 Selection via L_{ratio}

As a reminder, L_{ratio} measures if one cluster is well separated from the other data points. The lower is its value, the better the separation is (see section 4.5.2.1 for more details).

If the L_{ratio} values of all clusters are plotted according to the number of features belonging to the feature set, the decision of choosing a certain subset instead of another can be made. But let's imagine the situation in which the database contains 3 small well separated clusters and 1 cluster with a poor result in terms of L_{ratio} and representing a lot of spikes. Such a representation cannot allow the user to be conscious of the exact situation. To overcome this problem, the same representation is operated, but the size of a plotted L_{ratio} value of a cluster is modified to be proportional to the percentage of spikes that this cluster actually represents.

With such a representation, the choice of the subset to keep in the process is the one for which the L_{ratio} values are the minimum and for which the L_{ratio} values of the clusters representing the higher number of spikes are really small. A representation of the scaled L_{ratio} values are presented on figure 5.8 for Forward Selection 2 algorithm and a database containing 5 clusters. Similar graphs are proposed in appendix C on figures C.1, C.2 and C.3.

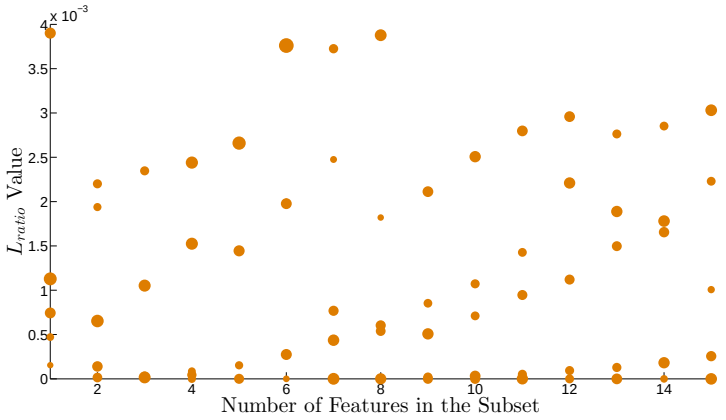

Figure 5.8: L_{ratio} values according to the number of features in the feature subset for Forward Selection 2, 5 clusters and a 0.15 noise level. The size of the data points are proportional to the number of spikes in the represented cluster.

On this figure, since 5 clusters are present, 5 values of L_{ratio} are plotted for each number of features in the subset. The size of each point shows qualitatively the quantity of spikes it represents. It can be observed that the choice of the number of features to retain is not trivial. Indeed, even with the size of the points, L_{ratio} values are spread on a large range. However, if a choice had to be made in this situation, the subsets with 2 and 9 features would be retained, because they possess similar values and configurations. Hence, this graph cannot indicate easily by itself the number of features to retain.

5.4.2 Selection via $R_{correlation}$

The help in the choice of the number of features to retain, another possibility consists in utilizing the average correlation value $R_{correlation}$. As a reminder, its measures the average of the Pearson's correlation coefficients, what represents how linearly dependent are two random variables, and this for each cluster (see subsubsection 4.5.2.2 for more details). Just as for L_{ratio}, these values are scaled in size to visually indicate how many spikes the cluster represents, to facilitate the decision. Hence, the choice of the feature subset to keep in the process is the one for which $R_{correlation}$ values are the maximum and $R_{correlation}$ values of the clusters representing the higher number of spikes are really high.

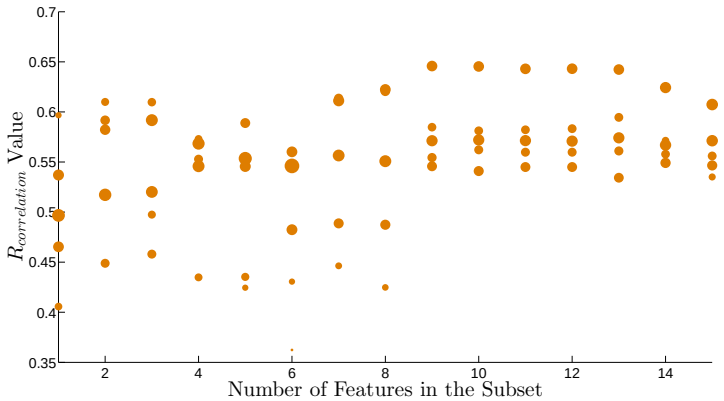

Figure 5.9: $R_{correlation}$ values according to the number of features in the feature subset for Forward Selection 2, 5 clusters and a 0.15 noise level. The size of the data points are proportional to the number of spikes in the represented cluster.

On figure 5.9 those values are plotted. Just as it was the case for L_{ratio}, $R_{correlation}$ values representation does not directly allow the choice of the number of features to

retain.

However, if a choice had to be made in this situation, the subsets with 9 to 12 features would be retained, because they possess similar values and configurations. Hence, again, this graph cannot indicate easily by itself the number of features to retain.

5.4.3 Proposed selection technique

Instead of taking these two evaluation measures individually, it may be interesting to confront their information so that a general and unique solution emerges. To do so, their representations are placed in parallel (see lower and middle graphs on figure 5.10).

