Lussac Spikesorting Optimization

Andrea Combette

ENS ULM

February 7, 2024

Introduction

- Neurons are the basic unit of the brain and the nervous system.
- They are responsible for receiving sensory input from the external world and sending information
- Communication between neurons is achieved through the synapse, through complex electrochemical processes.

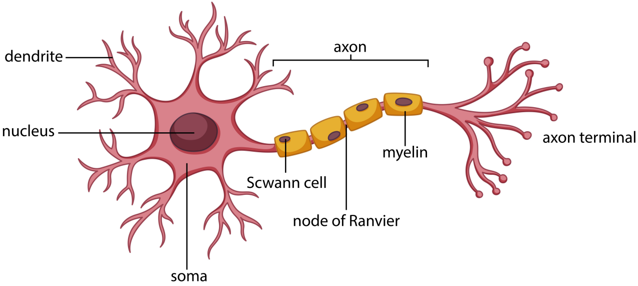


Figure: Neuron anatomy with different componentso

Introduction

- State of the neuron is defined by the potential of its membrane.
- The action potential (the electrical spike released by the neuron) is the result of the depolarization of the membrane.
- The action potential is characteristic of the neuron and is used to communicate with other neurons.
- The goal of a spikesorter is to identify the action potential of each neuron in the recorded data (record with electrodes).

Introduction

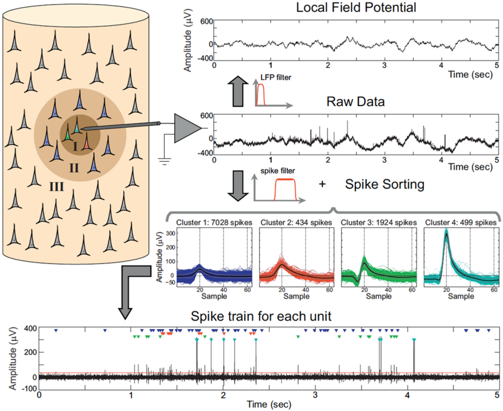


Figure: Spike sorting analysis Principles of neurons. The local field potential contains multiple neuron signals, the goal of the spilesorting algorithm is to isolate each signal and to attribute each spikes in the local field to a specific neuron. There is many methods to do so. Lussac goal is to deal with multiple of these algorithms to extract the best from each.

Introduction

- The spikesorting is a complex problem, and there is no unique solution.
- The goal of the Lussac project is to develop a new spikesorting algorithm that will be able to deal with multiple spikesorting algorithms.
- The goal of this project is optimize the Lussac choice between the different spikesorting algorithms.

CLUSTERING GROUP OF NEURONS

- Identify which neurons are identical between each analysis
- process each cluster of the same *real neuron*
- deal with relations between nodes of the Lussac graph
- each neuron of each analysis is a node of the graph
- each edge of the graph stands for relations between two neurons over different metrics

Graph Space

- The graph space is a i dimensional space with i the number of metrics to compare neurons.
- Metrics helps us to determined if a relation between two neurons is good or not
- If the relation is good, neurons are part of the same cluster

Clustering Metrics

- Similarity metrics :

$$\mathrm{sim}(n_i, n_j) = \frac{N_{ij}}{\min(N_i, N_j)}$$

  measures the spiking activity similarity between two neurons.

- Correlogram difference : Compares the correlogram of two neurons.

$$\mathrm{corr}_i = \frac{|\Gamma_i - \Gamma_{ij}|}{w_j - w_i}$$

- Template difference : Compares the waveform of the two neurons. It is just a euclidean distance, between the two curve.

Clustering Metrics

- Asymmetric metrics :

  $$\mathrm{asym}(n_i, n_j) = \frac{N_{ij}}{N_i}$$

  measures the spiking activity similarity between two neurons.
- Cross-contamination : Number of violation of the refractory period (rest time of the neuron), corrected by the censure time of the spike-sorter (cannot detect spike that are too close).

