UNIVERSITÀ DI PISA

Dipartimento
di Matematica

Final Task: analysis of EigenWorms dataset

Zito Giuseppe Pio 583233

1 Introduction

The study of unconstrained spontaneous behavior is the core of ethology (a branch of zoology that studies animals behavior), and it has also made significant contribution to behavioral genetics in model organism. Visible phenotypes based on locomotion and posture have played a critical role in understanding the molecular basis of behavior and developments in *caenorhabditis elegans* worms and other model organism. In [1] it is explained a method to automatically record and quantify spontaneous (without bacterial food) behavior of wild-type *caenorhabditis elegans* worms and some its mutant-type. This is very important to improve reproducibility, and to reduce the gap the ability to sequence and manipulate genomes and the ability to asses the effect of genetic variation and mutation on behavior. In particular, it has been shown that we can reduce the high dimensional space of shapes explored by *caenorhabditis elegans* worms during spontaneous behavior to a low dimensional (these dimension are called *EigenWorms*) that is sufficient to capture over 90% of the variance of worm shapes.

2 Dataset

The **EigenWorms** dataset (available at `https://www.timeseriesclassification.com/description.php?Dataset=EigenWorms`) consider a six dimensional space (*channel_0* to *channel_5*) and consider *caenorhabditis elegans* worms of the following types: *wild-type*, labeled as '1'; *goa-1* mutant-type, labeled as '2'; *unc-1* mutant-type, labeled as '3'; *unc-38* mutant-type, labeled as '4'; *unc-63* mutant-type, labeled as '5'. For every case in every channel it was recorded the motion respect the corresponding EigenWorm for 17984 time steps (25 frames per second). Moreover, the dataset contains 259 cases, split into 128 for training and 131 for test. In particular, there are:

- 55 cases classified as '1' in both training and test set;
- 22 cases classified as '2' in both training and test set;

- 17 and 18 cases classified as '3' in training and test set respectively;
- 22 and 23 cases classified as '4' in training and test set respectively;
- 12 and 13 cases classified as '5' in training and test set respectively.

In Figure 1 we can see and compare the path of one case for every class in each EigenWorm. Note that all paths are very irregular and this behavior gets worse as the EigenWorm labels grow. Furthermore, the paths are overlapping, even we can observe a different behavior for different class, in particular for the class '3' case; cause of the overlapping we can expect that the classical classification methods might have some difficult to classify the different cases. In the end, we can note that the data in different channels have different magnitudes, so a good idea can be to normalize the data before to apply the classification methods.

3 Methods

In the literature what we want to do is known as Time Series Classification (TSC). In our case we have 6 different data recorded as time series (channels), so this case is also known as Multivariate Time Series Classification (MTSC). To understand the main ideas of this learning task we start considering the case of one single channel; this case is known as Univariate Time Series Classification (UTSC).

When we want to produce a classifier for the UTSC task, a very used approach is to consider every time step as a feature. For example, in this way we can apply some classical methods, but due to the temporal structure of the data it is known that they are usually not well suited to work on raw time series (actually we noted the overlapping behavior in the previous section). Recently, many approach have been investigated, ranging from deriving new metrics to developing bag-of-words models to applying neural networks. In [2] is available a wide and detailed list of classifier for UTSC. It is known that these advanced methods works really better than the classical methods, but their computational cost can be very high.

For the MTSC task, the main idea is to bring it back to the UTSC case. The most used approaches are:

- to *concatenate* one after the other all channels and to obtain a single classifier;
- to apply a classifier for UTSC *channel-by-channel* and to obtain a classifier for each channel, from which we can obtain the final prediction in some way (the most classified class, the most probable class,...).

In addition to the classification task, we want to evaluate the relevance of EigenWorms/channels and time steps for the outcome of the classification method. A first simple measure of relevance for the channels can be the accuracy of the classifier applied to each channel; this strategy is valid for channel-by-channel approach, but these values can be useful for more general consideration about the classifier used.

