Kernel Representation Learning with Dynamic Regime Discovery for Time Series Forecasting

Kunpeng Xu[1(✉)], Lifei Chen[2(✉)], Jean-Marc Patenaude[3], and Shengrui Wang[1(✉)]

[1] Université de Sherbrooke, Sherbrooke, QC, Canada
{kunpeng.xu,shengrui.wang}@usherbrooke.ca
[2] Fujian Normal University, Fuzhou, Fujian, China
clfei@fjnu.edu.cn
[3] Laplace Insights, Sherbrooke, QC, Canada
jeanmarc@laplaceinsights.com

Abstract. Correlations between variables in complex ecosystems such as weather and financial markets lead to a great amount of dynamic and co-evolving time series data, posing a significant challenge to the current forecast methods. Discovering dynamic patterns (aka regimes) is crucial to an accurate forecast, especially for the interpretability of the outcome. In this paper, we develop a kernel-based method to learn effective representations for capturing dynamically changing regimes. Each such representation accounts for the non-linear interactions among multiple time series, thereby facilitating more effective regime discovery. On the basis of regime information, we build a regression model to forecast all the variables simultaneously for the next multiple time points. The results on six real-life datasets demonstrate that our method can yield the most accurate forecast (with the lowest root mean square error) in comparison with seven predictive models.

Keywords: time series forecasting · kernel · self-representation learning

1 Introduction

Time-series forecasting is an important topic that continuously attracts a great deal of interest in a myriad of areas such as finance, medicine, meteorology, ecology, sociology, and many industrial sectors. In real applications, time series often comprise numerous short segments, each recurring within the series. These segments generally correspond to particular regimes/patterns in dynamically changing environments – *e.g.*, on the volatile financial market [6,13], stock prices might decline during wartime and subsequently rise with the onset of peace talks. Discovering and leveraging these underlying regimes has become an essential research topic for generating accurate and interpretable time-series forecasts.

To capture the dynamic behaviors of time series, several machine learning methods have been developed to explore the regimes for time series forecasting, *e.g.*, RSVAR [8] for health management, WCPD-RS [4] in financial,

D.-N. Yang et al. (Eds.): PAKDD 2024, LNAI 14650, pp. 251–263, 2024.
https://doi.org/10.1007/978-981-97-2266-2_20

and ObritMap [11] in IoT/sensor streams analysis. These models suggest that the presence of structural discontinuities in time series leads to a regime shift, wherein each regime represents distinct behaviors that reveal the underlying dynamics throughout time. In general, these models first analyze the overall regime shifts present in the time series data and subsequently employ the derived models to forecast future regimes. However, certain characteristics of time series can be hard to capture when time series exhibit nonlinearity, mixing, or noise. Furthermore, the requirement to predefine the number of regimes in many models, such as the Markov-based switching model [4], limits their flexibility in dynamically inferring and estimating regimes from data.

Another significant challenge arises from the complexity inherent in identifying regimes within multiple time series forecasting tasks [14]. This difficulty primarily stems from the interdependence and co-evolution of the time series – *e.g.*, to forecast the traffic for a particular road, it is necessary to consider the impact of traffic on adjacent roads; similarly, the fluctuating user engagement with music streaming services-evidenced by the decline in Pandora's click rates and the simultaneous surge in Spotify's from 2012 to 2022-suggests competitive dynamics, with Spotify seemingly attracting Pandora's user base. Exploring the interrelationships between series at different time intervals is crucial for regime identification and prediction.

To address these challenges, we propose a novel approach for multiple time series forecasting, emphasizing modeling of their evolving interactions and regime identification. Our method redefines regime identification from the perspective of self representation learning and transforms the challenge into a subspace clustering problem. This transformation allows for a more nuanced and granular analysis of multiple time series data, leading to a more precise and interpretable forecasting model. Our method has the following desirable properties:

(1) **Adaptive:** Automatically identify and handle regimes (patterns) exhibited by multiple time series, without prior knowledge about regimes.
(2) **Interpretability:** Convert heavy sets of time series into a lighter and meaningful structure through kernel representation, depicting the continuous regime shift mechanism over multiple time series in nonlinear space.
(3) **Effective:** Operate on multiple time series, explore the nonlinear interactions, and forecast the future values within an ecosystem consisting of multiple time series.

2 Related Work

Traditional time series models, such as general state-space models [3], including ARIMA and exponential smoothing, excel at modeling the complex dynamics of individual time series with sufficiently long histories. Although these methods are widely used in time series forecasting due to their simplicity and interpretability, they are local in the sense that one model is learned for each time series. Consequently, they cannot effectively extract information across multiple time series.

