Representation Learning of Time Series Data with High-Level Semantic Features

Chengyang YE † Qiang MA ‡

† Graduate School of Informatics, Kyoto University Yoshida-Tachibana-Cho, Sakyo-Ku, Kyoto 606-8303, JAPAN
E-mail: † ye@db.soc.i.kyoto-u.ac.jp, ‡ qiang@i.kyoto-u.ac.jp

Abstract This study proposes a representation method of time-series data by considering its high-level semantic features to help build interpretable neural networks. This method constructs human cognition features as specific high-level semantic information to tackle the issue that cognitions are challenging to apply in representation learning of time series data. Our approach can improve the performance of traditional networks and make it possible to use time-series data in a variety of tasks.

Keyword Representation Learning, Time Series, High-Level Semantic Features, Neural Network

1. Introduction

Time series analysis is not a new study, and we have been using time series analysis for thousands of years. Time series data, such as daily prices of stocks [1], health care [2], and weather condition [3], are introductory classes of temporal data and have been widely used in a lot of applications.

This brings new challenges to the discovery of knowledge from big time series data. For example, in the stock market, time series analysis requires experienced and competent analysts to analyze the market changes and behavioral logic implied behind the extensive, complex, and detailed market data. There is widespread interest in providing efficient analysis using unlabeled or finitely labeled time series data. Consequently, Analysis of time series data is a quite important tool that could help us to understand how time series data works in our daily life. Based on this, several methods are proposed for different purpose. Such as indexing, classifying, clustering, and summarizing time series data [4, 5].

In this paper, we propose to study time series from a new angle. Our goal is to understand the hidden patterns between the time series data. Thus, instead of classify time series data to different groups or predicting the next time series value based on the pattern in the most recent time window, we focus on discovering the high-level semantic in time series data which could help us to understand the hidden connection between different subsequences of time series.

Our approach includes the following three subroutines. First, we identify the change points of different states in time series and create segmentation of time series data. It will help us to distinguish the different high-level semantic hidden in time series data. Then for each segmentation, intrinsic connections between different segmentation are learned. And it can be used as the encoding vector of high-level semantic features to join the input of the representation learning model or be directly used in the specific tasks as the extracted features. The details are shown in Fig.1.

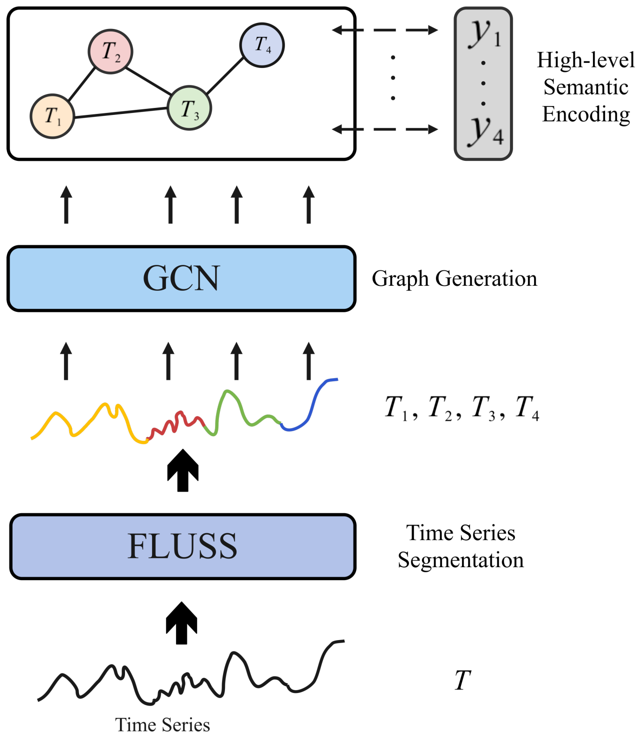

Fig. 1. Structure of representation learning of high-level semantic features.

A. State-of-the-art Solutions

Much work has been done on time series analysis, including time series prediction [6], time series classification [7], and similar time series matching [8]. Representation learning of time series data has become a popular research topic. Most models aim to discover the Spatio-temporal dependencies in the data. Time2Graph [9] begins from Shapelet [10], which can automatically mine time-series subsequences with representative features, and constructs graphs for representation learning by analyzing the direct relationship between different shapelets.

