ProdKG: Embedding Knowledge Graphs into Product Spaces

Eric Frankel, Dhruvik Parikh, Tommy Yang
Department of Computer Science
Stanford University
Stanford, CA 94305
{ericsf, dhruvik, t2yang}@cs.stanford.edu

Abstract

The manner in which data is represented can significantly affect performance for a variety of learning tasks. For knowledge graphs, which have applications in information retrieval and question-answering, high-dimensional Euclidean representations of their data have led to improved performance in predicting new relationships and query accuracy at the cost of increased dimensionality and computational requirements. Recently, product spaces, or the product of a combination of Euclidean, hyperbolic, and spherical spaces, have significantly minimized distortion and decreased representation dimensionality for graph embeddings. We assessed the effect of embedding knowledge graphs into different product spaces and subsequently examined the the effect of this data representation on query performance. We found that embedding knowledge graphs into product spaces forces a trade-off compared to Euclidean embeddings: it can decrease distortion, but does not guarantee all-around improvements on querying tasks, instead boosting predictive performance on certain niche tasks while reducing average rank performance.

1 Introduction

At the heart of many modern question-answering [1] and recommendation systems [2] are knowledge graphs, or collections of triples of the form (h, r, t), where a head entity h is related to a tail entity t by the relation r; an example is the triple (h=London, r=is_the_capital_of, t=England). Initial versions of knowledge graphs are often incomplete, making prediction of missing relations between unlinked entities increasingly important [3, 4]. The efficacy of link prediction is largely dependent on the way in which knowledge graphs are represented, with connectivity patterns of relations like symmetry (e.g., "is a sibling of"), transposition (e.g., "child of" and "parent of"), and composition (e.g. "my father's wife is my mother") being particularly essential to model [4]. Traditionally, knowledge graphs have been embedded into Euclidean space, with both entities and relationships being given a vector representation. While these methods have given meaningful and promising results, they are often high dimensional and accordingly require significant amounts of memory; efforts to reduce dimensionality of these Euclidean embeddings generally correspond with a decreased ability to represent the relationships between different entities.

However, recent studies have demonstrated the effectiveness of embedding graphical and hierarchical data into non-Euclidean space, including hyperbolic space and product spaces. Hyperbolic space is capable of representing hierarchical data like trees and graphs with low dimensionality due to its negative curvature. The intuition behind embedding data in product spaces, or the product of a combination of Euclidean, hyperbolic, and spherical space, is similar: data can often have different structures, ranging from linear, cyclical, and tree-like, which different geometries can better represent. Past studies of product spaces have demonstrated its ability to both effectively represent data in fewer dimensions and dramatically decrease data distortion relative to Euclidean space; however, these studies have have not evaluated the effectiveness of the new data representation on explicit tasks, like querying, so performance on said tasks relative to the data representation in Euclidean space is unknown.

We propose a new approach that seeks to take advantage of a product space's low-dimensional, data-fitting structure and evaluate whether this efficient embedding can continue to represent the complex relationships of knowledge graphs. To do so, we first evaluated the effectiveness of embedding knowledge graphs in product spaces and assessing its effect on data distortion using the Countries dataset, then analyzed the performance of product space-embedded knowledge graphs on standard knowledge graph assessment tasks using the FB15k dataset. More specifically, our algorithm took as input a knowledge graph dataset, and using Riemannian stochastic gradient descent, outputted a learned representation of the dataset in a product space. We then evaluated this knowledge graph representation on the metrics mean rank (MR), mean reciprocal rank (MRR), and Hits@N for

Preprint. Work in progress.

N$\in \{1, 3, 10\}$[1]. We found that knowledge graphs embedded in product spaces showed improved or effectively equal performance on Hits@N, but lower performance on MR and MRR compared to a purely Euclidean embedding, indicating a trade-off between being able to make accurate predictions and overall ranking performance.