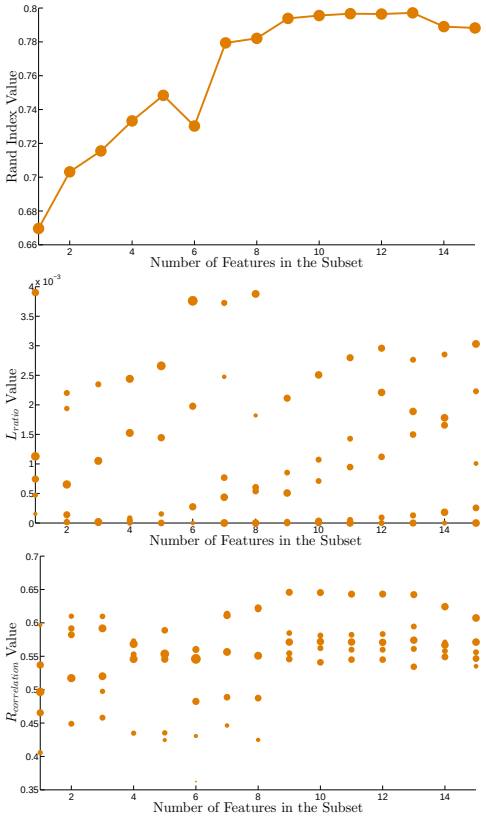

Figure 5.10: Representation of Rand Index (upper graph), L_{ratio} (graph in the middle) and $R_{correlation}$ (lower graph) values according to the number of features in the feature subset for Forward Selection 2, 5 clusters and a 0.15 noise level. The size of the data point are proportional to the number of spikes in the represented cluster for L_{ratio} and $R_{correlation}$ graphs.

The first element it can be interesting to notice is that L_{ratio} proposes a different choice (between 2 and 9) than $R_{correlation}$ (9 to 12) for the selection of the number of features to retain. However, with a quick look at those two graphs in parallel, it appears that 9 represents a good trade-off between these two evaluation measures.

Once the choice of the number of features to retain is made, the associated performance level must be examined. Indeed, if this selection leads to poor performance results, then it means that the selection technique is uninteresting. If this number of features is reported on the graph representing the Rand Index values (see upper graph on figure 5.10), it can be observed that the related Rand Index value is not the highest one, but that it is one of the best, the Rand Index indicating the performance level.

Therefore, to summarize, the process of selecting the number of features is the following. In a first step, L_{ratio} and $R_{correlation}$ values are examined independently to see if certain features show really good results, results above the mass. Then, the graphs provided by those two evaluation measures are analyzed in parallel and a choice is made from the best results of the two representations. Hence, this procedure uses well the information brought by L_{ratio} and $R_{correlation}$. Applying this technique on graphs of figures C.1, C.2 and C.3, it can be observed that high level of performance can be detected, without being the best every time.

5.5 Performances

To assess the quality of the feature subset selection step, a measure of performance is computed: the Rand Index. It compares two sets of indexes representing the belonging of points to clusters. The output value is the ratio between the number of point-pairs that are well classified and the total number of point-pairs. This measure stands in the $[0;1]$ interval. The higher the ratio is, the better the process is (see subsubsection 4.5.3.1 for more details).

This section illustrates that PCA is slightly superior to the other algorithms applied on the extracted features, with however certain exceptions in which PCA is outperformed. In addition to that, it shows that for the selected algorithms, there exists a number of features above with the performance level does not increase or at least not that much,

what means there is a kind of saturation in the performance level.

On figure 5.11, the Rand Index value obtained for each feature subset selection algorithm and for the feature subsets generated from the ranking process are presented. The number of clusters is equal to 5 and the noise level to 0.15. The results for the clusters 2 to 9 are presented in appendix B.1 on figure B.1.

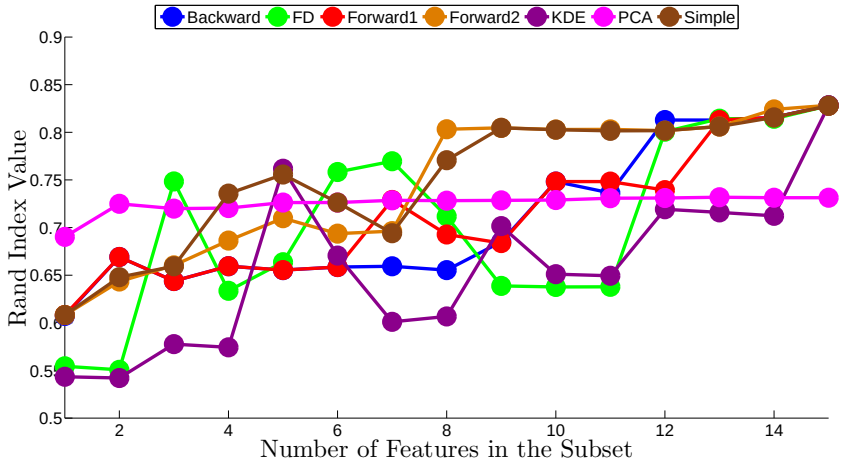

Figure 5.11: For the clustering of 4 clusters, representation of the Rand Index for an increasing number of features, for each algorithm and for a noise level equal to 0.15. The maximum number of features is equal to 15 for each algorithm except PCA since the database does not count more than 15 features. For PCA, this stop level has been arbitrarily set. The concordance is the following: Backward = Backward Elimination; Forward1 = Forward Selection 1; Forward2 = Forward Selection 2; Simple = Simple Ranking (for them, see subsection 4.4.1); KDE = Kernel Density Estimation of the Feature Density Similarity (see subsection 4.4.2); PCA = Principal Component Analysis.

The analysis of these results can be done of two ways. A first one consists in an observation of each curve, algorithm by algorithm. For Backward Elimination, the curve increases when the number of relevant features in the subset increases. It starts around 0.58 to increase fast until reaching 9 relevant features in the subset, from which its rise reduces to finally attain approximately 0.78. Feature Dispersion behaves a bit differently than the other curves since its performance level diminishes when the subset includes the second most relevant feature, to finally reach roughly 0.78. Forward Selection 1 is approximately constant (0.78) since more than 3 relevant features belong to the subset. Forward Selection 2 shows a similar behavior than Backward Elimination with however a sightlier increase. Kernel Density Estimation approximately follows Forward Selection

2. PCA a constant behavior, notably when more than 4 features belong to the subset. It shows in addition a high performance level (around 0.83). Finally, Simple Ranking suffers from a decrease when the fourth most relevant features is added to the subset and a reincrease more than 5 features are present in the subset.

