



POLITECNICO
MILANO 1863

**SCUOLA DI INGEGNERIA INDUSTRIALE
E DELL'INFORMAZIONE**

EXECUTIVE SUMMARY OF THE THESIS

An interpretable ordinal clustering for functional data with an application to antigen reaction profiles

LAUREA MAGISTRALE IN MATHEMATICAL ENGINEERING - INGEGNERIA MATEMATICA

Author: ALESSANDRO GANELLI

Advisor: PROF. FEDERICA NICOLUSSI

Co-advisor: PROF. ALESSANDRA MENAFOGLIO, DR. GIULIA PATANÈ

Academic year: 2023-2024

1. Introduction

Functional Data Analysis (FDA) has gained significant attention in recent years, earning a central role in statistical research. For functional data we often, but not always, intend a set of curves defined in an infinite-dimensional space, where each curve represents a realization of an underlying random process. Within this framework, functional clustering has also emerged as topic of interest, which formally refers to the unsupervised task of grouping a set of curves in distinct subsets, named clusters, in such a way that instances within a group are similar to each other while they are dissimilar to instances of other groups. Various methods have been developed, each leveraging different characteristics of functional data. Moreover, several classification schemes have been proposed in order to have a systematic description, such as those by [2]. Among the main categories, we can identify: raw data methods, which cluster functions based on their observed values on a discrete grid of their domain; filtering methods, which first approximate curves using a basis expansion, leading to a finite-dimensional representation, and then perform clustering on the resulting finite-dimensional objects; distance-

based methods, which rely on functional distance metrics to quantify similarities between curves. Notably, the latter category is overlapping with the previous two, as the choice of distance metric determines whether the method aligns more closely with raw data approaches or filtering approaches. While existing methods achieve well-separated functional clusters, they primarily focus on partitioning rather than uncovering the underlying mechanisms driving the observed curves. This limitation became evident to us during the analysis of a dataset containing the profiles of different antigen reactions. These curves seem to follow a shared pattern, consisting of an initial stable phase followed by a rapid decline at different intensity levels. The variation in these levels is driven by differences in the reagent concentration applied to the antigen, suggesting that the observed curves are governed by an ordinal latent structure, where a specific cluster membership corresponds to a different magnitude of the common underlying effect. Traditional clustering techniques do not explicitly capture this ordinal nature nor estimate the latent component influencing curve generation. To address this, we introduce K-Models, a new clustering approach tailored for functional data

with an underlying ordinal structure. By grouping functions based on a common latent effect and structuring clusters according to its intensity in an ordinal manner, K-Models not only produce meaningful partitions but also provide estimates of the underlying function and cluster-specific coefficients, enhancing interpretability.

2. Methodology

In the context of functional data clustering where latent groups are inherently endowed with an order relation, we consider a set of functional observations $\{f_i(t)\}_{i=1}^N$, where each function takes values in \mathbb{R} and is defined on a common domain $\mathcal{T} \subset \mathbb{R}$. Given a number of clusters K , the goal is to find an optimal partition $\mathcal{C}^* = \{C_1^*, \dots, C_K^*\}$ into which divide the curves.

2.1. K-Models

Unlike traditional clustering approaches, this procedure explicitly assumes that the latent classes follow an ordinal structure. Specifically, if C_i and C_j are two latent classes, we assume $C_i < C_j$ whenever the amplitude of the underlying effect g in C_i is smaller than in C_j . This ordering is reflected by the estimated parameters. In this setting, we consider the following cluster-specific functional linear model:

$$\begin{aligned} f_i(t) &= \beta_{0k} + \beta_{1k}g(t) + \varepsilon_i(t) \\ &= \psi((g, \beta_k), t) + \varepsilon_i(t) \quad t \in \mathcal{T}, \end{aligned} \quad (1)$$