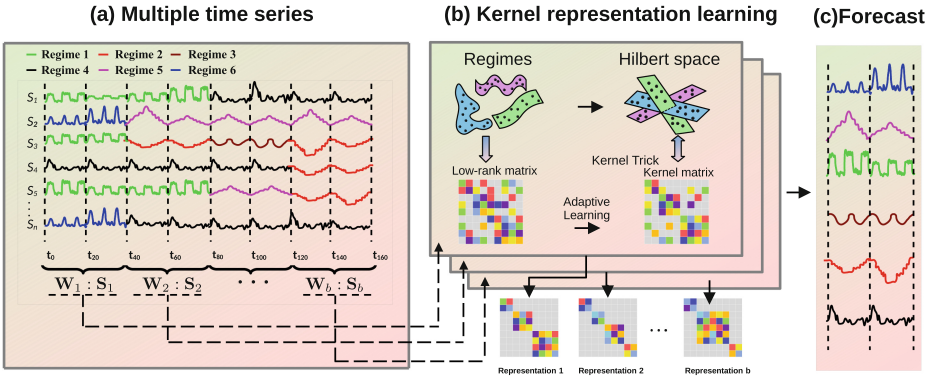

Fig. 1. Framework of our method. Given multiple co-evolving time series composed of various regimes in fixed-length windows, our method learns the kernel representation of these regimes for forecasting the next regimes.

Hochstein et al. [8] developed a multivariate smooth transition autoregression model to show how different time series are linearly dependent on each other. This model uses a vector autoregressive model for each regime. It is worth noting that this type of method attempts to capture the regime shift mechanism through a single transfer matrix, which unfortunately may be time-dependent for series that exhibit noncontiguous regimes. Matsubara et al. proposed the RegimeCast model [10], which learns the various patterns that may exist in a co-evolving environment at a given window and reports the pattern(s) most likely to be observed at a subsequent time. While the approach can report subsequent patterns, it does not capture possible dependencies between patterns. In their subsequent work [11], the authors introduced the deterministic OrbitMap model, designed to capture time-dependent transitions between exhibited regimes. However, their model relies on regimes that are labeled in advance. Recent research has demonstrated significant advancements in time series analysis through the use of deep neural networks [15,19–21]. However, the majority of these studies primarily focus on modeling and forecasting individual time series, often overlooking the interactions among multiple time series.

3 Preliminaries

3.1 Key Concepts

Time Series. A time series is a set of points ordered by a time index as follows: $S_i = \{(t_l, e_l^i)\}_{l=1}^m$, where t_l are regular time stamps, m the series length and e_l^i the series value at the specific time t_l. Here, the index i refers to the i-th time series in a set of N univariate time series $\mathbf{S} = \{S_i\}_{i=1}^N$.

Regime. In this paper, a regime is defined as the profile pattern of a group of similar subseries observed within a window instance. The term "profile pattern" refers to a subseries whose vector representation is the centroid of the similar subseries. Our model permits highly similar, or repetitive, patterns to occur across subseries at different windows. This repetition enables the identification of similar regimes at various window instances, facilitating effective regime tracking.

3.2 Self-representation Learning in Time Series

Time series often exhibit patterns that recur over time. One feasible way to capture these inherent patterns is through self-representation learning, a concept derived from subspace clustering [18]. This approach represents each data point in a series as a linear combination of others, formulated as $\mathbf{S} = \mathbf{SZ}$ or $S_i = \sum_j S_j Z_{ij}$, where $\mathbf{Z}$ is the self-representation coefficient matrix. In multiple time series, high Z_{ij} values indicate similar behaviors or regimes between S_i and S_j. The learning objective function is:

$$\min_{\mathbf{Z}} \frac{1}{2}||\mathbf{S} - \mathbf{SZ}||^2 + \Omega(\mathbf{Z}), \ s.t. \ \mathbf{Z} = \mathbf{Z}^{\mathrm{T}} \geq 0, \mathrm{diag}(\mathbf{Z}) = 0 \tag{1}$$

where $\Omega(\cdot)$ is a regularization term on $\mathbf{Z}$. The ideal representation $\mathbf{Z}$ should group data points with similar patterns, represented as block diagonals in $\mathbf{Z}$, each block signifying a specific regime. The number of blocks, k, corresponds to the distinct regimes.

The optimization of this problem can take various forms, influenced by the choice of $\Omega(\mathbf{Z})$. If $\Omega(\mathbf{Z}) = ||\mathbf{Z}||_1$, it results in classical Sparse Subspace Clustering [7]. Different norms for $\mathbf{Z}$ lead to various models like efficient dense subspace clustering (EDSC), the Frobenius norm in least-squares Regression (LSR) and the nuclear norm in Low-Rank Representation (LRR).

3.3 Kernel Trick for Modeling Time Series

Linear models in Euclidean space often struggle with capturing nonlinear relationships in multiple time series [16]. Kernelization techniques address this challenge by mapping data into higher, and in some cases, infinite-dimensional Hilbert spaces using suitable kernel functions [17]. This facilitates the identification of linear patterns within these transformed spaces. The process is facilitated by the "kernel trick", which employs a nonlinear feature mapping, $\Phi(\mathbf{S})$: $\mathcal{R}^d \rightarrow \mathcal{H}$, to project data $\mathbf{S}$ into a kernel Hilbert space $\mathcal{H}$. Direct knowledge of the transformation Φ is not required; instead, a kernel Gram matrix $\boldsymbol{\mathcal{K}} = \Phi(\mathbf{S})^\top \Phi(\mathbf{S})$ is used. The Gaussian kernel, which results in an infinitely dimensional feature space $\mathcal{H}$, is notably prevalent in this context.