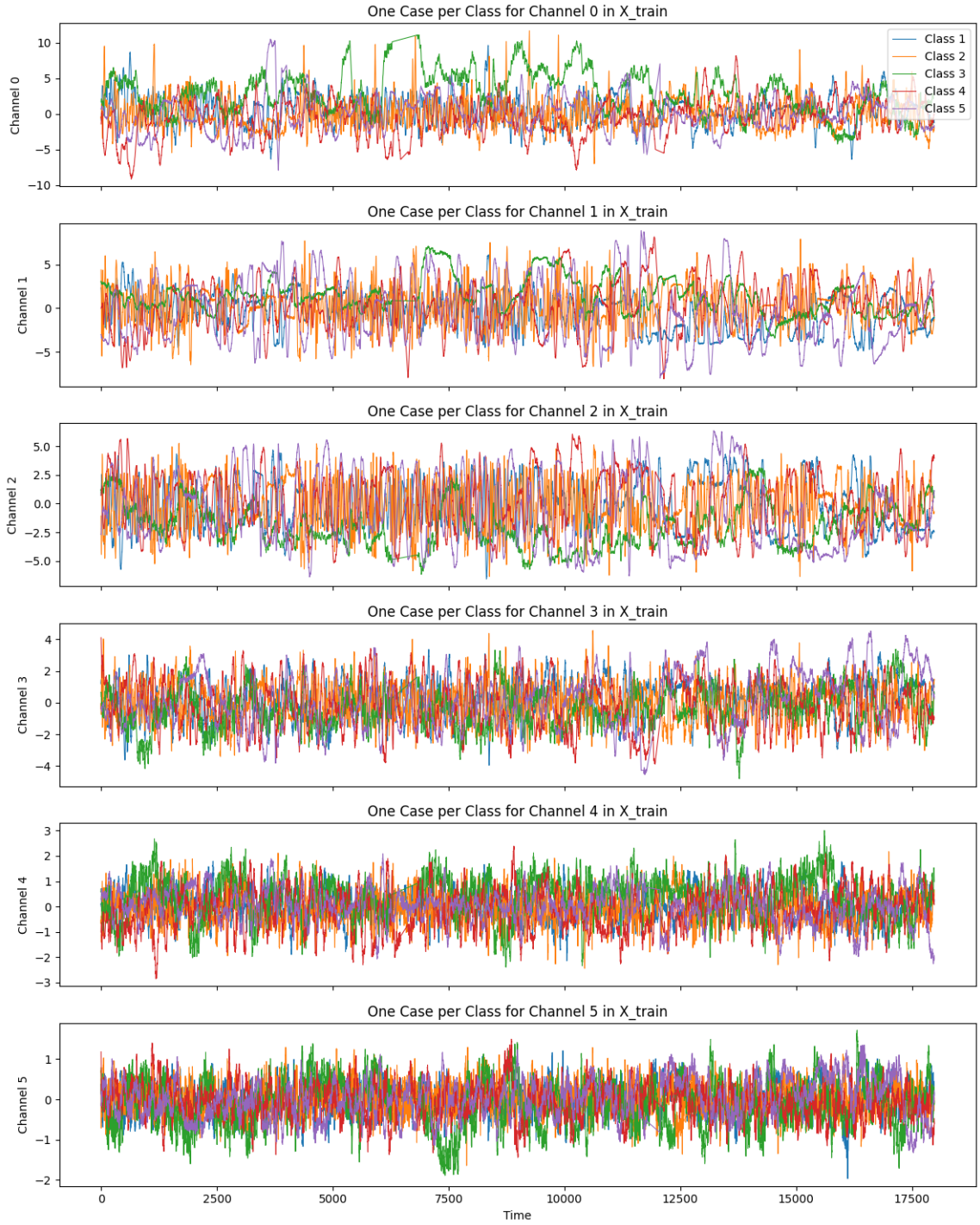


Figure 1: One case per class for each channel from training set.

An idea for the concatenate approach can be to use a regressive classifier (ridge, logistic) and to consider the coefficients of the regression as a measure of relevance for every time step; consequently, using suitable transformation (like mean, sum, squared sum,..) we can also obtain a measure for every channel. In particular, since the regressive classifiers work on binary classification, using for the multiclass case the *one-vs-rest* strategy we have coefficients for every class, so we can measure the relevance to classify each class. An interpretation of this measure can be the following: a positive coefficient means that time step is relevant to choose a given class; a negative coefficient means that time step is relevant to not choose a given class; the absolute value determines how that time step is relevant (in both sense).
There are other specific evaluation strategies for advanced methods (see [3]), but they can be very expensive computationally.

3.1 Classifiers used

First I decided to keep the default split in training and test set. For the channel-by-channel strategy I decide to use as final classifier the most classified class by the channel classifiers. To evaluate the classifiers I used the accuracy, the precision, the Matthews correlation coefficient (MCC) and the confusion matrix.

For its simplicity and wide use, the first classifier I used is the **1-nearest neighbor** with euclidean distance. I compared this classifier using both concatenate and channel-by-channel approaches, in particular for the concatenate approach I also considered the standardization of the channels before to concatenate. Due to the temporal structure and the overlapping behavior noted above, it is not necessary to consider more than 1 neighbor.
Since the values of time series are correlated but the euclidean distance compares the values at each time step independently, it is known that the euclidean distance is not the better choice for this task. In [2] is defined a specific metric (called *Dynamic Time Warping*) for the TSC task, but it is very expensive computationally.

Then I used **Logistic Regression** and **Ridge Regression** classifiers. For both I used only the concatenate approach in order to obtain more complete information from the coefficients, and I compared the models obtained with and without standardization of the channels. For the Ridge Regression, I compute a cross-validation to choose (using the accuracy) the best parameter α of the regularization term among a grid of 10 values from 0.001 to 100.