where the curve f_i belongs to the cluster \mathcal{C}_k , with $k \in \{1, \dots, K\}$. The parameters $\beta_k = [\beta_{0k}, \beta_{1k}]^\top$ consist of a scalar intercept β_{0k} and a scalar coefficient β_{1k} that linearly modulates the function g , defined on the same domain \mathcal{T} as the observed curves¹. Lastly, the term $\varepsilon_i(t)$ is a functional random process with the null function as its mean, i.e., $\mathbb{E}[\varepsilon_i(t)] = 0$ for all $t \in \mathcal{T}$. The function g captures the hidden phenomenon underlying the generation of the curves f_i , while the coefficients β_{1k} capture its effect within the specific cluster \mathcal{C}_k . Crucially, the ordinal nature of the latent classes is entirely determined by the values of β_{1k} , as they define a hierarchical ordering of the clusters, according to their increasing values. The complete procedure for

¹For notional convenience, we introduce the function $\psi((g, \beta_k), t) = \beta_{0k} + \beta_{1k}g(t)$, representing the cluster-specific structural component.

the estimation of Model 1 is outlined in Algorithm 1. Specifically, in step j.1, the coefficients β_k^j are estimated using a function-on-function regression approach [1]. At this stage of the algorithm, the function \hat{g}^{j-1} —estimated in the previous iteration—is used as a functional covariate. Subsequently, in step j.2, the perspective is reversed: a function-on-scalar regression [4] is employed to update g , using the newly estimated coefficients as scalar predictors. Lastly, in step j.3 the cluster memberships are updated and then reordered based on the coefficient β_{1k}^j . This step can be performed using one of two following reassignment strategies: (1) *Best-fitting model*, with each curve reassigned to the cluster $C_{\tilde{k}}$, whose model yields the lowest residual error, so that $\tilde{k} = \arg \min_{k \in \{1, \dots, K\}} \|f_i - \psi(\hat{g}^j, \hat{\beta}_k^j)\|_{L^2}$; (2) *Coefficients proximity*, with each curve reassigned to the cluster $C_{\tilde{k}}$, whose estimated coefficients are closest to those obtained for that curve alone. Specifically, $\tilde{\beta}_i$ is estimated using a function-on-function regression with only f_i as whole dataset, using \hat{g}^j as the functional covariate. The new cluster is then assigned as $\tilde{k} = \arg \min_{k \in \{1, \dots, K\}} \|\tilde{\beta}_i - \hat{\beta}_k^j\|_2$.

Algorithm 1 K-Models Pseudo Algorithm

Input: A functional dataset $\{f_i(t)\}_{i=1}^N$, number of clusters K

Output: Clusters partition \mathcal{C} , estimated common effect \hat{g} and cluster-specific coefficients $\{\hat{\beta}_k\}_{k=1}^K$

- 1 **Initialize:** Randomly divide the curves into K groups and let g equal to the pointwise mean of the dataset, i.e. $\hat{g}^0(t) = \frac{1}{N} \sum_{i=1}^N f_i(t)$
 - 2 **Iter j:**
 - j.1) Built K different models, one for each cluster, and estimate the coefficients $\hat{\beta}_k^j$, through function-on-function regression:

$$f_i(t) = \boxed{\beta_{0k}^j} + \boxed{\beta_{1k}^j} \hat{g}^{j-1}(t) + \varepsilon_i(t).$$
 - j.2) Estimate the function \hat{g}^j using the whole dataset, through function-on-scalar regression:

$$f_i(t) = \hat{\beta}_{0k}^j + \hat{\beta}_{1k}^j \boxed{g^j(t)} + \varepsilon_i(t).$$
 - j.3) Update the cluster membership for each curve, reorder according to the increasing values of $\hat{\beta}_{1k}^j$
 - 3 **until convergence;**
-

Based on these two reassignment strategies, we implement two versions of the K-Models, respectively referred to simply as K-Models (i.e. best-fitting model) and K-Models v2 ("version 2", i.e. coefficient proximity). Two additional refinements are introduced. The first is a repopulation mechanism. Since some clusters may experience a sharp reduction in the number of assigned curves after the first iterations², whenever the number of curves assigned to a cluster falls below a predefined threshold, an additional set of curves is randomly reassigned to that cluster³. The second is a multiple-start approach, employed to mitigate the impact of the random initialization.