To summarize, our contributions are as follows:

1. We propose a new method for embedding knowledge graphs into product spaces.
2. We assess the effectiveness of embedding knowledge graphs into product spaces on data representation and distortion.
3. We assess the effectiveness of knowledge graph representations in product spaces on standard querying metrics.

2 Related Works

2.1 Euclidean Knowledge Graph Embeddings.

Prior work on knowledge graph embedding into Euclidean space primarily revolved around representing relationships between head and tail nodes as translations[3, 5, 6, 7] or through tensor factorization approaches[8, 9]. These approaches are simple, but they fail to encode certain relation patterns. For example, translation approaches are unable to encode symmetry because no non-trivial translation is its own inverse in Euclidean space.

2.2 Complex Euclidean Knowledge Graph Embeddings.

The current state-of-the-art method for modeling relation patterns in knowledge graphs is RotatE[10]. The RotatE model embeds the members of a triplet (h, r, t) into a k-dimensional complex vector space $\mathbb{C}^k$, and defines a relation as a rotation of modulus 1 (i.e. $|r| = 1$) such that $t = h \circ r$, where $\circ$ denotes the Hadamard element-wise product. RotatE uses the scoring function $\|h \circ r - t\|^2$ during optimization of the model, where the optimal model will give a true triplet (h, r, t) a higher score than other possible "false" triplets like as (h', r, t) or (h, r, t'). Although these embeddings are able to achieve high performance, they also require high-dimensional spaces, which leads to a high memory cost.

2.3 Non-Euclidean Graph Embeddings.

Embedding graph data and other hierarchical information, like trees, in low-dimensional, non-Euclidean space, like spherical [11] and hyperbolic [12] space, has proven particularly effective for learning tasks. Hyperbolic space has proven especially effective due to its negative curvature, functioning as a continuous version of discrete tree-like structures [13].

2.4 Product Space Embeddings.

Recently, embedding graphical data into product spaces, or a Riemann product manifold combining hyperbolic, spherical and Euclidean components such that the product manifold has non-constant curvature, has significantly reduced the distortion of the embedding data relative to other embedding techniques, as well as improved performance on tasks like node similarity [14]. As such, our project seeks to investigate whether embedding a knowledge graph into a product space and studying its subsequent representation can improve performance in knowledge graph tasks.

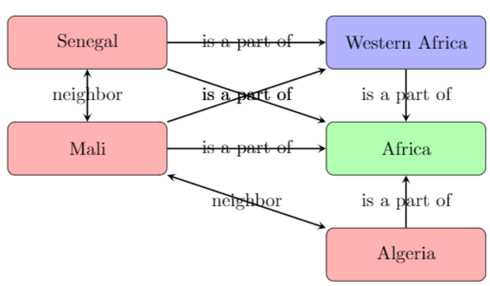

Figure 1: The Countries dataset. Note the two relations "neighbor" and "is a part of" relative to entities on the country, subcontinent, and continent level.

[1]The code for this paper can be found at `https://github.com/esfrankel/kg-prod-spaces`.

3 Datasets

3.1 Countries Dataset

The Countries dataset (Countries) is a knowledge graph of country relationships generated from public geographical data[15]. The dataset has 271 entities and 2 relations to create a small but nontrivial knowledge graph. We used 985 triplets of head entities, relations, and tail entities as a train set, 24 triplets as a validation set, and 24 triplets as a test dataset. A visualization of the structure of the dataset is present in Figure 1.

3.2 FB15k Dataset

The FB15k dataset (FB15k) is a subset of Freebase, a knowledge graph of over 1.2 billion triplets and 80 million entities, that was created by [3]. FB15k has 14,951 entities and 1,345 different relations among them. We used 60,393 triplets as a train set, 6,250 triplets as a validation set, and 7,384 triplets as a test dataset.