4 Proposed Method

For the sake of clarity, consider the set of time series depicted by Fig 1(a). We start by introducing our kernel representation learning. This process entails

searching for homogeneous patterns via self-representation learning with a block diagonal regularizer in kernel space. With a given sliding window of size w, we can split time series into contiguous subseries of length w, *i.e.*, $\mathbf{S} = \bigcup_{p=1}^{b} \mathbf{S}_p$ and get the number of distinct regimes k by counting the number of distinct profile patterns across all window-stamps, $W_1, ..., W_b$. Based on the discovered regimes, we will be able to predict the regime switch and series values.

4.1 Kernel Representation Learning: Modeling Regime Behavior

In our approach, we circumvent the obstacle of discovering regimes for multiple time series, by solving a self-representation learning problem. This approach allows us to effectively cluster subseries, retrieved using a sliding window technique, into distinct regimes. We begin with the simplest case, where we treat the whole series as a single window.

Given a set of time series $\mathbf{S} = (S_1, \ldots, S_N) \in \mathcal{R}^{T \times N}$ as described in Eq. (1), its self-representation $\mathbf{Z}$ would make inner product $\mathbf{SZ}$ come close to $\mathbf{S}$ if we adopted the linear approach. However, the objective (1) falls short in capturing the nonlinear relationships between series. To address this issue, the time series can be mapped, by "kernel tricks", into a high-dimensional RKHS, where a linear pattern analysis will be performed. By integrating the kernel mapping, we present a new kernel representation learning strategy (as shown in Fig 1(b), with the following objective function:

$$\min_{\mathbf{Z}} ||\Phi(\mathbf{S}) - \Phi(\mathbf{S})\mathbf{Z}||^2, \ s.t. \ \mathbf{Z} = \mathbf{Z}^{\mathrm{T}} \geq 0, \mathrm{diag}(\mathbf{Z}) = 0 \tag{2}$$

Here, the mapping function $\Phi(\cdot)$ need not be explicitly identified and is typically replaced by a kernel $\boldsymbol{\mathcal{K}}$ subject to $\boldsymbol{\mathcal{K}} = \Phi(\cdot)^{\top}\Phi(\cdot)$.

Ideally, we hope to achieve the matrix $\mathbf{Z}$ having k block diagonals under some proper permutations if $\mathbf{S}$ contains k regimes. To this end, we introduce a regularization term to $\mathbf{Z}$ and transform Eq. (2) to:

$$\begin{gathered} \min_{\mathbf{Z}} ||\Phi(\mathbf{S}) - \Phi(\mathbf{S})\mathbf{Z}||^2 + \gamma \sum_{i=N-k+1}^{N} \lambda_i(\mathbf{L}_{\mathbf{Z}}), \\ s.t. \ \mathbf{Z} = \mathbf{Z}^{\mathrm{T}} \geq 0, \mathrm{diag}(\mathbf{Z}) = 0 \end{gathered} \tag{3}$$

where $\gamma > 0$ defines the trade-off between the loss function and regularization terms, and $\lambda_i(\mathbf{L}_{\mathbf{Z}})$ contains the eigenvalues of Laplacian matrix $\mathbf{L}_{\mathbf{Z}}$ corresponding to $\mathbf{Z}$ in decreasing order. Here, the regularization term is equal to 0 if and only if $\mathbf{Z}$ is k-block diagonal (see Theorem 1 for details). Based on the learned high-quality matrix $\mathbf{Z}$ (containing the block diagonal structure), we can easily group the time series into k regimes using traditional spectral clustering technology [7].

Theorem 1. $\min \sum_{i=N-k+1}^{N} \lambda_i(\mathbf{L}_{\mathbf{Z}})$ *is equivalent to* $\mathbf{Z}$ *is* k*-block diagonal.*

Proof. Due to the fact that $\mathbf{Z} = \mathbf{Z}^{\mathrm{T}} \geq 0$, the corresponding Laplacian matrix $\mathbf{L_Z}$ is positive semidefinite, and thus $\lambda_i(\mathbf{L_Z}) \geq 0$ for all i. The optimal solution of $\min \sum_{i=N-k+1}^{N} \lambda_i(\mathbf{L_Z})$ is that all elements of $\lambda_i(\mathbf{L_Z})$ are equal to 0, which means that the k smallest eigenvalues are 0. Combined with the Laplacian matrix property, the multiplicity k of the eigenvalue 0 of the corresponding Laplacian matrix $\mathbf{L_Z}$ equals the number of connected components (blocks) in $\mathbf{Z}$, and thus the soundness of Theorem 1 has been proved.