2.2. Competitor methods

2.2.1. K-Means

The first clustering method used as a comparison is the well known K-Means. Its idea is to partition data by minimizing the total dissimilarity between curves and their cluster centroids, defined as $\sum_{k=1}^K \sum_{f_i \in C_k} d^2(f_i, z_k)$. Specifically, a centroid is a representative element of the cluster, computed as the pointwise mean of its observations, i.e. $z_k(t) = \frac{1}{|C_k|} \sum_{f_i \in C_k} f_i(t)$. The algorithm iteratively assigns each curve to the closest centroid—based on the distance metric d —and updates the centroids $\mathcal{Z} = \{z_1, \dots, z_K\}$ accordingly. In this study, similarity between curves is measured using the L^2 -norm, so that $d(f_i, z_k) = \|f_i - z_k\|_{L^2}$.

2.2.2. PAM

Partitioning Around Medoids (PAM), also known as K-Medoids, follows the same objective as K-Means—minimizing total dissimilarity within clusters—but replaces centroids with medoids, defining the so called Total Deviation as $TD := \sum_{k=1}^K \sum_{f_i \in C_k} d(f_i, m_k)$. A medoid m_k is the data point within each cluster that minimizes the total dissimilarity to all other points, i.e. $m_k := \arg \min_{f_i \in C_k} \sum_{f_j \in C_k} d(f_i, f_j)$. Notably, unlike centroids, which are computed as averages, medoids must be actual data points, making PAM more robust to outliers. The algorithm consists of two main steps: (1) *Build*, initializes the K medoids by choosing for K times

the instance which yields to the smallest TD (e.g. choosing at first the closest curves to all the others); (2) *Swap*, iteratively improves the clustering by considering all possible swaps between a medoid and a non-medoid instance and applying the one which reduces at most TD . As in K-Means, we use the L^2 -norm as dissimilarity measure for the curves.

2.2.3. FPCA & K-Means

Unlike previous methods, this approach falls into the category of filtering methods, since the idea is now to first reduce the dimensionality of the functional dataset using Functional Principal Component Analysis (FPCA), before applying clustering. The functional observations are so approximated as linear combinations of basis functions, by exploiting the Karhunen-Loève expansion: let F be a stochastic process, supposed belonging to $L^2(\mathcal{T})$, this can be expressed as $F(t) = \mu(t) + \sum_{l=1}^{\infty} \alpha_l \phi_l(t)$ for all $t \in \mathcal{T}$, where $\mu(t) := \mathbb{E}[F(t)]$ is its mean function, $\{\phi_l\}_{l=1}^{\infty}$ are the principal functions and $\{\alpha_l\}_{l=1}^{\infty}$ the corresponding principal component scores. Applying this expansion to a functional dataset $\{f_i(t)\}_{i=1}^N$, we are able to extract the first p principal component scores corresponding to each curve, with p suitably chosen, and build the multivariate dataset $\{\alpha_i\}_{i=1}^N$, with $\alpha_i \in \mathbb{R}^p$. Clustering is then performed using K-Means presented in Section 2.2.1, with the key difference that similarity is now measured using the Euclidean distance between score vectors instead of the L^2 -norm on curves.