Additionally, contrast learning has been introduced into this aspect of time series analysis [11]. Unsupervised representation learning of unlabeled time-series data is achieved by constructing positive and negative data pairs. On this basis, triplet loss is further combined with a CNN with dilation [12] to tackle long time-series data. This approach is fairly easy to implement and only requires distinguishing the main features.

The time series transformer (TST) model [13] is a recently proposed representation learning model for multivariate time series. This model essentially fills the gap in applying the transformer model to the representation learning of time series. This model achieves better learning performance than supervised training methods by introducing a transformer-based pre-training model.

However, not much attempt has been made to use the semantic features, especially the high-level semantic features of time series data, to understand the hidden patterns between the time series data. The high-level semantic will help us build interpretable neural networks and improve the performance of representation learning of time series data.

B. High-level Semantic Features

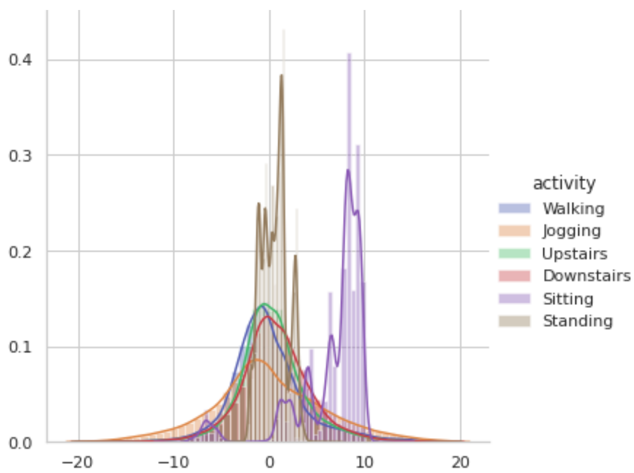

Fig. 2. Time series data of different human activities.

The high-level semantic feature is a fundamental notion in Computer Vision (CV). High-level semantic features are considered the prominent features for the images [14]. In other words, high-level semantic features can represent an image with the help of object details. These features also provide certain semantic information of the objects in images. Different hierarchical layers provide various types of features in deep learning.

There is a similar concept in time series data. Real world time series also have high-level semantic. Fig.2 shows a typical example of time-series series data captured from a human activities system include a set of human movements, e.g., sitting, standing, and walking. We can refer to specific patient movements as high-level semantic in the time series. Specifically, given a time series T, the goal is to find homogeneous segments, and then cluster similar segments to reveal high-level semantic features of T. The understanding of the high-level semantic features offers us a holistic insight into the time series. These features can be widely used in other time series mining and analysis applications. In this paper, we use high-level semantic features to join the representation learning of time series data to reveal the hidden patterns between the time series data.

The rest of this paper is organized as follows. We introduce some related works in Section 2 and elaborate our techniques in Section 3. And Section 4 shows experimental results. Finally, we conclude this paper and delineate the directions of future work in Section 5.

2. Related work

Some traditional and basic mathematical methods and models, such as ARIMA, have been used for many time series tasks, which is concerned with the value changes of time series X_t given observations $X_1, X_2, \cdots, X_{t-1}$. These approaches are designed with a frequent assumption that the time series has a certain memory with the past data, which means current values are only related to its neighboring node values. This results in that these approaches only focus on local information of time series data, and do not attempt to explain observations using the hidden pattern between time series data. Although many works focus on discovering certain patterns or regularities in time series [15], these patterns may not necessarily be regarded as important patterns, in terms of whether they can inform us of the latent semantic states. Many mining algorithms discover many patterns that are hard to interpret, which adds to the complexity of understanding the system instead of helping reduce it.

Therefore, finding high-level semantic features of time series data (i.e., latent semantic states) is one of the core subroutines in current time series mining applications. However, this task is not trivial. It has two challenges: (1) how to determine the correct change points between consecutive segments and cluster segments into different states (2) how to embed high-level features in time series data for representation learning. Fortunately, there are several related works focus on these two issues respectively.

A. Identification of different states

The first step of discovering high-level semantic features of time series data is identification of different states. There are three representative approaches under this issue.

pHMM [16] recognize different states by lines with different slopes and lengths. Although this is a quite simple but effective approach, this is not available to time series whose states are not linear.