Optimization. The problem (3) can be solved by the ALM with Alternating Direction Minimization strategy. Normally, the representation matrix $\mathbf{Z}$ in Eq. (3) needs to be nonnegative and symmetric, which are necessary properties for defining the block diagonal regularizer. However, these restrictions on $\mathbf{Z}$ will limit its representation capability. For this reason, we propose a modified model by introducing an intermediate-term $\mathbf{C}$:

$$\begin{aligned} \min_{\mathbf{Z},\mathbf{C}} \frac{1}{2}||\Phi(\mathbf{S}) - \Phi(\mathbf{S})\mathbf{C}||^2 + \frac{\beta}{2}||\mathbf{C} - \mathbf{Z}||^2 + \gamma \sum_{i=N-k+1}^{N} \lambda_i(\mathbf{L_Z}) \\ s.t.\ \mathbf{Z} = \mathbf{Z}^{\mathrm{T}} \geq 0, \mathrm{diag}(\mathbf{Z}) = 0 \end{aligned} \tag{4}$$

The above two models (Eq. (3) and Eq. (4)) are equivalent when $\beta > 0$ is sufficiently large. As will be seen in optimization, a benefit of the relaxation term $||\mathbf{C} - \mathbf{Z}||^2$ is that it makes the objective function separable. More importantly, the subproblems for updating $\mathbf{Z}$ and $\mathbf{C}$ are strongly convex, leading to final solutions that are unique and stable.

Note that $\sum_{i=N-k+1}^{N} \lambda_i(\mathbf{L_Z})$ is a nonconvex term, and by introducing the property about the sum of eigenvalues, we reformulate it as $\min_{\mathbf{W}} < \mathbf{L_Z}, \mathbf{W} >$, where $0 \preceq \mathbf{W} \preceq \mathbf{I}, \mathrm{Tr}(\mathbf{W}) = k$. So Eq. (4) is equivalent to

$$\begin{aligned} &\min_{\mathbf{Z},\mathbf{C},\mathbf{W}} \frac{1}{2}||\Phi(\mathbf{S}) - \Phi(\mathbf{S})\mathbf{C}||^2 + \frac{\beta}{2}||\mathbf{C} - \mathbf{Z}||^2 + \gamma < \mathrm{Diag}(\mathbf{Z1}) - \mathbf{Z}, \mathbf{W} > \\ &s.t.\ \mathbf{Z} = \mathbf{Z}^{\mathrm{T}} \geq 0, \mathrm{diag}(\mathbf{Z}) = 0, 0 \preceq \mathbf{W} \preceq \mathbf{I},\ Tr(\mathbf{W}) = k \end{aligned} \tag{5}$$

The optimization of Eq. (5) involves alternating updates of $\mathbf{W}$, $\mathbf{C}$, and $\mathbf{Z}$. Each subproblem is convex, allowing for closed-form solutions:

Updating W:

$$\mathbf{W}^{i+1} = \arg\min_{\mathbf{W}} < \mathrm{Diag}(\mathbf{Z1}) - \mathbf{Z}, \mathbf{W} >,\ s.t.\ 0 \preceq \mathbf{W} \preceq \mathbf{I}, \mathrm{Tr}(\mathbf{W}) = k \tag{6}$$

Updating C:

$$\mathbf{C}^{i+1} = (\boldsymbol{\mathcal{K}} + \beta\mathbf{I})^{-1}(\boldsymbol{\mathcal{K}} + \beta\mathbf{Z}) \tag{7}$$

Updating Z:

$$\mathbf{Z}^{i+1} = [(\hat{\mathbf{A}} + \hat{\mathbf{A}}^{\mathrm{T}})]_+, \text{where } \hat{\mathbf{A}} = \mathbf{A} - \mathrm{Diag}(\mathrm{diag}(\mathbf{A})), \mathbf{A} = \mathbf{C} - \frac{\gamma}{\beta}(\mathrm{diag}(\mathbf{W})\mathbf{1}^{\mathrm{T}} - \mathbf{W}) \tag{8}$$

4.2 Forecasting

We consider all windows to be known except the last one, for which we want to predict the series values. For b sliding windows $\{W_1, \ldots, W_b\}$, we obtain b kernel representations, each corresponding to a window. It is important to know that the regime R_i discovered from the subseries $\mathbf{S}_p$ within the $p^{th}(p \in [1, b])$ window might not be discovered (*i.e.*, there may be no series exhibiting this regime) in other subseries (*i.e.*, from other windows). This reveals the variety of regimes in time series and the demand for a dynamic representation. We predict the kernel representation for subseries in the next window W_{b+1} via a regression model λ, as follows:

$$\mathbf{Z}_{b+1} = \lambda(\mathbf{Z}_1, ..., \mathbf{Z}_b) + \mu_{b+1}, \tag{9}$$

where μ_{b+1} is the white Gaussian noise for reducing overfitting. Then, we forecast the value of the time series for the next window W_{b+1} based on the self-representation property:

$$\mathbf{S}_{b+1} = \hat{\mathbf{S}}_{b+1}\mathbf{Z}_{b+1} \tag{10}$$

Table 1. Data statistics

Data	# of series	Length of series
`Music`	20	219
`Electricity`	370	1,462
`Chlorine`	166	3,480
`Earthquake`	139	512
`Electrooculography`	362	1,250
`Rock`	50	2,844

In this case, we employ the same regression form to estimate $\hat{\mathbf{S}}_{b+1}$, *i.e.*, $\hat{\mathbf{S}}_{b+1} = \lambda(\mathbf{S}_1, ..., \mathbf{S}_b) + \mu_{b+1}$.