In the end I used the **ROCKET** (RandOm Convolutional KErnel Transform) classifier ([4]). This is one of the fastest methods among the advanced classification methods specific for the UTSC task, without to sacrifice the accuracy. This algorithm extracts features from time series using a large number of random convolutional kernels and using the transformed features to train a linear classifier (in our case the ridge regression classifier). The ROCKET classifier cover automatically the multivariate case with standardization. In order to compare and to evaulate the channels, I computed also the ROCKET with channel-by-channel approach. The default number of kernels used is 10000; since in [4] it is noted that there is a relative small difference in

accuracy between 5000 and 10000 kernels, in order to keep as low as possible the computational cost I decided to use 5000 kernels. In [3] it is illustrated a specific method for the ROCKET classifier to evaluate channels and time steps, but it is very expensive computationally.

4 Experiment

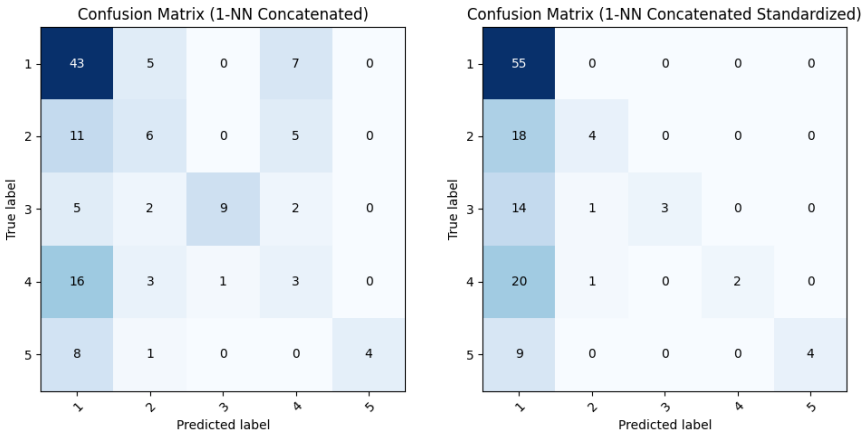
The full Python code is available at

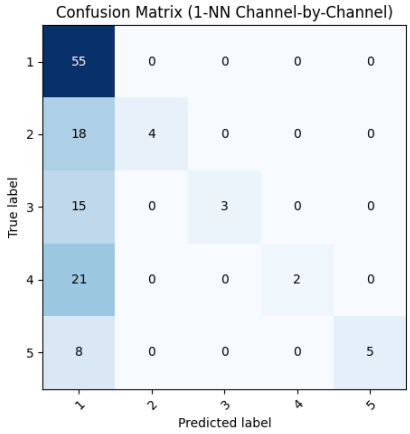
`https://colab.research.google.com/drive/1IpxMSIyT3UKtncAgPuQWhUOKzm-Xfxo9?usp=sharing`.

4.1 Classification

4.1.1 1-nearest neighbor

Method	Accuracy	Precision	MCC
Concatenate	0.496	0.589	0.268
Standard and concatenate	0.519	0.828	0.331
Channel-by-channel	0.523	0.894	0.358