AutoPlait [17] distinguishes different states by the mean and standard deviation of data values in each state. This type of approach makes full use of the statistical characteristics of the time series data. However, AutoPlait fails to work in situation when these states have similar mean and standard deviation values.

GRAB [18] exploits the similarity between subsequences. That is, they define the state as repeated subsequences, and the segments which have multiple similar subsequences are identical state. GRAB is motivated by FLUSS [19], which infers boundaries (or change points) between different states based on the intuition that the most similar subsequence pairs tend to belong to the same state. FLUSS produces a companion time series called Arc Curve (AC). Each value of AC is the number of arcs that connect the two most similar subsequences and spatially cross this point. Intuitively, most subsequences will find their nearest neighbor within the corresponding segment. Thus the "valley" points in AC are likely to be the locations of boundaries between different states.

However, GRAB was proposed to identify different states in time series data. The goal of this work is to use the high-level semantic features in representation learning of time series data. Nevertheless, GRAB offers an adaption solution by enriching GRAB with an embedding approach.

B. Embedding models for time series

Recently, some algorithms have been proposed with certain specific properties of time-series data, thus enabling the construction of model inputs using the relevant properties described by the time-series data. Signal2Vec [20] was the first to use the core idea of Word2Vec [21] to embed continuous-type data. First, Signal2Vec discretizes the continuous data. Next, the discretized data are processed by the skip-gram model to vectorize the input time-series data. Subsequently, the constructed embedding vectors are used for clustering the time-series data.

Multiple time-series data tasks require manually designed features, such as time windows and lag operators, which plausibly harm the usefulness and efficiency of models for time-series data. Time2Vec [22] was proposed to transform dynamic data into dimensional static embedding vectors and reduce the effort required for data pre-processing. This allows Time2Vec to capture the periodic features of time-series data. However, owing to the lack of a corresponding pre-training task, the embedding vectors learned by Time2Vec are easily overfitted to the training set, thus resulting in poor performance on the test set. In addition, this approach is not suitable for time-series data with insignificant periods, such as stock price data.

In addition, as the research on graph structure has intensified, some studies [23] have also been devoted to enhancing the effectiveness of existing models by applying graph mining methods using graphs constructed from the target time-series data. The earliest research using this approach can be traced back to the study on complex networks in 2008 [24]. The latest research [25] captures both, local and global features of time-series data by constructing sub-graphs to provide a more comprehensive representation. However, the significance of this embedding method has only been proven for conventional models, while there is no relevant existing research on deep learning models.

Based on these embedding models, high-level semantic features can also be regarded as the embedding vectors (like the part of position encoding in Transformer architecture) and be used in representation learning.

3. Methodology

To find high-level semantic features of time series, we propose a Frequency Decomposition-Based approach. It includes the following three subroutines. Fig.1 illustrates the details.

Time Series Segmentation: For a given time series T, we use FLUSS to identify change points of different states in time series and create segmentation of time series data.

FLUSS takes both a time series T and a user provided subsequence length in as input, and outputs an AC vector of length *n*, where at each index i contains the number of "arcs" that cross over *i*. We define an "arc" as follows. The i^{th} entry in the vector contains a positive integer *j*, which indicates the nearest neighbor location. So, for the i^{th} entry, containing a positive integer *j*, the nearest neighbor for the time series subsequence beginning at index i is the time series subsequence beginning at index *j*. We can visualize each entry pair (*i,j*) as an arc drawn from location i to *j*. The spatial layout of the arcs along with the number of "arc" crossing over of each index i is summarized by the Arc Curve. Specifically, index i of the Arc Curve contains a non-negative integer indicating the number of arcs that cross over *i*.

Graph Generation: After the identification of change points of different states in time series, we would create several segmentations of time series data. And each segmentation could represent a certain state of time series. Motivated by GRAB, we construct a graph representation of the original time series where each vertex corresponds to a segmentation. The weight of each edge is designed to reveal the similarity between each segmentation and others. It can also be regarded as the representation of the likelihood that two segmentations are under the same hidden state. With such graph generation, our approach could construct the specific graph structure of each time series. And because of the ability to establish the hidden relationship between different segmentation, the high-level semantic features of each time series could also be revealed.