3. Validation

To assess the performance of the proposed clustering methods, we employ two widely used evaluation metrics: the Variability Ratio and the Silhouette Score. These two measures provide complementary perspectives on clustering quality, offering both a global and a local evaluation of the resulting partitions. The first metric is defined as the ratio between the Between-Cluster Sum of Squares (BSS) and the Within-Cluster Sum of Squares (WSS), where $BSS := \sum_{k=1}^K N_k \|\mu_k - \mu_0\|_{L^2}^2$ and $WSS := \sum_{k=1}^K \sum_{f_i \in C_k} \|f_i - \mu_k\|_{L^2}^2$, with N_k the number of curves in cluster C_k , while $\mu_k(t) := \frac{1}{N_k} \sum_{f_i \in C_k} f_i(t)$ is their pointwise mean, and $\mu_0(t) := \frac{1}{N} \sum_{i=1}^N f_i(t)$ is the global pointwise

²Resulting in a stagnation of the algorithm.

³The threshold and the number of curves reassigned depend on the dimensionality of the dataset used.

mean. A higher ratio quantifies how well-separated the clusters are relative to their internal cohesion. The second metric is instead the well known Silhouette Score [5]. It captures local clustering behavior, allowing for the identification of misclassified observations for those positioned near cluster boundaries. This score ranges between -1 and 1, with higher values corresponding to better partitions of the curves.

4. Simulation study

Our novel method and the presented competitors are firstly tested on a series of Monte Carlo simulations, in order to assess the performances of K-Models and its ability to reconstruct the structural components of a stochastic process. Each of the 30 simulations consists of $N = 400$ curves, defined on $\mathcal{T} = [0, 1]$ and divided in $K = 4$ known groups. The data follow the structure of Equation 1, with $g(t) = \sin(5t)$ as common effect. The ordinal structure instead is defined by the coefficients $\beta_{1k} \stackrel{\text{iid}}{\sim} \mathcal{N}(0, 2^2)$, and in the same way $\beta_{0k} \stackrel{\text{iid}}{\sim} \mathcal{N}(0, 0.8^2)$. A stochastic noise component $\varepsilon_i(t)$ introduces variability across curves. Figure 1 illustrates an example dataset from a single simulation. The methods from Section 2

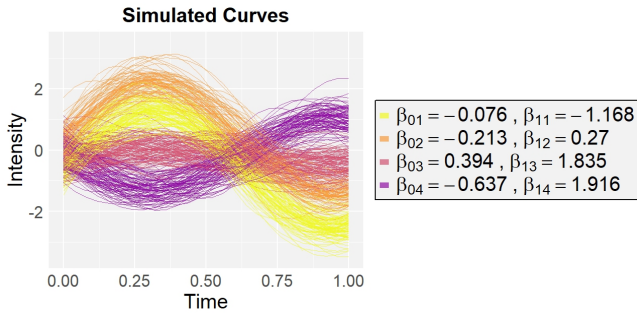


Figure 1: Simulated functional dataset, with cluster specific coefficients.

are applied for a number of clusters K as input from 2 to 7. The Monte Carlo estimation of metrics proposed in Section 3 is shown in Figure 2. Overall, the results are quite similar for all proposed methods, with the exception of K-Models v2, which generally performs slightly worse for most values of K . Meanwhile, K-Models demonstrates results comparable to the other methods, maintaining a good level of cluster separation while offering better interpretability of the clusters and the underlying stochastic process that generated the curves. An interesting observa-

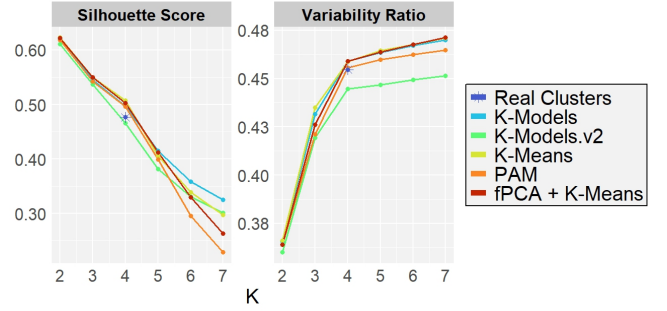


Figure 2: Monte Carlo estimation of the Silhouette score and of the Variability ratio.

tion emerges for $K = 4$, where the partitions obtained by the different methods can be compared to the original clustering of the curves. In this context, K-Models v2 appears to be the closest to the real partition when looking at the Silhouette Score. This method performs particularly well for the curves that are more 'borderline', showing its ability to handle ambiguous cases more effectively. Through the real partition of the data we are also able to compute different valuations commonly used for supervised classification models, with relative Monte Carlo estimations shown in Table 1. From these we see how both versions of K-Models demonstrate competitive performances, making them a viable alternative for clustering tasks where reconstructing the true partition is of interest.

