Meta-learning Implicit Neural Representation for Sparse Time Series Functional Data Analysis

Bofan Chen
Department of Pure Mathematics and Mathematical Statistics
University of Cambridge
cbfcbf.byron@gmail.com

1 Introduction

2 Related works

2.1 Functional Data

Multivariate data versus functional data Traditional multivariate statistics take finite-dimensional vectors as the objects of study to analyze their statistical properties, such as mean vectors and covariance matrices. This approach is appropriate and easy to compute when the dimension is relatively small. However, if the dimension of the space in which the objects are analyzed is increased to infinite, for example, in L^2 space, traditional methods face certain challenges. In such functional spaces, the focus is not only on the values of vectors in a particular dimension but also on properties such as continuity, smoothness and derivatives. At the same time, as the space grows to infinite dimensions, many useful tools in multivariate statistics, such as probability density functions, will lose their meaning. It is the differences in statistical properties and people's focus between infinite-dimensional and finite-dimensional spaces, that have led to widespread attention to functional data analysis (FDA) among researchers.

Time series as a representative example of functional data Time series is a typical object that needs to be addressed in an infinite-dimensional function space. This is because time series data collected in reality often exhibit continuity, and, in certain situations, people may focus on topics such as derivatives and time alignment (registration). Many previous studies have relied on multivariate statistics in the analysis of time series, such as [mention some specific examples]... From the perspective of function approximation, these multivariate-statistics-based studies can indeed reflect certain properties of time series. However, in many cases, this approach is challenging to generalize, as is the case with sparse time series data studied in this paper.

Sparse times series data analysis When the observation frequency of time series is low and irregular, sparse time series data is formed. We can represent the sparse dataset of n samples as

$$D = \{(t_{ij}, y_{ij})_{j=1}^{J_i}\}_{i=1}^{n},$$

where $y_{ij} = x_i(t_{ij}), i = 1, \cdots, n, j = 1, \cdots, J_i$ and J_i is the total observations of the i-th sample. This type of data is particularly common in the medical field, as the frequency of medical visits varies from person to person, with irregular patterns. When it comes to analyzing such "messy" data using multivariate statistics, people often find themselves at a loss. However, in functional data analysis, one can employ the method of local polynomial regression to estimate the mean function and covariance operator of functional data. Specifically, given our sparse observation of dataset D, we can define the loss function to estimate mean function $\mu(t)$

$$L_t^{\mu}(\beta) = \sum_{i=1}^{n}\sum_{j=1}^{J_i}\left[y_{ij} - \sum_{r=0}^{R}\beta_r\,(t - t_{ij})^r\right]^2 K\left(\frac{t - t_{ij}}{h}\right),$$

Preprint. Under review.

where $K(\cdot)$ is the kernel function and h is the bandwidth which shoud be carefully chosen based on data. By minimizing this function, we can get $\beta_t = (\beta_0(t), \cdots, \beta_R(t)) = arg\,minL_t^{\mu}(\beta)$. The estimator of mean function cam be defined as $\hat{\mu}(t) = \beta_0(t)$. Similarly, to estimate the covariance kernel $c(t, s)$, we define the loss function

$$L_{t,s}^c(\beta) = \sum_{i=1}^{n} \sum_{1 \leq j,l, \leq J_i} K_{t,s}(\frac{t - t_{ij}}{h}, \frac{s - t_{il}}{h}) \left[g(t_{ij}, t_{il}) - f\left(\beta, (t, s), (t_{ij}, t_{il})\right)\right]^2$$

where

$$g(t_{ij}, t_{il}) = (y_{ij} - \hat{\mu}(t_{ij}))(y_{il} - \hat{\mu}(t_{il}))$$

$$f\left(\beta, (t, s), (t_{ij}, t_{il})\right) = \beta_0 + \beta_{11}(t - t_{ij}) + \beta_{12}(s - t_{il})$$

By minimization of $L_{t,s}^c(\beta)$, we obtain $(\beta_0(t), \beta_{11}(t), \beta_{12}(t)) = arg\,minL_{t,s}^c(\beta)$. Then the estimator of the covariance kernel is defined as $\hat{c}(t, s) = \beta_0(t)$. Based on the estimation of mean and covariance, we can use the method of Principal Analysis by Conditional Expectation (PACE) to reconstruct each sample and compute their scores (see details in the appendix). Comparing to using local polynomial regression directly to reconstruct samples, It is noticeable that the reconstruction of the sample makes use of the total of information of all observations, but it hugely relies on the assumption that scores are normally distributed. In the following text, I will compare the estimates obtained based on this method as the baseline with the meta-learning-based approach proposed in this paper.

2.2 Implicit Neural Representation

Traditional grid-based representation versus implicit neural representation Traditionally, the representation of functional data involves using a standard grid-based method, wherein a finite set of coordinates and corresponding function values are given as pairs (t_i, y_i). However, this approach has certain drawbacks: the original functional data in an infinite-dimensional space is discretized into a finite-dimensional space. This not only hampers operations like differentiation but also significantly impacts the resolution of the function within a smaller time interval. With the emergence of deep learning, it has been discovered that utilizing deep neural networks can be an effective way to represent functions implicitly. Implicit neural representation (INR) parameterize the functional data by deep neural networks. This type of continuous representation not only proves to be more memory-efficient but also enables higher resolution and the handling of more complex operations on functional data.

