International Journal of Software and Informatics, ISSN 1673-7288
http://www.ijsi.org, ijsi@iscas.ac.cn, +86-10-62661048
IJSI, 2023, 13(4): 449–467, doi: 10.21655/ijsi.1673-7288.00310

Research Article

Time-aware Knowledge Hypergraph Link Prediction

Zirui Chen (陈子睿)[1,2], Xin Wang (王鑫)[1,2], Chenxu Wang (王晨旭)[1,2], Shaowei Zhang (张少伟)[1,2], Haoyu Yan (闫浩宇)[1,2]

[1] (College of Intelligence and Computing, Tianjin University, Tianjin 300350, China)
[2] (Tianjin Key Laboratory of Cognitive Computing and Application, Tianjin 300350, China)
Corresponding author: Xin Wang, wangx@tju.edu.cn

Abstract A knowledge hypergraph is a form of heterogeneous graph representing the real world through n-ary relations, but existing knowledge hypergraphs are usually incomplete in both general and vertical domains. Therefore, it is challenging to infer the missing links from the existing links in knowledge hypergraphs. Most of the current studies employ knowledge representation learning methods based on n-ary relations to accomplish link prediction in knowledge hypergraphs, but they only learn the embedding vectors of entities and relations from time-unknown hyperedges without considering the influence of temporal factors on the dynamic evolution of facts, which results in poor prediction performance in dynamic environments. Firstly, based on the definition of temporal knowledge hypergraphs proposed by this paper for the first time, this paper puts forward a link prediction model for temporal knowledge hypergraphs and learns static and dynamic representations of entities from their roles, positions, and timestamps of temporal hyperedges. Then these representations are merged in a certain proportion and utilized as final entity embedding vectors for link prediction tasks to realize the full exploitation of hyperedge temporal information. Meanwhile, it is theoretically proven that the proposed model is fully expressive with linear space complexity. Additionally, a temporal knowledge hypergraph dataset CB67 is constructed from the public business data of listed companies, and a large number of experimental evaluations are conducted on this dataset. The experimental results show that the proposed model can effectively perform link prediction tasks on the temporal knowledge hypergraph dataset.

Keywords temporal knowledge hypergraph; link prediction; knowledge representation; embedding learning; temporal information

Citation Chen ZR, Wang X, Wang CX, Zhang SW, Yan HY. Time-aware knowledge hypergraph link prediction, *International Journal of Software and Informatics*, 2023, 13(4): 449–467. http://www.ijsi.org/1673-7288/310.htm

Knowledge hypergraphs adopt n-ary relations to represent real-world facts, and each hyperedge consists of an n-ary relation ($n \geq 2$) and correspondingly ordered n entities. In contrast, knowledge graphs[1] are a special case of knowledge hypergraphs, which means they are a kind of knowledge hypergraphs in which the arity of each relation is two. In essence,

---

This is the English version of the Chinese article “面向时间感知的知识超图链接预测. 软件学报, 2023, 34(10): 4533-4547. doi: 10.13328/j.cnki.jos.006888”
Funding items: Science and Technology Innovation 2030 “New Generation of Artificial Intelligence” Major Project (2020AAA0108504); National Natural Science Foundation of China (61972275)
Received 2022-07-05; Revised 2022-08-18, 2022-12-14; Accepted 2022-12-28; IJSI published online 2023-12-28

knowledge hypergraphs are a more expressive form of knowledge representation than knowledge graphs, and also an important way to organize human knowledge effectively. Related work shows that traditional knowledge graphs based on 2-ary relations are deficient in representing higher-order semantic information in the real world; more than 1/3 of the entities in Freebase[2] are involved in the representation of n-ary relations[3], and more than 61% of the relations are represented in the form of n-ary relations[4]. This proves that knowledge hypergraphs are closer to real-world factual representation forms, and facilitate the development of a series of knowledge-based downstream applications[5], such as recommendation systems[6] and question answering systems[7]. Despite the large scale of existing knowledge hypergraphs, they are far from complete, and thus knowledge hypergraph link prediction, or automatic inference of missing facts between entities in knowledge hypergraphs, has gradually become an important research direction.

The research on knowledge hypergraph link prediction[8, 9] is mainly based on time-unknown hyperedges to complete embedding learning of entity relations, which is classified into four categories according to the learning methods.

- The soft rule-based method is interpretable and can leverage rules to explain the reason for inference results, such as the popular Markov Logic Network (MLN)[10] model.
- The translation-based method is mainly derived from the n-ary generalization of the knowledge graph embedding method, which learns the connections between entities and relations by embedding them into the same vector space. The first model of such a method is the m-TransH[3] model based on a generalization of TransH[11]. However, this model is not fully expressive, which means for any given disjoint set of true and false hyperedges, it cannot provide a parametric representation for each hyperedge to accurately express the type (true or false) to which the hyperedge belongs.
- The tensor decomposition-based method learns embedding vectors by decomposing the higher-order tensor into several lower-order tensors, and it is currently a higher-performance method for link prediction of knowledge hypergraphs. The latest SOTA model RAM[12] obtains the optimal performance indicators by digging into the relevance of entity roles.
- For the neural network-based methods, few models can be applied to the knowledge hypergraph structure. The latest work is the G-MPNN[13] model, which extends the Message Passing Neural Network (MPNN) to the knowledge hypergraph structure to solve link prediction of n-ary relations.

Despite the success of these existing embedding technologies, including both hypergraphs[14] and knowledge hypergraphs, they are based on the assumption that the hyperedges do not contain temporal properties, which means the learned entity embedding vector is a static representation or aggregation of entity features over a time period. However, as shown in Figure 1 which demonstrates the temporal knowledge hypergraph of “Xiaomi” Corporation, the 3-ary relations “Branch” indicate that the roles of entities in positions 1–3 are the temporal hyperedges of subsidiary, core company, and subsidiary respectively. With the Branch relation “Venus Venture”, “Xiaomi”, and “Aidong” 2013-12-26 as an example, the roles of entity 1 “Venus Venture” and entity 3 “Aidong” are both subsidiaries, the role of entity 2 “Xiaomi” is the core company, and the relations occur on December 26, 2013. Therefore, in the real world, many facts are not static, but highly transient. For example, the Branch relation “Venus Venture”, “Xiaomi”, “Yingpeng Entertainment”, and the Shareholder relation “Xiaomi”, “Mipay”, and “Xiaomi Credit” all occur on December 26, 2013. The Branch relation “Venus Venture”, “Xiaomi”, and “Mipay” should only be true after July 10, 2018. Intuitively, when knowledge hypergraph link prediction is performed, the temporal properties of facts should also play an important role. Learning only one static representation form for each entity may be suboptimal. This requires an in-depth study of the knowledge hypergraph embedding technology that includes temporal information to provide entities with their feature vectors at any given time.