	K-Models	K-Models v2	K-Means	PAM	fPCA & K-Me.
Accuracy	92.2 ± 10.2	90.3 ± 10.3	81.7 ± 16.5	92.1 ± 10.8	93.4 ± 8.5
Precision	92.1 ± 10.3	90.2 ± 10.3	80.8 ± 17.4	92.0 ± 10.9	93.4 ± 8.5
Recall	92.1 ± 10.4	90.7 ± 9.7	80.3 ± 17.9	92.3 ± 10.4	93.5 ± 8.4
ARI	0.9 ± 0.1	0.8 ± 0.2	0.8 ± 0.2	0.9 ± 0.2	0.9 ± 0.1

Table 1: Monte Carlo estimation of the classification performance for each method for $K = 4$. Values are reported as mean ± standard deviation. ARI stands for Adjusted Rand Index.

In this simulation study, we have the possibility to compare the true underlying quantities with the corresponding estimates obtained from our K-Models. As instance, Figure 3 illustrates the cluster-specific curves, structured according to Equation 1, where each curve is determined by the estimated coefficients $\hat{\beta}_{1k}$ and the shared component \hat{g} , for the run of our Monte Carlo study displayed in Figure 1. Examining the estimated \hat{g} , we observe that the model successfully captures the overall shape of the true function

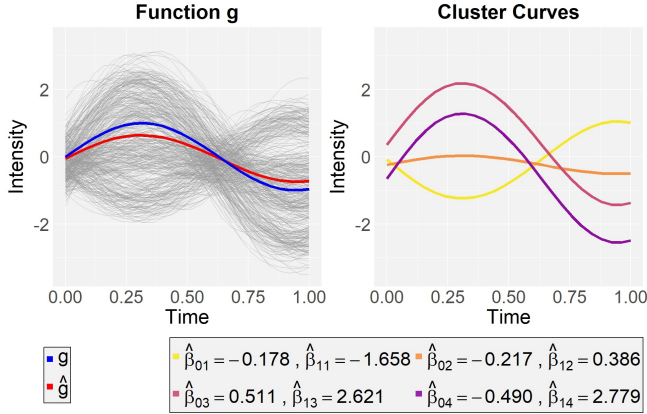


Figure 3: Results of the K-Models with $K = 4$ for an example run.

g . However, \hat{g} tends to exhibit a slightly lower amplitude, an effect compensated by the estimated coefficients $\hat{\beta}_{1k}$, which assume larger absolute values. This interplay suggests that while local distortions in amplitude and sign may occur, the model preserves the functional structure and relationships within the data.

5. Case study: ROI curves

This case study analyzes biosensor signals, represented by a dataset consisting of a video signal acquired at a rate of one frame per second, recorded using a reflectometric sensor. Each frame captures the intensity of reflected light at a given time point. Within the sensor, 1035 points were identified, each containing a specific concentration of the t-BSA antigen, arranged in a regular grid, thus defining 1035 regions of interest (ROI). To prepare the data for the analysis, a two-step preprocessing pipeline [3] is applied: (1) *light correction*, to account external illumination variations and sensor surface irregularities; (2) *functional data construction*, through a log-ratio transformation and a smoothing splines regression. The final dataset is displayed in Figure 4. The clustering method presented in Section 2 are applied with K as input varying from 2 to 7, with relative evaluations shown in Figure 5. Results indicate that no single method outperforms the others. However, K-Models v2 exhibits the weakest performance, regardless of the chosen K . In contrast, K-Models demonstrates significantly better results. A key insight is that while almost all methods perform competitively only in terms of the Variability

