Data Analysis and Model Classification

Mini-project 1

Authors:
Julie Laurent
Alice Leydier
Yara-Maria Proust

Group ID : 2

October 13, 2017

ÉCOLE POLYTECHNIQUE
FÉDÉRALE DE LAUSANNE

1 Introduction

Nowadays, the question of healing neurological diseases has become increasingly important, as more and more people are suffering from them. Therefore, neurosciences try to find answers about how the brain works, in order to provide possible solutions to treat it if needed. In order to do that, neural activity from a population of neural cells is usually assessed by the mean of many electrodes recording signals from several unknown neurons. Once it is recorded, a decoding has to be made so that information obtained can be studied and understood.
For this data analysis, different processing steps are needed. First, a spike detection is done, allowing detection of zones with neural activity. This step is followed by an epoching to isolate an extract of the full spike time course. A feature extraction is then applied in order to reduce the number of features from one hundred to three. To assign each spike to one type of neuron, a clustering method is applied. This enable to distinguish how many neurons are detected. Finally, a firing rate computation is done.
In this particular project, electrical activity was recorded from an electrode implanted in a non-human primate motor cortex. The file used for this research thus contained a large number of different detected spikes extracted (6000) at 100 time points. The dataset "spikes.mat" contained extracted after epoching spikes in the variable "spikes" as well as their projections after principal component analysis (PCA) in the variable "spikesPCA" and the "labels" corresponding at each activity. The use of PCA enables to convert a set of possibly correlated variables into a set linearly uncorrelated variables that are called principal components, explaining the most the difference between the samples. Thus the file "spikesPCA" contained only 3 features, while "spikes" contained 100. As the spike detection and the epoching were already done, the aim of this study was to compute and understand the clustering step. Since the data recorded are not "labeled", the type of clustering that was used here to identify individual neurons is called Unsupervised Machine Learning. Here, this was done notably by performing a k-means clustering that aims to separate n samples into k clusters in which each variable belongs to the cluster with the nearest center.
Finally, before computing the k-means clustering, an exploration of the dataset was performed in this project.

2 Methods

Data exploration The aim of this first step was to understand in depth what kind of data the project was based on. A file "spikes.mat" including the variables of "spikes", "spikesPCA" and "labels" was provided. "spikes" was representing the extracted after epoching spikes, while "spikesPCA" contained their projections after PCA.
The preliminary step consisted in identifying the size of the feature vector and how many samples it contained. Once this was determined, a 2D plot of all spikes (before PCA) in function of time was used to observe their general shapes and determine how many different neurons there could be.
The second step for data exploration was to analyze the features of the spikes after PCA. Only three principal components representative of each spikes were kept, leading to a considerable lighter dataset and thus, easier computation and analysis. First, PCA-components were plotted one in function of another to identify their influence on each other. Then, each feature distribution was assessed with a histogram to get information on the type of distribution and the feature separability, and with a boxplot to obtain the mean values and visualize the outliers.
Based on the observations, an assumption for the probable number of clusters was made. Since one neuron should produce a typical Action Potential shape, and thus a typical cluster, it could allow a first observation of the number of neurons recorded, which should be further confirmed.

k-means Clustering After exploring the dataset, spikes could be clustered, consider-

ing the fact that they were triggered by different neurons. Based on the similarity between some features, the k-means clustering algorithm could be computed and performed. It allowed to assign the data points (spikes) to a group (neurons). In the k-means method, a cluster is defined to be a set of points whose inter-point distances are small compared to points outside the ensemble. In other words, the within-cluster sum of square distances is minimized:

$$argmin \sum_{i=1}^{k} \sum_{s \in C_i} ||s - \mu_i||^2$$

with μ_i = mean of C_i. This algorithm functions as following : initialization determines the starting centroids μ_i of the clusters C_i followed by the assignment of samples s to the closest centroid and finally the centroids are updated by the mean of the assigned samples. The algorithm is repeated until there is no more change in the sample assignment anymore, meaning that there is no more change in the computed centroids. It is important to notice that the k-means clustering creates clusters having a similar size and each dataset can only belong to one cluster.
To justify the chosen number of neurons, the average spikes profiles of the different clusters were plotted. By repeating nine times the clustering with k-means, the variability of the clustering due to different initial conditions could be assessed. The algorithm process was repeated with different numbers of clusters and the within-cluster sum of squared errors was computed and plotted to evaluate the performance in function of the number of clusters. Finally, the MATLAB function "evalclusters" allowed to evaluate clustering algorithms and find the optimal number of clusters based on a chosen internal criterion. Here, several criterion were tested.

3 Results

Data exploration The feature vector contained 6000 samples (spikes), with 100 features (time). After PCA, it was reduced to 3 features explaining most of the variance between samples.
By looking at the plot representing the spikes (see Fig. 1), three types of spikes shapes (and thus neurons) could be assumed : some having a big amplitude, others small amplitudes and a group having a bump after the action potential. However, as this spikes representation does not allow a clear view of the spikes shapes, there is no certainty about having only three neurons.

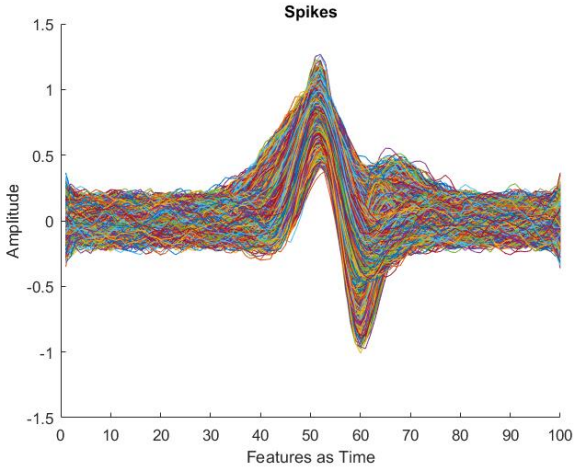

Figure 1: Shape of all the spikes.

To have a more precise idea of the number of neurons, results of PCA extraction were used. The plot of the first feature in function of the second one (see Fig. 2) was the most valuable as three different clusters could be observed, which seems to support the assumption made previously.

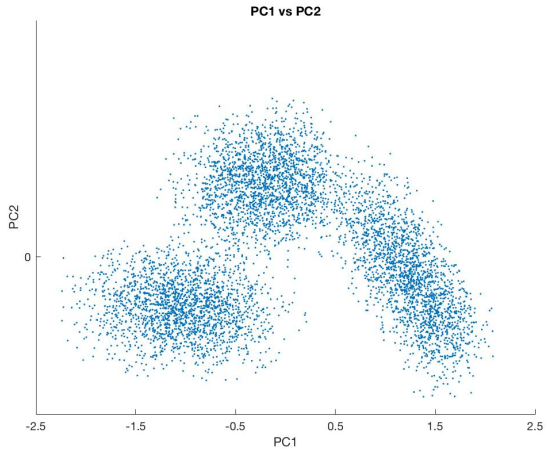

Figure 2: Plot of the first principal component (PC1) versus the second one (PC2). Each point represents one spike.

Using the histograms to evaluate the principal components' (PC) distributions (see Fig. 3, top row graphs), one could see that PC1 and PC2 seemed to have 2 sub-groups as a composition of two Gaussians was observed, thus signifying that those features could maybe regroup two smaller features. However, this phenomena seemed to be less distinct in the PC2 compared to PC1. On the contrary, the PC3 component had a Gaussian-looking distribution, thus representing only one feature. This last observation could maybe be related to the fact that PC3 component did not appear to be as useful as the other principal components. The boxplots of these principal components (see Fig. 3, bottom row graphs) allowed to see that PC3 had a lot of outliers values and that the median of each distribution was around 0.

