Mini-Project (ML for Time Series) - MVA 2022/2023

Jérémie Dentan jeremie.dentan@live.com
Gonzague de Carpentier decarpentierg@mail.com

March 27, 2023

1 Introduction and contributions

In this report, we present our work on the article *Laplacian Score for Feature Selection* [He et al.(2005)]. In this article, the authors present an unsupervised feature selection method, based on the graph Laplacian of the nearest neighbor graph of the data.

Problem formulation The article deals with the problem of selecting features representing data points, as a preprocessing step of a data analysis task, such as a classification task. this preprocessing step can have several advantages [Guyon and Elisseeff(2003)]: better predictive performances, computational efficiency, having to measure fewer features, and more interpretable models.

More precisely, we give ourselves a dataset X of m points $x_1, ..., x_m$ of a metric space $(\mathcal{M}, d)$, a set of features $F \in \mathbb{R}^{m \times R}$ computed for these points, and consider the problem of selecting the "best" features in a sense that remains to be specified:

- **Wrapper methods.** If the task to perform is specified in advance, one can optimize the feature selection to maximize the score obtained at this task. Those are called *wrapper methods*, since they are wrapped around the algorithm performing the task [Kohavi and John(1997)].
- **Filter methods.** Filter methods, on the contrary, evaluate the intrinsic properties of data to select features prior to a learning task. They can be divided into supervised and unsupervised filter methods: the former evaluates features with respect to the class labels, and the latter rely solely on the data itself. Common filter methods include data variance [Munson and Caruana(2009)], Pearson correlation coefficients [Freedman et al.(2007)], Fisher score [Gu et al.(2012)], and Kolmogorov-Smirnov test [Darling(1957)].
- **Laplacian Score.** In the article, the authors present a new unsupervised filter method, called Laplacian Score, which selects features according to their ability to preserve locality in the data. The goal of this project is to test this method on time series and evaluate the influence of several hyperparameters.

General information about the project Both students contributed equally to the project. We used no preexisting implementation for the computation of the Laplacian score. The source code of our experiments is available in this repository, and our main results are presented in this notebook. Our main contribution was to apply the method to a new kind of data, namely time series. In comparison to the original article, we also tested the influence of more parameters, such as the scaling of the Gaussian kernel used to compute the weighted nearest neighbor graph, or the number of nearest neighbors.

2 Method

Using Laplacian score to select features First, we present how the method works. We are given a dataset $x_1, ..., x_m$ of points of a metric space $(\mathcal{M}, d)$ under the form of their distance matrix $M_{i,j} = d(x_i, x_j)$, and a set $(f_{r,i})_{\substack{1 \leq r \leq R \\ 1 \leq i \leq m}}$ of features for these points. To select a subset $\mathcal{R} \subset [\![1, R]\!]$ of features:

1. **Compute the nearest neighbor graph** G of degree k, whose adjacency matrix is defined as:

$$G_{i,j} := \begin{cases} 1 & \text{if } x_i \text{ is among the } k \text{ nearest neighbors of } x_j \text{ or reciprocally} \\ 0 & \text{otherwise} \end{cases}$$

2. **Compute the weighted adjacency matrix** S**:**

$$S := G \odot \exp\left(-\frac{1}{\sigma^2} M^2\right) \in \mathbb{R}^{m \times m}$$

where $\odot$ denotes the element-wise product, exp the element-wise exponential and $\sigma > 0$ is a hyperparameter of the method.

3. **Compute the degree matrix** D**:**

$$D := \text{diag}(S\mathbb{1}) \in \mathbb{R}^{m \times m} \qquad \text{i.e.} \qquad D_{i,i} = \sum_j S_{i,j}$$

4. **Compute the centered features** $\tilde{f}$**:** Center the features by subtracting to them a weighted average of their values for all data points:

$$\tilde{f}_r = f_r - \frac{f_r^T D \mathbb{1}}{\mathbb{1}^T D \mathbb{1}} \mathbb{1} = f_r - \frac{\sum_{i,j} f_{r,i} S_{i,j}}{\sum_{i,j} S_{i,j}} \mathbb{1}$$

5. **Compute the Laplacian scores** L_r**:** After computing the graph Laplacian, defined as $L := D - S$, the Laplacian score L_r of the r-th feature is defined as:

$$L_r := \frac{\tilde{f}_r^T L \tilde{f}_r}{\tilde{f}_r^T D \tilde{f}_r} \in [0, 1]$$

6. **Select the features** that have the highest Laplacian scores.