If a general overview, based on figure B.1, is made, it can be observed that PCA curve seems to be relatively constant with no increase for a number of features superior to 2-3 features. Furthermore, the general tendency of the other algorithm curves tends to show a saturation of the performance value (Rand Index value) when the number of features in the subset increases. In addition, Feature Dispersion often depicts a decrease in its performance level when the second and/or the third feature are added.

A second analysis consists in a comparison between the curves. In this view, it can be observed that, for 1 feature in the subset, PCA is better than Forward Selection 1 and 2 and Simple Ranking, that are themselves better than Kernel Density Estimation, that is itself better than Feature Dispersion and Backward Elimination. When the second feature is added, Kernel Density Estimation crosses the algorithm that had better performance results, except PCA. When the third feature joins the subset, Feature Dispersion outperforms Backward Elimination. Between 4 and 9 features, the algorithms cross each other to finally reach a common performance level of approximately 0.77 when all features are in the subset.

If a general overview, based on figure B.1, is made, it can be observed that PCA provides the best results in 5 situations of the 8 ones. In the other cases, the PCA is outperformed or equalized. It however appears that each algorithm tends to reach the same performance level than the other ones when the number of features in the subset increases. Forward Selection 1 tends to be the algorithm that levels off the most quickly.

5.6 A word about noise

Noise on data generally reduces the performance. This can be easily understood by the fact that if noise impacts the spike representation, it also impacts feature extraction which provides other outputs than if there is no noise. To evaluate its consequences, the following two subsections discuss its impact on the feature ranking step and the

performances and show that both ranking and performance are impacted.

5.6.1 Impact on Ranking

The first step the noise can impact in the process investigated in this work is feature ranking. Since it is the preliminaries of the entire process, the consequences can spread throughout the different phases and finally degrade the results.

In appendix A, tables showing the ranking of the features by each algorithm are presented, for each database associated to each number of clusters and for 0.05, 0.015 and 0.30 noise levels. It turns out that the increase of the noise level dramatically modify the ranking of the features. Algorithms for which a relative constancy in the feature ranking was observed (see section 5.3) are also touched and loose it.

5.6.2 Impact on Performances

What really matters in the spike sorting process is the performance evaluated at the end. Since noise can impact it, it must be discussed and its consequences on each individual algorithm and the comparison between algorithm must be analyzed.

On figure 5.12, the Rand Index value obtained for Kernel Density Estimation (KDE) algorithm and for an increasing number of features in the subset are presented. The number of clusters is equal to 5 and the noise levels respectively equal to 0.05, 0.10, 0.15, 0.20 and 0.30. The results for the other algorithms and in the same situation are presented in appendix B.2 on figure B.3.

It can be observed that all curves start at a certain level, perform a trajectory and end at a higher level than the starting point. Furthermore, between 1 and 6 features, the curves cross each other meaning that the grading of performance level moves in that range. However, at the end, they do not and show that the higher performance levels are attributed to low noise level and that the performance decreases when the noise increases.

Based on figure B.3, if a general overview is made, it can be observed that the tendency evoked in the previous paragraph is still applicable. Indeed, it seems that until

a certain number of features the curves representing the performance for each noise level can cross each other, meaning that the noise does not specially influence their performance. However, when the number of features is sufficiently high as well as the performance level, there is a strict grading of the curves for which the higher performance levels are reached by lower noise levels and the contrary for lower performance levels. This is true for all algorithms, except for the Principal Component Analysis (PCA) for which the strict grading is present from 1 feature. This is also true for the other number of clusters. Some illustrations of this are proposed are proposed on figure B.2 and B.4 of the same appendix. This concludes to the fact that the noise degrades the performance of the spike sorting process.

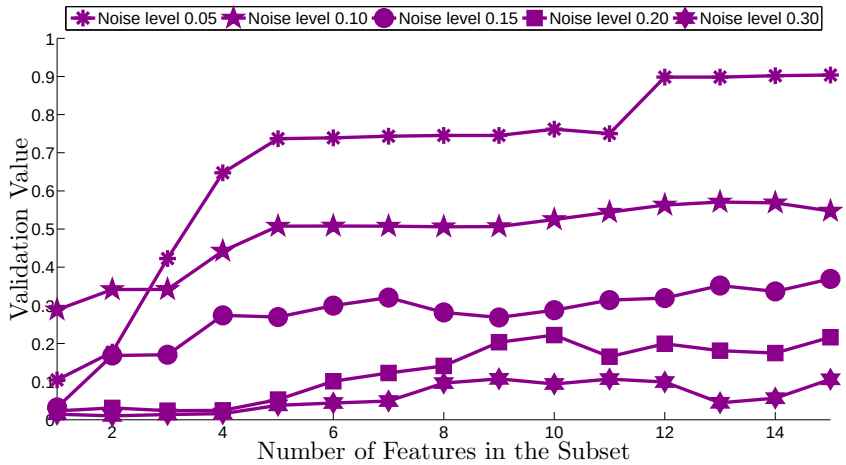

Figure 5.12: Representation of the Rand Index value for the Kernel Density Estimation (KDE) algorithm according to an increasing number of features, for 5 clusters and a noise levels being respectively equal to 0.05 (asterisk), 0.10 (star), 0.15 (circle), 0.20 (square) or 0.30 (hexagonal star).

Furthermore, based on figures B.5 and B.6 and in comparison with figure B.1 representing the graph of Rand Index for the algorithms and for noise level respectively equal to 0.05, 0.30 and 0.15, it can be observed that the explanations made on the comparison between the algorithms in section 5.5 are still applicable. In other words, it means that the behaviors of algorithms between each other are the same, whatever the noise level. The curves seem indeed to be shifted along the Rand Index axis.

