AHyTE: An Adaptive Hyperplane-based Method for Time-aware Knowledge Graph Embedding

Dawei Yang¹, Gang Zhou⁴*, Jicang Lu³, Yuanlong Ning⁴

¹Information Engineering University, Zhengzhou, Henan, 450001, China
²Information Engineering University, Zhengzhou, Henan, 450001, China
³Information Engineering University, Zhengzhou, Henan, 450001, China
⁴Information Engineering University, Zhengzhou, Henan, 450001, China

*Corresponding author’s e-mail: 141110095@smail.nju.edu.cn

Abstract: Knowledge graph (KG) embedding can learn the representations of entities and relations from a KG in low-dimensional continuous vector space, which is the foundation of knowledge acquisition and reasoning. Time-aware KG embedding is a hot research branch of KG embedding, which introduces time information while considering the knowledge expression ability to make knowledge dynamic. Among them, t-TransE model is the most classic. However, t-TransE has the following two flaws: one is that it only regards time information as a constraint and the embedding learned by this way is not explicitly temporally aware; the other is that it uses traditional Euclidean distance as a metric, and all feature dimensions are treated identically, hence, the model may not accurately reflect the priority of features. Currently, there has not been a single method resolves the flaws simultaneously, so we propose an adaptive hyperplane-based time-aware knowledge graph embedding model, namely, AHyTE. AHyTE associates each time stamp with the corresponding hyperplane, and introduces a diagonal weight matrix to assign weights to each feature dimension. Through link prediction experiments on the time data set extracted from the real world, we demonstrate the remarkable improvement of AHyTE.

1. Introduction

Recently, the explosion of various data on the Internet has been triggered by the rapid development of Internet technology and Web applications, which contains a large amount of valuable knowledge. How to find an effective way to organize, represent, and analyze these knowledge has attracted much attention [1]. Knowledge graph thus came into being. Knowledge Graphs (KGs) such as DBpedia, YAGO, and Freebase are multi-relational graphs describing entities and their relationships in the form of triple (entity, relation, entity), e.g., (Zhengzhou, islocatedin, Henan). Techniques of KG embedding can learn the representations of entities and relations in low-dimensional vector spaces, and these embeddings can be used to support knowledge acquisition and reasoning [2].

Time-aware KG embedding is an important branch of KG embedding as KGs in the real word are changing over time. For example, the triple (Barack Obama, ispresidentOf, America) is true only during 2009-2017. The earliest time-aware KG embedding model is t-TransE, which learns temporal order among relations, e.g., (wasBornIn->graduateFrom->workAt->diedIn) [3]. Through t-TransE was proved its effectiveness, the model still has two flaws: for one hand, it only regards time information as a constraint and the embedding learned by this way is not explicitly temporally aware; for another
hand, it uses traditional Euclidean distance as a metric, where all feature dimensions are treated identically, hence, the model may be interfered by irrelevant dimensions.

Based on the analysis above, in this paper, we propose an adaptive hyperplane-based time-aware KG embedding (AHyTE) method, which incorporates temporal information in the learned embeddings directly and takes the adaptive metric ideas for better knowledge representation.

2. Related work
The existing knowledge graph embedding models are mainly divided into translation-based models, semantic-matching models and models that consider additional information [4]. The semantic matching model is represented by the RESCAL [5], focusing on mining the latent semantics of the entities and relationships after vectorization; the models that consider additional information are differ from each other, there are models that consider entity types, relation paths (PTransE) [6], and text descriptions, etc. Models based on translating are the main focus of research, and representative work includes methods such as TransE [7], TransH [8], TransA[9], and TransAH [10].

Although the above-mentioned KG embedding methods are constantly updated and improved, they all ignore time information. As an independent dimension of knowledge, time plays a vital role in the construction and completion of the KG. The earliest time-aware KG embedding is t-TransE, which considers time sequence and time consistency on the basis of TransE. Specifically, the author defines a time evolution matrix \( \mathbf{T} \), and assumes that the prior relation \( r_1 \) can evolve into a successor relation \( r_2 \) through the time evolution matrix \( \mathbf{T} \), but the successor relation cannot evolve into a prior relation through the time evolution matrix, so the time information is taken as constraint [3].

Besides, [11] proposes the use of bilinear embedding learning method to model the nonlinear time evolution of KG elements--Know-Evolve. This model uses a recurrent neural network to learn the embedded nonlinear dynamic features. Compared with the traditional model, the model is significantly improved, but it is limited to event-based interaction type data and does not perform well on other datasets; [12] conducts to combine the traditional KG embedding model with the use of additional information in the permanent part of the graph for temporal embedding learning; [13] proposed to split the effective time of triples into slices and project entity and relation to their own time slice to learn the embedding, which inspired us to design our model; and [14] combines hyperplane model and duration model to effectively use the distribution of knowledge valid duration.