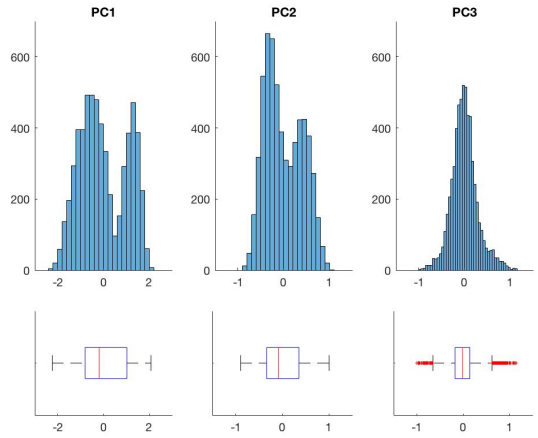

Figure 3: Histograms and boxplots of the principal components.

k-means clustering Once the number of three distinguishable clusters (thus neurons) was kept, each spike had to be attributed to its corresponding neuron. To do so, k-means clustering was used, as it can model a data distribution without desired target values. Indeed, it was useful in this project since there was no label from previous data showing to which neuron a particular spike was belonging to. Thus this technique uses what is called unsupervised machine learning, and permitted the clear view of which spike belonged to which cluster (see Fig. 4). In this Figure, we can see that the colors correspond well with the three hypothetic clusters, supporting the chosen number of clusters.

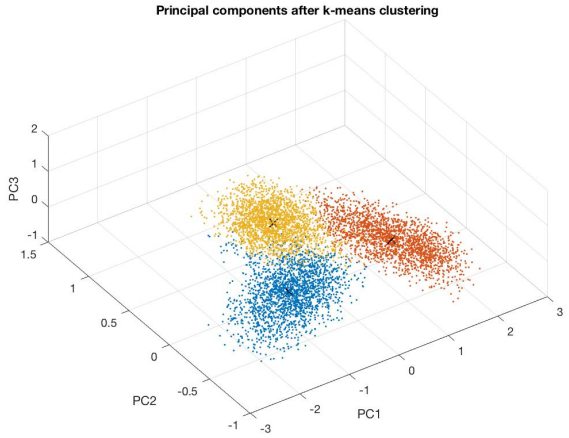

Figure 4: Plot of the principal components after the use of k-means clustering, assuming 3 clusters. The cross in the middle of each cluster represents its center. Each color represents the spikes/sample associated with a particular neuron/cluster.

The hypothesis of three clusters as the optimal number was supported by the fact that the shape of the spikes mean were really different from one cluster to another, as seen on Figure 5. However, more neurons could be involved, but as there were three very different shapes, one could say that three neurons were at least involved.

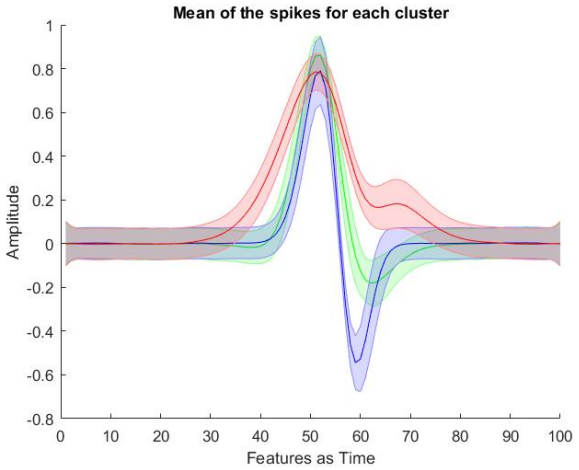

Figure 5: Plot of the mean spike for each cluster. The standard deviation is represented by the transparent contours.