5.7 Waveform Representation

Waveforms representation is another possible validation technique for spike sorting. It consists in plotting the different spikes of a cluster on a single graph. Hence, if the database contains n clusters, n graphs are created, one for each cluster (see subsubsection 4.5.3.2 for more details). The waveform representation strengthens the fact that the technique proposed earlier provides good results.

To illustrate that, figure 5.13 shows waveform representation for a database with 3 clusters, a 0.15 noise level, and a number of features providing a low level of performance (low value of Rand Index, graphs on the left) and for a number of features selected via the technique explained in section 5.4 and providing high level of performance (high value of Rand Index, graphs on the right). Here, since waveform representation is computed for each cluster, there are three graphs in each case. The average waveform, in cyan, is obtained by averaging the waveforms present in the cluster.

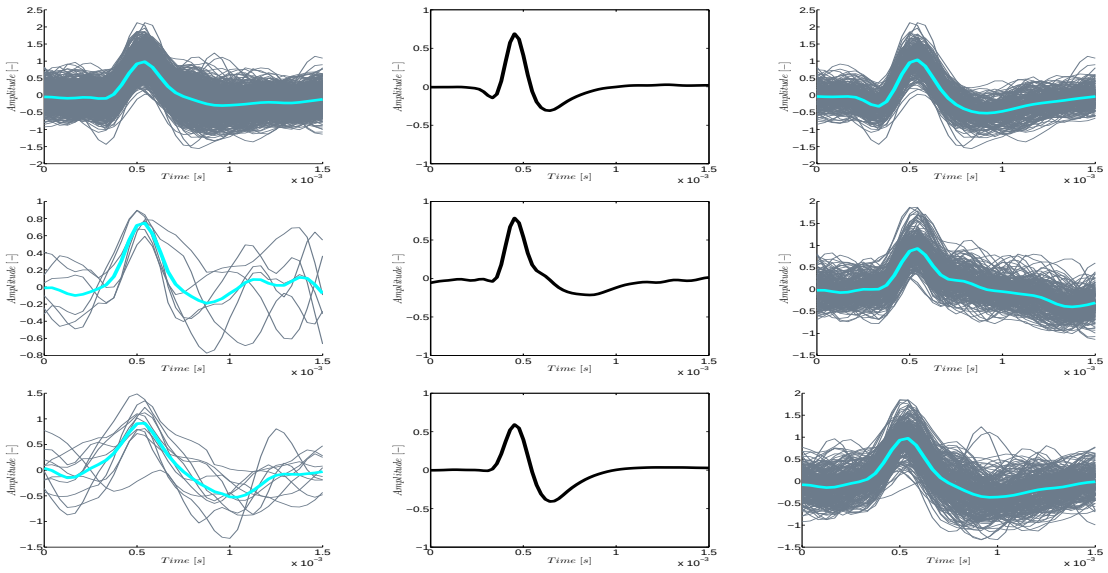

Figure 5.13: Waveforms representation (left and right in gray), average waveform (left and right in cyan) with the associated spikes (middle in black) for Feature Dispersion algorithm, a 0.15 noise level, a database containing 3 clusters and for 1 (left with poor performance) and 12 (right with higher performance) selected features. The selection of the 12 features are realized on the basis of the technique presented in section 5.4 while the first feature by examination of the Rand Index representation. The average waveform is obtained by averaging all the waveforms of the spikes present in the cluster.

It can be observed that the representations contain a high quantity of noise (characterized by signal out of the band). Due to that, false positives and negatives can be present, impacting the average waveform as well as the performance levels discussed in the previous sections. However, it can be seen on this figure that the waveform representation for Feature Dispersion and for 12 features, number determined by the technique presented in section 5.4, seems to provide a better average waveform. Indeed, it appears that this latter is closer to the original signal than the average waveform of the others. Waveform representation validates hence the proposed technique.

Part IV

Conclusion

CHAPTER 6

DISCUSSION

Throughout this work, it has been shown that the "smart" selection of features extracted from the spikes themselves provide similar performance results than Principal Component Analysis, and sometimes better performance levels, if the feature subset selection process is correctly realized. These features are positive relative area, positive relative amplitude, average relative value, relative peak-to-peak duration and relative energy applied on signal and first and second derivatives of the signal. The algorithms utilized to produce such results are Backward Elimination, Feature Dispersion, Forward Selection 1, Forward Selection 2, Kernel Density Estimation and Simple Ranking.

Furthermore, a technique for visual selection of the relevant number of features that must be conserved in the process has been proposed. It consists in making a pre-selection of a range of features based on low values of L_{ratio} and $R_{correlation}$to finally select the best number of features according the examination of their representations in parallel. This technique achieves to reaching high level of performance, without finding systematically the optimal one. Moreover, a visual inspection of the waveform representation confirms its efficiency.

In addition to that, it can be observed that Principal Component Analysis (PCA) reaches a constant level of performance when the first three components are kept whereas Feature Selection achieves its constant level for a higher number of components. This could be explained by the fact that, for PCA, the majority of the information is contained

in a few components while the information is spread on more variables for feature subset selection, meaning probably that the spikes are really represented by the combination of features.

Concerning the ranking, it has been shown that there was no strict agreement between the algorithms to grade the feature relevance, even if some tendency can be observed, more precisely concerning the less relevant features. This is certainly due to the different relevance definitions these algorithms use to determine this categorization. In addition, the fact that noise does not modify the behaviors of the algorithms between each other, even if it decreases the performance level, might indicate that the algorithms adapt to the noise level or that the combination of the features leads to the success of the process more than the features evaluated alone.