5 Experiments

5.1 Data

We collected six real-life datasets from various areas. The `Music` dataset from GoogleTrend event stream[1] contains 20 time series, each for the Google queries on a music-player spanning 219 months from 2004 to 2022. The `Electricity` dataset comprises 1462 daily electricity load diagrams for 370 clients, extracted from UCI[2]. From the UCR's public repository[3], we obtained four time-series datasets – *i.e.*, `Chlorine` concentration, `Earthquake`, `Electrooculography` signal, and `Rock`. Table 1 summarizes the statistics of the datasets.

[1] http://www.google.com/trends/.
[2] https://archive.ics.uci.edu/ml/datasets/.
[3] https://www.cs.ucr.edu/%7Eeamonn/time_series_data_2018.

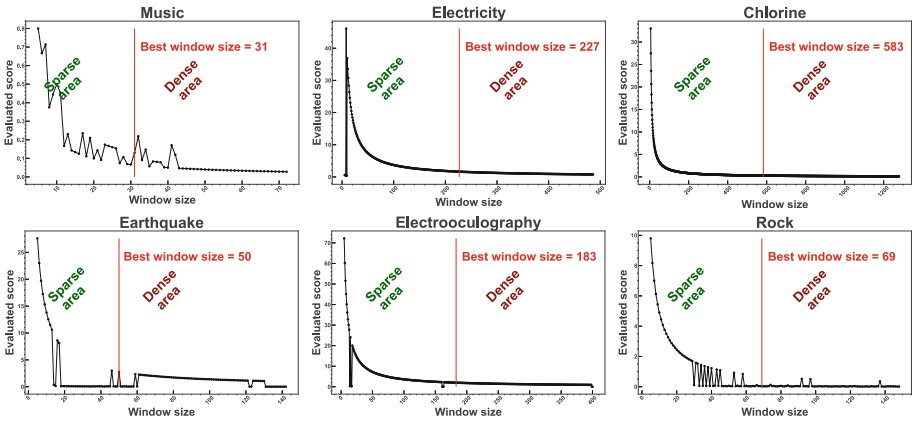

Fig. 2. The best window size (red line) for the six data sets (Color figure online)

5.2 Experimental Setup and Evaluation

In the kernel representation learning process, we used the Gaussian kernel of the form $\mathcal{K}(S_i, S_j) = exp(-||S_i - S_j||^2/d^2_{max})$, where d_{max} is the maximal distance between series. Parameters γ in Eq. (3) is selected over [0.1,0.4,0.8,1,4,10] and set to be $\gamma = 0.8$ for the best performance.

We evaluate the forecasting performance of the proposed model against seven different models. Among them, four are forecasting models (ARIMA [3], KNNR [5], INFORMER [21], and a state-of-the-art ensemble model N-BEATS [12]), the other three are RS models (MSGARCH [1], SD-Markov [2] and OrBitMap [11]).

5.3 Regime Identification

In this subsection, we evaluate the capability of our model to identify regimes. During the learning process, a fixed window slides over all the series, generating subseries under different windows. We then learn a kernel representation for the subseries in each window. The quality of our kernel representation depends highly on how well the time series is split. Due to space limitations, our method for automatically estimating the optimal size of the sliding windows to obtain suitable regimes using the Minimum Description Length (MDL) technique can be found in Supplementary. Figure 2 exhibits the selected window sizes for the respective datasets: the length of 31 (resp., 227, 583, 50, 183, 69) window used for the `Music` (resp., `Electricity, Chlorine, Earthquake, Electrooculography, Rock`) data.