Our experiments Based on this method, we decided to conduct several experiments:

- Evaluate the impact of the two hyperparameters of the method: the number of neighbors k, and the variance σ of the Gaussian kernel used to compute S.
- Given that we are dealing with time series, evaluate the impact of computing the distance matrix either with the normalized Euclidian distance, of with DTW [Giorgino(2009)].
- Compare the performance of a pipeline using Laplacian score for feature selection, to a pipeline using other classical methods. We chose to compare the Laplacian score to (1) a simple variance threshold, which is unsupervised like the Laplacian score, and (2) filtering on the ANOVA score [Scheffé(1999)], which is a supersized method.

To measure the performance of those pipelines, we chose to evaluate the classification accuracy of a Support Vector Classifier whose task is to classify a group of times series into two classes, based on features previously extracted from the series and then filtered with the method evaluated.

3 Data and feature extraction

The datasets To test the method on time series data, we consider three datasets taken from [Bagnall et al.([n. d.])]. The **Earthquakes** dataset [Bagnall([n. d.])] aims at classifying major earthquake events and their absence based on hourly readings from Northern California Earthquake Data Center. The **Wafer** dataset [Olszewski([n. d.])] contains inline process control measurements from various sensors during the processing of silicon wafers for semiconductor fabrication, with two classes of normal and abnormal and a large class imbalance. The **WormsTwoClass** dataset [Brown and Bagnall([n. d.])] aims at classifying individual worms as wild-type or mutant strains based on a projection of their motion on a dimension called "first eigenworm" and down-sampled to second-long intervals. Thus, all these datasets consist in *binary classification*.

In Figure 1, we can see that the Earthquakes and WormsTwoClass time series seem to be weakly stationary stochastic processes, whereas Wafer is not. Plotting the autocovariance functions (Figure 2) enables to see that the Earthquakes samples are very uncorrelated, whereas the WormsTwoClass samples are much more correlated. The Wafer autocovariance is more difficult to interpret since the process is not stationary.

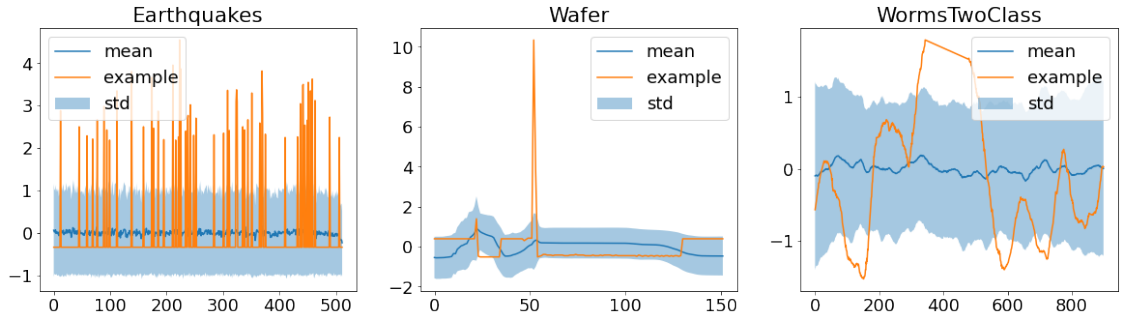

Figure 1: Visualization of the three datasets. For each dataset, we plot the average time series, the standard deviation at each timestamp and an example sampled randomly from the dataset.