However, the performances of this process are related to the database that was used and it is not proven here that the same level of performance can be achieved with other data, even it should be. In addition, to create feature ranking, feature space has been investigated through sequential search. Higher performance level could be found by exhaustive search and leads to a constant outperforming of the algorithms regarding to PCA.

Finally, this new approach gives rise to an help to spike sorting. Indeed, this techniques provides a powerful tool for a dimensionality reduction in a 15 dimensional space, leading to a decrease of the computation time. Furthermore, because the immediate selection of physical features from spikes leads to high level of performance, this work reopens this search field since it contradicts the statements of Wheeler and Heetderksl in [92].

CHAPTER 7

CONCLUSION

The goal of this work was to bring a contribution to the understanding of the visual system with a focus on neuronal responses and more precisely in studying spike sorting on electrophysiology recordings. To do so, smart features are extracted from spikes to show that the performance results of spike sorting from them is at least equal to the one of Principal Component Analysis (PCA), the most commonly used technique.

Chapter 2 provides the background on spike sorting and feature subset selection to the reader to finally show the interest of using these techniques together. Chapter 3 proposes a review of the literature, ending with the observation that there was a need for a new approach. Chapter 4 describes the material, methods and an explanation of the general process used to achieve the goal of this work.

Finally, chapter 5 exposes the results of the investigation. It was shown that the new approach often provided similar results than PCA, and sometimes better ones. To achieve such performance levels, a visual technique for the selection of the relevant features is also proposed and is able to detect high performance level. That gives rise to the fact that the application of feature subset selection on meaningful features leads to promising results for spike sorting.

Although the results presented here demonstrate the effectiveness of this approach, it could be further developed in a number of ways. The first one could be the unsupervised

implementation of the feature selection technique, based on L_{ratio} and $R_{correlation}$, since it can be time-consuming for neuroscientists. A second possibility could be the utilization of other feature subset selection algorithms that can reach better results and the study of their sensitivity to noise in the ranking process. Finally, so that this technique be officially recognized to provide high performance results, it could be interesting to use other databases and other validation criteria, even if the results presented here are convincing.

BIBLIOGRAPHY

[1] D. A. ADAMOS, E. K. KOSMIDIS, AND G. THEOPHILIDIS, *Performance evaluation of pca-based spike sorting algorithms*, Computer Methods and Programs in Biomedicine, 91 (2008), pp. 232 – 244.

[2] M. ALIBEIGI, S. HASHEMI, AND A. HAMZEH, *Unsupervised feature selection using feature density functions*.

[3] O. ALTER, P. O. BROWN, AND D. BOTSTEIN, *Singular value decomposition for genome-wide expression data processing and modeling*, Proceedings of the National Academy of Sciences, 97 (2000), pp. 10101–10106.

[4] H. AZAMI, J. ESCUDERO, A. DARZI, AND S. SANEI, *Extracellular spike detection from multiple electrode array using novel intelligent filter and ensemble fuzzy decision making*, Journal of Neuroscience Methods, 239 (2015), pp. 129 – 138.

[5] J. BASAK, R. K. DE, AND S. K. PAL, *Unsupervised feature selection using a neuro-fuzzy approach*, Pattern Recognition Letters, 19 (1998), pp. 997 – 1006.

[6] R. BELLMAN AND R. BELLMAN, *Adaptive Control Processes: A Guided Tour*, Princeton Legacy Library, Princeton University Press, 1961.

[7] R. BESTEL, A. W. DAUS, AND C. THIELEMANN, *A novel automated spike sorting algorithm with adaptable feature extraction*, Journal of Neuroscience Methods, 211 (2012), pp. 168 – 178.

[8] C. M. BISHOP, *Pattern Recognition and Machine Learning*, Springer-Verlag, New York, NY, 2006.

[9] C. BOUTSIDIS, M. MAHONEY, AND P. DRINEAS, *Unsupervised Feature Selection for the k-means Clustering Problem*, in Advances in Neural Information Processing Systems 22, Y. Bengio, D. Schuurmans, J. Lafferty, C. K. I. Williams, and A. Culotta, eds., 2009, pp. 153–161.

[10] G. BRASSARD AND P. BRATLEY, *Fundamentals of Algorithmics*, Prentice-Hall, Inc., Upper Saddle River, NJ, USA, 1996.

[11] R. J. BRYCHTA, S. TUNTRAKOOL, M. APPALSAMY, N. R. KELLER, D. ROBERTSON, R. G. SHIAVI, AND A. DIEDRICH, *Wavelet methods for spike detection in mouse renal sympathetic nerve activity*, IEEE Transactions on Biomedical Engineering, 54 (2007), pp. 82–93.

[12] J. BUCK, *Jamis buck blog*.
http://www.jamisbuck.org/presentations/rubyconf2011/, Online on 2011, consulted on June 5th 2016.

[13] R. CARUANA AND D. FREITAG, *Greedy attribute selection*, in In Proceedings of the Eleventh International Conference on Machine Learning, Morgan Kaufmann, 1994, pp. 28–36.

[14] 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, 8 (2011), p. 016006.

[15] R. B. D'AGOSTINO AND M. A. STEPHENS, eds., *Goodness-of-fit Techniques*, Marcel Dekker, Inc., New York, NY, USA, 1986.

[16] S. DAS, *Filters, wrappers and a boosting-based hybrid for feature selection*, in Proceedings of the Eighteenth International Conference on Machine Learning, ICML '01, San Francisco, CA, USA, 2001, Morgan Kaufmann Publishers Inc., pp. 74–81.

[17] M. DASH, K. CHOI, P. SCHEUERMANN, AND H. LIU, *Feature selection for clustering - a filter solution*, in Data Mining, 2002. ICDM 2003. Proceedings. 2002 IEEE International Conference on, 2002, pp. 115–122.