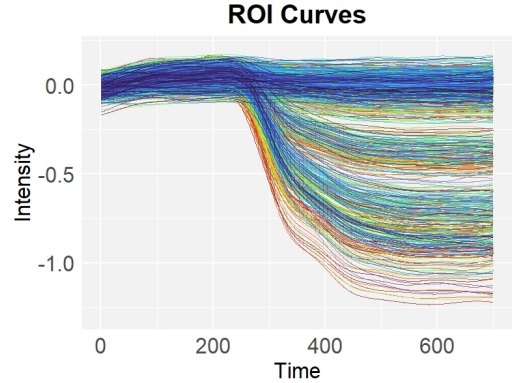


Figure 4: ROI curves.

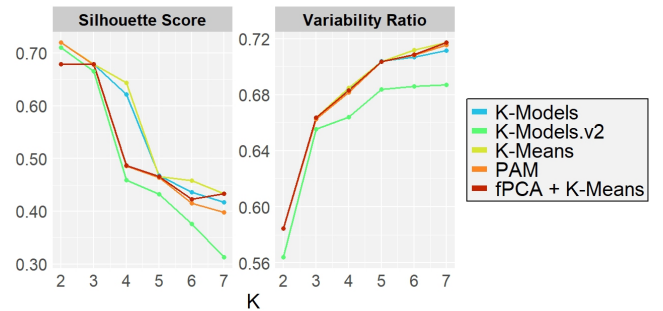


Figure 5: Evaluation of the clustering methods applied to ROI curves, for $K = 2, \dots, 7$.

Ratio, only K-Models and K-Models achieve optimal performance across both metrics. This indicates that, in this case study, our approach effectively balances two crucial aspects of clustering: maximizing the separation between groups and ensuring accurate assignments, particularly for observations near cluster boundaries. Other methods tend to compromise on at least one of these aspects, whereas K-Models achieves a more optimal trade-off. Both metrics show an inflection at $K = 5$, suggesting this as the optimal choice via the Elbow Rule. Given this fact, Figure 6 displays the estimated \hat{g} and $\hat{\beta}_k$ from our K-Models, with the corresponding cluster-specific curves. This plot clearly shows how the latent phenomenon g is modulated across clusters through the coefficients β_{1k} . The increasing trend confirms the ordinal nature of the clustering, where each group represents a different level of the common effect. Notably, $\hat{\beta}_{0k}$ remains stable across clusters, indicating similar baseline responses, while $\hat{\beta}_{1k}$ determines the variation. The first cluster ($\hat{\beta}_{11} = 0.0531$) shows a nearly flat response, suggesting minimal reagent concentration, whereas the last cluster ($\hat{\beta}_{15} = 3.2081$)

exhibits strong amplification, corresponding to the highest concentration. This progression in $\hat{\beta}_{1k}$ confirms that K-Models captures reactivity levels effectively, rather than just partitioning curves.

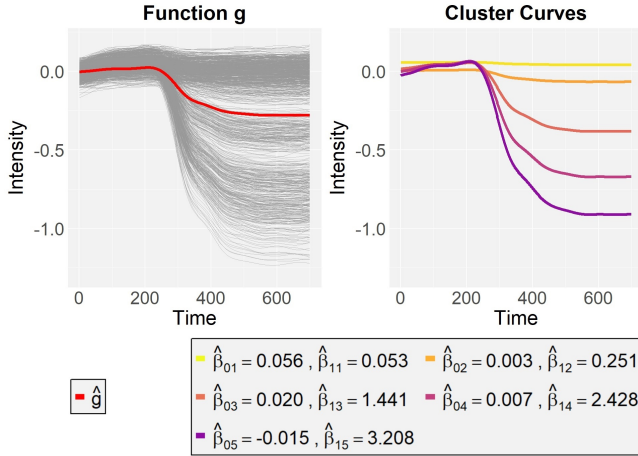


Figure 6: Results of the K-Models applied to ROI curves, with $K = 5$.