With these window sizes, we can plot the profile pattern to visualize the regime. In Fig. 3, each row displays the distinct regimes exhibited by co-evolving series in the six datasets, respectively. We discovered 4 different regimes in the `Music` time series, 3 in `Electricity`, 4 in `Chlorine`, 3 in `Earthquake`, 3 in `Electrooculography`, and 5 in `Rock`. For these real cases, we lack the ground

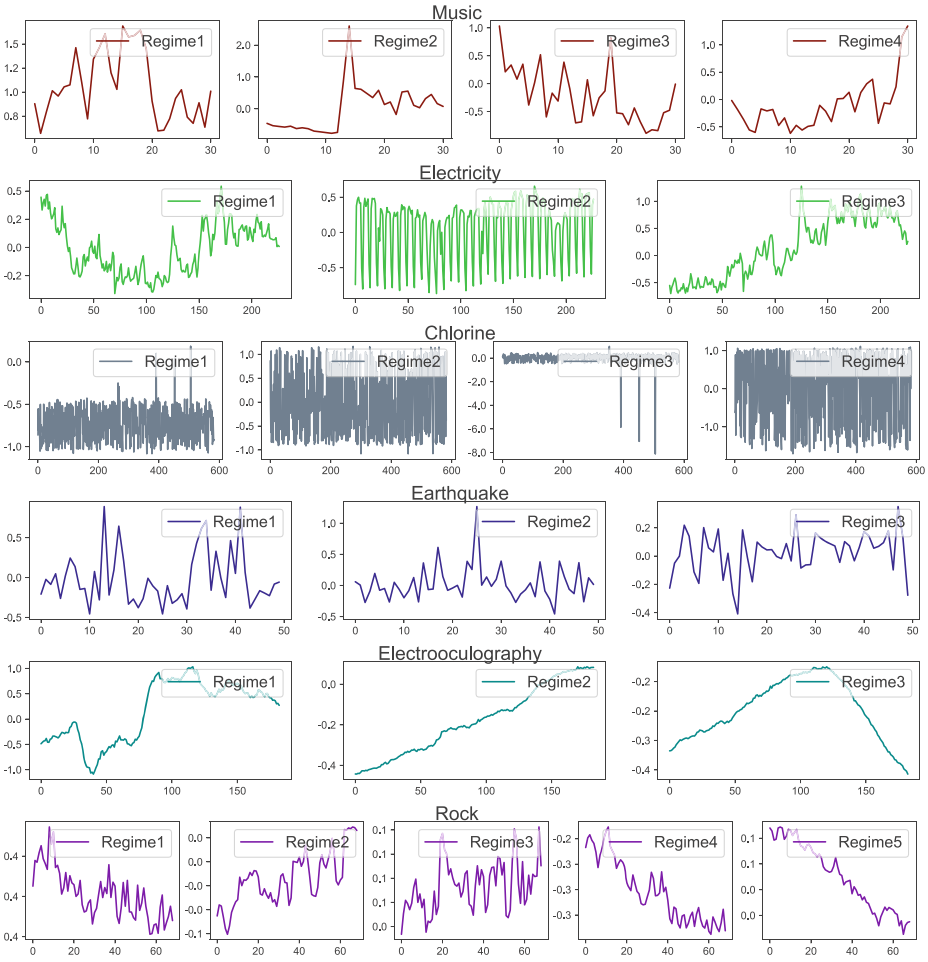

Fig. 3. Discovered regimes for the six real datasets

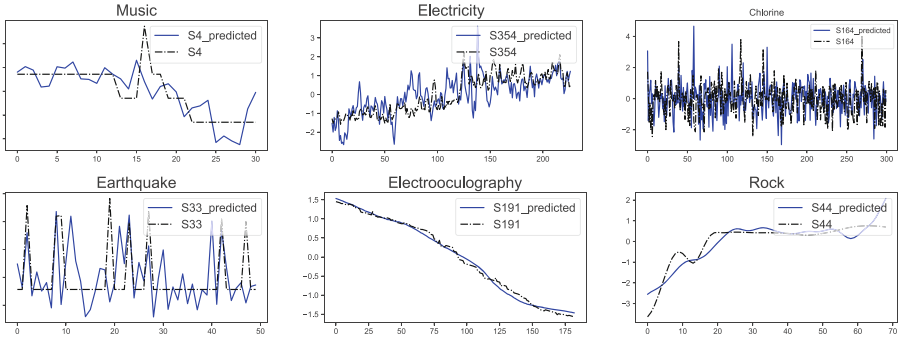

Fig. 4. True (black) and forecasted (blue) values for the six time series, each from a real dataset. (Color figure online)

truth for validating the obtained regimes. Fortunately, according to the forecasting which depends on the identified regimes, we will be able to better validate whether the identified regimes are the right ones.

Figure 4 illustrates the forecasted outcomes for six arbitrarily selected time series from the respective datasets, offering a demonstrative insight into the notable efficacy of our model in forecasting time series. It is important to note that this illustration is intended to showcase the proficient results achieved via our regime-based forecasting, a detailed evaluation of the forecasting ability will be presented in Sect. 5.4.

