Predicting Temporal Sets with Deep Neural Networks

Le Yu[1], Leilei Sun[1*], Bowen Du[1], Chuanren Liu[2], Hui Xiong[3], Weifeng Lv[1]
[1]SKLSDE and BDBC Lab, Beihang University, Beijing 100083, China
[2]Department of Business Analytics and Statistics, University of Tennessee, Knoxville, USA
[3]Department of Management Science and Information Systems, Rutgers University, USA
Contact: {yule,leileisun}@buaa.edu.cn Lab: https://www.brilliantasus.com/

Abstract

We propose an integrated solution based on the deep neural networks for temporal sets prediction. A unique perspective of our approach is to learn element relationship by constructing set-level co-occurrence graph and then perform graph convolutions on the dynamic relationship graphs. Moreover, we design an attention-based module to adaptively learn the temporal dependency of elements and sets. Finally, we provide a gated updating mechanism to find the hidden shared patterns in different sequences and fuse both static and dynamic information to improve the prediction performance.

Introduction

Three types of temporal data:

- Time Series: a sequence of numerical values.
- Temporal Event: a sequence of nominal events.
- Temporal Sets: a sequence of sets with timestamps, where each set contains an arbitrary number of elements.

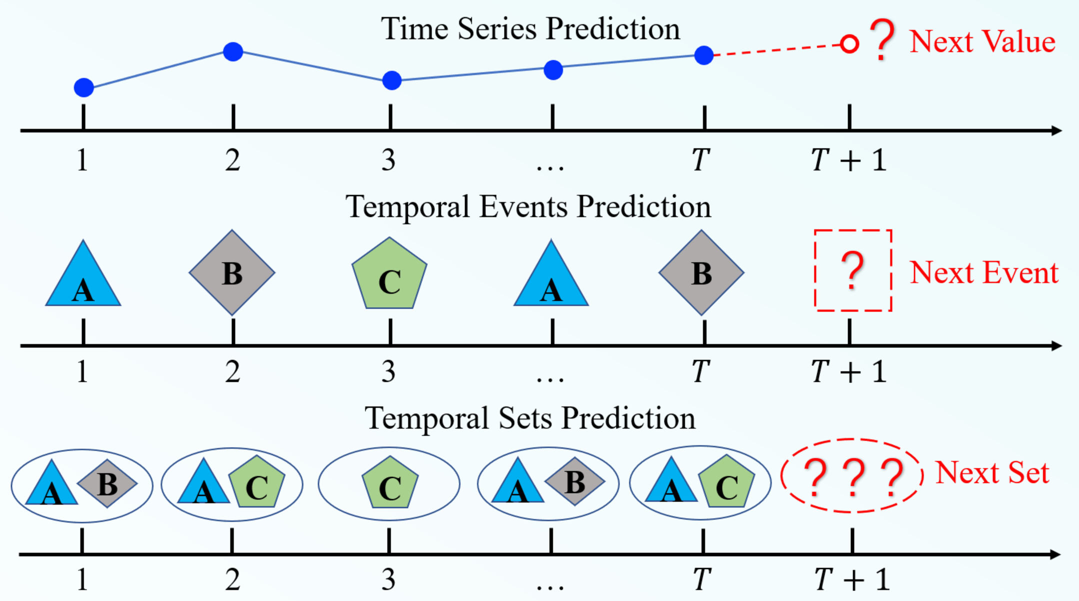

Figure 1: Prediction of three types of temporal data: time series, temporal events and temporal sets.

For temporal sets prediction:

- Methods designed for time series can not handle semantic relationships among elements.
- Methods designed for temporal events prediction cannot deal with multiple elements within a set.

Hence, it is necessary to design a dedicated method for predicting temporal sets.

Related Work

Recent literature for temporal sets prediction usually follow a two-step strategy:

1) Set Embedding.
2) Sequential Behaviors Learning.