Unsupervised Clustering

- The goal of the unsupervised clustering is to find the best cluster of neurons.
- Unsupervised clustering is a method of clustering that does not require the user to specify clusters for training of the model.
- KMEANS is a popular unsupervised clustering algorithm

Unsupervised Clustering

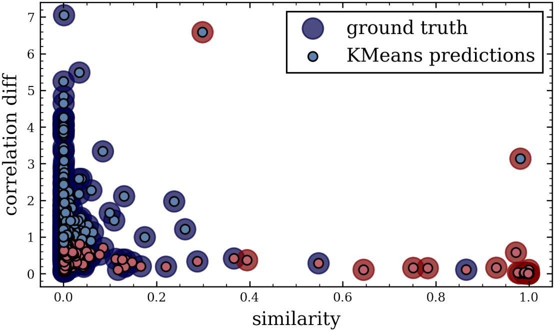
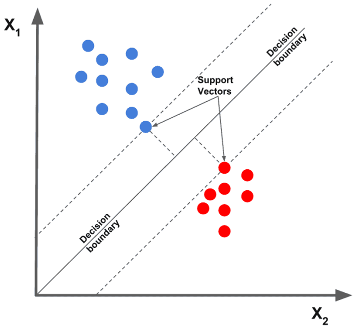


Figure: Big circles are the ground truth label, red one are known good relations between neurons. The results of the unsupervised clustering is represented by the little blue and red circle, the red one are the good relations, and the blue one are the bad ones. The unsupervised clustering is not able to separate the data in a good way.)

Supervised Clustering

- Here the idea is to first use a classification methods to first determine if the edge is good or not.
- Then we can use the result of the classification to determine the cluster of the neurons.
- The classification methods are trained on a known set of good and bad relations.
- Use of SVM classifier (support vector machine)

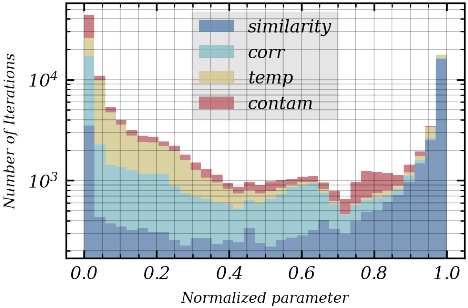
Supervised Clustering



- The SVM classifier find the best hyperplane to separate the data.
- Kernel trick is used to separate the data in a higher dimensional space. (non-linear speration)

$$K(x_i, x_j) = \exp(-\gamma||x_i - x_j||^2)$$

APPLICATION



- features distribution study
- appears that template difference and contamination are quite redundant in their distribution

APPLICATION

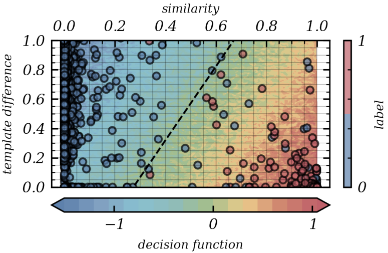


Figure: Decision Function for the given SVM, the correlation dependence is omitted due to its weak impact on the decision boundaries

- able to separate the data in a good way
- score of 0.9997 on the test set : really bad neurons are numerous and easy to separate from the good ones.
- Introduce new metrics to have better insights of the classification

APPLICATION

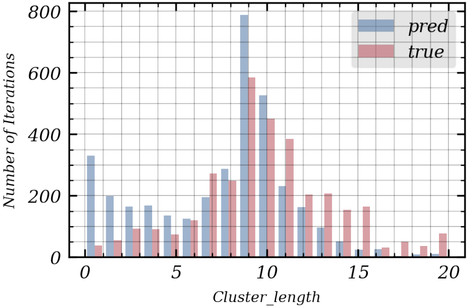


Figure: Here we can see the length histogram of the clusters, the length of the clusters is the number of neurons inside this latter. In blue the predicted cluster length and in red the true one.

- predicted clusters have the same distribution as the true one
- smaller size of the clusters : relevant for eliminate bad units

Clustering Efficiency

- quantify clustering quality
- use the following formula :

$$\text{score}(Y^{\text{pred}}, Y^{\text{true}}) = \sum_{k=0}^{N} \frac{\#(G_k^{\text{pred}} \cap G_k^{\text{true}})}{\max(\#G_k^{\text{pred}}, \#G_k^{\text{true}})}$$

- from this we get a score of 0.94 for the SVM classifier

Clustering on high dimensional space

- expand dimsensions to separate the data in a better way
- Next we will just keep the similarity, template difference, correlogram difference metrics to simplify the problem. The features space is then of dimension 3.
- gives the following representation of the data :

$$\underbrace{\begin{pmatrix} |1,0,0| & \dots & \dots \\ \dots & |1,0,0| & \dots \\ \dots & \dots & |1,0,0| \end{pmatrix}}_{3N}$$

- In this representation we do not consider the link validity but how the neuron are connected together.