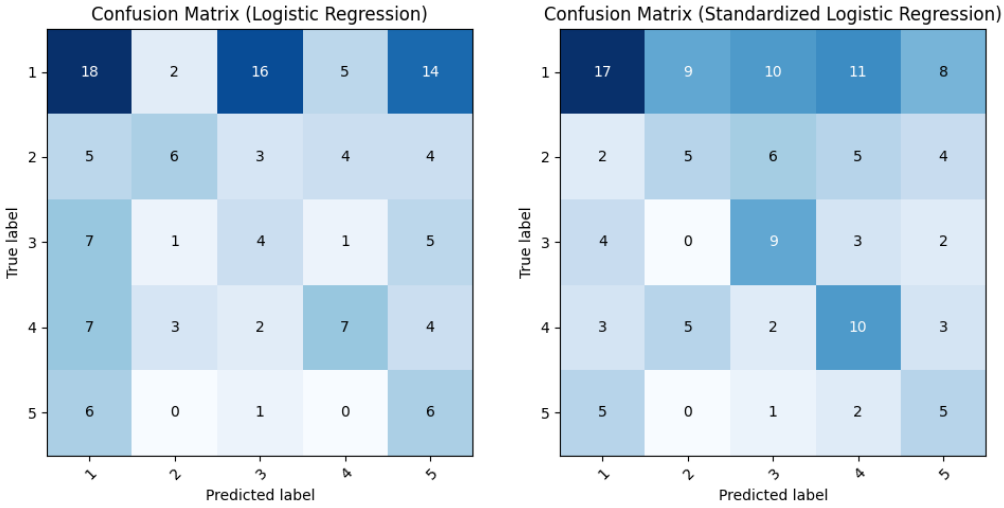




In this case, the best classifier for all the metrics is obtained by the channel-by-channel strategy, but it is not a so good classifier (round 50% accuracy and low MCC).
However, there are some interesting phenomena. First, note that the standardization increase all the metrics for concatenated strategy. Moreover, the precision for concatenated standardized strategy and channel-by-channel strategy is very high due to the sparsity of their confusion matrices: indeed, the precision is a binary metric and for multiclass problems it is calculated as the mean of one-vs-one precision, so every zero entry correspond to an high value of precision. Then in this case the precision can be misleading for the evaluation of the classifiers. In particular, from the confusion matrices we can note that almost every error in the classification is classified as '1', that is the most popular class in the dataset, highlighting the difficulty of these classifiers to distinguish the different classes. In the end, note that the channel-by-channel strategy seems to work slightly better than the concatenate strategies.

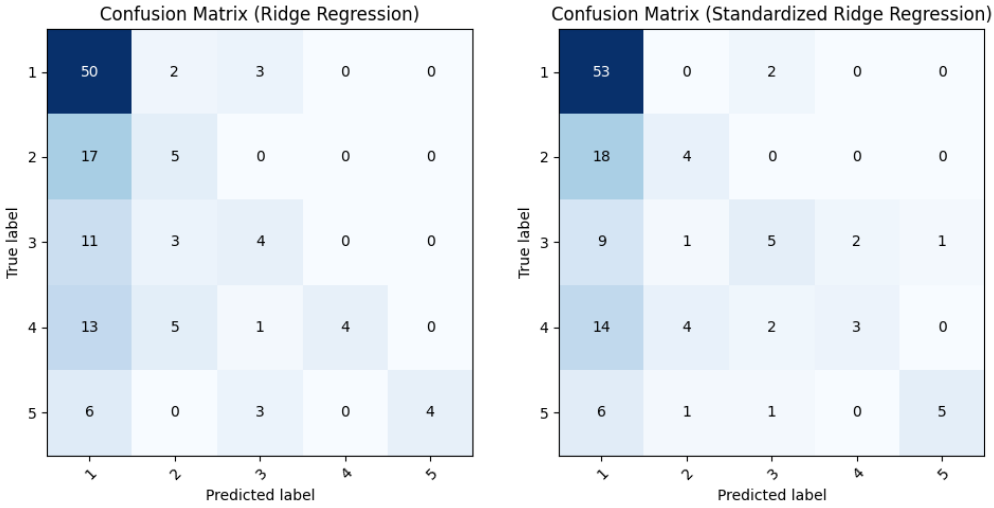
4.1.2 Regressive classifiers

Method	Accuracy	Precision	MCC
Concatenate	0.313	0.333	0.113
Standard and concatenate	0.351	0.337	0.183



The first regressive classifier used is the logistic regression classifier. This classifier works really bad for both direct and standardized strategies. In particular, the very low MCC suggests that these classifiers work almost randomly. However, also in this case the standardization increase all the metrics.

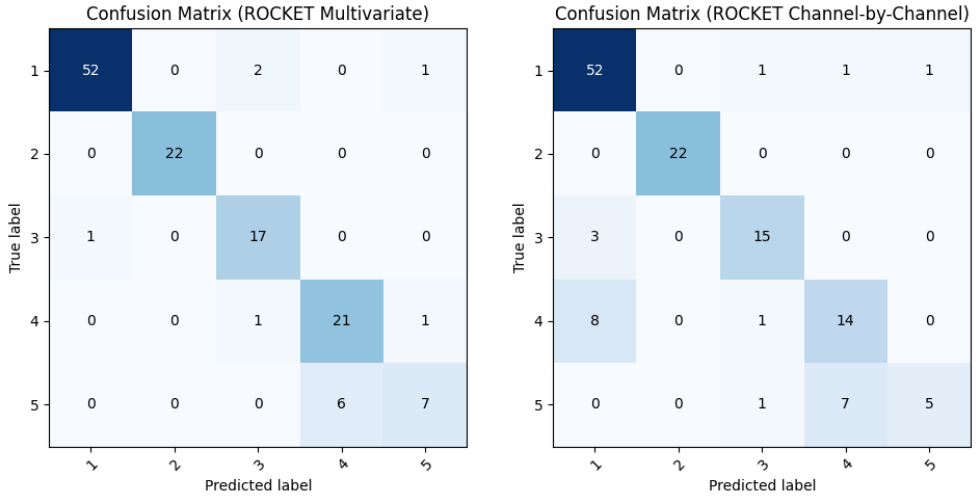
Method	α	Accuracy	Precision	MCC
Concatenate	0.001	0.511	0.642	0.287
Standard and concatenate	100	0.573	0.546	0.329



The ridge regression classifiers work better then logistic regression ones. They achieved an accuracy similar to the best 1-nearest neighbor classifiers, but with lower precision due to the less sparsity of the confusion matrices. In particular, the lower MCC denote an interpretation of this less sparsity as a more random classification, and also in this case there is a dominance of cases classified as '1'. Also in this case the standardization increase the metrics except for the precision (the confusion matrix of the standardized case is less sparse). In particular, the real impact of the standardization is evident in the completely opposite choice of the regularization parameter α (the lowest for the direct strategy, the highest for the standardized strategy).