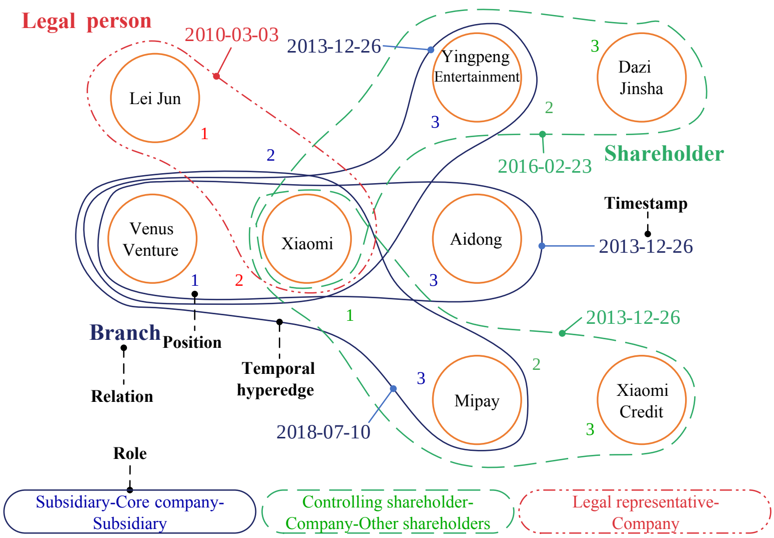


Figure 1 Example of temporal knowledge hypergraph

Firstly, the concept of temporal knowledge hypergraph proposed by this paper for the first time and the definition of its link prediction task are clarified, and then the concept is extended to n-ary relations based on the knowledge graph embedding model DistMult[15]. Meanwhile, entity information is extracted from the perspective of static embedding and dynamic embedding to conduct information merging, and the embedding model THM (Temporal knowledge Hypergraph Model) is proposed for link prediction in temporal knowledge hypergraphs. This model aims to obtain static structural information and dynamic temporal information of the temporal knowledge hypergraph. Firstly, the static embedding vectors of entities and relations are obtained by the differences in roles and positions of entities in the temporal hyperedges, and then the dynamic embedding vectors of entities at specific timestamps are obtained by the differences in temporal information of the temporal hyperedge where the entity is located. Finally, the two types of embedding vectors are merged in a certain proportion and employed for the downstream link prediction task. It is theoretically proven that the THM model is fully expressive with linear space complexity. The validity, rationality, and robustness of THM are verified by comparing it with the knowledge hypergraph embedding model on the real temporal knowledge hypergraph dataset CB67. It is known that this is the first fully expressive model for link prediction in temporal knowledge hypergraphs.

The main contributions of this paper are as follows.

(1) Based on the definition of temporal knowledge hypergraphs proposed by this paper for the first time, the embedding model THM for link prediction in temporal knowledge hypergraphs is put forward to simultaneously learn static embedding vectors from entity role and position information. Additionally, dynamic embedding vectors are learned from temporal hyperedge timestamps, and the two embedding vectors are merged in a certain proportion as the final entity representation to fully exploit hyperedge temporal information and improve the link prediction performance.

(2) It is theoretically proven that the THM model is fully expressive and has linear space

complexity, and can provide an embedding representation for any temporal knowledge hypergraph dataset to accurately classify facts and non-facts. Meanwhile, the model is scalable when processing large-scale temporal knowledge hypergraph data.

(3) The first temporal knowledge hypergraph dataset CB67 is constructed from the public business data of listed companies. It contains up to 7-ary relations and hyperedges with timestamp properties and can be adopted as a benchmark dataset for link prediction tasks of temporal knowledge hypergraphs. A large number of experiments have been carried out on CB67 with the model in this paper, and the validity, rationality, and robustness of the model are verified by comparing it with the performance of the knowledge hypergraph embedding model.

In this paper, Section 1 and Section 2 introduce the related work and the preliminary knowledge of temporal knowledge hypergraphs and link prediction in temporal knowledge hypergraphs respectively. Section 3 proposes a link prediction model in temporal knowledge hypergraphs, which is formulated in terms of structural static embedding, temporal dynamic embedding, and model training. Section 4 theoretically proves that the model is fully expressive and has linear space complexity. Section 5 conducts experiments on the real dataset CB67, with a summary in Section 6.

1 Related Work

In recent years, academic and industrial circles have gradually realized the importance of graph data applications, and link prediction has become a research hotspot in this direction. There is no research on link prediction based on temporal knowledge hypergraph structures in academia. This section will introduce the current research related to link prediction in temporal knowledge hypergraphs, which is mainly divided into temporal knowledge graphs and knowledge hypergraphs.

1.1 Link prediction in temporal knowledge graphs

Several studies have put forward methods to address link prediction in temporal knowledge graphs, and the methods generally adopt temporal embedding to encode the evolution of entities and relations over time. Know-Evolve[16] models entity representations by evolution modeling of timestamps to represent the influence of temporal properties on entities. TA-TransE[17] employs a modeling method that embeds temporal information, embeds time in textual form into relation representations via a Recurrent Neural Network (RNN), and utilizes the score function of TransE[18] for entity prediction. Temporal TransE[19] represents the embedding vectors of entities and relations in the same vector space and leverages the score function similar to TransE for link prediction. ChronoR[20] similarly projects entities and relations into other spaces by transformation, rotates the space according to time, and employs a novel score function for prediction. However, these dynamic reasoning models cannot capture temporal correlation and generalize the structural information of temporal knowledge graphs to a future time.

Subsequently, several studies analyze the structure of temporal knowledge graphs and apply a combination of Graph Neural Networks (GNNs) and RNN modules to link prediction tasks in temporal knowledge graphs. Recursive Event Network (RE-Net) can predict concurrent facts at multiple time points in the temporal knowledge graph and build temporal correlation models. Although the adopted RGCN aggregator[21] could obtain the relational adjacency information at a certain time as a whole, the adjacency information of non-target entities would be added during the aggregation, resulting in a decreased predictive ability of the model. Subsequent work improves the RE-Net model and proposes the RE-GCN model[22] to shorten the training time of the model by optimizing the RGCN aggregation, and entity static constraints are added to the model for link prediction to improve the model performance.

1.2 Link prediction in knowledge hypergraphs

Link prediction in knowledge hypergraphs mainly accomplishes the prediction of unknown hyperedges by representing hyperedges and entities as embedding vectors in a low-dimensional vector space. The solutions for this task mainly include soft rule-based, translation-based, tensor decomposition-based, and neural network-based methods.

The soft rule-based method is interpretable, employs entities and relations in the knowledge hypergraph as variables and predicates respectively, and sets logical constraints for relational reasoning to reason about unknown hyperedges. MLN first combines first-order predicate logic with a probabilistic graphical model and assigns different weights to logic rules to effectively process knowledge hypergraph data and rule uncertainty. Relational Logistic Regression (RLR)[23] applies logistic regression algorithms to relational models to improve the predictive performance of the MLN model.

The translation-based method embeds entities and relations into the same vector space and translates the entity vectors based on the relational embedding. As a result, the model learns the representations of entities and relations in the embedding space and further utilizes the learned embedding vectors to accomplish the link prediction task. m-TransH is the first knowledge hypergraph embedding model, which extends the TransH based on 2-ary relations to n-ary relations for supporting the reasoning about unknown hyperedges. Then, RAE[24] adopts the m-TransH model as a basis to extend the performance of the m-TransH method by adding two probability values indicating joint participation in the same hyperedge to the loss function. The above models are the two main preliminary works for link prediction of knowledge hypergraphs and are not fully expressive with limitations in relation modeling. Subsequently, the BoxE model based on spatial translation[25] is proposed. It represents entities and relations as points and hyper-rectangles in the embedding space respectively, and predicts the probability value of hyperedge existence by calculating the distance from the entity point to the center of the hyper-rectangle corresponding to the hyperedge relations. This is the only model with full expressiveness among the current translation-based methods.

Methods based on tensor decomposition generally decompose a higher-order tensor into a sum of several lower-order tensors. The first application of such methods is the GETD model[26], which is an extension of the knowledge graph TuckER[27] model. This model is fully expressive but can only process k-uniform knowledge hypergraphs, and cannot simultaneously process knowledge hypergraphs with n-ary relations. It is necessary to divide the dataset according to the arity of the relations before training, and then train the models corresponding to such arities. Subsequently, inspired by the tensor decomposition-based knowledge graph embedding method, the extended models m-CP[29] and m-DistMult[29] based on CP[28] and DistMult algorithms were proposed. Then, the HSimplE[29] model based on SimplE[30], which can be applied to n-ary relations, was put forward. Meanwhile, to consider the role of entity position semantic information in link prediction of knowledge hypergraphs, the HypE[29] model learning different embedding representations for different entities based on positions was proposed. Both models are fully expressive. Subsequently, for the solution to the over-parameterization of tensor decomposition-based methods, the S2S[31] model was proposed to reduce the model parameters by sparsifying the core tensor while preserving its expressiveness through neural architecture search technology to improve the performance of such methods. The RAM model finds that current studies neglect the important semantic property of entity roles, and thus "role-aware modeling" at the role level is proposed to encourage semantically related roles to have similar embedding representations. This model is the current SOTA method for link prediction in knowledge hypergraphs and is also fully expressive.