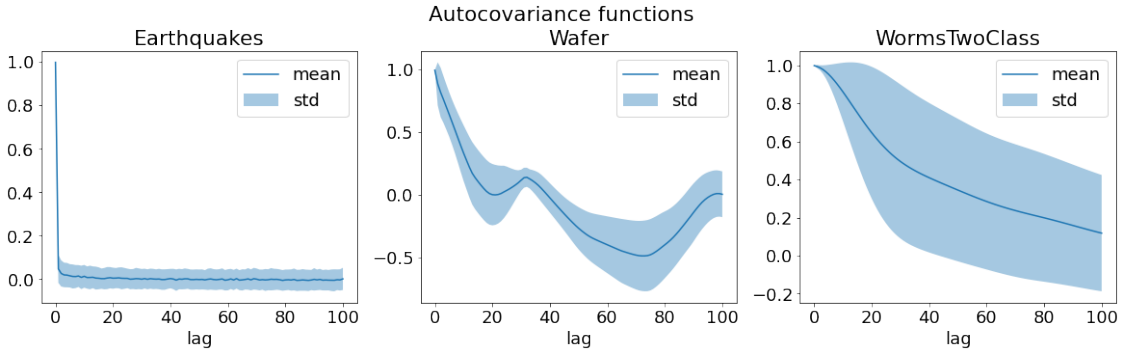

Figure 2: Average autocovariance functions for the three datasets.

The features Finally, since feature extraction is not the task we are interested in here, we used TSFEL [Barandas et al.(2020)], a library specialized in feature extraction from time series. The number of features extracted per series depends on the time series, and in the case of our data it varies between 210 and 389.

4 Results

4.1 Impact of parameters σ and k

Histograms of the Laplacian scores First, we plot some histograms of the distributions of the Laplacian scores when sigma varies. To do so, we choose a dataset (here, the Wafer one), we fix the number of neighbors, for example to 25. Moreover, we focus on the ratio $\sigma/\overline{M}$, where $\overline{M}$ is the average of the values of M, which makes more sense than the raw value of sigma. The result can be seen in Figure 3. We observe on this figure (and the similar ones for the other dataset, that are easy to generate with the notebook) that there are two quasi-stationary phases: one when σ is really small, and one when σ is really large. This is not surprising. With the notations of section 2:

- When σ tends to zero, $S \to 0$, so $D \to 0$ and $L \to 0$. Given that we added a small ε in the division for the computation of the Laplacian score, it is thus normal that the scores converge to zero. This corresponds to a situation when every point can only "see" itself due to the Gaussian kernel.
- When σ is really large, $S \to G$. This corresponds to a situation where all neighbors have the same importance due to the high radius of the Gaussian kernel. Thus, the Laplacian scores tends to their values with S was perfectly equal to G.

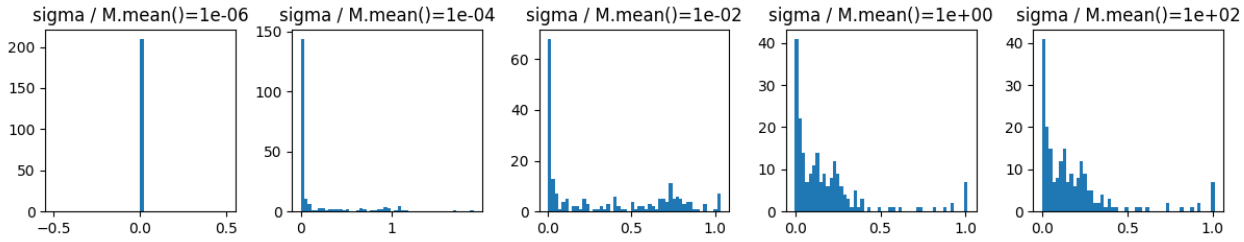

Figure 3: Histograms of the value of the Laplacian score for several values of $\sigma/\overline{M}$

Impact of σ on the classification accuracy These histograms allowed us to understand the general behavior of the score. However, to better understand the impact of σ, we directly measured its impact on the classification accuracy. To do so, we fixed the number of neighbors, and we made σ vary between 10^{-7} and 10^{+3}. We repeated this experiment with several values of the number of feature that were finally selected. The results are presented in figure 4. We observe that:

- For some datasets (namely, Wafer and Earthquakes), the accuracy is quite stable, no matter the value of σ or the number of features selected. This is probably because the classification problem is either too simple or too hard with the features we have. For example with dataset Earthquakes, we observe a performance of $0.8 \simeq 368/461$, which is the proportion of label "1" in the dataset. No matter the features that are selected, they are not informative enough, and the best that can be done by the SVC is to predict the same label for every sample.
- In most cases, the best performance comes for low sigma values, around 10^{-4}. In the case of 150 features selected, the maximum is reached around 10^{-2}, however this maximum is only equal to the maximum with 30 features.

Thus, for the stability of the method, it is safe to always take a quite small value for $\sigma/\overline{M}$, yet not too small to to avoid that every Laplace score is equal to 0.

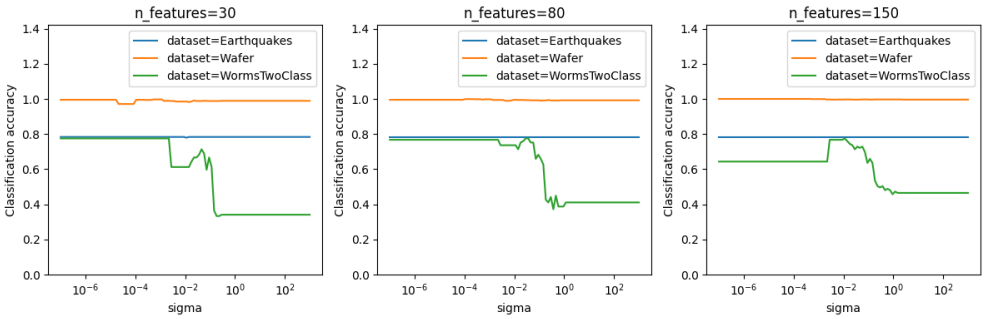

Figure 4: Evolution of the classification accuracy against the value of $\sigma/\overline{M}$.

Impact of the number of neighbors k on the classification accuracy Then, we measured the impact of the number of neighbors on the performances. The result is presented in figure 5. We observe that if we stick to small values of $\sigma/\overline{M}$, the number of neighbors does not really the classification accuracy. To conclude, a good heuristic for the choice of σ and k is to take $\sigma/\overline{M}$ small enough, around 10^{-4}, and k medium, of the order of ten.

4.2 Comparison with other methods

Finally, we compared this filter method with two other: a variance threshold (unsupervised), and the ANOVA score (supervised). The result is presented in Figure 5. We observe that the method based on the Laplace score has similar performances to these two techniques, and is even better on certain ranges of values. However, for a limited number of features, the performance gap remains important, and constitutes a major disadvantage of the method. Finally, we observe that contrary to what we could have intuitively imagined, the performances are not improved, on the contrary, by using the DTW, and this even if DTW is generally considered more relevant for time series.

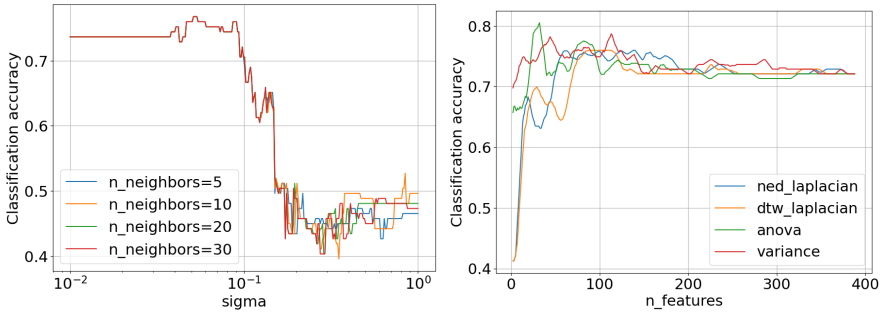

Figure 5: Left: Evolution of the classification accuracy against the value of sigma. Right: Evolution of the classification accuracy against the number of features.