4.1.3 ROCKET

Method	Accuracy	Precision	MCC
Multivariate	0.908	0.877	0.877
Channel-by-channel	0.824	0.819	0.759



The ROCKET classifiers work really well respect to the other classifier we seen above, highlighting the difference between a specific method for the TSC task and the other generic methods. In particular, the specific multivariate classifier achieves really high score for all metrics, demonstrating it is a very good classifier. Moreover, note that in this case the multivariate strategy works better then the channel-by-channel strategy (contrary to the 1-nearest neighbor case).

Remark 4.1. The precision of ROCKET classifiers is similar to the precision of the best 1-nearest neighbor classifiers. This remind that, for multiclass classification, the precision is a metric that evaluate the sparsity of the confusion matrix, but this is not always a good property for the classification: the ROCKET confusion matrices are example of *good* sparsity, the 1-nearest neighbor confusion matrices are example of *bad* sparsity. So to give to the precision the right interpretation for the evaluation of the classification classification it is necessary to look the confusion matrix.

4.2 Evaluation

Now there are showed some measure of relevance obtained from every different methods used in the previous section. In the conclusion I will compare and interpret all them.

4.2.1 1-nearest neighbor

This method produces an evaluation of the relevance of each channel that is the accuracy of the classifier given for each channel.

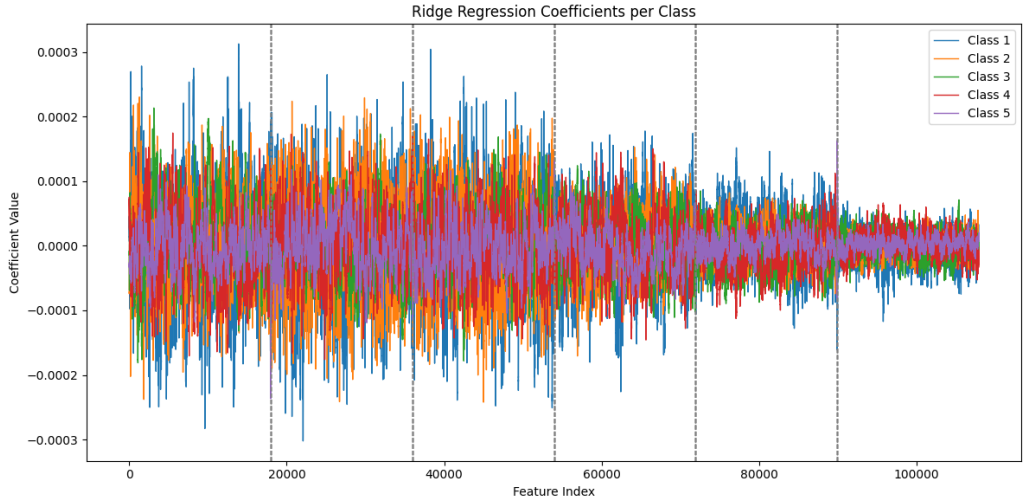
Channel	Relevance
channel_0	0.321
channel_1	0.343
channel_2	0.359
channel_3	0.534
channel_4	0.519
channel_5	0.519

Using this measure, the most relevant channel is the fourth, but in general it can be distinguish two group of channel that are similar relevance: the last three are more relevant than the first three.