Various Unsupervised Algorithm

- Affinity Propagation
- HDBSCAN
- both methods lead to poor score compared to the SVM classifier (respectively 0.687 and 0.781)
- eliminate neurons for mid-size cluster which is not relevant

Nodes Clustering

- the nodes clustering is the next step of the Lussac algorithm
- After creating clusters based on neurons relations
- We want to isolate the best neurons of each cluster, to have the best representation of the real neurons.

metrics Overview

- **rb contamination** : The contamination gives a corrected number of violations metrics, indeed it's calculated using a censure time. Indeed, spike sorters are not able to detect spikes that are too close to each other. A way of correct the rate of number of violation is to not consider a specific time window around each spike. This time window is called the censure time.
- **SNR** : The SNR is the ratio between the mean of the spike amplitude and the standard deviation of the noise. The SNR is a measure of the quality of the spike detection. A high SNR means that the spike detection is good.

metrics Overview

- **presence ratio** : The presence ratio is the ratio between the number of spikes detected and the number of spikes expected. A high presence ratio means that the spike detection is good.
- **firing rate** : The firing rate is the number of spikes detected divided by the duration of the recording. A high firing rate means that the spike detection is good.
- **synchrony** : The synchrony is the ratio between the number of coinciding spikes and the number of spikes detected. A high synchrony means that the spike detection is good.

metrics Overview

- **sd ratio** : The sd ratio is the ratio between the standard deviation of the spike amplitude and the standard deviation of the noise. A high sd ratio means that the spike detection is good.
- **quality score metrics** : The quality metrics is score taking into account the contamination and the number of spikes (firing rate). It is defined as following :

  $$S = N(1 - (k+1)C)$$

  witth N the number of spikes, C the contamination and k a constant. Generally we take $k = 1$ for minimizing the accuracy, but one could argue that it's not the best choice, indeed a false positive has a stronger impact on the spike sorting quality, so we could take $k = 2.5$ to accentuate the dependence of false positive on the score. Indeed, the former formula leads to :

  $$S = N - FN - kFP$$

  .

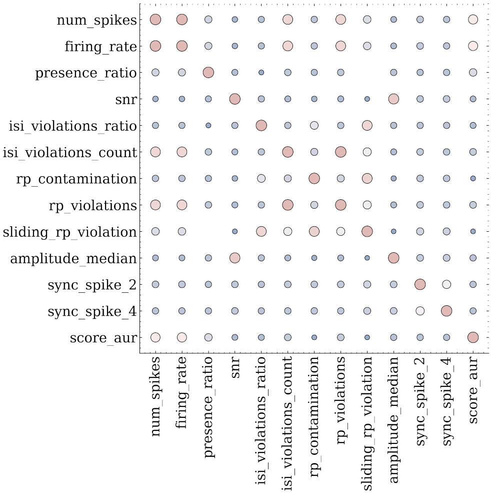


Figure: Correlation matrix of the different metrics

- goal is to keep the most relevant uncorrelated metrics
- sd ratio and SNR are highly correlated
- presence ratio and firing rate are highly correlated

Clustering on every nodes

- Dropping the cluster dependence allows us to use all the nodes for the global clustering
- Some metrics are cluster dependent, so we have to drop them
- We used the following : SNR, contamination, presence ratio

Weighting Function

- To avoid a strong impact of near best neurons on the clustering, we use a weighting function

$$f(x) = \left[\tanh\left(\frac{x - \mathrm{max}_c / W_C}{\sigma}\right) + \frac{1}{2}\right]^2$$

  x is the known accuracy score of the training set

- We consider just the best neurons of each cluster and bad neurons.
- Best neuron and bad neurons are determined by the accuracy score.

Clustering

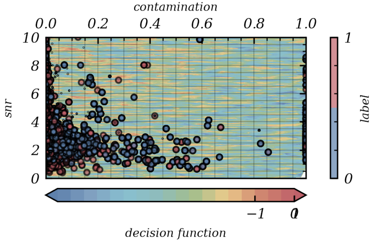


Figure: The data is impossible to separate into 2 clusters, the SVM is overfitting the data

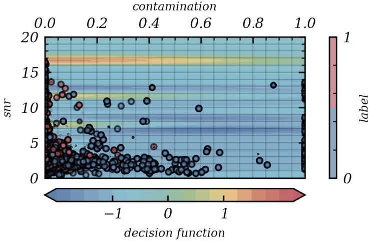


Figure: dropping the presence ratio leads to the same results need to introduce relative cluster metrics

Relative Clusters Metrics

- one way of solving this is to renormalize the parameter by the mean and the std of the cluster

$$\frac{p_i - \mu}{\sigma}$$

This way we can compare the different clusters.