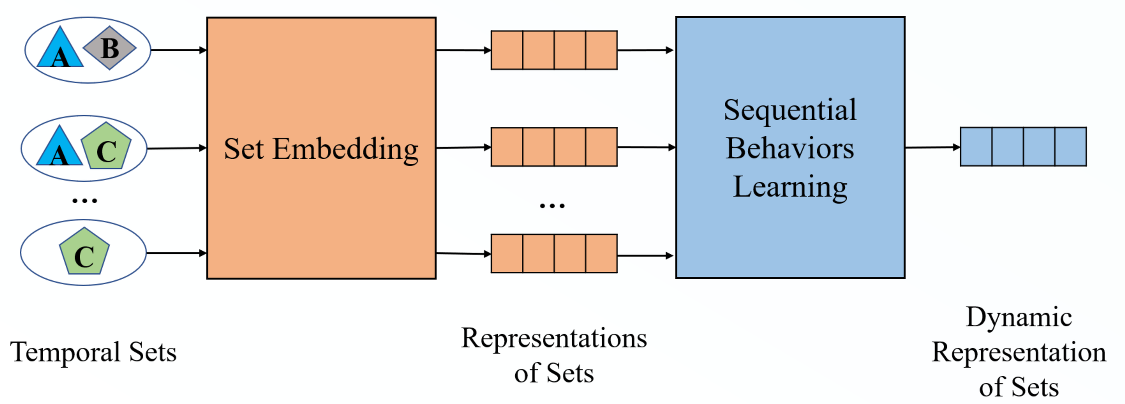

Figure 2: The two-step strategy that existing methods adopt.

The two-step strategy would lead to information loss, which results in unsatisfactory prediction performance.

Methodology

Problem Formalization

Let $\mathbb{U} = \{u_1, u_2, \cdots, u_n\}$, $\mathbb{V} = \{v_1, v_2, \cdots, v_m\}$ denote the set of n users and m elements, a set $S \subset \mathbb{V}$ denotes the collection of elements. Given a sequence of sets $\mathbb{S}_i = \{S_i^1, S_i^2, \cdots, S_i^T\}$ that records the historical behaviors of user $u_i \in \mathbb{U}$. The goal is to predict the next-period set of u_i,

$$\hat{S}_i^{T+1} = f(S_i^1, S_i^2, \cdots, S_i^T, \boldsymbol{W}),$$

where $\boldsymbol{W}$ is the trainable parameter.

Framework

The proposed model consists of three components:

1) Element relationship learning: learn set-level element relationship.
2) Attention-based temporal dependency learning: learn temporal dependency of each element in different sets.
3) Gated information fusing: fuse static and dynamic representations together.