Concretely, based on the time series segmentation, we construct an undirected weighted graph $G = (V, E)$, where each vertex corresponds to a segmentation and the weight of edge connecting two segmentations is the similarity between them.

A time series $T = (t_1, t_2, \cdots, t_n)$ is an ordered sequence of real-valued numbers $t_i \in \mathbb{R}, n = |T|$ is the length of T. A subsequence $T_{i,l} = (t_i, t_{i+1}, \cdots, t_{i+l-1})(1 \leq in - l + 1)$ denotes the continuous sequence of length l starting from the i^{th} position in T.

For each segmentation S_m, we define a vector ω_n. The length of ω_n is m, the number of segmentation subsequences. The i^{th} entry in ω_n indicates the similarity between S_m and S_i. The similarity could be calculated as the Arc Curve number in FLUSS algorithm.

High-level Semantic Encoding: Based on the graph generated in last step, a new issue is how to extract the graph features of time series data. The most straightforward method is Graph Convolutional Networks (GCN) [26].

Recently, many papers focused on generalizing neural networks to work on arbitrarily structured graphs [27-31]. Some methods of them, like graph-based and kernel-based techniques, lead the hot topic and excited achievements in such domain.

Graph Convolutional Networks (GCN) [26] is a widely used architecture for most graph neural network models. Motivated by Convolutional Neural Network (CNN), GCN is specially designed for graph structure. For these models, the goal is then to learn a function of signals or features on a graph $G = (V, E)$, which takes as input: A feature description x_i for every node i; summarized in a $N \times D$ feature matrix X, where N represent number of nodes, and D is number of input features. And a representative description of the graph structure in matrix form; typically, in the form of an adjacency matrix A (or some function thereof).

Based on this description, every neural network layer can then be written as a non-linear function.

$$H^{(l+1)} = f\left(H^{(l)}, A\right) \tag{1}$$

where $H^{(0)} = X$ and $H^{(L)} = Z$. L being the number of layers. The specific models then differ only in how$f(\cdot)$ is chosen and parameterized.

In this paper, we chose GCN to extract the features of generated graph of time series. And the output of GCN can be regarded as the high-level semantic features used in many tasks of time series data.

After we calculate the features of generated graph based on time series, it can be used as the part of the input of the TST model, like position encoding in Transformer architecture. Different segmentations with high-level semantic features will have a different distribution of frequency decomposition. In this way, we can add high-level semantic features to input vectors of TST. On the other hand, it can also be used directly as the extracted features in tasks of time series data like classification and.

4. Experiment

In this section, we tested the effectiveness of our model on the UEA&UCR dataset [32], using the classification task as a downstream task.

TABLE I. A SUMMARY OF UEA&UCR DATASETS

Dataset	Train Seize	Test Seize	Length	Classes	Type
Birdchicken	20	20	512	2	IMAGE
CBF	30	900	128	3	SIMULATED
ECG200	100	100	96	2	ECG
Car	60	60	577	4	SENSOR
Earthquakes	322	139	512	2	SENSOR

We selected a set of five univariate datasets from the UEA&UCR time series classification archives, which provides multiple datasets from different domains, with a varying number of dimensions, unequal length dimensions, and missing values. Meanwhile, they also provide an initial benchmark for the existing models. This provides accurate baseline information.

In 2002, the UCR time series classification archive was first released with sixteen datasets. It gradually expanded, until 2015 when it increased in size from 45 datasets to 85 datasets. In October 2018 more datasets were added, bringing the total to 128 [32]. The new archive contains a wide range of problems, including variable length series. One of the motivations for introducing the archive was to encourage researchers to perform a more rigorous evaluation of newly proposed time series classification (TSC) algorithms. It has worked: most recent research into TSC uses all 85 datasets to evaluate algorithmic advances [33]. The newest update version is 2018 version, which consists of 30 datasets with a wide range of cases, dimensions and series lengths. And for the first iteration of the archive, they format all data to be of equal length, include no series with missing data and provide train and test splits. In conclusion, UEA&UCR dataset is a significant dataset for research of analysis of time series time data.

And in order to demonstrate the performance of our approach directly, we chose a simple SoftMax model [34] as a classifier to classify the extracted features of GCN. Based on the performance metrics provided by the UEA&UCR archives, we chose the SoftMax classifier with and without our high-level semantic feature extract algorithm as experimental subjects.