The neural network-based model is an effective method to solve link prediction of knowledge hypergraphs. As the first application of such a method, NaLP[32] introduced the role-entity form of n-ary relations and employed CNN and FCN modules to measure the compatibility of entities with their roles. Subsequent works HINGE[33] and NeuInfer[34] decomposed n-ary relation hyperedges into a knowledge graph triplet and several role-entity pairs, which mainly relied on CNN and FCN models for compatibility determination respectively. The recent work StarE[35] adopted the CompGCN module to model the decomposed knowledge graph triplet. The G-MPNN model solved the link prediction of n-ary relations by extending the MPNN to knowledge hypergraph structures. However, the neural network model utilizes more parameters to represent the knowledge hypergraph structure, which is prone to overfitting during the training and makes the training difficult to proceed.

Unlike the above methods, the temporal knowledge hypergraph structure proposed in this paper contains both the temporal information in the temporal knowledge graph and the n-ary relations in the knowledge hypergraph. Additionally, since the concept of temporal knowledge hypergraph is put forward by this paper for the first time, no existing methods can be directly applied to the temporal knowledge hypergraph structure to accomplish the link prediction task. By combining the main features of the above two related works, this paper obtains entity embedding from both the static structure of the knowledge hypergraph and the dynamic time series of the temporal knowledge graph, which is combined as the final embedding representation and is applied to the downstream link prediction task.

2 Preliminary Knowledge

This section will give a detailed introduction of relevant background knowledge, including the definitions of the temporal knowledge hypergraph structure and the temporal knowledge hypergraph link prediction task, which are proposed by this paper for the first time to lay a foundation for understanding the model architecture of this paper. The main symbols adopted in this paper and their meanings are given in Table 1. Variables are expressed in italics, and vectors and matrices are expressed in bold.

Table 1 Symbol list

Symbol	Meaning
e, r, p, t	Entity, relation, role, timestamp
n	Arity of the relation
m, d	Number of layers and dimension of entity embedding
L, l	Role set and number
$\boldsymbol{z}, \boldsymbol{e}, \boldsymbol{c}, \boldsymbol{e}', \boldsymbol{e}^t$	Entity, entity basis, role, entity role, entity temporal embedding vector
$\boldsymbol{w}$	Role weight
$\boldsymbol{B}, \boldsymbol{B}_a, \boldsymbol{R}$	Role, relation basis, role relation matrix
$\boldsymbol{F}^t, \boldsymbol{D}^t$	Time frequency, weight matrix
ϕ	Score function
$\odot$	Element product function
σ	Normalization function
cat	Concatenation function
φ	Activation function
η	Loss function

2.1 Temporal knowledge hypergraph

There is currently no study that gives a specific definition of the temporal knowledge hypergraph structure. Before defining link prediction tasks in temporal knowledge hypergraphs, the definition of the temporal knowledge hypergraphs is given first.

Definition 1 (temporal knowledge hypergraph)**.** A temporal knowledge hypergraph $\mathcal{H}$ is a heterogeneous graph with timestamped hyperedges, and is expressed as $\mathcal{H} = \{(r, p_1^r : e_1, p_2^r : e_2, \cdots, t) | r \in R, e_i \in E, t \in T\}$, where R is the relation set, E is the entity set, and T is the set of all possible timestamps. A temporal hyperedge is expressed as an m tuple $h = (r, p_1^r : e_1, p_2^r : e_2, \cdots, p_n^r : e_n, t)$, consisting of a n-ary relation r, n-role p_i^r of the entity at each position corresponding to the relations, entity e_i, and a timestamp t, (so, $m = n + 2$), to represent an event occurring at time t. In a temporal knowledge hypergraph, there can be multiple temporal hyperedges with different timestamps but the same relation type among the same entities. For example, the temporal hyperedges (Supply, Company A, Company B, Company C, 2022-01-01) and (Supply, Company A, Company B, Company C, 20220-02-01) can exist in the same temporal knowledge hypergraph simultaneously.

2.2 Link prediction in temporal knowledge hypergraphs

After clarifying the definition of temporal knowledge hypergraphs, the definition of the link prediction tasks in the temporal knowledge hypergraph structure is further given.

Definition 2 (Link prediction in temporal knowledge hypergraphs)**.** Given a temporal knowledge hypergraph $\mathcal{H}$ containing all observable temporal hypergraphs, the goal of link prediction in temporal knowledge hypergraphs is to predict the existence of a temporal hyperedge $(r, p_1^r : e_1, p_2^r : e_2, \cdots, t)$ consisting of an entity sequence replacing an entity under the n-ary relation and at time t through the existing observable temporal hypergraphs, and the position i of the replaced entity can be any one under n.

3 Model

This section mainly introduces the link prediction model THM based on temporal knowledge hypergraphs, which employs both static structural information of the temporal knowledge hypergraph and dynamic temporal information to generate two types of entity embedding and mixes them in a certain proportion to generate the final entity embedding vectors. This proportion is adopted as a hyperparameter to adjust the proportion of the two types of embedding vectors in the final entity representation. The overall architecture of the THM model is shown in Figure 2.

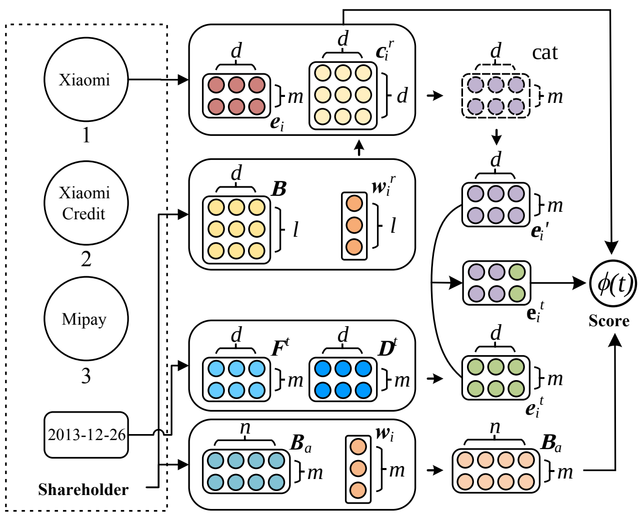


Figure 2 Overall architecture of THM

3.1 Structural static embedding

This subsection aims to obtain the static structural information of the temporal knowledge hypergraph and make full use of the differences in the roles and positions of entities in different temporal hyperedges to obtain the static embedding vectors of entities and relations. Based on the DistMult model of the knowledge graph, generalization is first conducted to support the representation of n-ary relations, then the differences in roles and positions between entities are highlighted for entity embedding, and finally, the differences in roles and positions in different entity positions are also highlighted for relation embedding. As a result, the information of the temporal knowledge hypergraph on the static structure can be fully exploited through two types of information.

As a classical method for knowledge graph representation learning, DistMult obtains the probability value of the triplet being true by simultaneously learning the embedding vectors of relations, head entities, and tail entities, and multiplying the three linearly. By generalizing the model to the knowledge hypergraph structure, the relation embedding with each entity embedding in the entity list can be naturally multiplied successively to realize

$$\phi(r(e_1, e_2, \cdots, e_i)) = \odot(\boldsymbol{r}, \boldsymbol{e}_1, \boldsymbol{e}_2, \cdots, \boldsymbol{e}_i) \tag{1}$$

For the entity embedding part of Eq. (1), the role difference and position difference of the entities are obtained successively.

