Machine Learning for Graphs

Do the Embeddings Translate on Hyperplanes?

Zheng Chen

studentnumber: 2853664

Abstract. The knowledge graph is a widely applied information storage structure. To capture potential knowledge and leverage numeric methods(e.g. deep learning), graphs need a numeric and continuous representation. TransH is an efficient method to map knowledge graphs into vector spaces. However, a gap between the motivation and the experiment setting of TransH can not be ignored. The projection operation might not hold with the soft constraints settings. To explore the effectiveness of the soft constraints, we reproduced TransH and TransE and then conducted an ablation study. We also provided a theoretical analysis of the expressiveness of TransH on different relation types.

Keywords: Knowledge Representation and Reasoning · Graph Embedding · Representation learning

1 Introduction

The knowledge graph is a multi-relational graph that stores relations(edges) between entities(nodes). It is essential to areas from question-answering[3] to protein-target prediction[9]. To better explore potential knowledge(e.g. missing links) with numeric methods, a mapping from graphs to numeric representation is needed.

Early representation learning methods, like RESCAL [10], are more factorization-based, which are in a dilemma of expressiveness and complexity, as [2] pointed out. The emergence of energy-based methods [5, 1] significantly reduced the parameter and validated the effectiveness of simplified relation modeling.

A more intuitive model, TransE [2], attempts to model the relation as translating in the space. However, TransE fails to model complex relation patterns such as many-to-1 relations. TransH [13] is a pioneer in enhancing the expressiveness of TransE by mapping the translation into other (sub)spaces. It leverages the projection on different hyperplanes to model different relations. Despite the recent advances in utilizing different translations in the space[11, 12] and more powerful modeling methods like graph neural networks[8], early translating-based methods remain competitive for their simplicity.

Motivated by the fact that different embeddings can share the same projection on hyperplanes, TransH models relations as translation between the projections. In order to restrict the translation on the hyperplane, TransH proposes a soft constraint strategy, which combines the restriction into the loss function with a weight parameter. In practice, the weight parameter is small, which conflicts with the theory that a large enough weight is needed to make the constraint effective. An underlying question appears: **"Do the embeddings really translate on the hyperplane?"** or does the performance improve only for the increase in parameters? No official source code released adds to the difficulty in interpreting the experiment setting.

This report will focus on the reproduction of TransH and attempt to explain the gap between the theory and experiment setting in the original paper through an ablation study. Moreover, a theoretical analysis of TransH's expressiveness is also included.

1.1 Contribution

- This paper reproduced TransE and TransH[1]. TransH has no official code released. Our reproduction might provide a possible way of better understanding the TransH paper.
- Our ablation study reveals the sensitivity of the hyperparameter setting of TransH, especially the constraint parameter. Possible solutions are discussed.
- A theoretical analysis, including mathematical proof, of the expressive on different relation types is conducted.

[1] `https://github.com/LilOpal/ML4G.git`

2 Related Work

The most related work is TransH [13]. Our report attempts to reproduce the results and provide possible explanations for unclear parts. In the original, the soft constraint parameter setting appears inconsistent with the theory. An ablation study is conducted to verify the necessity of the constraints.

TransE [2], as the first translation-based method, is a widely applied baseline. TransH only the result from the TransE paper in the link prediction task under a different setting. Our reports also reproduced TransE in a relatively uniform setting to compare with TransH.

Our mathematical analysis of the expressiveness of different relation types is inspired by RotatE [11]. However, the result differs with RotatE. In RotatE, it tries to offer a universal expressiveness of the TransX family with the loss of specific cases including TransH. Strict proof is also not provided by RotatE. Our report limits the scope to TransH and tries to offer a special case proof.