TABLE II. ACCRUACY ON CLASSIFICATION DATASETS

Dataset	GCN-SoftMax	SoftMax
Birdchicken	0.90	0.80
CBF	0.80	0.80
ECG200	0.75	0.71
Car	0.70	0.67
Earthquakes	0.65	0.50

We selected a set of five univariate datasets from the UEA&UCR time series classification archives, which provides multiple datasets from different domains, with a varying number of dimensions, unequal length dimensions, and missing values. Meanwhile, they also provide an initial benchmark for the existing models. This provides accurate baseline information. The summary of these datasets can be found in Table Ⅰ.

The results of classification task are shown in Table Ⅱ. As shown in Table Ⅱ, demonstrated the better performance on three out of the five datasets. Our approach can work better in the dataset with small train size (Birdchicken and Car), And dataset with specific jump change (Earchquakes). This result shows that the GCN based algorithm can increase the performance of model. And high-level semantic features do have a positive impact in time series data tasks.

5. Future work

In this paper, we have proposed an approach to add high-level semantic features to representation learning of time series data. As the algorithm suggests, we construct a distribution of high-level semantic features and add these features to the time series representation learning model.

Future research should focus on more experiments of this approach and certainly further test the feasibility and scalability of our proposed approach.

Acknowledgements

This work is partly supported by KAKENHI (19H04116).

References

[1] Rezaei H, Faaljou H, Mansourfar G. Stock price prediction using deep learning and frequency decomposition[J]. Expert Systems with Applications, 2021, 169: 114332.

[2] Hicks S A, Isaksen J L, Thambawita V, et al. Explaining deep neural networks for knowledge discovery in electrocardiogram analysis[J]. Scientific reports, 2021, 11(1): 1-11.

[3] Chou J S, Truong D N. Multistep energy consumption forecasting by metaheuristic optimization of time‐series analysis and machine learning[J]. International Journal of Energy Research, 2021, 45(3): 4581-4612.

[4] Witten I H, Frank E, Hall M A, et al. Practical

machine learning tools and techniques[C]//DATA MINING. 2005, 2: 4.

[5] Keogh E. A decade of progress in indexing and mining large time series databases[C]//Proceedings of the 32nd international conference on Very large data bases. 2006: 1268-1268.

[6] Gu X, Wang H. Online anomaly prediction for robust cluster systems[C]//2009 IEEE 25th International Conference on Data Engineering. IEEE, 2009: 1000-1011.

[7] Wang P, Wang H, Liu M, et al. An algorithmic approach to event summarization[C]//Proceedings of the 2010 ACM SIGMOD International Conference on Management of data. 2010: 183-194.

[8] Takbiri N, Goeckel D L, Houmansadr A, et al. Asymptotic limits of privacy in Bayesian time series matching[C]//2019 53rd Annual Conference on Information Sciences and Systems (CISS). IEEE, 2019: 1-6.

[9] Z. Cheng, Y. Yang, W. Wang, W. Hu, Y. Zhuang and G. Song, "Time2Graph: Revisiting Time Series Modeling with Dynamic Shapelets", Proceedings of the AAAI Conference on Artificial Intelligence, vol. 34, no. 04, pp. 3617-3624, 2020. Available: 10.1609/aaai.v34i04.5769.

[10] C. Ji et al., "A fast shapelet selection algorithm for time series classification", Computer Networks, vol. 148, pp. 231-240, 2019. Available: 10.1016/j.comnet.2018.11.031.

[11] J. Aren et al., "Unsupervised learning of semantic audio representations", in 2018 IEEE international conference on acoustics, speech and signal processing (ICASSP), IEEE, 2018, pp. 126-130.

[12] F. Jean-Yves, D. Aymeric and J. Jaggi, "Unsupervised scalable representation learning for multivariate time series", arXiv preprint arXiv:1901.10738, 2019.

[13] Z. George, J. Srideepika, P. Dhaval, B. Anuradha and E. Carsten, "A transformer-based framework for multivariate time series representation learning", in Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery ¥& Data Mining, 2021, pp. 2114-2124.

[14] X.Cheng,J.Lu,J.Feng,B.Yuan,andJ.Zhou,‘‘Scenereco gnitionwith objectness,’’ Pattern Recognit., vol. 74, pp. 474–487, Feb. 2018.