In the dataset of temporal knowledge hypergraphs, the same entity may correspond to multiple positions and roles simultaneously. For example, as shown in Figure 1, the entity “Xiaomi” corresponds to positions 1 and 2 and the roles of holding company and legal representative in the temporal hyperedges with a relation of shareholder and legal person respectively. To fully represent the role information of an entity, each entity $e_i \in E$ is mapped into an entity embedding matrix, and it is assumed that $\boldsymbol{e}_i \in \mathbb{R}^{m\times d}$ is the entity embedding, m is the number of embedding layers, and d is the embedding dimension. A vector space $\boldsymbol{B}$ of dimension $l \times d$ is created for roles to make full use of the semantic differences of roles among entities, where l is the total number of roles and $\boldsymbol{w}_i^r \in \mathbb{R}^l$ is the role weight vector. The semantic relevance is implicitly parameterized by the role weights:

$$\boldsymbol{c}_i^r = \sum_{l=1}^{L} \boldsymbol{B}[l] \cdot \sigma(\boldsymbol{w}_i^r)[l] \tag{2}$$

After obtaining the entity's role embedding vector, the entity representation $\boldsymbol{e}_i' = \boldsymbol{e}_i \cdot \boldsymbol{c}_i^r$ with role semantics is obtained by multiplying the entity base embedding vector and the role embedding vector. With the inspiration from the knowledge graph embedding method SimplE, the semantic information of position i is combined within a single temporal hyperedge by a concatenation function cat, and the $\mathrm{cat}(\boldsymbol{v}, x)$ function shifts the vector $\boldsymbol{v}$ to the left by x steps:

$$\boldsymbol{e}_i' = (\boldsymbol{e}_i^1, \mathrm{cat}(\boldsymbol{e}_i^2, m \cdot d/n), \cdots, \mathrm{cat}(\boldsymbol{e}_i^\alpha, m \cdot d \cdot (n-1)/n)) \tag{3}$$

For the relation embedding part in Eq. (1), a relation matrix is generated to measure the compatibility between relations and entities.

The roles at each position in the relation are represented by a relation matrix to measure the compatibility between positions, roles, and all participating entities under a certain relation. For the relation $r \in R$, the relation matrix of the role at the i-th position is expressed by $\boldsymbol{R}_i^r \in \mathbb{R}^{n\times m}$. $\boldsymbol{R}_i^r[j,:]$ on the j-th row represents the compatibility of the role with the entity embedding at the j-th position, which is the fitness degree of the role played by the current entity at the current position with the semantic information of the temporal hyperedge. It is assumed that $\boldsymbol{B}_a \in \mathbb{R}^{n\times m}$ is the relation base matrix associated with the role base vector $\boldsymbol{B}[l]$ that represents the semantic relevance between roles in the same relation. This matrix is formed by splicing the role base vectors contained in the relation, aligned with the role base vector $\boldsymbol{B}[l]$, and normalized by the σ function to calculate the role relation matrix[12]:

$$\boldsymbol{R}_i^r = \sum_{l=1}^{L} \sigma(\boldsymbol{w}_i^r)[l] \cdot \sigma(\boldsymbol{B}_a[l]) \tag{4}$$

3.2 Temporal dynamic embedding

In addition to fully digging into the structural information of the temporal knowledge hypergraph, it is also crucial to exploit the temporal information. Aiming at obtaining the dynamic temporal information of the temporal knowledge hypergraph, this subsection makes full use of the differences in temporal information of different temporal hyperedges of the entities to obtain the dynamic embedding vectors of entities under specific timestamps. Two temporal feature matrices are set for the timestamps of the temporal hyperedges, and they represent the occurrence frequency of the current timestamp and the weight respectively. For the temporal dynamic embedding vector of the entity based on the temporal feature matrices, the merging proportion with the structural static embedding is determined by setting a hyperparameter to obtain the final entity representation. The role embedding, the final entity representation, and the role relation matrix are input to the score function to obtain the probability value of the existence of the predicted temporal hyperedge.

Intuitively, by learning the temporal feature matrices $\boldsymbol{F}^t$ and $\boldsymbol{D}^t$, the model knows how to turn on and off the entity features at different time points, and then the weights of the features can be controlled at any timestamp and the unknown temporal hyperedges can be predicted accurately. The sine function is employed as the activation function for the following formula since it can simulate multiple on/off states, and the temporal dynamic embedding of entities at specific timestamps is obtained by the following formula:

$$\boldsymbol{e}_i^t = \varphi(\boldsymbol{F}^t \cdot t + \boldsymbol{D}^t) \tag{5}$$

After obtaining the static and dynamic embedding vectors of the entity, the model mixes the two types of vectors in a certain proportion and generates the final entity embedding vector. A dynamic embedding function is proposed, whose output is the mixed entity representation, and the final entity embedding vector $\boldsymbol{z}_i$ is defined as follows. The first λd elements of the vector capture the temporal dynamic features of the entity, and the remaining $(1-\lambda)d$ elements capture the structural static features of the entity, where $0 \leq \lambda \leq 1$ is a hyperparameter to control the proportion of feature types:

$$\boldsymbol{z}_i[k] = \begin{cases} \boldsymbol{e}_i'[k] \cdot \varphi(\boldsymbol{F}^t[k] \cdot t + \boldsymbol{D}^t[k]), & 0 \leq k \leq \lambda d \\ \boldsymbol{e}_i'[k], & \lambda d < k \leq d \end{cases} \tag{6}$$

Since the strategy of obtaining the structural static embedding of the temporal knowledge hypergraph starts from the position and role of the entity and generates unique representations for the roles of the entity at different positions in the n-ary relations, the relation embedding in the score function of Eq. (1) can be replaced by the role embedding. After obtaining the final entity embedding vector, the role embedding, the final entity embedding, and the role relation matrix are input to the score function to obtain the score value of true temporal hyperedge:

$$\phi(h) = \sum_{i=1}^{n} \langle \boldsymbol{c}_i^r, \boldsymbol{R}_i^r[1,:]\boldsymbol{z}_1, \cdots, \boldsymbol{R}_i^r[n,:]\boldsymbol{z}_n \rangle \tag{7}$$

3.3 Model training

The temporal hyperedges in the temporal knowledge hypergraph are divided into training, validation, and test sets, and the model parameters are learned by a stochastic gradient descent method combined with a small-batch sampling technique. Assuming that S is a small batch sampling of the training set, for each temporal hyperedge $h = (r, p_1^r : e_1, p_2^r : e_2, \cdots, t) \in S$, a total of n queries are generated: $(r, p_1^r : e_1, \cdots, p_i^r :?, \cdots, t)$. For each generated query, a candidate answer set C_e is given, which contains randomly selected q entities different from the entity e_i replaced at that entity position. The minimized cross-entropy is then adopted as the loss function, which has been utilized for link prediction tasks in knowledge graphs and temporal knowledge graphs with sound results:

$$\eta = -\left(\sum_{h\in S}\sum_{i=1}^{\alpha}\frac{\exp(\phi(h))}{\sum_{e_i\in C_e}\exp(\phi(r, p_1^r : e_1, \cdots, p_i^r : e_i, \cdots, t))}\right) \tag{8}$$

4 Analysis of Full Expressiveness and Space Complexity

Full expressiveness is an important property of models and the research content of knowledge hypergraphs. The ideal property of a model is full expressiveness. Given the correct and incorrect temporal hyperedge sets of any temporal knowledge hypergraph dataset, a model will be fully expressive if there is a parametric representation in the model that can correctly classify the true or false value of the temporal hyperedge in both sets. For knowledge hypergraph embedding models, several models have been proven to be fully expressive, but there is no temporal knowledge hypergraph embedding model that is fully expressive. The full expressiveness of the THM is established by the following theorem.

Theorem 1. THM is fully expressive for temporal knowledge hypergraphs.