[18] M. DASH AND H. LIU, *Feature selection for classification*, Intelligent Data Analysis, 1 (1997), pp. 131 – 156.

[19] M. DASH AND H. LIU, *Feature selection for clustering*, in Pacific-Asia Conference on knowledge discovery and data mining, 2000, pp. 110–121.

[20] M. DASH, H. LIU, AND J. YAO, *Dimensionality reduction for unsupervised data*, in Tools with Artificial Intelligence, 1997. Proceedings., Ninth IEEE International Conference on, Nov 1997, pp. 532–539.

[21] M. DEVANEY AND A. RAM, *Efficient feature selection in conceptual clustering*, in Proc. 14th International Conference on Machine Learning, Morgan Kaufmann, 1997, pp. 92–97.

[22] P. A. DEVIJVER AND J. KITTLER, *Pattern Recognition: A Statistical Approach*, Prentice Hall, Englewood,Cliffs, 1982.

[23] 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, ICML '07, New York, NY, USA, 2007, ACM, pp. 521–528.

[24] J. DOAK, *An Evaluation of Feature Selection Methods and Their Application to Computer Security*, tech. report, UC Davis Department of Computer Science, 1992.

[25] J. G. DY AND C. E. BRODLEY, *Feature subset selection and order identification for unsupervised learning*, in Proc. 17th International Conf. on Machine Learning, Morgan Kaufmann, 2000, pp. 247–254.

[26] J. G. DY AND C. E. BRODLEY, *Visualization and interactive feature selection for unsupervised data*, in In Proceedings of the International Conference on Knowledge Discovery and Data Mining (KDD), 2000, pp. 360–364.

[27] A. FERREIRA AND M. A. T. FIGUEIREDO, *Unsupervised feature selection for sparse data*, in European Symp. on Artificial Neural Networks - ESANN, vol. 1, April 2011, pp. 339–344.

[28] K. FUKUNAGA, *Statistical Pattern Recognition (second edition)*, Academic Press, San Diego, CA, 1990.

[29] Y. GHANBARI, L. SPENCE, AND P. PAPAMICHALIS, *A graph-laplacian-based feature extraction algorithm for neural spike sorting*, in Engineering in Medicine and Biology Society, 2009. EMBC 2009. Annual International Conference of the IEEE, Sept 2009, pp. 3142–3145.

[30] S. GIBSON, J. W. JUDY, AND D. MARKOVIƒÁ, *Spike sorting: The first step in decoding the brain: The first step in decoding the brain*, IEEE Signal Processing Magazine, 29 (2012), pp. 124–143.

[31] E. M. GLASER AND W. B. MARKS, *Online separation of interleaved neuronal pulse sequences*, Data Acquisition Process Bio Med, 5 (1968), pp. 137–156.

[32] P. R. GRAY, *Conditional probability analyses of the spike activity of single neurons*, Biophysical Journal, 7 (1967), pp. 759–777.

[33] I. GUYON AND A. ELISSEEFF, *An introduction to variable and feature selection*, J. Mach. Learn. Res., 3 (2003), pp. 1157–1182.

[34] I. GUYON, S. GUNN, M. NIKRAVESH, AND L. A. ZADEH, *Feature Extraction, Foundations and Applications*, Springer, 2006.

[35] M. A. HALL, *Correlation-based feature selection for discrete and numeric class machine learning*, in Proceedings of the Seventeenth International Conference on Machine Learning, ICML '00, San Francisco, CA, USA, 2000, Morgan Kaufmann Publishers Inc., pp. 359–366.

[36] K. D. HARRIS, D. A. HENZE, J. CSICSVARI, H. HIRASE, AND G. BUZSÁKI, *Accuracy of tetrode spike separation as determined by simultaneous intracellular and extracellular measurements*, Journal of Neurophysiology, 84 (2000), pp. 401–414.

[37] K. D. HARRIS, H. HIRASE, X. LEINEKUGEL, D. A. HENZE, AND G. BUZSAKI, *Temporal interaction between single spikes and complex spike bursts in hippocampal pyramidal cells*, Neuron, 32 (2001), pp. 141–149.

[38] D. N. HILL, S. B. MEHTA, AND D. KLEINFELD, *Quality metrics to accompany spike sorting of extracellular signals*, J. Neurosci, (2011), pp. 8699–8705.

[39] A. L. HODGKIN AND A. F. HUXLEY, *A quantitative description of membrane current and its application to conduction and excitation in nerve*, Bulletin of Mathematical Biology, 52 (1990), pp. 25–71.

[40] E. HULATA, R. SEGEV, AND E. BEN-JACOB, *A method for spike sorting and detection based on wavelet packets and shannon's mutual information*, Journal of Neuroscience Methods, 117 (2002), pp. 1 – 12.

[41] E. HULATA, R. SEGEV, Y. SHAPIRA, M. BENVENISTE, AND E. BEN-JACOB, *Detection and sorting of neural spikes using wavelet packets*, Phys. Rev. Lett., 85 (2000), pp. 4637–4640.

[42] G. H. JOHN, R. KOHAVI, AND K. PFLEGER, *Irrelevant features and the subset selection problem*, in MACHINE LEARNING: PROCEEDINGS OF THE ELEVENTH INTERNATIONAL, Morgan Kaufmann, 1994, pp. 121–129.

[43] M. JOSHUA, S. ELIAS, O. LEVINE, AND H. BERGMAN, *Quantifying the isolation quality of extracellularly recorded action potentials*, Journal of Neuroscience Methods, 163 (2007), pp. 267 – 282.

[44] 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, 47 (2000), pp. 1406–1411.

[45] ——, *A wavelet-based method for action potential detection from extracellular neural signal recording with low signal-to-noise ratio*, IEEE Trans. Biomed. Engineering, 50 (2003), pp. 999–1011.