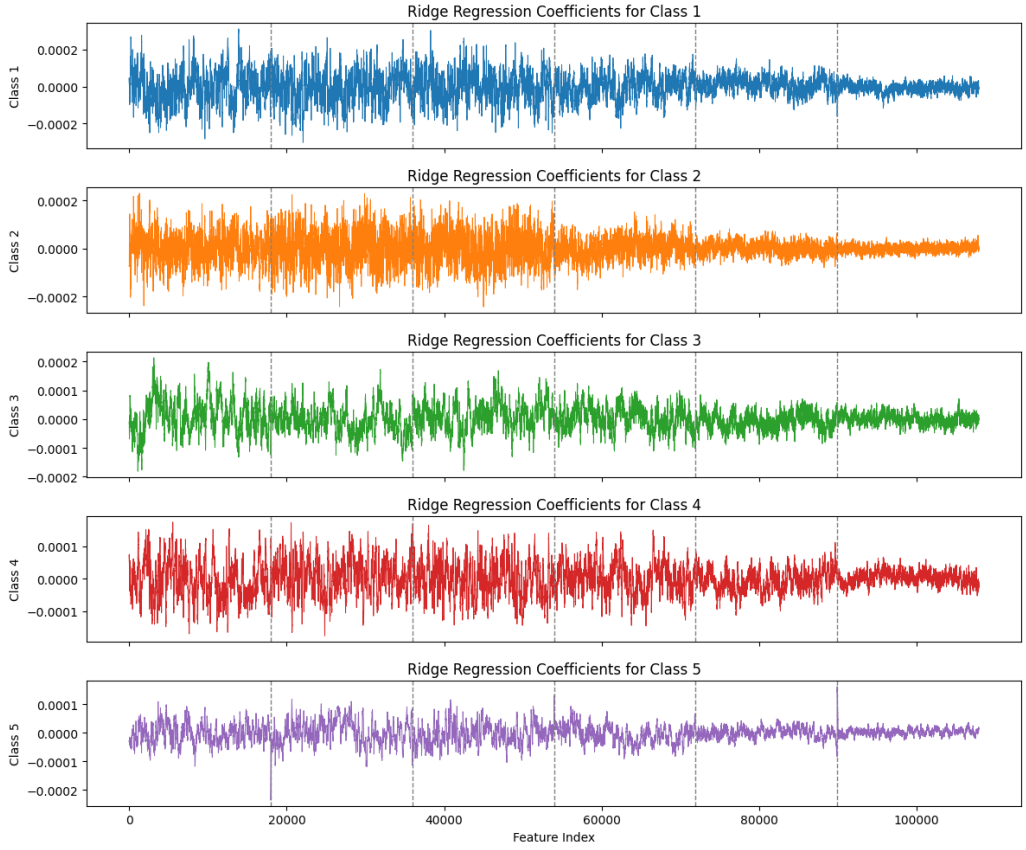
4.2.2 Regressive classifiers

I decided to analyze the relevance measures given to the ridge regression classifiers only: indeed the logistic classifiers are very bad, so their measures of relevance are not really significant.

In this case, the measure of relevance is given by the coefficients of the regression, in particular this is a measure for the time steps. Since the ridge regression classifier works using the one-vs-rest approach, there are coefficients for each class.



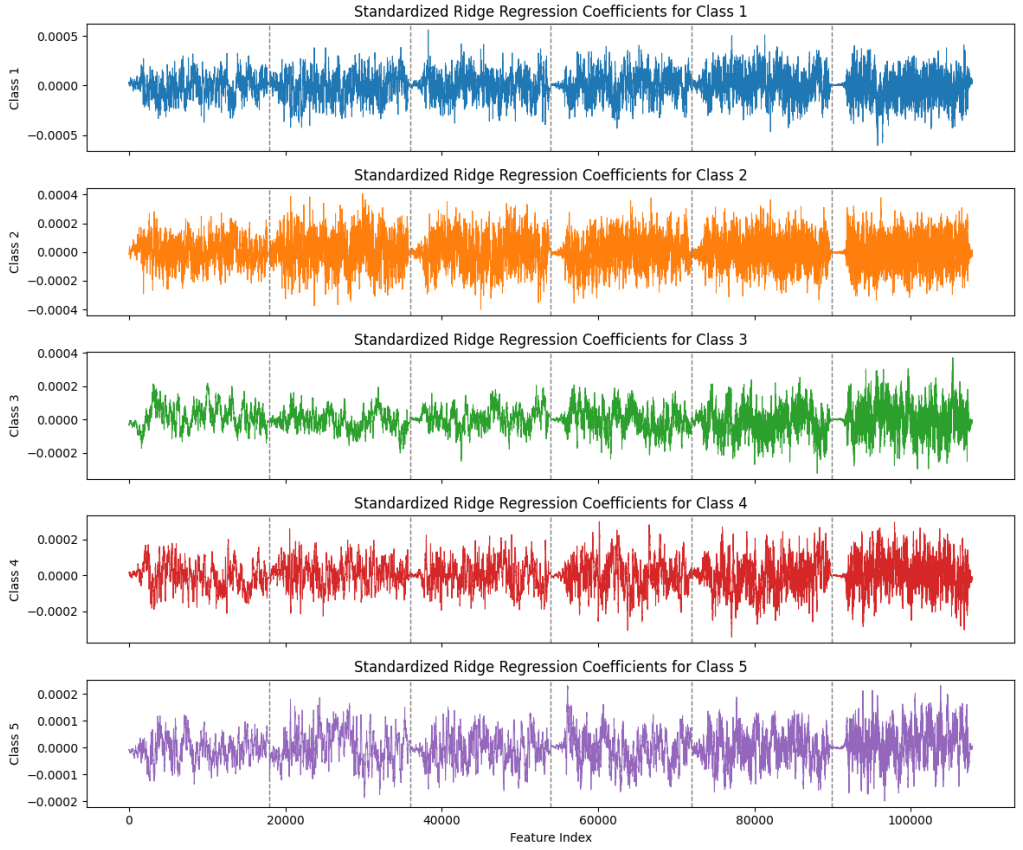
(a) All classes in the same graph.