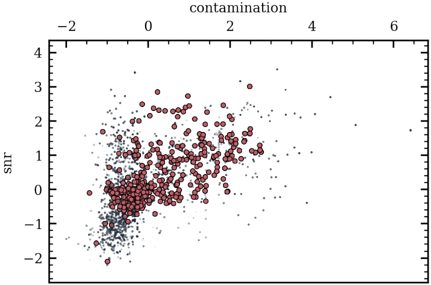
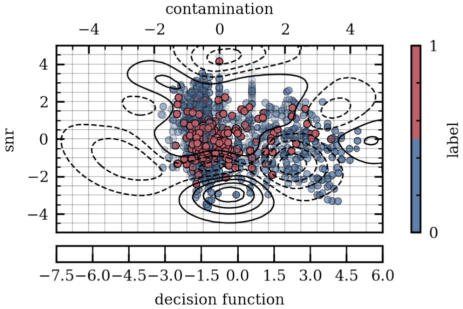


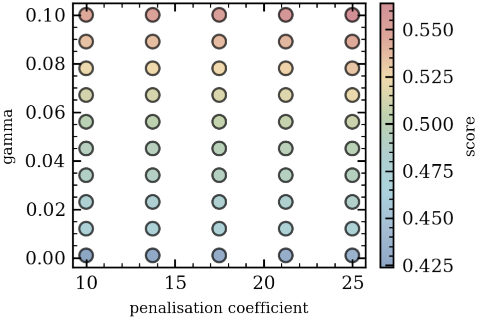
Figure: In red the good neurons and in black the bad ones, size of neurons are proportional to their weights. **Introduce class weights.**

SVM classifier



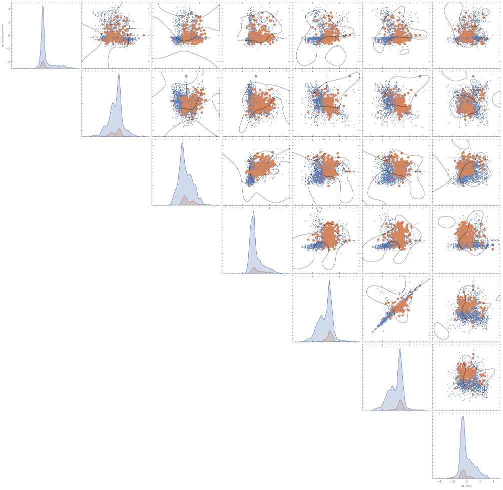
- More homogeneous separation of the data
- However the separation is not perfect and one way of improving the separation is to tweak the kernel parameters of the SVM, and the penalty parameter.

SVM classifier



- Perform a grid search to find the best parameters for the SVM
- this leads to relatively high Γ and C parameters

SVM classifier



SVM classifier

- The separation line is too complex
- fix the problem by choosing small $\Gamma = 0.03$

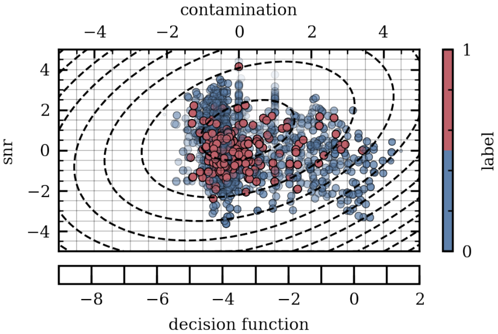
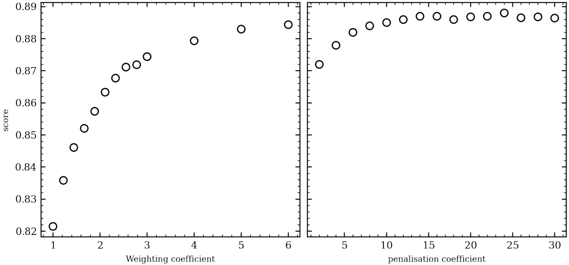


Figure: Here we choose $\Gamma = 0.03$ and $C = 1$ (i.e. no penalty), the separation line is much more homogeneous.