[15] Mueen A, Keogh E, Bigdely-Shamlo N. Finding time series motifs in disk-resident data[C]//2009 Ninth IEEE International Conference on Data Mining. IEEE, 2009: 367-376.

[16] Wang P, Wang H, Wang W. Finding semantics in time series[C]//Proceedings of the 2011 ACM SIGMOD International Conference on Management of data. 2011: 385-396.

[17] Matsubara Y, Sakurai Y, Faloutsos C. Autoplait: Automatic mining of co-evolving time sequences[C]//Proceedings of the 2014 ACM SIGMOD international conference on Management of data. 2014: 193-204.

[18] Lu Y, Wang P, Tang B, et al. GRAB: Finding Time Series Natural Structures via A Novel Graph-based Scheme[C]//2021 IEEE 37th International Conference on Data Engineering (ICDE). IEEE, 2021: 2267-2272.

[19] Gharghabi S, Ding Y, Yeh C C M, et al. Matrix profile VIII: domain agnostic online semantic segmentation at superhuman performance levels[C]//2017 IEEE international conference on data mining (ICDM). IEEE, 2017: 117-126.

[20] N. Christoforos and V. Dimitris, "Signal2vec: time-series embedding representation", in International Conference on Engineering Applications of Neural Networks, Springer, 2019, pp. 80-90.

[21] J. Bhatta, D. Shrestha, S. Nepal, S. Pandey and S. Koirala, "Efficient Estimation of Nepali Word Representations in Vector Space", Journal of Innovations in Engineering Education, vol. 3, no. 1, pp. 71-77, 2020. Available: 10.3126/jiee.v3i1.34327.

[22] K. Seyed Mehran et al., "Time2vec: Learning a vector representation of time", arXiv preprint arXiv:1907.05321, 2019.

[23] P. Boniol and T. Palpanas, "Series2Graph", Proceedings of the VLDB Endowment, vol. 13, no. 12, pp. 1821-1834, 2020. Available: 10.14778/3407790.3407792.

[24] L. Lacasa, B. Luque, F. Ballesteros, J. Luque and J. Nuño, "From time series to complex networks: The visibility graph", Proceedings of the National Academy of Sciences, vol. 105, no. 13, pp. 4972-4975, 2008. Available: 10.1073/pnas.0709247105.

[25] M. Zheng, S. Domanskyi, C. Piermarocchi and G. Mias, "Visibility graph based temporal community detection with applications in biological time series", Scientific Reports, vol. 11, no. 1, pp. 1-12, 2021. Available: 10.1038/s41598-021-84838-x.

[26] Kipf T N, Welling M. Semi-supervised classification with graph convolutional networks[J]. arXiv preprint arXiv:1609.02907, 2016.

[27] Bruna J, Zaremba W, Szlam A, et al. Spectral networks and deep locally connected networks on graphs[C]//2nd International Conference on Learning Representations, ICLR 2014. 2014.

[28] Henaff M, Bruna J, LeCun Y. Deep convolutional networks on graph-structured data[J]. arXiv preprint arXiv:1506.05163, 2015.

[29] Duvenaud D K, Maclaurin D, Iparraguirre J, et al. Convolutional networks on graphs for learning molecular fingerprints[J]. Advances in neural information processing systems, 2015, 28.

[30] Li Y, Zemel R, Brockschmidt M, et al. Gated Graph Sequence Neural Networks[C]//Proceedings of ICLR'16. 2016.

[31] Defferrard M, Bresson X, Vandergheynst P. Convolutional neural networks on graphs with fast localized spectral filtering[J]. Advances in neural information processing systems, 2016, 29.

[32] B. Anthony et al., "The UEA multivariate time series classification archive, 2018", arXiv preprint arXiv:1811.00075, 2018. [Accessed 30 November 2021.

[33] Bagnall A, Lines J, Bostrom A, et al. The great time series classification bake off: a review and experimental evaluation of recent algorithmic advances[J]. Data mining and knowledge discovery, 2017, 31(3): 606-660.

[34] Gold S, Rangarajan A. Softmax to softassign: Neural network algorithms for combinatorial optimization[J]. Journal of Artificial Neural Networks, 1996, 2(4): 381-399.