(b) One graph for each class.

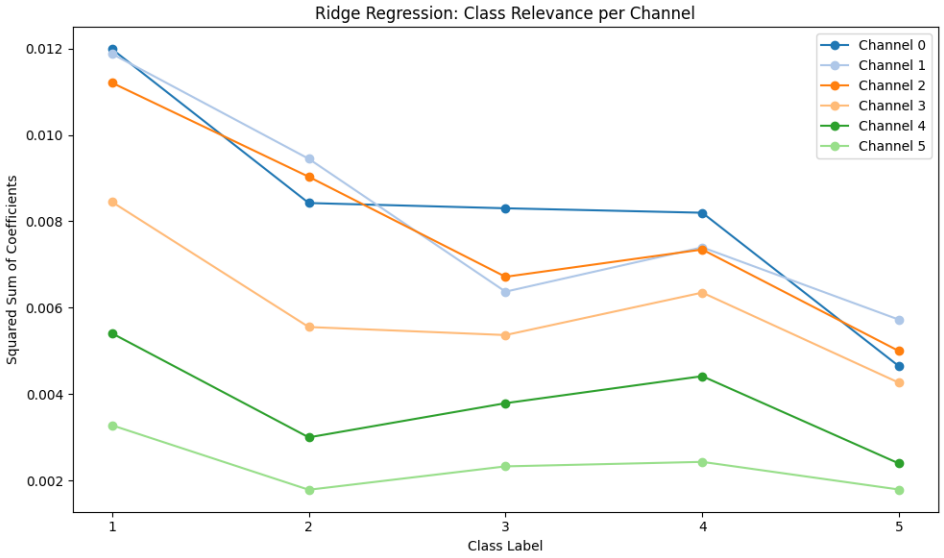
Figure 2: Coefficients of ridge regression. Since the ridge regression classifier works using the one-vs-rest approach, there are coefficients for each class. The gray vertical lines separate each channel in the concatenated model.

(a) All classes in the same graph.

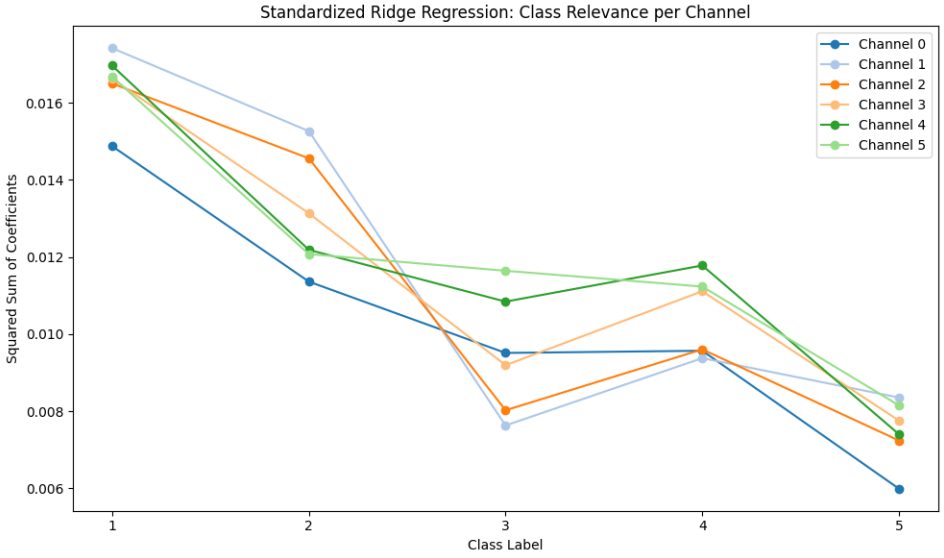


(b) One graph for each class.

Figure 3: Coefficients of ridge regression for standardized channels. Since the ridge regression classifier works using the one-vs-rest approach, there are coefficients for each class. The gray vertical lines separate each channel in the concatenated model.



(a) Ridge regression classifier.



(b) Ridge regression classifier with standardized channels.

Figure 4: Relevance of each channel given from the ridge regression classifiers. Since the ridge regression classifier works using the one-vs-rest approach, there is one relevance for each class.

Figure 2 and 3 show the coefficients for direct and standardized strategy respectively. In both cases, note that the coefficients follow the irregular behavior of the time series. Furthermore, the magnitude of the coefficient decrease when the label of the class increase, in particular this behavior is more evident in the standardized strategy; this can be due to the different (and decreasing) number of cases of each class.
In the direct strategy, note that the coefficient also follow the decreasing behavior of the magnitude when the index of channel increase, while in the standardized strategy the magnitude seems slightly increasing. Moreover, in the standardized strategy, note that for every channel the first time steps are very close to zero: This can be due to the fact that the worms start the experiment in different position, so it is more important for the classification how they move during the experiment, while the starting position is not relevant. Instead in the direct strategy there is not any particular structure.