3 Background

A knowledge graph can be represented by ordered triples as (h, r, t), where h and t are entities(nodes) in the graph and r represents the relation between h and t. To map the entities and relations into a vector space while preserving the semantics in the graph, r is mapped to a translating vector $\mathbf{r}$. h and t are also mapped as vectors $\mathbf{h}$ and $\mathbf{t}$. $\mathbf{r}$ should translate $\mathbf{h}$ to $\mathbf{t}$. Mathematically, $\mathbf{h}+\mathbf{r} = \mathbf{t}$. To fit the optimization and generalization, a function $f_r(\mathbf{h}, \mathbf{t})$ is defined in the translation-based methods to measure how "well" the transformation is applied. in TransE, $f_r(\mathbf{h}, \mathbf{t}) = \|\mathbf{h} - \mathbf{t} + \mathbf{r}\|$ and then minimized to induce the vector embedding.

However, an unneglectable flaw is observed. Relations can be classified into one-to-one, many-to-one, one-to-many, and many-to-many, according to the number of head entities and tail entities. Relationships, except for the one-to-one relation, cannot be represented by TranE in principle. Consider a simple 1-to-Many relation: `('Body', 'hasPart', 'Hand')` and `('Body', 'hasPart', 'Foot')`. If the TranE is perfectly optimized, `'Hand'` should share the same embedding with `'Foot'`, which can hardly benefit the downstream tasks. To conquer these relationships, other translations are proposed.

TransH leverages projection on a hyperplane as a solution. The intuition is that a vector can have theoretically infinitive projections on different hyperplanes and different vectors can share the same projection. Instead of measuring the distance between the vectors themselves, we measure the distance between projections on a relation-specific hyperplane. Mathematically, the translation function $\|(\mathbf{h} - \mathbf{w}_r^T\mathbf{h}\mathbf{w}_r) + \mathbf{r} - (\mathbf{t} - \mathbf{w}_r^T\mathbf{h}\mathbf{w}_r)\|$, where $\mathbf{w}_r$ is the normal vector of the hyperplane produced by r.
TranH also introduces two techniques to improve the performance:

- *Soft Constraints.* To project the vector onto the hyperplanes and measure the distance, $\mathbf{r}$ and $\mathbf{w}_r$ should be orthogonal. This constraint is converted to

a soft constraint combined with the loss with a weighting hyperparameter C to adjust the importance.
- *Bernoulli Sampling*. To reduce the false negative samples in the training process, a positive sample (h, r, t) is replaced by (h, r, t') or (h', r, t) proportionally to the number of heads and tails the relation have. It is intuitive that if the relation has more heads, the randomly sampled (h', r, t) is more possibly in positive samples.

4 Research Reproduction

This report implements two models: TransE and TransH, as described in the previous section. Besides, *Bernoulli Sampling* introduced by TransH is also implemented to compare with uniform sampling.

While implementing TransH, due to the unclarified points in the original paper and no official code, the implementation might not be consistent with the original work. The most concerning part is the soft constraints. In the original paper, all the embeddings of entities and relations in the graph are combined in the loss function. However, this operation is computation-consuming, and taking the scale of the sum of all entities, the optimization can be numerically unstable. Especially in the early stage, most of the embeddings are just randomly assigned. Other open-source implementations, such as *OpenKE*[6] and *Pykg2vec*[14], also only consider the constraints within each batch. Our implementation follows their setting.

Other details in the implementation of the constraints still highly vary among those open-source implementations. In summary, there are two strategies. One tends to keep all the constraints soft and the other uses normalization instead. In our implementation, for TransE, we use normalization on both entity embedding and relation embedding. Relation embeddings are unnormalized in the original TransE paper but normalized in the open-source implementation. For TransH, only the normal vector of the hyperplanes is normalized and the embedding of the entities and the orthogonal condition are normalized through the soft constraints, which aligns with the paper.

4.1 Experiments & Results

We focus on the link prediction task. For a triple (h, r, t), h or t is replaced by all other entities. Without loss of generality, in the steps below h is assumed to be replaced. Then $f_r(\mathbf{h}, \mathbf{t})$ scores each (h', r, t). The triples are ranked according to the scores. Two metrics are applied to measure the performance. *Mean Rank*, namely *MR*, averages the rank of the ground truth. *Hits@10* is the percentage that the ground truth is among the top 10. Noticeably, though not the original triple (h, r, t), (h', r, t) can be in the knowledge graph. If all the other true (h', r, t) are filtered out during the training and evaluation, then it is called filtered, denoted as 'Filt.' in the result. Otherwise, the setting is referred to as 'Raw'.