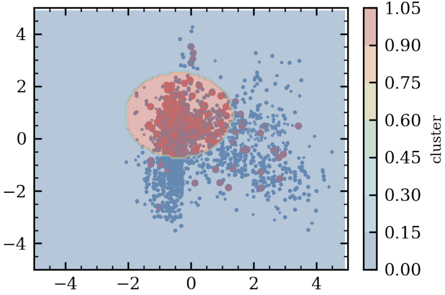
SVM Classifier



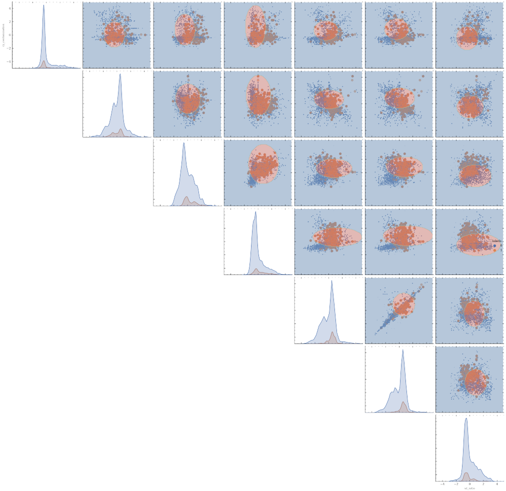
- optimize with $\Gamma = 0.03$
- Must take large penalty and weighting parameter
- With $C = W_C = 5$ we got a final score of 0.84

Naive Bayes classifier

- Naive Bayes classifier is a simple probabilistic classifier based on applying Bayes' theorem with strong (naive) independence assumptions between the features.
- Clustering over **synchrony** and **SNR** gives us the following (With the same weighting process as the SVM) :



Naive Bayes classifier

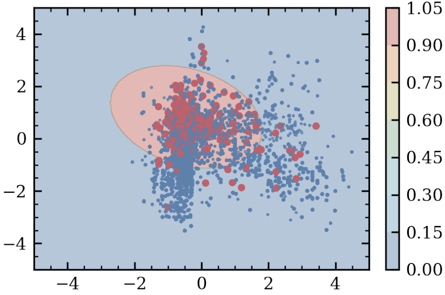


Naive Bayes classifier

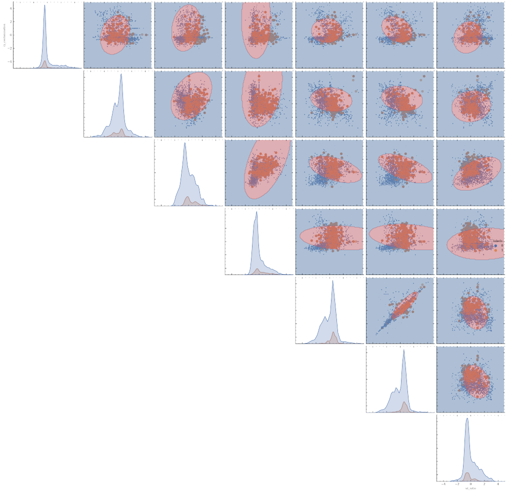
- for some diensions the clustering is not perfect.
- However, clusters centers are always well separated.
- The multidimensional analysis leads to a global score of 0.87, given the weighting metrics.

Quadratic Discriminant Analysis

- Quadratic Discriminant Analysis assumes that the probability density function of the features given the class follows a Gaussian distribution
- Clustering over **synchrony** and **SNR** gives us the following (With the same weighting process as the SVM) :



Quadratic Discriminant Analysis



Quadratic Discriminant Analysis

- for some diensions the clustering is not perfect.
- leads quite to the same results as for the Bayesian approach
- one can show that the boundaries are a little less strict
- The final score is 0.88.

Comparison of the different methods

- The different methods lead quite to the same score
- However boundaries are a little more strict for the QDA
- avoid False negative is more important than avoiding False positive, because we have many points
- quadratic discriminant analysis is the best method for tackling the classification problem

Conclusion

- The Lussac algorithm is a complex problem, and there is no unique solution.
- The goal of the Lussac project is to develop a new spikesorting algorithm that will be able to deal with multiple spikesorting algorithms.
- ptimize the Lussac choice between the different spikesorting algorithms.
- Two clustering steps are necessary : Clustering of good relations to isolate the neurons clusters and clustering of the neurons to isolate the best neurons of each cluster.
- We find the best method in our scope of study