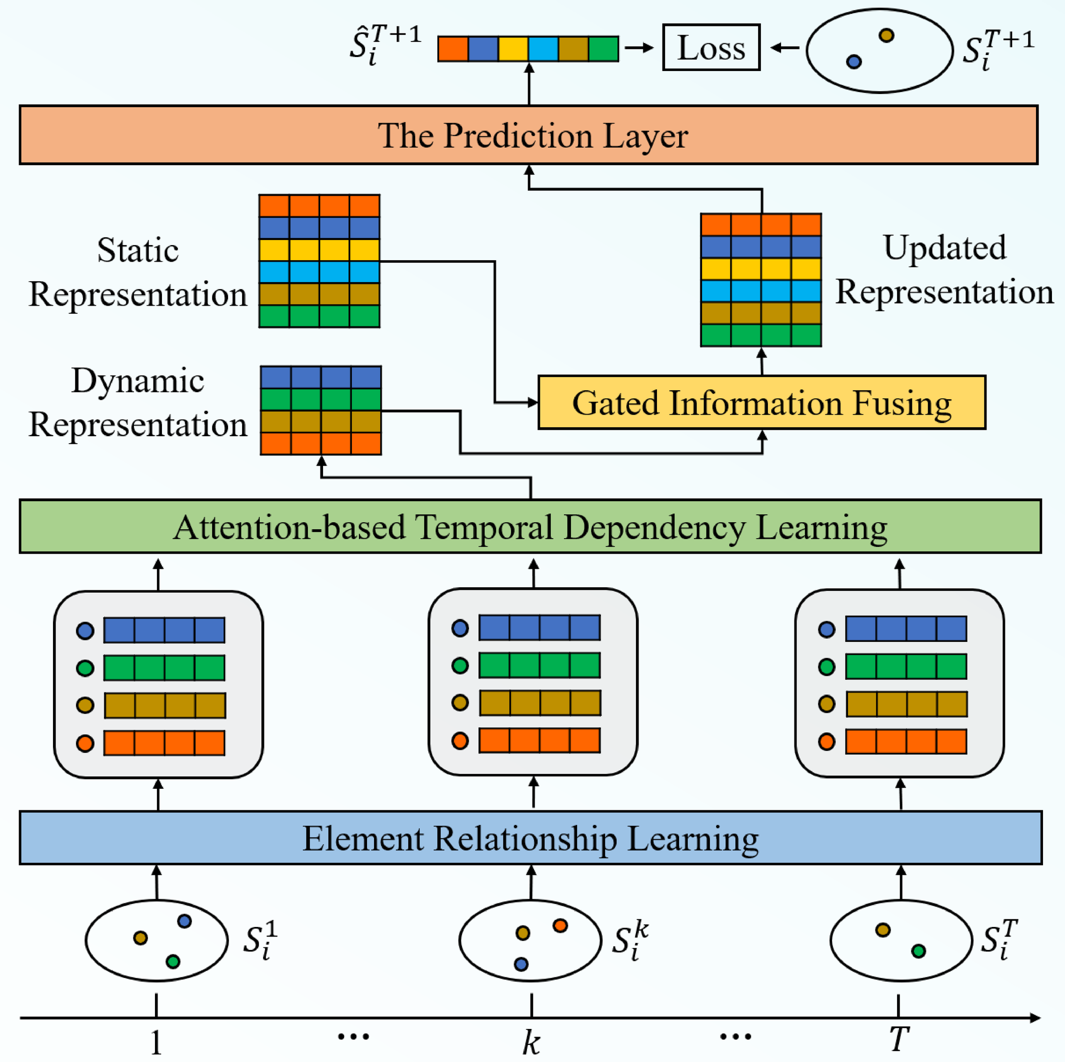

Figure 3: Framework of the proposed model.

By focusing more on element relationship, the integrated architecture could leverage useful information of elements as much as possible, which alleviates the information loss issue in existing methods.

Element Relationship Learning

- Weighted Graphs Construction:

a) Generate pairs of elements.
b) Get unique pairs and add self-connection.
c) Normalization.
d) Construct weighted graphs and assign representations.

- Weighted Convolutions on Dynamic Graphs with Parameter Sharing:

$$\boldsymbol{c}_{i,j}^{t,l+1} = \sigma\left(\boldsymbol{b}^l + \sum_{k \in N_{i,j}^t \cup \{j\}} A_i^t[j,k] \cdot \left(\boldsymbol{W}^l \boldsymbol{c}_{i,k}^{t,l}\right)\right)$$

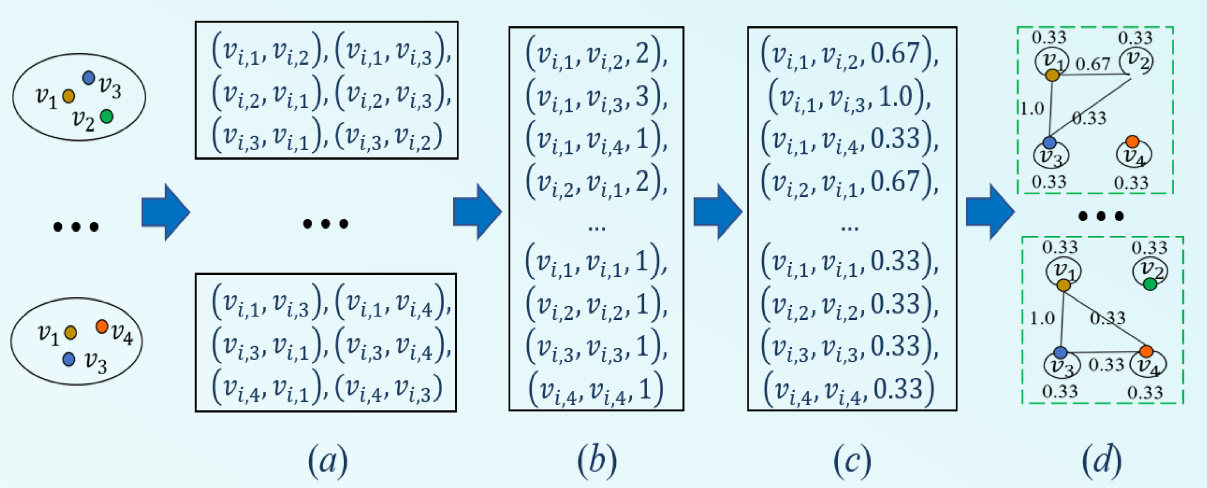

Figure 4: The process of weighted graphs construction.