3. AHyTE model
This section specifically introduces the principle and algorithm of our method AHyTE in detail. Aiming at the two flaws of t-TransE, AHyTE conducts a time-oriented hyperplane method and an adaptive measurement method respectively, combining the two under the same framework, which significantly improves knowledge express ability.

3.1 Time-oriented hyperplane method
As static KG describes entities and their relations in the form of triple, we represent knowledge with temporal annotations by quadruples, e.g., \((h, r, t, [\tau_s, \tau_e])\), where \(\tau_s\) and \(\tau_e\) denote the start and end time during which the triple \((h, r, t)\) is valid. Due to the large scale, long time span, and high complexity of the KG, an entity or relation is usually only valid in a part of time in the graph and considering it in all time states can easily affect the accuracy of the model. For this we dismantle KG into several subgraphs by the given timestamps, e.g., \(G = \bigcup_{i=1}^{T} G_{\tau_i}\) for \(\{\tau_i | i = 1, 2, \ldots T\}\). A quadruple \((h, r, t, [\tau_s, \tau_e])\) is considered to be positive for some timestamp \(\tau_i\) only when \(\tau_s \leq \tau_i \leq \tau_e\), and the set of positive quadruple of \(\tau_i\) is denoted as \(D_{\tau_i}^+\).
Furthermore, we represent time as the hyperplane, and the T discrete time points in the KG divide the graph into T different spaces, each corresponding to a hyperplane, and the hyperplane is represented by its normal vector, denoted as \( \{n_{t_i}\}_{i=1}^{T} \). All positive quadruples will be projected onto the time hyperplane, where their translational distance is minimized and the entity or relation embedding will be different on different hyperplanes for their normal vector differ from each other, as shown in Figure 1.

![Figure 1. Simple illustration of Time-oriented hyperplane.](image)

Specifically, for any i, the embedding on the hyperplane is expressed as follows:

\[
\begin{align*}
    h^+ &= h - (n_{t_i}^T h)n_{t_i} \\
    r^+ &= r - (n_{t_i}^T r)n_{t_i} \\
    t^+ &= t - (n_{t_i}^T t)n_{t_i}
\end{align*}
\]

(1)

Where we restrict \( ||n_{t_i}||_2 = 1 \) to prevent overfitting.

Since time is the reason of most different many-to-one, one-to-many, or many-to-many relations, by projecting to the time-oriented hyperplane, not only can dynamic quadruples be transformed into relatively static triples, but also can effectively solve the ambiguity of knowledge to make the model perform better in complex relations. Further, we consider the design of score function.

3.2 Adaptive measurement method

The existing translation-based embedding methods obey the same principle: \( \text{head} + \text{relation} \approx \text{tail} \) and share a similar score function that:

\[
f_r(e_h, e_r) = ||e_h + e_r - e_t||^2 = (e_h + e_r - e_t)^T (e_h + e_r - e_t)
\]

(2)

Where \( ||e_h + e_r - e_t|| = \prod_{i=1}^{k} ||e_{h_i} + e_{r_i} - e_{t_i}|| \) and \( k \) is the length of feature dimension.

However, the Euclidean distance measurement form a spherical equipotential surface shape in space, taking all feature dimensions into consideration, which is often inflexible in real applications and the accuracy may be effected [9]. Therefore, we introduce a diagonal weight matrix to the score function, which has the following three advantages:

- In the equipotential surface, the one near the center is the positive sample. Taking weight matrix can replace the spherical equipotential surface with the elliptical equipotential surface, and with its stretch characteristic, the positive samples can be increased while avoiding negative samples to improve the accuracy of the model.
• Usually, a relation is only affected by a few specific potential features, and the addition of others will cause negative effects. The traditional score function treats all feature dimensions equally and cannot filter irrelevant dimensions, making the model not accurate enough. The weight matrix can make each dimension be reasonably controlled by the weight to avoid the occurrence above.

• The diagonal weight matrix is used instead of the general weight matrix because the comparison between features is meaningless and will increase the computational cost.