Dataset. Following the original paper, we use the same two datasets used in the original paper. *FB15K* is a subset of the *Freebase* which is a general facts database. *WN18* is a subset of Wordnet which contains lexical relations between words. Note the test set of FB15K is cut to 5000 triples due to the time limit. Otherwise, a single test will cost more than 40 hours by estimation.
Implementation. In TransH paper, it didn't re-implement TransE through the same fine-tuning process. This could lead to unfair comparisons. Due to the limited resources, hyperparameter selection is not feasible for our reproduction. We take the settings from the TransH paper and apply the same hyperparameter to TransE under the same dataset and sampling strategy. This is not regular and usually not the best way to achieve fair comparison. But by controlling the norm for scoring function, training epochs, and embedding size, the reproduction can still hopefully give some comparable results. Here we copy the setting from the TransH paper, in which α is the learning rate for stochastic gradient descent optimizer, γ is the margin in the loss, C is the constraint weight, B is the batch size:

Under the uniform sampling setting, the optimal configurations are:$\alpha = 0.01$, $\gamma = 1$, $k = 50$, $C = 0.25$, and $B = 75$ on WN18; $\alpha = 0.005$, $\gamma = 0.5$, $k = 50$, $C = 0.015625$, and $B = 1200$ on FB15k.

Under the Bernoulli sampling setting, the optimal configurations are: $\alpha = 0.01$, $\gamma = 1$, $k = 50$, $C = 0.25$, and $B = 1200$ on WN18; $\alpha = 0.005$, $\gamma = 0.25$, $k = 100$, $C = 1.0$, and $B = 4800$ on FB15k.

When applied to TransE, C is just ignored. The norm applied to score functions is $\mathcal{L}_2$. All the training goes for 500 epochs. Following the original papers, Xavier normalization[4] is adopted to initialize the embeddings. All the experiments are done on the VU compute server with Intel Xeon Silver 4110 (32) @ 3.000GHz CPU and NVIDIA Tesla P4 GPU.

Unfortunately, the significant test is not quite feasible for a single evaluation will take 1 to 4 hours depending on the machine and the settings.

The main results are shown in Table 1. The abnormal performance of the TransH model is not because of the flaw in implementation but the numerical instability of the soft constraints. Three pieces of evidence can be provided. First, on a simpler dataset *nations*[7], the performance is invariant to the setting of the parameter C. We test $C = 0.015625$ and $C = 0.25$, *hits@10* scores are higher than 65, comparable to the performance of TransE. Second, on WR18, under Bernoulli sampling, simply changing C to 0.001 will produce a meaningful result, where filtered *hits@10* is 40.14 and *MR* is 777.40. Third, in the training log, a numerical overflow is frequent with the original settings. The soft constraints are unproportionally large to the main loss, causing the overflowing.

5 Research Extension

Our extension contains two parts. First, we conduct a theoretical analysis of the expressiveness of TransH. Second, we conduct an ablation study on the soft constraints.

Table 1. Link prediction results

Dataset	WN18				FB15k			
Metrics	MR		Hits@10		Hits@10		Hits@10	
Filtered	Raw	Filt.	Raw	Filt.	Raw	Filt.	Raw	Filt.
TransE (unif.)	418.37	403.39	74.12	75.74	189.53	185.70	40.12	39.89
TransE (bern.)	441.99	430.31	73.47	74.3	464.86	465.61	40.79	40.19
TransH (unif.)	13117	12722	0.08	0.12	4461.19	4288.80	0.67	0.73
TransH (bern.)	16129	14904	0.05	0.17	4458.82	4527.00	0.91	0.89
TransH (unif. abl.)	308.99	314.44	78.61	78.78	188.54	189.02	40.23	40.12
TransH (bern. abl.)	321.30	308.21	77.10	77.23	293.7	277.71	39.00	39.98

5.1 Expressiveness Analysis

Following RotatE[11], we define 4 types of essential relation patterns here: symmetric, antisymmetric, inverse, and compositive. Denote the triple set as Δ.