To evaluate the performance of this model, the impact of the initial conditions on the clus-

tering was assessed and an optimal number of clusters was searched.
To study the impact of initial conditions (as this algorithm depends on them), k-means clustering was executed nine times with random initial conditions, using three and fours clusters, in order to see if it changed the repartition and the center of the clusters (see Fig. 6 and Fig. 7). The number of two clusters was not tested as the obtained data showed that at least three neurons were present.

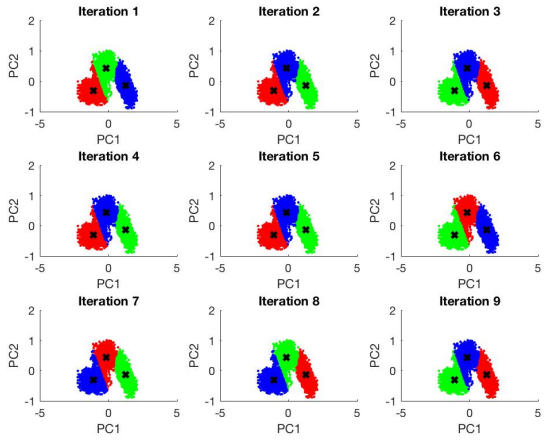

Figure 6: Plot of nine times the k-means clustering with random initial conditions, using three clusters by default: the model seems to be robust to different initial conditions

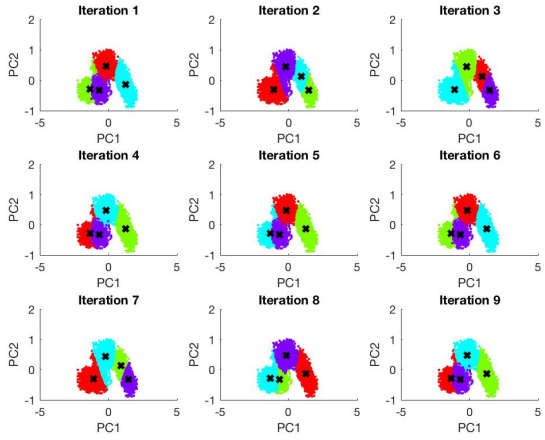

Figure 7: Plot of nine times the k-means clustering with random initial conditions, using four clusters by default: clusters are very different from one another with different initial conditions

From these two graphs, one could observe that when the number of clusters was optimal, the choice of the initial conditions had no or very little impact on the clustering. This was indeed the case with three clusters. In the case with four clusters, the k-means clustering was not robust: different initial conditions had a large impact on the clusters and their centers, meaning that four was not an optimal number of clusters.
The optimal number of clusters based on an internal criterion was then assessed. This means that the analysis was based on the same data that were used to cluster. The sum of the squared distances to centroids in function of the number of clusters was plotted (see Fig. 8). This error decreased as the number of clusters increased, meaning that a better performance with this criteria was obtained as the number of clusters increased.

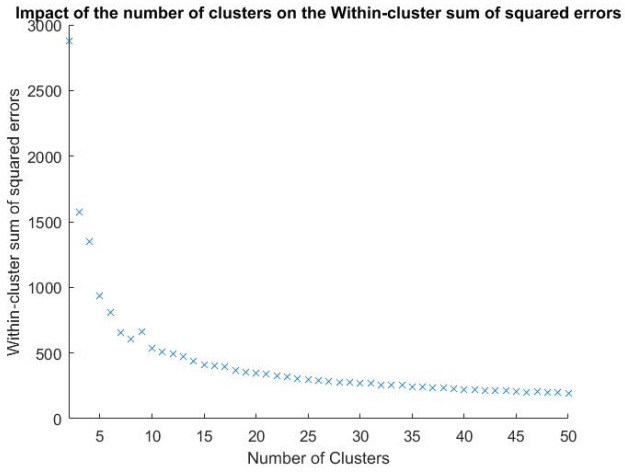

Figure 8: Sum of the squared distances to centroids in function of the number of clusters.