4 Methods

Our main method is to embed knowledge graph data into of product spaces by representing the knowledge graph as a directed graph with unique edge weights that correspond to each relation; that is, a triple of the form (h=London, r=is_the_capital_of, t=England) could be expressed as a node, London, with a directed edge to another node, England, with a weight of $w(r)$, with $w : \mathcal{R} \to \mathbb{R}^+$ an injective map from the set of relations to a unique positive weight. After we have constructed our new graph, given the set of graph distance $\{d_G(X_i, X_j)\}_{ij}$, we optimized embedding our graph using the loss function

$$\mathcal{L}(x) = \sum_{1 \le i \le j \le n} \left| \left(\frac{d_{\mathcal{P}}(x_i, x_j)}{d_G(X_i, X_j)} \right)^2 - 1 \right|$$

which is representative of the average distortion of our embedding. Given this loss function, we then perform Riemannian stochastic gradient descent (R-SGD) with a fixed learning rate to optimize our loss function, which cannot be done using traditional stochastic gradient descent [16]. We assess this embedding on the metric **distortion**[2] and **mean average precision.**[3]

After R-SGD is performed and the points of our graph converge to relatively fixed locations, we want to ensure that our points still have "knowledge graph" structure. To do this, we define a "scoring" function that rewards triplets that are true in the knowledge graph with a higher score than triplets that are not true. For our product space embedding, we use the function

$$f((h, r, t)) = \|h + r - t\|_{\mathcal{P}}$$

where $\|\cdot\|_{\mathcal{P}}$ is the product space norm. In particular, for $\mathcal{P} = M_1 \times M_2 \times \dots M_k$ and $p = m_1 \times m_2 \times \cdots \times m_k$, its norm is

$$\|p\|_{\mathcal{P}} = \sqrt{\|m_1\|_{M_1}^2 + \|m_2\|_{M_2}^2 + \cdots + \|m_k\|_{M_k}^2}$$

Past methods for learning vector representations of knowledge graphs have used **negative sampling loss**, a technique that's proved effective across both knowledge graphs and word embeddings:

$$\mathcal{L} = -\log \sigma(\gamma - f((h, r, t))) - \sum_{i=1}^{n} \frac{1}{k} \log \sigma(f((h_i', r, t_i')) - \gamma)$$

However, we use an additional approach for drawing negative samples called **self-adversarial negative sampling**. The intuition is as follows: as training continues when optimizing using negative sampling loss, traditional negative sampling becomes inefficient because more examples are obviously wrong. Instead, we sample negative triples from the distribution

$$p(h_j', r, t_j' | \{(h_i, r_i, t_i)\}) = \frac{\exp(\alpha f((h_j', r, t_j')))}{\sum_i \exp(\alpha f((h_j', r, t_j')))}$$

an approach that introduced in [10], where α is the adversarial temperature. Therefore, we express our modified negative sampling loss with self-adversarial training optimized with the **Adam optimizer** as follows:

$$\mathcal{L} = -\log \sigma(\gamma - f((h, r, t))) - \sum_{i=1}^{n} p(h_i', r, t_i') \log \sigma(f((h_i', r, t_i')) - \gamma)$$

[2]A mapping $f : X \to Y$ of a metric space (X, d_X) to a metric space (Y, d_Y) is an embedding with distortion α if there exists a constant $r > 0$ such that for every $x_1, x_2 \in X$, $r \cdot d_X(x_1, x_2) \le d_Y(f(x_1), f(x_2)) \le \alpha r \cdot d_X(x_1, x_2)$

[3]Mean average precision is the mean of the average precision scores for each query.

where γ is a fixed margin, σ is the sigmoid function, and (h'_i, r, t'_i) is the i-th negative triplet.

Following the successful embedding of the graph into our product space, we assess the quality of the embeddings using the metrics Mean Rank (MR), Mean Reciprocal Rank (MRR), Hits at 1 (H@1), Hits at 3 (H@3), and Hits at 10 (H@10), which are standard evaluation metrics for similar problems in existing literature. These metrics are constructed as follows:

1. For each triple (h_i, r_i, t_i) in our test set, remove the head.
2. For every possible head h in our entire data set, compute the score $f(h, r_i, t_i)$.
3. Sort all of the scores in decreasing order.
4. Store the rank of the correct head entity h_i within the sorted score list.