5 Conclusion

Our experiments identified heuristics for choosing sigma and the number of neighbors in the absence of additional knowledge about the data. Moreover, our experiments have shown that this method leads to relatively good performances compared to other widely used methods such as the ANOVA score, although they remain slightly inferior.

References

[Bagnall([n. d.])] Anthony Bagnall. [n. d.]. Earthquakes dataset. https://timeseriesclassification.com/description.php?Dataset=Earthquakes

[Bagnall et al.([n. d.])] Anthony Bagnall, Eamonn Keogh, Jason Lines, Aaron Bostrom, James Large, and Matthew Middlehurst. [n. d.]. Time Series Classification Website. https://timeseriesclassification.com/dataset.php

[Barandas et al.(2020)] Marília Barandas, Duarte Folgado, Letícia Fernandes, Sara Santos, Mariana Abreu, Patrícia Bota, Hui Liu, Tanja Schultz, and Hugo Gamboa. 2020. TSFEL: Time Series Feature Extraction Library. *SoftwareX* 11 (Jan. 2020), 100456. https://doi.org/10.1016/j.softx.2020.100456

[Brown and Bagnall([n. d.])] Andre Brown and Anthony Bagnall. [n. d.]. WormTwoClass dataset. https://timeseriesclassification.com/description.php?Dataset=WormsTwoClass

[Darling(1957)] D. A. Darling. 1957. The Kolmogorov-Smirnov, Cramer-von Mises Tests. *The Annals of Mathematical Statistics* 28, 4 (1957), 823–838. http://www.jstor.org/stable/2237048 Publisher: Institute of Mathematical Statistics.

[Freedman et al.(2007)] David Freedman, Robert Pisani, and Roger Purves. 2007. Statistics (international student edition). *Pisani, R. Purves, 4th edn. WW Norton & Company, New York* (2007).

[Giorgino(2009)] Toni Giorgino. 2009. Computing and Visualizing Dynamic Time Warping Alignments in *R* : The **dtw** Package. *Journal of Statistical Software* 31, 7 (2009). https://doi.org/10.18637/jss.v031.i07

[Gu et al.(2012)] Quanquan Gu, Zhenhui Li, and Jiawei Han. 2012. Generalized Fisher Score for Feature Selection. https://doi.org/10.48550/arXiv.1202.3725 arXiv:1202.3725 [cs, stat].

[Guyon and Elisseeff(2003)] Isabelle Guyon and André Elisseeff. 2003. An introduction to variable and feature selection. *The Journal of Machine Learning Research* 3, null (March 2003), 1157–1182.

[He et al.(2005)] Xiaofei He, Deng Cai, and Partha Niyogi. 2005. Laplacian Score for Feature Selection. In *Advances in Neural Information Processing Systems*, Vol. 18. MIT Press. https://proceedings.neurips.cc/paper/2005/hash/b5b03f06271f8917685d14cea7c6c50a-Abstract.html

[Kohavi and John(1997)] Ron Kohavi and George H. John. 1997. Wrappers for feature subset selection. *Artificial Intelligence* 97, 1 (Dec. 1997), 273–324. https://doi.org/10.1016/S0004-3702(97)00043-X

[Munson and Caruana(2009)] M. Arthur Munson and Rich Caruana. 2009. On Feature Selection, Bias-Variance, and Bagging. In *Machine Learning and Knowledge Discovery in Databases (Lecture Notes in Computer Science)*, Wray Buntine, Marko Grobelnik, Dunja Mladenić, and John Shawe-Taylor (Eds.). Springer, Berlin, Heidelberg, 144–159. https://doi.org/10.1007/978-3-642-04174-7_10

[Olszewski([n. d.])] Robert Thomas Olszewski. [n. d.]. Wafer dataset. https://timeseriesclassification.com/description.php?Dataset=Wafer

[Scheffé(1999)] Henry Scheffé. 1999. *The Analysis of Variance*. John Wiley & Sons. Google-Books-ID: z9yUEAAAQBAJ.