Attention-based temporal dependency learning

- Self-attention mechanism:

$$\boldsymbol{Z}_{i,j} = softmax\left(\frac{(\boldsymbol{C}_{i,j}\boldsymbol{W}_q)(\boldsymbol{C}_{i,j}\boldsymbol{W}_k)^{\mathrm{T}}}{\sqrt{F''}} + \boldsymbol{M}_i\right) \cdot (\boldsymbol{C}_{i,j}\boldsymbol{W}_v)$$

$$M_i^{t,t'} = \begin{cases} 0, & if\ t \le t' \\ -\infty, & otherwise \end{cases}$$ is a masked matrix.

- Weighted aggregation: $\boldsymbol{z}_{i,j} = \left(\left(\boldsymbol{Z}_{i,j} \cdot \boldsymbol{w}_{agg}\right)^{\mathrm{T}} \cdot \boldsymbol{Z}_{i,j}\right)^{\mathrm{T}}$

Gated Information Fusing

- Static information: original element representation.
- Dynamic information: learned compact representation.

The gated updating mechanism:

$$\boldsymbol{E}_{i,I(j)}^{update} = \left(1 - \beta_{i,I(j)} \cdot \gamma_{I(j)}\right) \cdot \boldsymbol{E}_{i,I(j)} + \left(\beta_{i,I(j)} \cdot \gamma_{I(j)}\right) \cdot \boldsymbol{z}_{i,j}$$

The Prediction Layer

The possibilities of elements appearing in the next-period set is calculated by,

$$\hat{\boldsymbol{y}}_i = sigmoid\left(\boldsymbol{E}_i^{update}\boldsymbol{w}_o + \boldsymbol{b}_o\right)$$

Experimental Results

- Datasets: TaFeng, DC, TaoBao and TMS.
- Baselines: TOP, PersonalTOP, ElementTransfer, DREAM and Sets2Sets.
- Evaluation metrics: Recall, NDCG and PHR.

Table 1: Comparisons with different methods.

Datasets	Methods	K=10			K=20			K=30			K=40		
		Recall	**NDCG**	**PHR**	**Recall**	**NDCG**	**PHR**	**Recall**	**NDCG**	**PHR**	**Recall**	**NDCG**	**PHR**
TaFeng	Top	0.1025	0.0974	0.3047	0.1227	0.1033	0.3682	0.1446	0.1104	0.4256	0.1561	0.1140	0.4474
	PersonalTop	0.1214	0.1128	0.3763	0.1675	0.1280	0.4713	0.1882	0.1336	0.5063	0.2022	0.1398	0.5292
	ElementTransfer	0.0613	0.0644	0.2255	0.0721	0.0670	0.2519	0.0765	0.0676	0.2590	0.0799	0.0687	0.2671
	DREAM	0.1174	0.1047	0.3088	0.1489	0.1143	0.3814	0.1719	0.1215	0.4383	0.1885	0.1265	0.4738
	Sets2Sets	0.1427	0.1270	0.4347	0.2109	0.1489	0.5500	0.2503	0.1616	0.6044	0.2787	0.1700	0.6379
	DNNTSP	**0.1752**	**0.1517**	**0.4789**	**0.2391**	**0.1720**	**0.5861**	**0.2719**	**0.1827**	**0.6313**	**0.2958**	**0.1903**	**0.6607**
DC	Top	0.1618	0.0880	0.2274	0.2475	0.1116	0.3289	0.3204	0.1288	0.4143	0.3940	0.1448	0.4997
	PersonalTop	0.4104	0.3174	0.5031	0.4293	0.3270	0.5258	0.4499	0.3318	0.5496	0.4747	0.3332	0.5785
	ElementTransfer	0.1930	0.1734	0.2546	0.2280	0.1816	0.3017	0.2589	0.1929	0.3417	0.2872	0.1955	0.3783
	DREAM	0.2857	0.1947	0.3705	0.3972	0.2260	0.4964	0.4588	0.2407	0.5613	0.5129	0.2524	0.6184
	Sets2Sets	0.4488	0.3136	0.5458	0.5143	0.3319	0.6162	0.5499	0.3405	0.6517	0.6017	0.3516	0.7005
	DNNTSP	**0.4615**	**0.3260**	**0.5624**	**0.5350**	**0.3464**	**0.6339**	**0.5839**	**0.3578**	**0.6833**	**0.6239**	**0.3665**	**0.7205**
TaoBao	Top	0.1567	0.0784	0.1613	0.2494	0.1019	0.2545	0.3166	0.1164	0.3220	0.3679	0.1264	0.3745
	PersonalTop	0.2190	0.1535	0.2230	0.2260	0.1554	0.2306	0.2354	0.1575	0.2402	0.2433	0.1590	0.2484
	ElementTransfer	0.1190	0.1153	0.1217	0.1253	0.1166	0.1284	0.1389	0.1197	0.1427	0.1476	0.1214	0.1516
	DREAM	0.2431	0.1406	0.2491	0.3416	0.1657	0.3483	0.4060	0.1796	0.4129	0.4532	0.1889	0.4606
	Sets2Sets	0.2811	0.1495	0.2868	0.3649	0.1710	0.3713	0.4267	0.1842	0.4327	0.4672	0.1922	0.4739
	DNNTSP	**0.3035**	**0.1841**	**0.3095**	**0.3811**	**0.2039**	**0.3873**	**0.4347**	**0.2154**	**0.4406**	**0.4776**	**0.2238**	**0.4843**
TMS	Top	0.2627	0.1627	0.4619	0.3902	0.2017	0.6243	0.4869	0.2269	0.7222	0.5605	0.2448	0.8007
	PersonalTop	0.4508	0.3464	0.6440	0.5274	0.3721	0.7146	0.5453	0.3765	0.7339	0.5495	0.3771	0.7374
	ElementTransfer	0.3292	0.2984	0.4752	0.3385	0.3038	0.4828	0.3410	0.3034	0.4863	0.3423	0.3036	0.4889
	DREAM	0.3893	0.3039	0.6090	0.4962	0.3379	0.7279	0.5677	0.3570	0.794	0.6155	0.3690	0.8315
	Sets2Sets	**0.4748**	**0.3782**	**0.6933**	0.5601	**0.4061**	0.7594	0.6145	**0.4204**	0.8131	0.6627	**0.4321**	0.8570
	DNNTSP	0.4693	0.3473	0.6825	**0.5826**	0.3839	**0.7880**	**0.6440**	0.4000	**0.8439**	**0.6840**	0.4097	**0.8748**