Figure 4 show the relevance of each channel to predict each class for the direct strategy (4a) and standardized strategy (4b) obtained from the coefficient of the regression. Since the coefficient are distributed around and close to zero, in order to make more evident as possible I decided to use the squared sum of the coefficient as measure of relevance.
For the direct strategy, the relevance follows (except for classes '2' and '5') the channel indexes. It is not surprising since the different magnitudes noted above.
The standardized strategy looks more interesting. First, for every class the differences among the relevance of each channel is lower than the direct strategy (not surprising since the similar magnitudes). Moreover, in this case the two last channels play an important role every class (except for '2'), while the first channel play a marginal role.

In the end, summing the relevance for each class, I obtained the total relevance of the channels.

Channel	Relevance Direct	Relevance Std
channel_0	0.042	0.0513
channel_1	0.041	0.0580
channel_2	0.039	0.0559
channel_3	0.030	0.0578
channel_4	0.019	0.0592
channel_5	0.012	0.0598

As expected, for the ridge regression classifier the ranking of relevance follow the indexes of the channels. While for the ridge regression classifier with standardized channels the ranking is almost the opposite (except for the second channel), in particular the last channel is the most relevant.

4.2.3 ROCKET

This method produces an evaluation of the relevance of each channel that is the accuracy of the classifier given for each channel.

Channel	Relevance
channel_0	0.817
channel_1	0.748
channel_2	0.756
channel_3	0.847
channel_4	0.832
channel_5	0.717

In this case, the most relevant channel is the fourth one and the lowest relevant is the sixth one. In particular, there are two group of channels having a similar relevance (fourth, fifth and first are the more relevant ones, sixth, second and third are the less relevant ones).

5 Conclusion

I analyzed the multivariate time series dataset *EigenWorms* using some classical classifiers as 1-nearest neighbor, logistic and ridge regression, and the specific TSC classifier ROCKET. As expected due to the temporal structure of the data, the classical classifiers do not achieve good result in terms of accuracy, in particular the logistic regression classifiers work almost randomly, while the others have a tendency to choose the most popular class in the training set (in this case '1'). Instead, the ROCKET classifiers work really well.

Another difference between the classical methods and the specific methods is the different behavior between the channel-by-channel and concatenate strategies. Indeed, for the 1-nearest neighbor the channel-by-channel strategy works better, while for the ROCKET works better the concatenate strategy. This fact highlights again the difficulty of the classical methods to extract information from the temporal data.

Since the channels have different magnitudes, the comparison between a direct concatenation of the channels and the standardization of each channel before their concatenation shows an increasing performance for the classifiers when standardize the channels. The reason of this phenomenon have to be searched in the evaluation of the relevance of the channels. The next table summarize the rank of relevance of each channel for every classifier used.

Channel	Rank 1-NN	Rank Ridge	Rank Ridge Std	Rank ROCKET
channel_0	6th	1st	6th	3rd
channel_1	5th	2nd	3rd	5th
channel_2	4th	3rd	5th	4th
channel_3	1st	4th	4th	1st
channel_4	2nd	5th	2nd	2nd
channel_5	2nd	6th	1st	6th

From the rankings of 1-nearest neighbor and ROCKET (computed from the accuracy of each channel), note that some of the last channels (in particular channel_3 and channel_4) are very

relevant for the classification for both classifiers. So it can be deduce that these channels contain some important information about the classes, but, since they have a lower magnitude than the previous channels, it is more difficult for the classifiers to use these information without the standardization of the channels.

A last interesting phenomenon to note is given in the evaluation of the relevance of each time step from the ridge regression classifier with standardized channels. Indeed, in that case the relevance of the first time steps is very close to zero, and this means that the starting position of the worms is not relevant to classify the type of the worm.

References

[1] A. Brown, E. Yemini, L. Grundy, T. Jucikas, and W. Schafer, *A dictionary of behavioral motifs reveals clusters of genes affecting caenorhabditis elegans locomotion*, Proceedings of the National Academy of Sciences of the United States of America (PNAS), vol. 10, no. 2, pp. 791–796, 2013.

[2] J. Faouzi, *Time Series Classification: A Review of Algorithms and Implementations*, 10.5772/intechopen.1004810, 2024.

[3] D. I. Serramazza, T. T. Nguyen, T. L. Nguyen, G. Ifrim, *Evaluating Explanation Methods for Multivariate Time Series Classification*, arXiv:2308.15223, 2023.

[4] A. Dempster, F. Petitjean, G. I. Webb, *ROCKET: exceptionally fast and accurate time series classification using random convolutional kernels*, Data Min Knowl Disc 34, 1454–1495, 2020.