However, it appeared not to be a reliable system to evaluate the performance. Indeed, as the number of clusters was increasing, these clusters started to become smaller, and thus, the distance between the points and the clusters centers was decreasing, which resulted in the distortion of the outcome until one cluster corresponded to one sample. Furthermore, according to this criteria, the higher the number of clusters, the better the performance, but also, the higher the complexity. A weight in the increase of performance with the increase

of complexity had to be made as a way to regulate was needed.
For that purpose, the Calinski-Harabasz criterion, which evaluates the optimal number of clusters, was selected. The resulting plot (see Fig. 9) showed the highest Calinski-Harabasz value occurring for three clusters. Thus, it suggested the optimal number of clusters to be three. This criterion was chosen amongst others as it uses the Euclidian distance, as k-means, but also because it showed an optimal number of clusters of three, as proposed before. Therefore, it supported that the previous hypothesis was coherent.

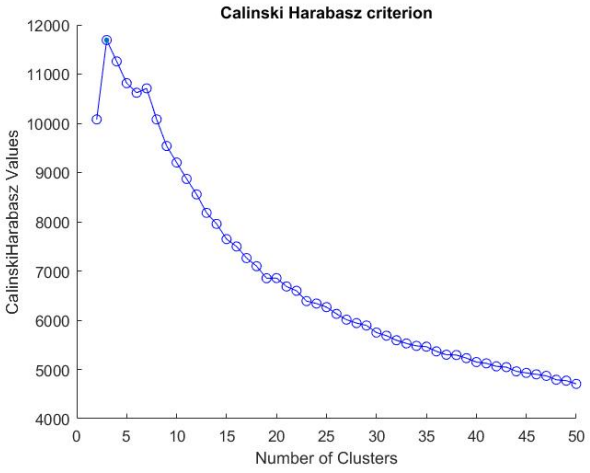

Figure 9: Evaluation of the optimal number of clusters with the Calinski-Harabasz criterion.

4 Discussion

At first, the only information for the project were neural activity recordings of several neurons. However, the number of neurons recorded as well as which neural activity was corresponding to which neuron was unknown. As the number of time points at which spikes were recorded was quite high (100), a PCA was needed in order to reduce the complexity of the model. To conclude, this project allowed a better understanding of unsupervised learning machine. Indeed, the exploration of the dataset as well as their projections after PCA permitted the elaboration of the k-means algorithm which allowed in the end to label the data, that was at first "unlabeled". The number of three neurons was determined and each spike (representing thus the neural activity) was affected to one cluster. However, the k-means algorithm presents some limitations. As a matter of fact, boundaries between clusters are not necessarily well-defined as seen in Figure 10. This could thus constitutes a limitation to the clear understanding of the data, and some information could be lost. As seen before, the algorithm depends on the initial conditions (if the number of clusters is not optimal), meaning the result of our clustering might not be optimal. However, with three clusters, the initial conditions had no influence on the clustering and this number was supported by the Calinski Harabasz criterion, suggests the k-means algorithm allows an acceptable understanding of the data.

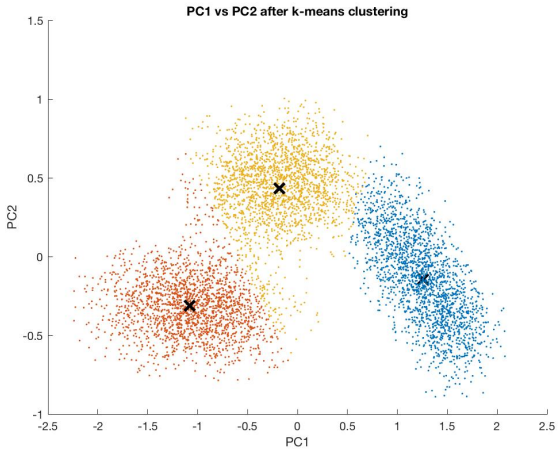

Figure 10: Plot of PC1 vs PC2 after k-means clustering: red and yellow clusters overlap a little.