Proof: According to the definition of Eq. (6), the embedding function of THM maps the representation of a single entity at different positions into a vector, and the theorem is proven by a special case of THM, which means the entity embedding is a purely temporal dynamic embedding vector and a purely structural static embedding vector. This special case can be achieved by assuming $\lambda = 1$ or $\lambda = 0$ and all entities $e \in E$ at $0 \leq k \leq d$.

If THM can achieve full expressiveness in this special case, the final embedding of entities in these two cases can be written as

$$\boldsymbol{z}_i[k] = \boldsymbol{e}'_i[k] \cdot \sin(\boldsymbol{F}^t[k] \cdot t + \boldsymbol{D}^t[k]) \tag{9}$$

$$\boldsymbol{z}_i[k] = \boldsymbol{e}'_i[k] \tag{10}$$

To further simplify the proof, based on the research[30], this theorem can be proven by showing why the score of the temporal hyperedge can be a positive or negative number when $(r, p_1^r : e_1, p_2^r : e_2, \cdots, t) \in S$ or $(r, p_1^r : e_1, p_2^r : e_2, \cdots, t) \notin S$ respectively.

Assuming that $d = |R| \cdot |E| \cdot |T| \cdot L$ where L is a natural number, these entity embedding vectors can be considered as $|R|$ blocks with a length of $|E| \cdot |T| \cdot L$, and for the j-th relation r_j, all elements of the entity vector $\boldsymbol{z}$ are set to be 0 except 1 for the j-th block. By this assignment, the score of the temporal hyperedge is only related to the j-th block of the embedding vector. Now, the focus is on the j-th block of the embedding vector.

The length of the j-th block (similar to all other blocks) is $|E| \cdot |T| \cdot L$, and it can be considered as $|E|$ subblocks with a length of $|T| \cdot L$. For the i-th entity e_i, the value of $\boldsymbol{z}$ is assumed to be zero in all the subblocks except the i-th subblock. With such a value assignment, only the i-th sub-block of the j-th block is important to obtain the score of a temporal hyperedge.

It should be noted that this subblock is unique for each temporal hyperedge containing entity e_i and relation r_j. Now, the focus is on the i-th subblock of the j-th block.

The length of the i-th subblock of the j-th block is $|T| \cdot L$, which can also be considered as $|T|$ subblocks with a length of L. According to the Fourier sine series, under the large enough L, the sum of the $\boldsymbol{z}$ elements of the p-th subblock is 1 when $t = t_p$ and is 0 when t is a timestamp other than t_p through setting the values of $\boldsymbol{z}$, $\boldsymbol{F}^t$, and $\boldsymbol{D}^t$. It should be noted that this subblock is unique for each temporal hyperedge containing entity e_i, relation r_j, and timestamp t.

With the above value assignment, the score of the temporal hyperedge can be a positive or negative number when $(r, p_1^r : e_1, p_2^r : e_2, \cdots, t) \in S$ or $(r, p_1^r : e_1, p_2^r : e_2, \cdots, t) \notin S$.

In addition to full expressiveness, the model is also scalable in the case of large-scale temporal knowledge hypergraph data. The linear space complexity of the THM is established by the following theorem.

Theorem 2. The space complexity of the THM model is $\mathrm{O}(m_e d + L m_r m_\alpha)$, where m_e is the total number of entities, m_r is the total number of relations, and m_α is the maximum number of relation arities in the temporal knowledge hypergraph.

Proof: Since the number of relations in the temporal knowledge hypergraph is rarely higher than 7 (cf. Table 2), the assigned value to parameter m will not be more than 3 and the maximum number of parameters in the role embedding vector is $m_e d$. Additionally, the maximum number of parameters in the relation base matrix is $L m m_\alpha$, the maximum number of parameters in the role weight vector is $L m_r m_\alpha + Ld$, and the maximum number of total parameters in the model is $\mathrm{O}(m_e d + L m_r m_\alpha + Ld + L m m_\alpha) = \mathrm{O}(m_e d + L m_r m_\alpha)$. Therefore, the space complexity of the THM model is $\mathrm{O}(m_e d + L m_r m_\alpha)$.

Table 2 Statistical information of datasets

Dataset	Number of entities	Number of relations	Number of 2-arity	Number of 3-arity	Number of 4-arity	Number of 5-arity	Number of 6-arity	Number of ≥7-arity	Time type	Interval
JF17K	29,177	322	56,332	34,550	9,509	2,230	37	0	None	—
FB-AUTO	3,388	8	3,786	0	215	7,212	0	0	None	—
M-FB15k	10,314	71	400,027	26	11,220	0	0	0	None	—
WikiPeople	47,765	707	337,914	25,820	15,188	2,514	718	75	None	—
CB67	7,840	67	6,200	957	476	300	205	138	Time point	2014.10.1–2021.8.4

5 Experiments

The experiments in this section verify the validity, rationality, and robustness of THM on a real dataset. The design thought is as follows. The macroscopic and specific performance is compared with the previous knowledge hypergraph methods on the real temporal knowledge hypergraph dataset to prove the validity of the model design, and ablation experiments are performed to verify the rationality of the model design. Meanwhile, the robustness of the model design is verified through a parameter sensitivity experiment. Before analyzing the experimental results, how the real dataset of the temporal knowledge hypergraph is constructed should be first explained. Secondly, the compared knowledge hypergraph embedding models and the adopted experimental settings should be introduced, and finally, the experimental evaluation indicators should be given.

5.1 Dataset

In this paper, the CB67 dataset (Company Business, CB) was constructed by company groups according to the annual reports published by listed companies, and the data was desensitized. There were 7,840 entities including company type and person type, 67 relations including intra-company job relations, inter-company supply, and affiliation relations, and relation arity ranging from 2 to 7. The first entity in the relation-entity list is the company described by this temporal hyperedge, and the relation name is a combination of "relation name" and the number "relationship elements - 1" (where "-1" stands for the first element representing the described company in the entity list) to distinguish the fact that the same relation may be composed of different numbers of entities, which shows that "Supervisor 3" and "Supervisor 4" indicate a company's supervisors consisting of three and four persons respectively. The entities in a single temporal hyperedge are sorted in increasing order of the joining time. Company D and Company E joined the supply chain of Company A on March 14, 2020 and July 21, 2020 respectively, and Company B exited the supply chain of Company A on December 2, 2020; then the temporal hyperedges of supply relation of Company A in 2020 and 2021 were expressed as (Supply 2, A, B, C, 2020-01-01) and (Supply 3, A, C, D, E, 2021-01-01) respectively. The whole dataset had 8,276 temporal hyperedges, and the timestamp spanned from October 1, 2014 to August 4, 2021. The set of temporal hyperedges was shuffled, 89% of the hyperedges were randomly selected as the training set, and the remaining data were employed as the validation and test sets. Table 2 shows the statistics of this temporal knowledge hypergraph dataset and the currently public knowledge hypergraph dataset, and the "number of X-ary" in the table indicates the number of X-ary relation (temporal) hyperedges. At present, the CB67 dataset has been publicly available on https://github.com/zirui-chen/CB67.

5.2 Baseline model and experiment settings

Since no link prediction model can be directly applied to the temporal knowledge hypergraph structure, to make a fair comparison, this paper only selects the models that provide open source code and optimal experimental hyperparameters in link prediction of knowledge hypergraphs based on tensor decomposition, and further screens out the models that can process data with relations of different arity:

(1) m-TransH[3]. The TransH algorithm is extended from 2-ary to n-ary relations, and the n-ary relations are modeled at the hyperedge level to enable the score function to score the existence of unknown hyperedges.

(2) m-CP[29]. CP (Canonical polyadic) decomposition is a tensor decomposition-based method that can only process 2-arity relations, while the m-CP model can accommodate relations of any order of magnitude to achieve relational modeling of knowledge hypergraphs.

(3) m-DistMult[29]. The DistMult algorithm is extended to generalize the bilinear score function to n-ary relations to score unknown hyperedges.