Table 2. Models' forecasting performance, in terms of RMSE, for the nine datasets

Models		Forecasting models			
		ARIMA	KNNR	INFORMER	N-BEATS
Datasets	`Music`	6.571	4.021	2.562	0.956
	`Electricity`	2.458	2.683	2.735	**1.593**
	`Chlorine`	8.361	6.831	3.746	1.692
	`Earthquake`	5.271	3.874	4.326	1.681
	`Electrooculography`	3.561	3.452	4.562	2.487
	`Rock`	6.836	6.043	5.682	2.854
Models		RS models			
		MSGARCH	SD-Markov	OrbitMap	Ours
Datasets	`Music`	2.641	3.234	1.244	**0.663**
	`Electricity`	2.425	2.439	1.835	1.644
	`Chlorine`	5.712	3.462	1.753	**1.387**
	`Earthquake`	4.213	3.573	**1.386**	1.392
	`Electrooculography`	3.566	3.571	3.251	**1.198**
	`Rock`	5.924	4.587	4.571	**1.699**

5.4 Benchmark Comparison

In this subsection, we evaluate the forecasting performance of our proposed model against seven different models, utilizing the Root Mean Square Error (RMSE) as an evaluative metric. Table 2 shows the forecasting performance of the models. We see that our model consistently outperforms the other models, achieving the lowest forecasting error on all datasets (except for the `Earthquake`, because of the weak correlation between the time series). ARIMA has the ability to capture seasonality patterns within time series; however, when the various seasonalities are noncontiguous, the models face difficulties in capturing complex, nonlinear dynamic interactions between time series. Notably, N-BEATS, the state-of-the-art deep network model, is generally the second-best performer owing to its ensemble-based strengths. However, it falls short in capturing complex regime transitions within multiple time series, revealing the limitations of

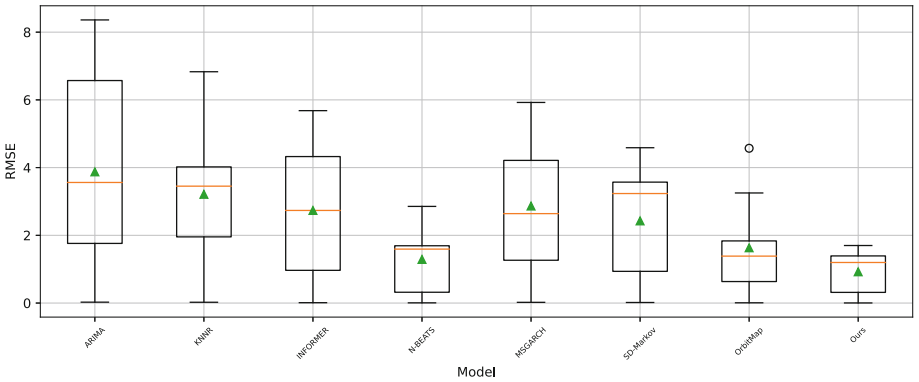

Fig. 5. Box Plot of RMSE Values for Each Model Across Datasets

a model geared solely for single time series forecasting. Meanwhile, OrbitMap, while also regime-aware, is hindered by its necessity for predefined regimes and struggles with handling multiple time series. Figure 5 illustrates the distribution of RMSE values for each model across all datasets, and it is evident that our model achieves the most favourable outcomes overall.

5.5 Ablation Study

We conduct ablation experiments to validate the efficacy of our model's kernel representation learning. We focus particularly on the regularization and kernelization techniques employed in (3). Our approach is compared against existing self-representation learning methods[9] - SSC, LSR, LRR, BDR, EDSC, SSQP, and SSCE, under varying values of γ. The comparative results are illustrated

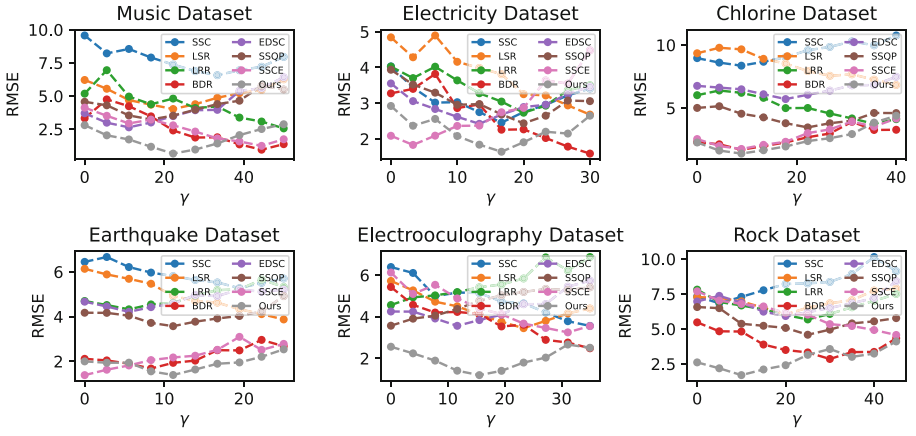

Fig. 6. RMSE across datasets for different regularizations at varying γ

in Fig. 6. This comparison clearly demonstrates that our model achieves superior performance, outperforming the other methods in terms of RMSE across different datasets for a range of γ values.