SIREN architecture for time series Many prior works have utilized ReLU-based multilayer perceptrons (MLP) as the neural network architecture for Implicit Neural Representations (INR). However, ReLU-based INR often lack the ability to capture fine details in signal and struggle to effectively express higher-order derivatives. In addressing this limitation, [1] introduced the SIREN architecture as an alternative to traditional ReLU-MLPs. It employs the sine function as the activation function, effectively capturing intricate details in functions. A remarkable observation is that even after taking derivatives of a function represented using SIREN, it can still be expressed using the SIREN architecture. While previous research predominantly applied SIREN to represent images and videos, its application in time series representation has been limited. This paper explores the utilization of SIREN in the analysis of time series, revealing that its distinctive properties confer unique advantages in representing temporal sequences.

SIREN for time series: advantage...

2.3 Meta-learning

Meta-learning framework

Two different types different architecture: MAML / SNAIL

3 Method

advantage

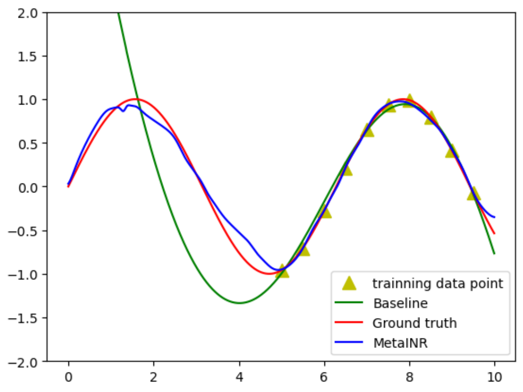

Figure 1: Learning periodicity

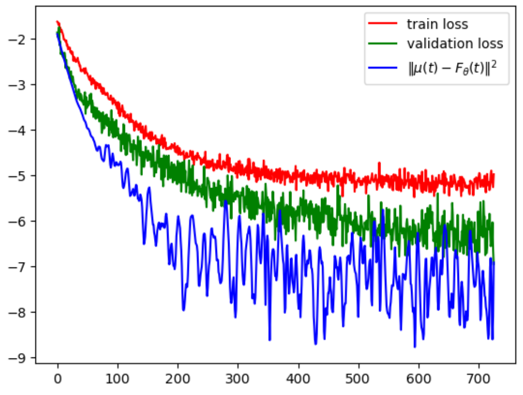

Figure 2: Learning mean

4 Experiments

4.1 Synthetic data

4.1.1 what meta-learning learned in timeseries

global+differential property lead to a better PC estimation

1. periodicity
2. differential equations: No?
3. average phenomenon for meta prior model
4. based on information from whole dataset to get a estimation

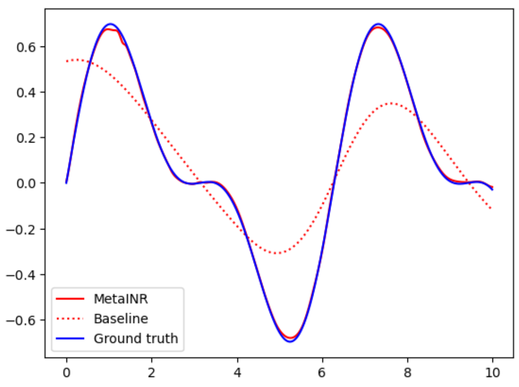

Figure 3: Mean estimation

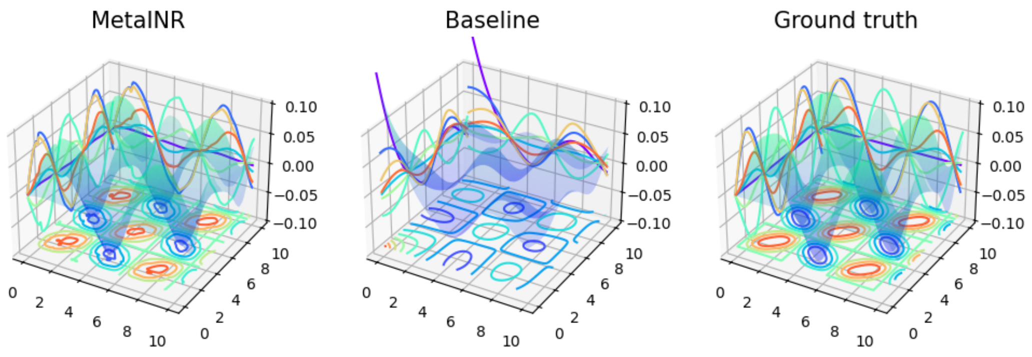

Figure 4: Covariance estimation

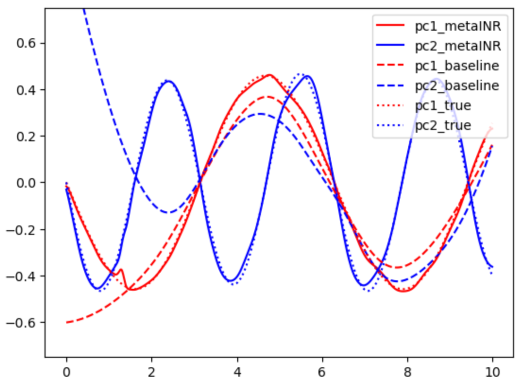

Figure 5: FPCA

4.1.2 Mean estimation

4.1.3 Covariance estimation

4.1.4 fPCA

4.2 Real-world data

5 Conclusion

6 Appendix

6.1 Traditional FDA for sparse functional data

6.2 Brief introduciton to SIREN

References

References

[1] V. Sitzmann, J. Martel, A. Bergman, D. Lindell, and G. Wetzstein, “Implicit neural representations with periodic activation functions,” *Advances in neural information processing systems*, vol. 33, pp. 7462–7473, 2020.