R5-Reviewer-rJau

Con1:

The proposed solution is originally proposed for time-specific next POI recommendation task.

Better to analyze its capability for general time-aware recommendation tasks.

Response-to-Con1:

The proposed solution is not limited to the POI recommendation task. Similar to commonly used addition or concatenation operations, the time2rotation technique incorporates temporal influence through rotation operations, making it a model-agnostic general technique. It can be readily applied to other time-aware recommendation tasks. We plan to further explore the Time2rotation technique for general temporal studies in the future.

We provide the experimental results of the latest method (TPG [25]) and ROTAN in the following table. In this experimental setting, we only consider the time-aware POI recommendation task, i.e., predicting the POIs at specific times. It doesn't utilize the context information in the trajectories. We can see that ROTAN significantly outperforms TPG, indicating the advantages of the Time2rotation technique.

Table1. The results of TPG and ROTAN for general time-aware recommendation task on three datasets.

Datasets	**Models**	**Acc@1**	**Acc@5**	**Acc@10**	**MRR**
NYC	TPG time-aware	0.1812	0.3623	0.4387	0.2665
	ROTAN time-aware	0.2247	0.4084	0.4893	0.3128
TKY	TPG time-aware	0.0819	0.2142	0.2723	0.1448
	ROTAN time-aware	0.1071	0.2256	0.2745	0.1656
CA	TPG time-aware	0.0838	0.1548	0.1889	0.1198
	ROTAN time-aware	0.1015	0.1925	0.2315	0.1446

Con2:

The differences between Time2rotation and the time2vec technique [17] can be further elaborated.

Response-to-Con2:

Time2Rotation is a method of utilizing temporal information, while time2vector [17] is a method of representing temporal information, i.e., learning a vector representation of time. In time2vec [17], it mainly utilizes periodic activation functions for time representation. Conventionally, timeslots are represented with time embedding vectors, and then these vectors are integrated using concatenation or addition operations. In this work, we propose an innovative approach to integrate temporal information using rotation techniques. Instead of conventional embedding vectors, we represent timeslots with rotation vectors. Moreover, we incorporate temporal influence via rotation operations, which differs from existing methods.

These two techniques can be jointly utilized. For instance, we could use the time2vec technique to initialize the rotation vectors, which are learnable and can be updated during the training phases.

Q1:

As far as I know, the rotation technique has been used for periodical time series predictions [2]. What is the difference between time2rotation and learning-to-rotate [2]? The authors should provide more explanations regarding this issue.

Response-to-Q1:

The learning-to-rotate [2] is also proposed for temporal information, mainly designed for complicated periodical time series forecasting. It primarily utilizes a Quaternion Transformer to capture multiple periods, variable periods, and phase shifts in real-world datasets. Learning-to-rotate [2] indicates that the rotation operation can effectively incorporate temporal information, especially periodical patterns. However, our work differs from learning-to-rotate in three aspects. (1) They address different tasks. Our ROTAN solves the time-specific recommendation task, while [2] focuses on time series forecasting. The proposed solution of [2] cannot directly address our task. (2) The approach to rotation is different. In [2], quaternions are used for the attention module. Quaternions are extensions of complex numbers, represented as □ = □ + □i + □j + □k, where □, □, □, □ are real numbers and i, j, k are imaginary units. It exploits rotation with unit quaternions. In our work, the rotation is based on complex space, which is relatively simpler. (3) The strategy for exploiting rotation to incorporate temporal information is different. To measure similarity in time series, [2] designs a novel Learning-to-Rotate Attention mechanism, which replaces the canonical attention mechanism. Our work applies rotation operations to the extracted representations, inspired by RotatE [35].

Q2:

Figure 3 is difficult to interpret. It requires some effort to get the different patterns from the hour distributions of users/POIs.

Response-to-Q2:

To investigate the temporal information of check-in data, we conduct data observations based on real-world datasets. Specifically, we study the distributions of the number of hours for users and POIs in Figure 3. Figure 3(a) shows the statistics of the number of hours that users are active. We can see that most users have check-ins between 10 and 18 hours, indicating their activity throughout various periods. Meanwhile, as shown in Figure 3(b), most POIs were visited between 2 and 8 hours within the 24-hour period, which is narrower than the distribution of Figure 3(a). This distribution suggests that most POIs are accessible by users during specific hours. For instance, restaurants may primarily be visited during lunch or dinner times. The distinct temporal distributions of users and POIs necessitate separate modeling approaches. Since users and POIs exhibit distinct temporal patterns, we consider them independently in Section 4.3: User Temporal Sequence Modeling and POI Temporal Sequence Modeling.