[46] Y. KIM, W. N. STREET, AND F. MENCZER, *Feature selection for unsupervised learning via evolutionary search*, in In Proceedings of the Sixth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, ACM Press, 2000, pp. 365–369.

[47] Y. KIM, W. N. STREET, AND F. MENCZER, *Evolutionary model selection in unsupervised learning*, Intell. Data Anal., 6 (2002), pp. 531–556.

[48] K. KIRA AND L. A. RENDELL, *The feature selection problem: Traditional methods and a new algorithm*, in Proceedings of the Tenth National Conference on Artificial Intelligence, AAAI'92, AAAI Press, 1992, pp. 129–134.

[49] R. KOHAVI AND G. H. JOHN, *Wrappers for feature subset selection*, Artificial Intelligence, 97 (1997), pp. 273 – 324.
Relevance.

[50] G. KREIMAN, C. KOCH, AND I. FRIED, *Category-specific visual responses of single neurons in the human medial temporal lobe*, Nature Neurosciences, 3 (2000), pp. 946–953.

[51] V. KUMAR AND S. MINZ, *Feature selection: A literature review*, Smart CR, 4 (2014), pp. 211–229.

[52] P. LANGLEY, *Selection of relevant features in machine learning*, in In Proceedings of the AAAI Fall symposium on relevance, AAAI Press, 1994, pp. 140–144.

[53] M. S. LEWICKI, *A review of methods for spike sorting: the detection and classification of neural action potentials*, Network: Computation in Neural Systems, 9 (1998), pp. 53–78.

[54] H. LIU AND H. MOTODA, *Feature Selection for Knowledge Discovery and Data Mining*, Kluwer Academic Publishers, Norwell, MA, USA, 1998.

[55] H. LIU, H. MOTODA, R. SETIONO, AND Z. ZHAO, *Feature selection: An ever evolving frontier in data mining*, Journal of Machine Learning Research(JMLR) - Workshop and Conference Proceedings, 10 (2010), pp. 4–13.

[56] H. LIU AND R. SETIONO, *A Probabilistic Approach to Feature Selection - A Filter Solution*, in International Conference on Machine Learning, 1996, pp. 319–327.

[57] H. LIU, J. SUN, L. LIU, AND H. ZHANG, *Feature selection with dynamic mutual information*, Pattern Recognition, 42 (2009), pp. 1330 – 1339.

[58] H. LIU AND L. YU, *Toward integrating feature selection algorithms for classification and clustering*, IEEE TRANSACTIONS ON KNOWLEDGE AND DATA ENGINEERING, 17 (2005), pp. 491–502.

[59] L. LIU, J. KANG, J. YU, AND Z. WANG, *A comparative study on unsupervised feature selection methods for text clustering*, in 2005 International Conference on Natural Language Processing and Knowledge Engineering, Oct 2005, pp. 597–601.

[60] S. LLOYD, *Least squares quantization in pcm*, IEEE Trans. Inf. Theor., 28 (2006), pp. 129–137.

[61] J. MACQUEEN, *Some methods for classification and analysis of multivariate observations*, in In 5-th Berkeley Symposium on Mathematical Statistics and Probability, 1967, pp. 281–297.

[62] R. MILLECCHIA AND T. MCINTYRE, *Automatic nerve impulse identification and separation*, Computers and Biomedical Research, 11 (1978), pp. 459 – 468.

[63] A. J. MILLER, *Subset Selection in Regression*, Chapman and Hall/CRC, London, 1990.

[64] P. MITRA, S. MEMBER, C. A. MURTHY, AND S. K. PAL, *Unsupervised feature selection using feature similarity*, IEEE Transactions on Pattern Analysis and Machine Intelligence, 24 (2002), pp. 301–312.

[65] S. MUKHOPADHYAY AND G. C. RAY, *A new interpretation of nonlinear energy operator and its efficacy in spike detection*, IEEE Transactions on Biomedical Engineering, 45 (1998), pp. 180–187.

[66] P. M. NARENDRA AND K. FUKUNAGA, *A branch and bound algorithm for feature subset selection*, IEEE Trans. Comput., 26 (1977), pp. 917–922.

[67] Z. NENADIC AND J. W. BURDICK, *Spike detection using the continuous wavelet transform*, IEEE TRANSACTIONS ON BIOMEDICAL ENGINEERING, 52 (2005), pp. 74–87.

[68] A. Y. NG, *On feature selection: Learning with exponentially many irrelevant features as training examples*, in Proceedings of the Fifteenth International Conference on Machine Learning, ICML '98, San Francisco, CA, USA, 1998, Morgan Kaufmann Publishers Inc., pp. 404–412.

[69] J. O'KEEFE AND M. L. RECCE, *Phase relationship between hippocampal place units and the eeg theta rhythm*, Hippocampus, 3 (1993), pp. 317–330.

[70] R. OSTROVSKY, Y. RABANI, L. J. SCHULMAN, AND C. SWAMY, *The effectiveness of lloyd-type methods for the k-means problem*, J. ACM, 59 (2013), pp. 28:1–28:22.

[71] C. POUZAT, O. MAZOR, AND G. LAURENT, *Using noise signature to optimize spike-sorting and to assess neuronal classification quality*, Journal of Neuroscience Methods, 122 (2002), pp. 43 – 57.

[72] N. PRADHANANGA, *Effective Linear-Time Feature Selection*, Department of Computer Science, University of Waikato, Hamilton, New Zealand, 2007.

[73] D. PURBES, G. J. AUGUSTINE, D. FITZPATRICK, W. C. HALL, A.-S. LAMANTIA, J. O. MCNAMARA, AND S. M. WILLIAMS, *Neuroscience - Third Edition*, Sinauer Associates, Inc, Sunderland, Massachusetts USA, 2004.

[74] R. Q. QUIROGA, *Spike sorting*, 2 (2007), p. 3583.