Definition 1. *Relation r is symmetric if* $\forall h, t,\ (h, r, t) \in \Delta \Rightarrow (t, r, h) \in \Delta$.

Definition 2. *Relation r is anti-symmetric if* $\forall h, t,\ (h, r, t) \in \Delta \Rightarrow (t, r, h) \notin \Delta$.

Definition 3. *Relation r_1 is inverse to r_2 if* $\forall h, t,\ (h, r_1, t) \in \Delta \Rightarrow (t, r_2, h) \notin \Delta$.

Definition 4. *Relation r_3 is composive of r_1 and r_2 if* $\forall h, t_1, t_2,\ (h, r_1, t_1)$ *and* $(t_1, r_2, t_2) \in \Delta \Rightarrow (h, r_3, t_2) \in \Delta$.

Theorem 1. *TransH can infer anti-symmetric patterns but cannot infer symmetric patterns.*

Proof. By the definition of symmetric, the following formula holds for h and t:

$$\begin{cases} \mathbf{h} - \mathbf{w}^T\mathbf{h}\mathbf{w} + \mathbf{r} - \mathbf{t} + \mathbf{w}^T\mathbf{t}\mathbf{w} & = 0 \\ \mathbf{t} + \mathbf{w}^T\mathbf{t}\mathbf{w} + \mathbf{r} + \mathbf{h} - \mathbf{w}^T\mathbf{h}\mathbf{w} & = 0 \end{cases}$$

$\mathbf{r} = 0$ is the only feasible solution, which means TransH fails to model the symmetric patterns and when $\mathbf{r} \neq 0$, the relation is naturally anti-symmetric.

The intuition is that although projection enhances expressiveness, one embedding still only has one projection on a hyperplane. Thus, the translation is forced to 0 and all non-zero translations will lead to anti-symmetric.

Theorem 2. *TransH can infer inversive patterns.*

Proof. By the definition of inversive, the following formula holds for any h and t:

$$\begin{cases} \mathbf{h} - \mathbf{w_{r_1}}^T\mathbf{h}\mathbf{w_{r_1}} + \mathbf{r}_1 - \mathbf{t} + \mathbf{w_{r_1}}^T\mathbf{t}\mathbf{w_{r_1}} & = 0 \\ \mathbf{t} + \mathbf{w_{r_2}}^T\mathbf{t}\mathbf{w_{r_2}} + \mathbf{r_2} + \mathbf{h} - \mathbf{w_{r_2}}^T\mathbf{h}\mathbf{w_{r_2}} & = 0 \end{cases}$$

By observation, we find a trivial solution to the formula that is:

$$\begin{cases} \mathbf{w_{r_1}} = \mathbf{w_{r_2}} \\ \mathbf{r_1} = -\mathbf{r_2} \end{cases}$$

This proof shows a trivial modeling of inversive patterns. The inversive translations translate on a shared hyperplane(determined by the same norm) and in the opposite direction.

Theorem 3. *TransH can infer inversive patterns.*

Proof. Here we omit the formula and directly give a trivial solution.

$$\begin{cases} \mathbf{w_{r_1}} = \mathbf{w_{r_2}} = \mathbf{w_{r_3}} \\ \mathbf{r_3} = \mathbf{r_1} + \mathbf{r_2} \end{cases}$$

Similar to the previous proof, the translations degenerate to a shared hyperplane. Compositive is inherent in the translation operation.

The summary of the expressiveness is in Table2. With the analysis above, a conclusion is that a TransH with embedding size $k+1$ is at least as expressive as TransE over the analyzed patterns, only considering the trivial solutions. This result differs from the one in [11] because of the scope of analysis. In RotatE, a more general conclusion is drawn with the loss of specific cases.

Table 2. The pattern modeling and inference abilities of TransE and TransH. The result for TransE is from [11]. Symmetric is equivalent to reflexive.