6 Conclusion

This paper introduces a new approach for modeling non-linear interactions in an ecosystem comprising multiple time series. This approach enhances time series forecasting for subsequent periods, thanks to a notable ability which is its capacity to identify and handle multiple time series dominated by various regimes. This is accomplished by devising a kernel representation learning method, from which the time-varying kernel representation matrices and the block-diagonal property are utilized to determine regime shifts. Furthermore, our model automatically uncovers various hidden regimes without requiring any prior knowledge about the series under investigation. Validation with real-world datasets has shown that our model surpasses existing models in terms of forecast accuracy.

References

1. Ardia, D., Bluteau, K., Boudt, K., Catania, L., Trottier, D.A.: Markov-switching garch models in r: the MSGARCH package. J. Stat. Softw. (2019)
2. Bazzi, M., Blasques, F., Koopman, S.J., Lucas, A.: Time-varying transition probabilities for Markov regime switching models. J. Time Series Anal. **38**, 458–478 (2017)
3. Box, G.: Box and jenkins: time series analysis, forecasting and control. In: A Very British Affair, pp. 161–215. Springer, Heidelberg (2013). https://doi.org/10.1057/9781137291264_6
4. Chatigny, P., Chen, R., Patenaude, J.M., Wang, S.: A variable-order regime switching model to identify significant patterns in financial markets. In: ICDM, pp. 887–892. IEEE (2018)
5. Chen, R., Paschalidis, I.: Selecting optimal decisions via distributionally robust nearest-neighbor regression. Adv. Neural Inf. Process. Syst. **32** (2019)
6. Dong, X., Dang, B., Zang, H., Li, S., Ma, D.: The prediction trend of enterprise financial risk based on machine learning arima model. J. Theory Pract. Eng. Sci. **4**(01), 65–71 (2024)
7. Elhamifar, E., Vidal, R.: Sparse subspace clustering: algorithm, theory, and applications. IEEE TPAMI **35**(11), 2765–2781 (2013)
8. Hochstein, A., Ahn, H.I., Leung, Y.T., Denesuk, M.: Switching vector autoregressive models with higher-order regime dynamics application to prognostics and health management. In: PHM, pp. 1–10. IEEE (2014)
9. Lu, C., Feng, J., Lin, Z., Mei, T., Yan, S.: Subspace clustering by block diagonal representation. IEEE TPAMI **41**(2), 487–501 (2018)
10. Matsubara, Y., Sakurai, Y.: Regime shifts in streams: real-time forecasting of co-evolving time sequences. In: ACM SIGKDD, pp. 1045–1054 (2016)
11. Matsubara, Y., Sakurai, Y.: Dynamic modeling and forecasting of time-evolving data streams. In: ACM SIGKDD, pp. 458–468 (2019)

12. Oreshkin, B.N., Carpov, D., Chapados, N., Bengio, Y.: N-beats: Neural basis expansion analysis for interpretable time series forecasting. arXiv preprint arXiv:1905.10437 (2019)
13. Qiao, Y., Jin, J., Ni, F., Yu, J., Chen, W.: Application of machine learning in financial risk early warning and regional prevention and control: a systematic analysis based on shap. World Trends Realities Accompany. Prob. Dev. **331** (2023)
14. Song, X., et al.: Zeroprompt: streaming acoustic encoders are zero-shot masked lms. arXiv preprint arXiv:2305.10649 (2023)
15. Xu, C., Yu, J., Chen, W., Xiong, J.: Deep learning in photovoltaic power generation forecasting: Cnn-lstm hybrid neural network exploration and research. In: The 3rd International scientific and practical conference "Technologies in education in schools and universities", Athens, Greece, 23–26 January 2024, vol. 363, p. 295. International Science Group (2024)
16. Xu, K., Chen, L., Wang, S.: Data-driven kernel subspace clustering with local manifold preservation. In: 2022 IEEE International Conference on Data Mining Workshops (ICDMW), pp. 876–884. IEEE (2022)
17. Xu, K., Chen, L., Wang, S.: A multi-view kernel clustering framework for categorical sequences. Expert Syst. Appl. 116637 (2022)
18. Xu, K., Chen, L., Wang, S., Wang, B.: A self-representation model for robust clustering of categorical sequences. In: U, L.H., Xie, H. (eds.) APWeb-WAIM 2018. LNCS, vol. 11268, pp. 13–23. Springer, Cham (2018). https://doi.org/10.1007/978-3-030-01298-4_2
19. Ye, J., et al.: Multiplexed oam beams classification via fourier optical convolutional neural network. In: 2023 IEEE Photonics Conference (IPC), pp. 1–2. IEEE (2023)
20. Ye, J., Solyanik, M., Hu, Z., Dalir, H., Nouri, B.M., Sorger, V.J.: Free-space optical multiplexed orbital angular momentum beam identification system using fourier optical convolutional layer based on 4f system. In: Complex Light and Optical Forces XVII, vol. 12436, pp. 70–80. SPIE (2023)
21. Zhou, H., et al.: Informer: beyond efficient transformer for long sequence time-series forecasting. In: Proceedings of the AAAI Conference on Artificial Intelligence, vol. 35, pp. 11106–11115 (2021)