(4) HSimplE[29]. Inspired by the SimplE model, this algorithm provides different embedding representations for entities according to their different positions in the corresponding hyperedges of different relations.

(5) HypE[29]. A corresponding position convolution weight converter is learned for each possible position of an entity in the knowledge hypergraph dataset. Meanwhile, entity embedding is obtained for entities at different positions through the corresponding converter and is subsequently combined with relation embedding and input into the score function to generate the existence probability of unknown hyperedges.

(6) RAM[12]. It is the current SOTA model for link prediction in knowledge hypergraphs. It starts from the level of entity role, proposes "role-aware modeling" for hyperedges in knowledge hypergraphs, and encourages close representation of semantically related roles.

These models will only adopt the relation and entity data in the temporal hyperedges for parameter optimization. Additionally, the optimal hyperparameters of each model on the JF17K dataset will be employed by default to complete the model training and predict the temporal hyperedge, and the obtained experimental data will be used for model performance comparison. The proposed model will additionally utilize the timestamp information in the temporal hyperedges for model training.

5.3 Evaluation indicators

The model performance is evaluated on the CB67 dataset by the two evaluation indicators mean reciprocal rank *MRR* and Hits rate *Hits@k* $(k = 1, 3, 10)$. The negative sampling method is as follows: given a set of temporal hyperedges, any positive example of temporal hyperedge in the test set is expressed by h. The entity e_i at position i in h is replaced by an entity in $E - \{e_i\}$ to construct q temporal hyperedges of the hyperparameter. Finally, a set of temporal hyperedge negative examples $N_q(h)$ corresponding to the temporal hyperedge positive example h at position i are constructed. Assuming $H_q(h) = \{h\} \cup N_q(h)$, $rank_q(h)$ means the ranking of temporal hyperedge positive example h in $H_q(h)$ based on the score function $\phi(\cdot)$. Assuming r is the relation corresponding to the temporal hyperedge positive example h and $\mathrm{cond}(\cdot)$ is the conditional function when the condition is valid, the value is 1, otherwise, it is 0. The specific formulas for *MRR* and *Hits@k* are as follows:

$$MRR = \frac{1}{\sum_{h \in H_{\text{test}}} |r|} \sum_{h \in H_{\text{test}}} \sum_{p=1}^{|r|} \frac{1}{rank_p(h)}$$

$$Hits@k = \frac{\sum_{h \in H_{\text{test}}} \sum_{p=1}^{|r|} \mathrm{cond}(rank_p(h) \leq k)}{\sum_{h \in H_{\text{test}}} |r|}$$

5.4 Experimental results

The following five questions are studied to evaluate the effectiveness of the link prediction model THM based on the temporal knowledge hypergraphs.

- Q1: Compared with the knowledge hypergraph embedding model which only employs the structural information of the temporal knowledge hypergraphs, can the THM model which additionally adopts the temporal information in the dataset yield better prediction results?
- Q2: What is the performance difference in the link prediction with relations of different arity between the temporal knowledge hypergraph embedding model THM and the knowledge hypergraph SOTA model RAM?
- Q3: Is the combination of static structure module, dynamic frequency module, and dynamic weight module in THM the best combination for link prediction?
- Q4: Does the parameter size of the THM model lead to overfitting of training and further affect the generalization ability of the model?
- Q5: Is the THM model robust in model training with different hyperparameter combinations?

5.4.1 *Q*1: *Effectiveness of THM*

Table 3 shows the experimental results of comparing the THM model with other baseline models of knowledge hypergraphs selected for the experiment on the dataset CB67. The experimental results indicate that the proposed THM model outperforms the baseline models in all evaluation indicators on the temporal knowledge hypergraph dataset. In terms of MRR indicators, the THM model improves by 14.01% compared with the SOTA model RAM embedded in the knowledge hypergraphs. This indicates that the model can effectively utilize the temporal information in the temporal hyperedges to accomplish the link prediction task and achieve effective utilization of temporal information in the dataset. According to the results in Table 3, the experimental results of these knowledge hypergraph algorithms on the temporal knowledge hypergraph dataset are not satisfactory. These algorithms not only need to employ static features such as entity roles and positions as the embedding learning contents but also make full use of the unique temporal dynamic information in the temporal hyperedges to accomplish link prediction based on the temporal knowledge hypergraph structure. From the perspective of static structure information, RAM is the current SOTA model, but the experimental results of this model on the temporal knowledge hypergraph dataset CB67 constructed in this paper show that its results are not optimal under the temporal knowledge hypergraph dataset. On the contrary, THM can achieve optimal results. Considering the differences between the design of the knowledge hypergraph embedding model and the CB67 dataset, the main reason for this phenomenon is that the entities within a single temporal hypergraph in the CB67 dataset are sorted in increasing order of entity joining time, rather than absolutely assigning the entity position according to the entity roles, and different entities in the same temporal hyperedge may have the same role. For example, in the temporal hyperedge (Supply 3, A, C, D, E, 2021-01-01), except for Company A, a company with a customer role in the supply chain, the latter three companies have a supplier role in the supply chain, and the difference in their orders is only the time of joining the supply chain. In this context, the score function design of THM, which completes the embedding product in entity order, can better capture the temporal differences in the order of temporal hyperedge entities in the CB67 dataset than RAM, which mainly takes into account the difference in entity role information. This experimental phenomenon further demonstrates the necessity of designing embedding models for temporal knowledge hypergraph data that conform to their structural properties. From the perspective of dynamic structural information, there are two main reasons for the lower results of RAM than THM. First, RAM does not further enhance the semantic influence of entity position information on the hyperedge compared with the static embedding in this paper. Second, RAM does not consider the dynamic trend of the hyperedge compared with the dynamic embedding in this paper. Based on comprehensively analyzing experimental results, the main reason for the optimal performance of the THM model is not only the consideration of the static structural information of the hyperedge but also the incorporation of unique temporal information in the temporal knowledge hypergraph as the dynamic embedding. Additionally, the combination of the two types of information through mixing in a certain proportion to make full use of the structural information and temporal information of the knowledge hypergraph to enhance the link prediction performance is also a reason.

Table 3 Link prediction results

Model	*MRR*	*Hits*@1	*Hits*@3	*Hits*@10
m-TransH	0.063	0.053	0.065	0.076
m-CP	0.196	0.171	0.215	0.234
m-DistMult	0.238	0.216	0.256	0.268
HSimplE	0.222	0.199	0.236	0.257
HypE	0.214	0.193	0.228	0.246
RAM	0.221	0.192	0.237	0.267
THM	0.257	0.235	0.281	0.289

5.4.2 *Q2*: *Performance of THM under n-ary relations*

Figure 3 shows the specific performance of THM under different relation arities. The performance of THM on CB67 under different relation arities is consistent and exceeds that of the RAM model. Its performance on the temporal hyperedge of higher arities is relatively weak, which is due to the uneven distribution of data arities in the dataset. Details can be referred to the temporal hyperedge statistics under relations with different arities in Table 2. Meanwhile, THM obtains significant results on the 2-ary relation data to verify its generalization ability on the 2-ary relations.