We then calculate MR as being the average of the ranks stored for each triple in our test set, MRR as the average of the reciprocal of the ranks, and Hits at N (H@N) as the proportion of the correct entities that are ranked within the top N in the sorted list.

$$MR = \frac{1}{n}\sum_{i=1}^{n} \text{rank}_i, \quad MRR = \frac{1}{n}\sum_{i=1}^{n} \frac{1}{\text{rank}_i}, \quad H@N = \frac{1}{n}\sum_{i=1}^{n} \mathbf{1}\{\text{rank}_i \leq N\}$$

where n is the number of triples in our test set, rank_i is the rank of the i-th triple, and $\mathbf{1}$ is the indicator function.

5 Results

To see if embedding knowledge in product spaces was worth pursuing, we embedded the Countries dataset in three spaces: one which was exclusively Euclidean, one of which was exclusively hyperbolic, and one of which was a product space, using a constant total dimension of 10. Using a Google Cloud VM with 4 CPUs and 2 NVIDIA Tesla V100 GPUs, we computed the following metrics for the embeddings in the respective spaces with a learning rate of 5, batch size of 64, and subsample of 16:

Table 1: Results of embedding the Countries dataset in various spaces.

Space	Distortion	MAP	Training Loss
$\mathbb{R}^{10}$	0.347	0.472	0.299
$\mathbb{H}^{10}$	0.509	0.440	0.402
$\mathbb{R}^5 \times \mathbb{H}^5$	**0.341**	**0.498**	**0.298**

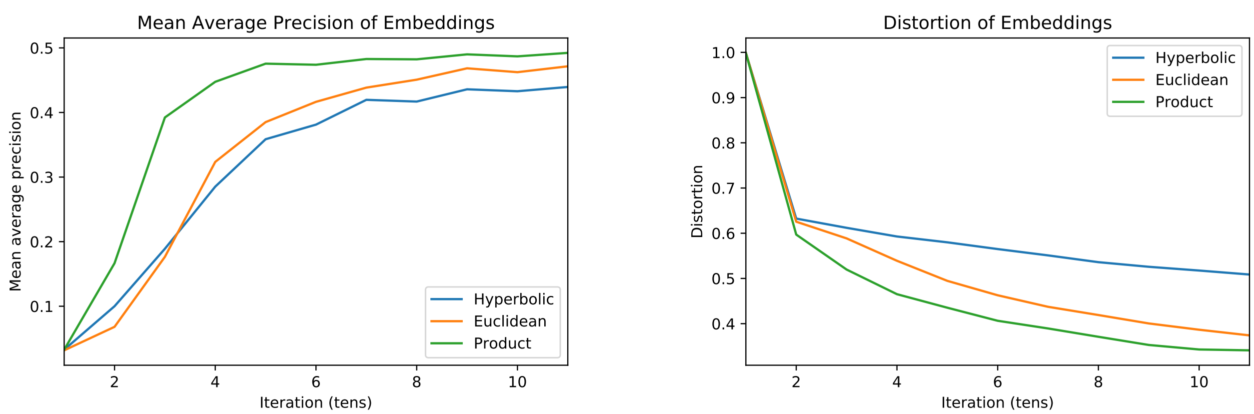

Table 2: Left: plot of the mean average precision for different embeddings over time. Right: plot of the distortion for different embeddings over time. Note that the distortion is lowest for the product space embedding.

We can see that the product space embedding outperformed both Euclidean and hyperbolic embeddings with lower distortion, which suggests that attributes of both types of embeddings were utilized. The lower distortion also suggests that the relations between the data are being represented better, which could result in product space embeddings outperforming Euclidean and hyperbolic embeddings on the metrics defined in the previous section.