Experimental results demonstrate that our approach could outperform existing methods with a significant margin.

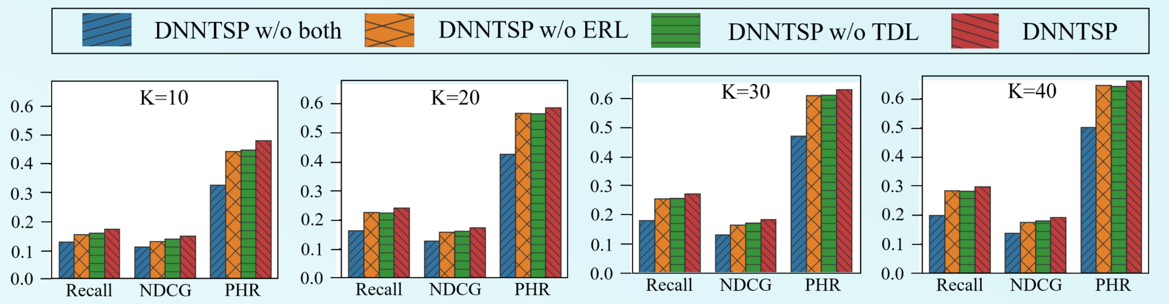

Figure 5: Ablation study of the proposed model.

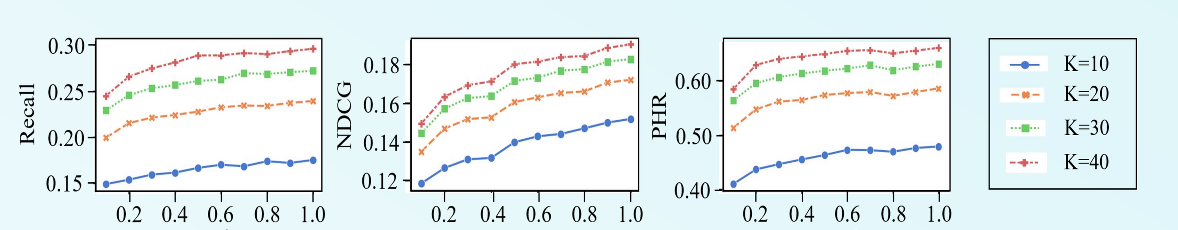

Figure 6: Performance on different ratios of training data.

Conclusion

This paper studies predictive modelling of a new type of temporal data, namely, temporal sets. Different from the existing methods, our method is founded on the multiple and comprehensive set-level element representations. Experimental results demonstrate that our method could circumvent the information loss problem suffered by the set-embedding based methods, and achieve higher prediction performance than the state-of-the-art methods.