The elements of diagonal weight matrix are non-negative to prevent negative influence to optimization goals and could be worked out directly by setting the derivation to zero. The diagonal weight matrix is denoted as $D_r$ and its initial value is:

$$D_r = \text{diag}(-\sum_{(h,r,t) \in D^+_r} |h + r - t|^2 + \sum_{(h',r',t') \in D^-_r} |h' + r' - t'|^2)$$

(3)

3.3 AHyTE model

Based on the contents above, AHyTE puts the time-oriented hyperplane method and adaptive measurement method under a unified framework, and its scoring function and optimization goals are:

$$f_{ti}(h,r,t) = ((h - (n_i^h \cdot h_{n_i}^r) + r - (t - n_i^t \cdot t_{n_i}^r))^T \bullet D_r \bullet ((h - (n_i^h \cdot h_{n_i}^r) + r - (t - n_i^t \cdot t_{n_i}^r))$$

(4)

$$\mathcal{L} = \min \sum \sum \sum \max(f_{ti}(x) - f_{ti}(y) + \gamma, 0)$$

s.t.

$$\begin{align*}
\forall e \in E, \|e\|_2 & \leq 1, \\
\forall r \in R, \|r\|_2 & \leq 1, \\
\forall i \in T, \|n_i^r\|_2 & = 1 \\
\forall i \in T, \|n_i^r \cdot \tau_i \|_2 & \leq \varepsilon \\
\| D_r \|_2 & \leq 1,
\end{align*}$$

(5)

$$D^+_r = \{(h',r,t,\tau_i) | h' \in E, (h',r,t) \notin D^+ \}$$

(6)

$$\bigcup \{(h,r,t',\tau_i) | t' \in E, (h,r,t') \notin D^+ \}$$

Where $E$ is the set of all entities, $D^+$ is the set of positive samples, $D^-_r$ is the negative set of $\tau_i$, $\gamma$ is the margin and the constraints ensure that the entity or relation vector is not greater than 1 and the translation vector is on the hyperplane. In order to comprehensively consider the constraints in the calculation, we choose to integrate them into the goal function and add parameters $\lambda$ and $\mu$ to prevent overfitting. The final objective function is as follows:

$$\mathcal{L} = \min \sum \sum \sum \max(f_{ti}(x) - f_{ti}(y) + \gamma, 0) + \lambda(\sum_{r \in R} \|D_r\|_F^2) +$$

$$\mu(\sum_{e \in E} \|e\|^2 + \sum_{r \in R} \|r\|^2 + \sum_{i \in T} \left(\frac{(n_i^r \cdot \tau_i)^2}{\| \tau_i \|^2} - \varepsilon^2 \right))$$

(7)
In model training, we use the classic stochastic gradient descent method to optimize the objective function and positive examples will loop multiple times to generate negative examples according to different tasks. After a period of training, the model will be automatically updated.

4. Experiment and Analysis

4.1 Datasets

We tested our model on YAGO11k and Wikidata12K, the temporally rich subgraphs extracted from Wikidata and YAGO, which are common datasets in the field of KG embedding. To make sure of the healthy connectivity within the graph, edges containing entity with only one mention are removed recursively [13]. The statistics about the datasets are shown in the Table 1.

| Dataset          | #R | #E       | #Train | #Test | #Valid |
|------------------|----|----------|--------|-------|--------|
| YAGO11K          | 10 | 10623    | 16408  | 2051  | 2046   |
| Wikidata12K      | 24 | 12554    | 32497  | 4062  | 4062   |

4.2 Entity prediction

Predicting the missing entity is the aim of this task, including head entity and tail entity. Specifically, given an incomplete quadruple \((?, r, t, \tau)\) or \((h, r, ?, \tau)\), we predict the \(h\) or \(t\) correspondingly. The negative samples are generated by randomly replace an entity as eq.6 illustrated. We compute scores for the candidate set, which consists of positive and negative samples, and rank them in the increasing order to get the rank of positive triples. We report the mean rank (MR) over all the test queries and proportion of correct entities in top 10 rank (Hits@10) [7].

4.3 Relation prediction

The task here is to predict the relation between two entities, i.e., given \((h, ?, t, \tau)\), we predict the relation \(\rho\). Similar to the entity prediction, we corrupt the quadruples with all possible relations and report the rank of actual relation [7]. Due to the number of relations are much less in both datasets, here, we report Hits@1 rather than Hits@10 for this task and MR is reported as well.

4.4 Results

We have tried several settings on datasets to get the best configuration for AHyTE. The optimal configurations are: the batch size \(b = 50k\), the dimensions of embeddings \(d = 128\), learning rate used for SGD \(l_\gamma = 0.0001\), the margin \(\gamma = 1\), the weight \(\mu = 0.85\) on Wikidata12K; and \(b = 50k\), \(d = 128\), \(l_\gamma = 0.0005\), \(\gamma = 5\), \(\mu = 0.35\) on YAGO11K.