Model	Score Function	Symmetry	Antisymmetry	Inversion	Composition
TransE	$\|\mathbf{h} - \mathbf{w}^T\mathbf{h}\mathbf{w} + \mathbf{r} - \mathbf{t} + \mathbf{w}^T\mathbf{t}\mathbf{w}\|$	✗	✓	✓	✓
TransH	$\|\mathbf{h} + \mathbf{r} - \mathbf{t}\|$	✗	✓	✓	✓

5.2 Ablation of Soft Constraints

The ablation study is to verify the influence of soft constraints. In the original paper, a gap between the motivation and the parameter setting can be observed. That is, we believed the soft constraints parameter was too small to be effective before we conducted the experiment. However, with the reproduction result, an overflowing problem was found during the reproduction. It partly explained the parameter setting in the original paper.

5.3 Experiments & Results

Based on the normal settings of TransH, we set $C = 0$ in every scenario to conduct the ablation study. The results are also shown in Table1, annotated with

(abl.). There are two constraints treated as soft constraints in our implementation: $\mathbf{r}$ and $\mathbf{w}_r$ should be orthogonal and all the embeddings should be unit vectors. We argue that the constraints are either redundant to the main loss or can be replaced by normalization. Unit vector constraints can be easily dealt with normalization and the orthogonal constraint is redundant. The main loss implicitly requires $\mathbf{r} \approx \mathbf{t}_\perp - \mathbf{h}_\perp$, where the $\perp$ sign is the projection of embedding with respect to $\mathbf{w}_r$. Then $\mathbf{w}_r \cdot \mathbf{r}$ should be close to $(\mathbf{t}_\perp - \mathbf{h}_\perp) \cdot \mathbf{r} = 0$. This is confirmed by calculating the sum of all normalized $\mathbf{r} \cdot \mathbf{w}_r$ obtained from ablated TransH. The absolute value is 0.0018, which means, the model still learns the implicit orthogonal condition without soft constraints. At least we can say, that without the soft constraints, the training process is more numerically stable, which is directly confirmed by the results.

6 Discussion & Conclusion

The reproduction results, along with the ablation study, reveal the TranH's sensitivity to the constraint parameter C and the redundancy of the soft constraints. If properly optimized, regardless of the soft constraints, the translation will be restricted in the hyperplane. Our analysis of the expressiveness illustrates the theoretical strength of the TransH, confirming the effectiveness of the intuition.

Due to the time and computational limit, the experiment part is not solid enough. The absence of finetuning and significant tests definitely contributes to the abnormal performance of TransH.

A further analysis of TransX's non-trivial solution expressiveness could be conducted, which can be the real source of the expressiveness. While the trivial solutions are easily found by humans but might not be the same for optimizers. If more non-trivial solutions exist, the optimization can be easier. A solid experiment setting with full finetuning and significant test can be conducted.

References

1. Bordes, A. *et al.*: Learning Structured Embeddings of Knowledge Bases. In: Burgard, W., Roth, D. (eds.) Proceedings of the Twenty-Fifth AAAI Conference on Artificial Intelligence, AAAI 2011, San Francisco, California, USA, August 7-11, 2011, pp. 301–306. AAAI Press (2011). https://doi.org/10.1609/AAAI.V25I1.7917
2. Bordes, A. *et al.*: Translating Embeddings for Modeling Multi-relational Data. In: Burges, C.J.C. (ed.) Advances in Neural Information Processing Systems 26: 27th Annual Conference on Neural Information Processing Systems 2013. Proceedings of a meeting held December 5-8, 2013, Lake Tahoe, Nevada, United States, pp. 2787–2795 (2013). https://proceedings.neurips.cc/paper/2013/hash/1cecc7a77928ca8133fa24680a88d2f9-Abstract.html (visited on 02/01/2025)