Next, we embedded the FB15k dataset into four spaces with a constant total dimension of 600. Three of these spaces are product spaces composed of Euclidean and hyperbolic spaces, while the fourth is purely Euclidean. After training the embeddings for 500 epochs using a learning rate of 0.0001, a negative sample size of 256, an adversarial temperature of 1.0, a gamma of 24, and a batch size of 512, the following results were obtained:

Table 3: Results of the ProdKG model evaluated on the FB15k dataset in various spaces.

Space	MR	MRR	Hits@1	Hits@3	Hits@10
$\mathbb{R}^{100} \times \mathbb{H}^{500}$	92.5	0.626	**0.609**	0.715	0.810
$\mathbb{R}^{500} \times \mathbb{H}^{100}$	59.8	0.671	0.564	0.752	**0.847**
$\mathbb{R}^{300} \times \mathbb{H}^{300}$	95.97	0.501	0.342	0.622	0.752
$\mathbb{R}^{600}$ (TransE baseline)	**49.4**	**0.684**	0.582	**0.759**	0.845

Our results demonstrate differing best performance on each task depending on the space it was embedded in. In particular, embedding FB15k in the product space $\mathbb{R}^{100} \times \mathbb{H}^{500}$ produced the best Hits@1 score, while embedding in $\mathbb{R}^{500} \times \mathbb{H}^{100}$ produced the best Hits@10 score; on all other metrics (MR, MRR), $\mathbb{R}^{600}$ performed better.

6 Conclusion

In this project, we explored embedding knowledge graphs in product spaces to evaluate 1) whether distortion of knowledge graph representations could be meaningfully reduced and 2) whether these representations could represent the complex relationships of knowledge graphs. After performing experiments on the Countries and FB15k dataset, we ultimately found that while embedding knowledge graphs in product spaces does seem to reduce distortion, the reduction in distortion does not necessarily result in better all-around performance on querying tasks.

More specifically, we saw that product space embeddings performed well on the Hits@N metrics by either outperforming or nearing the performance of the Euclidean embedding, but did not perform well on the MR and MRR metrics. The good Hits@N performance suggests that product space embeddings do well for certain types of structures and relations within knowledge graphs, which results in a high score for these structures, but perform very poorly on queries that include relations that are not represented well by the embedding, which greatly lowers the average performance as shown in the MR and MRR scores. Moreover, this shows that the product space embedding was able to improve the ability of the model to make predictions, because Hits@N is a measure of the embeddings ability to recall true relations, but decreased the ability of the model to rank these predictions, because the lower-ranked relations were ordered very inaccurately resulting in a poor MR score.

The trade-off in performance between the product space embeddings and the purely Euclidean embedding suggests that product space embeddings should only be used when the structure of the data matches the product space, e.g. embedding knowledge graphs with hierarchical relations, which are tree-like, in product spaces involving hyperbolic spaces. Also, the specific application should be taken into account. Perhaps, for exclusively prediction-based tasks such as question-answering, product spaces might outperform other representations, whereas they may perform poorly on rank-based applications such as search.

7 Future Work

This work was severely limited by the computational costs in time and money for training various models. With more resources, we suggest various future experiments:

1. Embedding knowledge graphs in product spaces with components in spherical space.
2. Using Sarkar's construction for identifying the optimal embedding size for each dataset.
3. Allowing the embeddings to train for more epochs could cause meaningful changes in important metrics.
4. Performing more robust combinatorial analysis on hyperparameters to tune/improve performance.

8 Contributions

Eric and Dhruvik contributed by implementing the experiments required for our project and each contributing to writing the paper. Tommy contributed by building the infrastructure of the project and helping develop the poster and paper. We also received advice and assistance from Dr. Fred Sala from the DAWN Lab and Adva Wolf from the Stanford Department of Mathematics.

References

[1] Chenyan Xiong, Russell Power, and Jamie Callan. Explicit semantic ranking for academic search via knowledge graph embedding. In *Proceedings of the 26th International Conference on World Wide Web*, WWW '17, pages 1271–1279, Republic and Canton of Geneva, Switzerland, 2017. International World Wide Web Conferences Steering Committee.