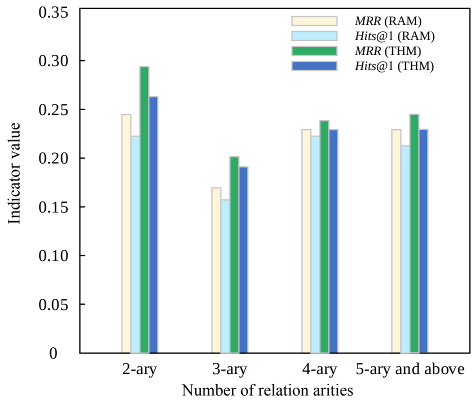


Figure 3 Specific performance under different numbers of relation arities

5.4.3 *Q*3: *Ablation experiment*

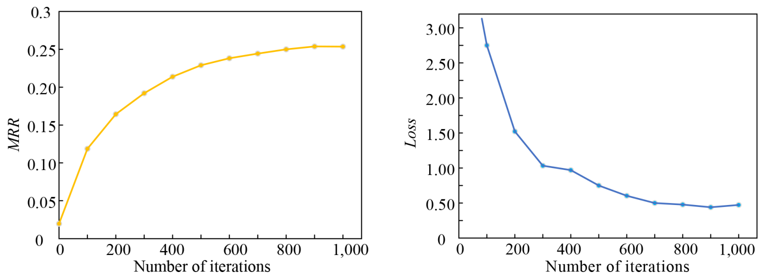
Six variants of the THM model were designed in this paper to verify the effect of each module on the model performance. In Table 4, the model variants from top to bottom are pure structure, pure frequency, pure weight, structure + frequency, structure + weight, and frequency + weight respectively. According to Table 4, the THM model outperforms its six variants in all indicators. Specifically, in terms of the *Hits*@1 indicator, the THM model outperforms the six variants by 8.93%, 94.47%, 95.32%, 1.70%, 2.12%, and 93.61% respectively. It is obvious that the lack of any module makes the model less effective, and the effect of link prediction using static embedding only is better than that of link prediction using dynamic temporal information only. This indicates that static embedding is still the main information adopted by the model for link prediction, while the addition of temporal information can enrich the information from the dynamic perspective and improve the prediction performance of the model. The static embedding, temporal frequency, and temporal weight of the THM model can well capture the interaction between entities and relations and the changing trend of facts over time.

Table 4 Results of ablation experiments on CB67 dataset

Model	*MRR*	*Hits*@1	*Hits*@3	*Hits*@10
THM-S	0.233	0.214	0.251	0.259
THM-F	0.014	0.013	0.016	0.017
THM-W	0.010	0.011	0.016	0.016
THM-SF	0.253	0.231	0.273	0.279
THM-SW	0.251	0.230	0.274	0.281
THM-FW	0.016	0.015	0.017	0.018
THM-SFW	0.257	0.235	0.281	0.289

5.4.4 *Q*4: *Whether there is overfitting in THM training*

Models with a large number of parameters are prone to overfitting the training data, which can impair the generalization performance and affect the performance of the test set. The model was tested for overfitting through the early stop technique during the training to verify the performance of THM in terms of overfitting. The training curves in terms of *MRR* and loss are plotted in Figure 4(a) and 4(b) respectively, and the *MRR* gradually converges after a rapid increase as the training iterations proceed. For the loss curve, the loss decreases rapidly and slowly during the first 300 iterations and 300–700 iterations and gradually converges after 700 iterations.



(a) *MRR* vs. Number of iterations (b) *Loss* vs. Number of iterations

Figure 4 Overfitting analysis according to *MRR* and *loss*

5.4.5 *Q5*: *Robustness of THM*

The influence of the hyperparameter settings on the algorithm performance was analyzed on the CB67 dataset in terms of three hyperparameters of the model design to verify the robustness of the THM model. The hyperparameters include embedding dimension d, negative sampling rate q, and dynamic embedding ratio λ. The experimental tests were conducted with the embedding dimension $d \in \{50, 100, 150, 200, 250, 300, 350, 400\}$, the negative sampling rate $q \in \{2, 4, 6, 8, 10, 12, 14, 16\}$, and the dynamic embedding ratio $\lambda \in \{0, 0.2, 0.4, 0.6, 0.8, 1\}$.

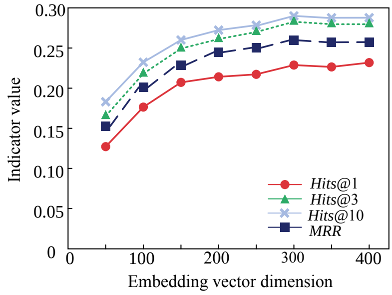
Figure 5(a) shows the line chart of the model performance at different embedding dimensions. With the increasing embedding dimension, the model performance firstly improves rapidly, and the four performance indicators stabilize after the embedding dimension reaches 300.

Figure 5(b) shows the line chart of the model performance at different negative sampling rates, which shows that the negative sampling rate has little effect on the performance of the THM model. This hyperparameter is stable for the model performance, and the changes in the negative sampling rate will not cause large performance differences.

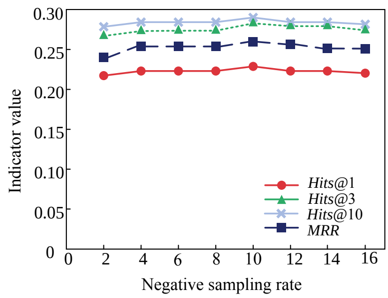
Figure 5(c) shows the line chart of the model performance at different dynamic embedding ratios, which shows that the dynamic embedding ratio is a sensitive hyperparameter for the model. When the dynamic embedding ratio is low, the model mainly learns the static structural information of the temporal knowledge hypergraph, and the performance of the four indicators of the model gradually improves with the rising dynamic embedding ratio. However, the model performance decreases sharply when the dynamic embedding ratio is larger than 0.8. This indicates that the prediction performance of the model can be improved based on the static structural information and supplemented by the temporal information, while the under-consideration of the difference in entity role positions may not be helpful to model learning. Therefore, different values should be assigned to this parameter according to the actual applications.

6 Conclusion

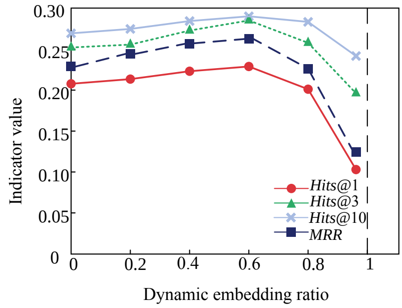
This paper proposed a link prediction model of temporal knowledge hypergraphs THM based on the definition of temporal knowledge hypergraphs. The model employs the static structural information of the temporal knowledge hypergraph, including the role and position



(a) Influence of parameter embedding vector dimension on the indicator



(b) Influence of parameter negative sampling rate on the indicator



(c) Influence of parameter dynamic embedding ratio on the indicator

Figure 5 Parameter sensitivity analysis

information of entities, and the compatibility information of relations with entity roles. Meanwhile, the dynamic temporal information of the temporal knowledge hypergraph is also adopted, containing the timestamp frequency and weight information, and the two types of embedding vectors are mixed in a certain proportion to generate the final entity representation. The structural temporal information in the temporal knowledge hypergraph was made full use of to improve the performance indicators of the downstream link prediction task. At the same time, it was theoretically proven that THM is fully expressive and has linear space complexity. In addition, the first temporal knowledge hypergraph dataset CB67 was constructed from the public business data, and a large number of experimental evaluations were conducted on this dataset. The experimental results showed that THM can effectively perform the link prediction task on the temporal knowledge hypergraph dataset. Feasible future research directions include modeling the temporal knowledge hypergraph as a complete dynamic graph, instead of combining static structures with dynamic temporal representations. Additionally, inductive learning can be implemented to possess the ability to embed representation for entity relations at timestamps in non-training sets.

References

[1] Wang X, Zou L, Wang CK, Peng P, Feng ZY. Research on knowledge graph data management: A survey. Ruan Jian Xue Bao/Journal of Software, 2019, 30(7): 2139–2174 (in Chinese with English abstract). http://www.jos.org.cn/1000-9825/5841.htm [doi: 10.13328/j.cnki.jos.005841]

[2] Bollacker K, Evans C, Paritosh P, Sturge T, Taylor J. Freebase: A collaboratively created graph database for structuring human knowledge. Proc. of the 2008 ACM SIGMOD Int'l Conf. on Management of Data. New York: Association for Computing Machinery, 2008. 1247–1250.