The results reported in Table 2 and Table 3 demonstrate the efficacy of our model. To illustrate the results, the best values are in bold, and better value in the MR is lower while the Hits@10/Hits@1 is higher. In comparison with the traditional model TransE and TransH, AHyTE performs better due to the direct inclusion of time in the relation-entity semantic space. Also, AHyTE outperforms the time-aware model t-TransE and HyTE as we introduce a diagonal weight matrix to reduce the interference of irrelevant dimensions.

| Dataset          | YAGO11K | Wikidata12K |
|------------------|---------|-------------|
| method           | Mean Rank | Hits@10(%) | Mean Rank | Hits@10(%) |
| TransE           | head    | tail        | head    | tail        | head    | tail    |
|                  | 2020    | 504         | 1.2     | 4.4         | 740     | 520     | 6.0     | 11.0     |
|       | YAGO11K |             | Wikidata12K |             |
|-------|---------|-------------|-------------|-------------|
|       | Mean Rank | Hits@1(%)  | Mean Rank   | Hits@1(%)   |
| TransE | 1.7      | 78.4        | 1.35        | 88.4        |
| TransH | 1.53     | 76.1        | 1.4         | 88.1        |
| T-TransE | 1.66    | 75.5        | 1.97        | 74.2        |
| HyTE  | 1.23     | 81.2        | 1.13        | 92.6        |
| AHyTE | **1.12** | **87.2**    | **1.08**    | **93.1**    |

### 5. Conclusion

We propose AHyTE, an adaptive hyperplane-based time-aware knowledge graph embedding model. Our model exploits temporally information to perform link prediction with an adaptive matrix and we demonstrate the superiority of AHyTE on the real world datasets. In future, we would like to combine the description of entities and relations with our model to get the further improvement.

### Acknowledgments

This paper was supported by the Science and Technology Research Project of Henan Province under Grant Nos. 192102210129.

### References

[1] Saiping Guan, Xiaolong Jin, Yuanzhuo Wang, Yantao Jia, Huawei Shen, Zixuan Li, Xueqi Cheng. (2018) Self-learning and embedding based entity alignment , Knowledge and Information Systems

[2] Guan S, Jin X, Jia Y, Wang Y, Cheng X. (2018) Knowledge graph oriented knowledge inference methods: A survey. Ruan Jian Xue Bao/Journal of Software, 29(10) (in Chinese).

[3] Tingsong Jiang, Tianyu Liu, Tao Ge, Lei Sha, Sujian Li, Baobao Chang, and Zhifang Sui. (2016) Encoding temporal information for time-aware link prediction. In: Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing, pages 2350–235.

[4] Wang H.F, Qi G.L, Chen J. (2019) Knowledge Graph. PHEI Publishing, The beijing.

[5] Maximilian Nickel, Volker Tresp, Hans-Peter Kriegel. (2011) A Three-way Model for Collective Learning on Multi-Relational Data.ICML,:809-816

[6] LIN Yankai, LIU Zhiyuan, LUAN Huanbo, et al. (2015) Modeling Relation Paths for Representation Learning of Knowledge Bases. In : EMNLP, 205-714

[7] Bordes A, Usunier N, Garcia-Duran A, et al. (2013) Translating embeddings for modeling multi- relational data, Proceedings of Advances in Neural Information Processing Systems. Red Hook. Curran Associates Inc. NY, 2787-2795.

[8] Wang Z, Zhang J, Feng J, et al.Knowledge graph embedding by translating on hyperplanes, In: Twenty-eighth AAAI Conference on Artificial Intelligence 2014.

[9] Xiao Han, Huang Minlie, Hao Yu, et al. (2016) TransA: An adaptive approach for knowledge graph embedding. Computer Science.

[10] Fang Y, Zhao X, Tan Z.et al. (2018) A revised translation-based method for knowledge graph representation. Journal of Computer Research and Development, 55(1): 139-150 (in Chinese)
[11] Rakshit Trivedi, Hanjun Dai, Yichen Wang, and Le Song. (2017) Know-evolve: Deep temporal reasoning for dynamic knowledge graphs. In: Proceedings of the 34th International Conference on Machine Learning.

[12] Tim Dettmers, Pasquale Minervini, Pontus Stenetorp, and Sebastian Riedel. 2018. Convolutional 2d knowledge graph embeddings. In: Proceedings of the Thirty-Second AAAI Conference on Artificial Intelligence.

[13] Dasgupta S S, Ray S N, Talukdar P. (2018) Hyte: Hyperplane based temporally aware knowledge graph embedding. In: Proc of the 2018 Conf on Empirical Methods in Natural Language Processing. Stroudsburg, ACL: 2001-2011

[14] Cui Y N, Li J, Shen L, et al. (2020) Duration-HyTE: A Time-Aware Knowledge Representation Learning Method Based on Duration Modeling. Journal of Computer Research and Development, 57(6): 1239-1251