[2] Fuzheng Zhang, Nicholas Jing Yuan, Defu Lian, Xing Xie, and Wei-Ying Ma. Collaborative knowledge base embedding for recommender systems. In *Proceedings of the 22Nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '16, pages 353–362, New York, NY, USA, 2016. ACM.

[3] Antoine Bordes, Nicolas Usunier, Alberto Garcia-Duran, Jason Weston, and Oksana Yakhnenko. Translating embeddings for modeling multi-relational data. In C. J. C. Burges, L. Bottou, M. Welling, Z. Ghahramani, and K. Q. Weinberger, editors, *Advances in Neural Information Processing Systems 26*, pages 2787–2795. Curran Associates, Inc., 2013.

[4] Zhen Wang, Jianwen Zhang, Jianlin Feng, and Zheng Chen. Knowledge graph embedding by translating on hyperplanes, 2014.

[5] Guoliang Ji, Shizhu He, Liheng Xu, Kang Liu, and Jun Zhao. Knowledge graph embedding via dynamic mapping matrix. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 687–696, Beijing, China, July 2015. Association for Computational Linguistics.

[6] Zhen Wang, Jianwen Zhang, Jianlin Feng, and Zheng Chen. Knowledge graph embedding by translating on hyperplanes. In *Proceedings of the Twenty-Eighth AAAI Conference on Artificial Intelligence*, AAAI'14, pages 1112–1119. AAAI Press, 2014.

[7] Timothee Lacroix, Nicolas Usunier, and Guillaume Obozinski. Canonical tensor decomposition for knowledge base completion. In Jennifer Dy and Andreas Krause, editors, *Proceedings of the 35th International Conference on Machine Learning*, volume 80 of *Proceedings of Machine Learning Research*, pages 2863–2872, Stockholmsmässan, Stockholm Sweden, 10–15 Jul 2018. PMLR.

[8] Maximilian Nickel, Volker Tresp, and Hans-Peter Kriegel. A three-way model for collective learning on multi-relational data. In *Proceedings of the 28th International Conference on International Conference on Machine Learning*, ICML'11, pages 809–816, USA, 2011. Omnipress.

[9] Bishan Yang, Wen-tau Yih, Xiaodong He, Jianfeng Gao, and Li Deng. Embedding entities and relations for learning and inference in knowledge bases. In *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*, 2015.

[10] Zhiqing Sun, Zhi-Hong Deng, Jian-Yun Nie, and Jian Tang. Rotate: Knowledge graph embedding by relational rotation in complex space. In *International Conference on Learning Representations*, 2019.

[11] Weiyang Liu, Yandong Wen, Zhiding Yu, Ming Li, Bhiksha Raj, and Le Song. Sphereface: Deep hypersphere embedding for face recognition. *CoRR*, abs/1704.08063, 2017.

[12] Christopher De Sa, Albert Gu, Christopher Ré, and Frederic Sala. Representation tradeoffs for hyperbolic embeddings. *CoRR*, abs/1804.03329, 2018.

[13] Dmitri Krioukov, Fragkiskos Papadopoulos, Maksim Kitsak, Amin Vahdat, and Marián Boguñá. Hyperbolic geometry of complex networks. *Phys. Rev. E*, 82:036106, Sep 2010.

[14] Albert Gu, Frederic Sala, Beliz Gunel, and Christopher Ré. Learning mixed-curvature representations in product spaces. In *International Conference on Learning Representations*, 2019.

[15] Guillaume Bouchard, Sameer Singh, and Théo Trouillon. On approximate reasoning capabilities of low-rank vector spaces. In *Knowledge Representation and Reasoning: Integrating Symbolic and Neural Approaches, AAAI Spring Symposium Series*, 2015.

[16] S. Bonnabel. Stochastic gradient descent on riemannian manifolds. *IEEE Transactions on Automatic Control*, 58(9):2217–2229, Sep. 2013.