3. Cao, S. *et al.*: KQA Pro: A Dataset with Explicit Compositional Programs for Complex Question Answering over Knowledge Base. In: Muresan, S., Nakov, P., Villavicencio, A. (eds.) Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), pp. 6101–6119. Association for Computational Linguistics, Dublin, Ireland (2022). https://doi.org/10.18653/v1/2022.acl-long.422. https://aclanthology.org/2022.acl-long.422/ (visited on 02/01/2025)
4. Glorot, X., Bengio, Y.: Understanding the difficulty of training deep feedforward neural networks. In: Teh, Y.W., Titterington, D.M. (eds.) Proceedings of the Thirteenth International Conference on Artificial Intelligence and Statistics, AISTATS 2010, Chia Laguna Resort, Sardinia, Italy, May 13-15, 2010. JMLR Proceedings, pp. 249–256. JMLR.org (2010). http://proceedings.mlr.press/v9/glorot10a.html (visited on 02/03/2025)
5. Glorot, X. *et al.*: A Semantic Matching Energy Function for Learning with Multi-relational Data. In: Bengio, Y., LeCun, Y. (eds.) 1st International Conference on Learning Representations, ICLR 2013, Scottsdale, Arizona, USA, May 2-4, 2013, Workshop Track Proceedings (2013). http://arxiv.org/abs/1301.3485 (visited on 02/01/2025)
6. Han, X. *et al.*: OpenKE: An Open Toolkit for Knowledge Embedding. In: Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing: System Demonstrations, pp. 139–144. Association for Computational Linguistics, Brussels, Belgium (2018). https://doi.org/10.18653/v1/D18-2024. http://aclweb.org/anthology/D18-2024 (visited on 02/02/2025)
7. Kemp, C. *et al.*: Learning Systems of Concepts with an Infinite Relational Model. In: Proceedings, The Twenty-First National Conference on Artificial Intelligence and the Eighteenth Innovative Applications of Artificial Intelligence Conference, July 16-20, 2006, Boston, Massachusetts, USA, pp. 381–388. AAAI Press (2006). http://www.aaai.org/Library/AAAI/2006/aaai06-061.php (visited on 02/01/2025)
8. Kipf, T.N., Welling, M.: Semi-Supervised Classification with Graph Convolutional Networks. In: 5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24-26, 2017, Conference Track Proceedings. OpenReview.net (2017). https://openreview.net/forum?id=SJU4ayYgl (visited on 02/01/2025)
9. Mohamed, S.K., Nováček, V., Nounu, A.: Discovering protein drug targets using knowledge graph embeddings. Bioinformatics **36**(2), 603–610 (2020). https://doi.org/10.1093/bioinformatics/btz600
10. Nickel, M., Tresp, V., Kriegel, H.-P.: A Three-Way Model for Collective Learning on Multi-Relational Data. In: Getoor, L., Scheffer, T. (eds.) Proceedings of the 28th International Conference on Machine Learning, ICML 2011, Bellevue, Washington, USA, June 28 - July 2, 2011, pp. 809–816. Omnipress (2011). https://icml.cc/2011/papers/438%5C_icmlpaper.pdf (visited on 02/01/2025)
11. Sun, Z. *et al.*: RotatE: Knowledge Graph Embedding by Relational Rotation in Complex Space, (2019). https://doi.org/10.48550/arXiv.1902.10197. http://arxiv.org/abs/1902.10197 (visited on 01/17/2025). arXiv:1902.10197 [cs].
12. Trouillon, T. *et al.*: Complex Embeddings for Simple Link Prediction. In: Balcan, M.-F., Weinberger, K.Q. (eds.) Proceedings of the 33nd International Conference on Machine Learning, ICML 2016, New York City, NY, USA, June 19-24, 2016. JMLR Workshop and Conference Proceedings, pp. 2071–2080. JMLR.org (2016). http://proceedings.mlr.press/v48/trouillon16.html (visited on 02/01/2025)

13. Wang, Z. *et al.*: Knowledge Graph Embedding by Translating on Hyperplanes. Proceedings of the AAAI Conference on Artificial Intelligence **28**(1) (2014). https://doi.org/10.1609/aaai.v28i1.8870. https://ojs.aaai.org/index.php/AAAI/article/view/8870 (visited on 01/12/2025)
14. Yu, S.Y. *et al.*: Pykg2vec: A Python Library for Knowledge Graph Embedding, (2019). https://doi.org/10.48550/arXiv.1906.04239. http://arxiv.org/abs/1906.04239 (visited on 02/02/2025). arXiv:1906.04239 [cs].