[3] Wen J, Li J, Mao Y, Chen S, Zhang R. On the representation and embedding of knowledge bases beyond binary relations. arXiv:1604.08642, 2016.

[4] Fatemi B, Taslakian P, Vazquez D, Poole D. Knowledge hypergraphs: Prediction beyond binary relations. arXiv:1906.00137, 2019.

[5] Ernst P, Siu A, Weikum G. Highlife: Higher-arity fact harvesting. Proc. of the 2018 World Wide Web Conf. Int'l World Wide Web Conf. Steering Committee, 2018. 1013–1022.

[6] Zhang F, Yuan NJ, Lian D, Xie X, Ma WY. Collaborative knowledge base embedding for recommender systems. Proc. of the 22nd ACM SIGKDD Int'l Conf. on Knowledge Discovery and Data Mining. New York: Association for Computing Machinery, 2016. 353–362.

[7] Lukovnikov D, Fischer A, Lehmann J, Auer S. Neural network-based question answering over knowledge graphs on word and character level. Proc. of the 26th Int'l Conf. on World Wide Web. Steering Committee, 2017. 1211–1220.

[8] Hou ZN, Jin XL, Chen JY, Guan SP, Wang YZ, Cheng XQ. Survey of interpretable reasoning on knowledge graphs. Ruan Jian Xue Bao/Journal of Software, 2022, 33(12): 4644–4667 (in Chinese with English abstract). http://www.jos.org.cn/1000-9825/6522.htm [doi: 10.13328/j.cnki.jos.006522]

[9] Yang DH, He T, Wang HZ, Wang JB. Survey on knowledge graph embedding learning. Ruan Jian Xue Bao/Journal of Software, 2022, 33(9): 3370–3390 (in Chinese with English abstract). http://www.jos.org.cn/1000-9825/6426.htm [doi: 10.13328/j.cnki.jos.006426]

[10] Richardson M, Domingos P. Markov logic networks. Machine Learning, 2006, 62(1): 107–136.

[11] Wang Z, Zhang J, Feng J, Chen Z. Knowledge graph embedding by translating on hyperplanes. Proc. of the 2014 AAAI Conf. on Artificial Intelligence. Palo Alto: AAAI, 2014. 1112–1119.

[12] Liu Y, Yao Q, Li Y. Role-aware modeling for n-ary relational knowledge bases. Proc. of the 2021 Web Conf. 2021. New York: Association for Computing Machinery, 2021. 2660–2671.

[13] Xu F, He F, Xie E, Li L. Fast OBDD reordering using neural message passing on hypergraph. arXiv:1811.02178, 2018.

[14] Hu BD, Wang XG, Wang XY, Song ML, Chen C. Survey on hypergraph learning: Algorithm classification and application analysis. Ruan Jian Xue Bao/Journal of Software, 2022, 33(2): 498–523 (in Chinese with English abstract). http://www.jos.org.cn/1000-9825/6353.htm [doi: 10.13328j.cnki.jos.006353]

[15] Yang B, Yih WT, He X, Gao J, Deng L. Embedding entities and relations for learning and inference in knowledge bases. arXiv:1412.6575, 2014.

[16] Trivedi R, Dai H, Wang Y, Song L. Know-evolve: Deep temporal reasoning for dynamic knowledge graphs. Proc. of the 2017 Int'l Conf. on Machine Learning. 2017. 3462–3471.

[17] García-Durán A, Dumančić S, Niepert M. Learning sequence encoders for temporal knowledge graph completion. arXiv:1809.03202, 2018.

[18] Bordes A, Usunier N, Garcia-Duran A, Weston J, Yakhnenko O. Translating embeddings for modeling multi-relational data. Proc. of the 26th Int'l Conf. on Neural Information Processing Systems. New York: Curran Associates Inc., 2013. 2787–2795.

[19] Leblay J, Chekol MW. Deriving validity time in knowledge graph. Proc. of the 2018 Int'l World Wide Web Conf. 2018. 1771–1776.

[20] Sadeghian A, Armandpour M, Colas A, Wang DZ. ChronoR: Rotation based temporal knowledge graph embedding. Proc. of the 2021 AAAI Conf. on Artificial Intelligence. Palo Alto: AAAI, 2021. 6471–6479.

[21] Schlichtkrull M, Kipf TN, Bloem P, Berg RV, Titov I, Welling M. Modeling relational data with graph convolutional networks. Proc. of the 2018 European Semantic Web Conf. Cham: Springer, 2018. 593–607.

[22] Li Z, Jin X, Li W, Guan S, Guo J, Shen H, Wang Y, Cheng X. Temporal knowledge graph reasoning based on evolutional representation learning. Proc. of the 44th Int'l ACM SIGIR Conf. on Research and Development in Information Retrieval. New York: Association for Computing Machinery, 2021. 408–417.

[23] Kazemi SM, Buchman D, Kersting K, Natarajan S, Poole D. Relational logistic regression. Proc. of the 14th Int'l Conf. on the Principles of Knowledge Representation and Reasoning. Vancouver: University of British Columbia, 2014.

[24] Kazemi SM, Buchman D, Kersting K, Natarajan S, Poole D. Relational logistic regression. Proc. of the

14th Int'l Conf. on the Principles of Knowledge Representation and Reasoning. Vancouver: University of British Columbia, 2014.

[25] Abboud R, Ceylan I, Lukasiewicz T, Salvatori T. Boxe: A box embedding model for knowledge base completion. Advances in Neural Information Processing Systems. New York: Curran Associates Inc., 2020. 9649–9661.

[26] Liu Y, Yao Q, Li Y. Generalizing tensor decomposition for n-ary relational knowledge bases. Proc. of the 2020 Web Conf. New York: Association for Computing Machinery, 2020. 1104–1114.

[27] Balažević I, Allen C, Hospedales TM. Tucker: Tensor factorization for knowledge graph completion. arXiv:1901.09590, 2019.

[28] Hitchcock FL. The expression of a tensor or a polyadic as a sum of products. Journal of Mathematics and Physics, 1927, 6(1–4): 164–189.

[29] Fatemi B, Taslakian P, Vazquez D, Poole D. Knowledge hypergraphs: Prediction beyond binary relations. arXiv:1906.00137, 2019.

[30] Kazemi SM, Poole D. Simple embedding for link prediction in knowledge graphs. Advances in Neural Information Processing Systems. Red Hook: Curran Associates Inc., 2018. 4289–4300.

[31] Di S, Yao Q, Chen L. Searching to sparsify tensor decomposition for n-ary relational data. Proc. of the 2021 Web Conf. New York: Association for Computing Machinery, 2021. 4043–4054.

[32] Guan S, Jin X, Wang Y, Cheng X. Link prediction on n-ary relational data. Proc. of the 2019 World Wide Web Conf. New York: Association for Computing Machinery, 2019. 583–593.

[33] Rosso P, Yang D, Cudré-Mauroux P. Beyond triplets: Hyper-relational knowledge graph embedding for link prediction. Proc. of the 2020 Web Conf. New York: Association for Computing Machinery, 2020. 1885–1896.

[34] Guan S, Jin X, Guo J, Wang Y, Cheng X. Neuinfer: Knowledge inference on n-ary facts. Proc. of the 58th Annual Meeting of the Association for Computational Linguistics. Association for Computational Linguistics, 2020. 6141–6151.

[35] Galkin M, Trivedi P, Maheshwari G, Usbeck R, Lehmann J. Message passing for hyper-relational knowledge graphs. arXiv:2009.10847, 2020.

Zirui Chen, master. His research interests include knowledge representation learning.

Shaowei Zhang, master. His research interests include knowledge representation learning and knowledge graph construction.

Xin Wang, Ph.D., professor, doctoral supervisor. His research interests include knowledge graph data management, graph database, distributed processing of big data.

Haoyu Yan, master. His research interest is knowledge representation learning.

Chenxu Wang, master. His research interest is knowledge representation learning.