6. Discussion

This study provides a broad evaluation of the K-Models framework, highlighting its strengths and limitations relative to established functional clustering methods. K-Models v2 exhibits weaker clustering performance, likely due to its update strategy, which relies solely on estimated coefficients $\hat{\beta}_k$, disregarding important functional variations. However, it excels in identifying borderline cases, as seen in the simulated scenarios, suggesting its potential in applications where subtle cluster distinctions are crucial, and also gives robust results in recreating the true partition. Conversely, K-Models (first version) proves to be a competitive alternative to existing methods, achieving comparable or superior results while offering an additional interpretative advantage. Unlike traditional clustering techniques, K-Models estimates both the latent structure \hat{g} and cluster-specific coefficients $\hat{\beta}_k$, providing deeper insights into the data structure. This dual benefit—strong clustering performance and enhanced interpretability—makes K-Models particularly valuable in contexts where understanding the underlying process is as important as the clustering itself.

7. Conclusions

The vast array of clustering methods for functional data underscores the field’s importance within mathematical and engineering communities. However, in many contexts—such as the ROI curves analyzed in this study—existing techniques do not always provide the most suitable tools for capturing both partitioning structure and underlying data-generating mechanisms. The K-Models approach introduced in this work successfully addresses both aspects by defining a clustering framework based on a linear functional model. This allows for the estimation of a latent effect driving the observed curves and evaluates its manifestation across clusters. A key advantage of this formulation is its ability to define an ordinal partition, where clusters are naturally ranked based on the strength of the common effect—an absent feature in traditional functional clustering methods. As discussed in Section 6, we are satisfied with the clustering performance of K-Models, which demonstrates competitive results while offering enhanced interpretability. Both versions of the method exhibit distinct strengths, suggesting that future research could explore a hybrid approach, combining their respective update strategies to maximize their advantages. Additionally, extending the functional model beyond the linear setting—e.g. exponential, logarithmic, or polynomial—could enhance its flexibility and applicability to more complex real-world scenarios. In conclusion, integrating these advancements could further improve the method’s effectiveness while preserving its strong interpretative capacity, making K-Models a valuable tool for functional clustering in diverse applications.

8. Acknowledgements

This research has received funding by the European Commission under the “HORIZON-CL4-2021-DIGITALEMERGING-01 project BioProS - Biointelligent Production Sensor to Measure Viral Activity” (grant agreement no. 101070120), 2022-2026”. FN, AM, and GP acknowledge the initiative “Dipartimento di Eccellenza 2023–2027”, MUR, Italy, Dipartimento di Matematica, Politecnico di Milano.

References

- [1] Andrada E. Ivanescu, Ana-Maria Staicu, Fabian Scheipl, and Sonja Greven. Penalized function-on-function regression. *Computational Statistics*, 30(2):539–568, June 2015.
- [2] Julien Jacques and Cristian Preda. Functional data clustering: A survey. *Advances in Data Analysis and Classification*, 8:231–255, 09 2013.
- [3] Giulia Patanè, Federica Nicolussi, Alexander Krauth, Günter Gauglitz, Bianca Maria Colosimo, Luca Dede', and Alessandra Menafoglio. Functional-ordinal canonical correlation analysis with application to data from optical sensors, 2025.
- [4] Philip T Reiss, Lei Huang, and Maarten Mennes. Fast function-on-scalar regression with penalized basis expansions. *The International Journal of Biostatistics*, 6(1):Article 28, 2010. Research supported by U.S. Gov't, Non-P.H.S.
- [5] Peter Rousseeuw. Rousseeuw, p.j.: Silhouettes: A graphical aid to the interpretation and validation of cluster analysis. *comput. appl. math.* 20, 53-65. *Journal of Computational and Applied Mathematics*, 20:53–65, 11 1987.