[75] R. Q. QUIROGA, *What is the real shape of extracellular spikes?*, Journal of Neuroscience Methods, 177 (2009), pp. 194 – 198.

[76] R. Q. QUIROGA, Z. NADASDY, AND Y. BEN-SHAUL, *Unsupervised spike detection and sorting with wavelets and superparamagnetic clustering*, Neural Comput., 16 (2004), pp. 1661–1687.

[77] R. Q. QUIROGA, L. REDDY, C. KOCH, AND I. FRIED, *Decoding visual inputs from multiple neurons in the human temporal lobe*, Journal of Neurophysiology, 98 (2007), pp. 1997–2007.

[78] W. RAND, *Objective criteria for the evaluation of clustering methods*, Journal of the American Statistical Association, 66 (1971), pp. 846–850.

[79] L. RENDELL AND R. SESHU, *Learning hard concepts through constructive induction: framework and rationale*, Computational Intelligence, 6 (1990), pp. 247–270.

[80] H. G. REY, C. PEDREIRA, AND R. Q. QUIROGA, *Past, present and future of spike sorting techniques*, Brain Research Bulletin, 119, Part B (2015), pp. 106 – 117. Advances in electrophysiological data analysis.

[81] U. RUTISHAUSER, E. M. SCHUMAN, AND A. N. MAMELAK, *Online detection and sorting of extracellularly recorded action potentials in human medial temporal lobe recordings, in vivo*, Journal of Neuroscience Methods, 154 (2006), pp. 204 – 224.

[82] T. SATO, T. SUZUKI, AND K. MABUCHI, *Fast automatic template matching for spike sorting based on davies-bouldin validation indices*, in 2007 29th Annual International Conference of the IEEE Engineering in Medicine and Biology Society, Aug 2007, pp. 3200–3203.

[83] N. SCHMITZER-TORBERT, J. JACKSON, D. HENZE, K. HARRIS, AND A. REDISH, *Quantitative measures of cluster quality for use in extracellular recordings*, Neuroscience, 131 (2005), pp. 1 – 11.

[84] N. SCHMITZER-TORBERT AND A. D. REDISH, *Neuronal activity in the rodent dorsal striatum in sequential navigation: Separation of spatial and reward responses on the multiple t task*, Journal of Neurophysiology, 91 (2004), pp. 2259–2272.

[85] D. M. SCHWARZ, M. S. ZILANY, M. SKEVINGTON, N. J. HUANG, B. C. FLYNN, AND L. H. CARNEY, *Semi-supervised spike sorting using pattern matching and a scaled mahalanobis distance metric*, Journal of Neuroscience Methods, 206 (2012), pp. 120 – 131.

[86] S. TAKAHASHI, Y. ANZAI, AND Y. SAKURAI, *A new approach to spike sorting for multi-neuronal activities recorded with a tetrode,Äîhow ica can be practical*, Neuroscience Research, 46 (2003), pp. 265 – 272.

[87] T. TAKEKAWA, Y. ISOMURA, AND T. FUKAI, *Accurate spike sorting for multi-unit recordings*, European Journal of Neuroscience, 31 (2010), pp. 263–272.

[88] Z. TIGANJ AND M. MBOUP, *Spike detection and sorting: combining algebraic differentiations with ICA*, in ICA 2009, Paraty, Brazil, Mar. 2009.

[89] D. TZEMACH, *David tzemach blog*.
http://www.dtvisiontech.com/2015/12/types-of-performance-tests-david-tzemach.html, Online on December 24th dec 2015, consulted on June 5th 2016.

[90] R. VARSHAVSKY, A. GOTTLIEB, M. LINIAL, AND D. HORN, *Novel unsupervised feature filtering of biological data*, BIOINFORMATICS, 19 (2006), pp. e507–e513.

[91] S. VISALAKSHI AND V. RADHA, *A literature review of feature selection techniques and applications: Review of feature selection in data mining*, in Computational Intelligence and Computing Research (ICCIC), 2014 IEEE International Conference on, Dec 2014, pp. 1–6.

[92] B. C. WHEELER AND W. J. HEETDERKS, *A comparison of techniques for classification of multiple neural signals*, IEEE Transactions on Biomedical Engineering, BME-29 (1982), pp. 752–759.

[93] J. WILD, Z. PREKOPCSAK, T. SIEGER, D. NOVAK, AND R. JECH, *Performance comparison of extracellular spike sorting algorithms for single-channel recordings*, Journal of Neuroscience Methods, 203 (2012), pp. 369 – 376.

[94] E. WOOD, M. FELLOWS, J. R. DONOGHUE, AND M. J. BLACK, *Automatic spike sorting for neural decoding*, in Engineering in Medicine and Biology Society, 2004. IEMBS '04. 26th Annual International Conference of the IEEE, vol. 2, Sept 2004, pp. 4009–4012.

[95] F. WOOD, S. GOLDWATER, AND M. J. BLACK, *A non-parametric bayesian approach to spike sorting*, in Engineering in Medicine and Biology Society, 2006. EMBS '06. 28th Annual International Conference of the IEEE, Aug 2006, pp. 1165–1168.

[96] E. P. XING, M. I. JORDAN, AND R. M. KARP, *Feature selection for high-dimensional genomic microarray data*, in Proceedings of the Eighteenth International Conference on Machine Learning, ICML '01, San Francisco, CA, USA, 2001, Morgan Kaufmann Publishers Inc., pp. 601–608.

[97] L. YU AND H. LIU, *Feature selection for high-dimensional data: a fast correlation-based filter solution*, in In: Proceedings of the 20th International Conferences on Machine Learning, 2003.

Rue Archimède, 1 bte L6.11.01, 1348 Louvain-la-Neuve **www.uclouvain.be/epl**