Abstract

Knowledge graph (KG) elements are relational triples of the form (head, relation, tail).

Predicting new relations and returning correct queries depends on the data representation.

Embedding graphical data in a product of model spaces has led to decreased data distortion.

We embed knowledge graphs on product spaces and assessed distortion and query performance.

Embedding KGs in product spaces can decrease distortion, but won’t guarantee all-around improvements on querying tasks.

Non-Euclidean Embeddings

Traditionally, data for machine learning has been represented in Euclidean space as tensors.

Three key benefits to Non-Euclidean embeddings:
  - Data can have a better representation
  - Better models built on better geometries
  - Lower dimensionality and faster operations

Different spaces, different data
  - Cycles → spherical
  - Tree → hyperbolic

Have to use non-Euclidean methods for working over Riemannian manifolds
  - Riemannian SGD (RSGD)

KG-Product Space Distortion

KG embeddings must represent three relation patterns: Symmetry, Inversion, Composition

A product space is a Riemannian manifold $P$ defined by $P = \mathbb{S}^n_1 \times \cdots \times \mathbb{S}^n_l \times \cdots \times \mathbb{S}^n_m$.

Goal: embed in a product space that minimizes distortion with using the loss function

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

optimized using Riemannian SGD.

Datasets: Countries
  - Countries: relationships between countries and regions (testing all three patterns)

We embedded the data into three spaces, two of which were exclusively Euclidean and hyperbolic, and one product space, using a constant total dimension.

Product space outperformed both Euclidean and Hyperbolic embeddings (lower distortion), suggesting attributes of both types of embeddings were utilized.

KG-Product Space Querying

KG embeddings have a scoring function $f_r(h, t)$ that ranks true triples the highest.

\[
f_{\text{Sibling of}}(\text{Eli Manning, Peyton Manning}) = 1\]

\[
f_{\text{Sibling of}}(\text{Eli Manning, Kanye West}) = 2\]

\[
f_{\text{Sibling of}}(\text{Eli Manning, Obama}) = -1\]

We assessed our product space embeddings on the query metrics mean rank (MR), mean reciprocal rank (MRR), and Hits@N.

Goal: learn vector representations of triples (h, r, t) to give high Hits@N, MRR, low MR using the loss function

\[
L = -\log(\sigma(\gamma - f_r(h, t))) - \sum_{i=1}^{k} p(h_i, r, t_i) \log(\sigma(f_r(h_i, t_i) - \gamma))
\]

\[
f_r(h, r, t) = -\frac{1}{n} - \frac{1}{n} \|h + r - t\|_p
\]

where the second loss term is composed of self-adversarial negative sampling and the norm is taken with respect to the product space.

We embedded the data into four spaces, only one of which was purely Euclidean and again keeping the total dimension constant, training for 500 epochs.

Product spaces outperformed the others at different tasks:
  - Product spaces outperformed Euclidean space at Hits@1 and Hits@10 and were close for Hits@3, but underperformed at MR and MRR.

Conclusion

Knowledge graph elements are relational triples that can be modeled in non-Euclidean spaces.

We embedded knowledge graphs in different product spaces and assessed their distortion and performance on query tasks.

We found that embedding knowledge graphs in product spaces will decrease distortion, but will not guarantee improved performance on query tasks - some will be improved while others worsen.

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:
  - Using Sarkar’s construction for identifying the optimal embedding size for each dataset.
  - Allow the embeddings to train for more epochs could cause meaningful changes in important metrics (double descent).
  - Perform more robust combinatorial analysis on hyperparameters to tune/improve performance.

References

Future Work

Acknowledgements

We would like to thank Andrew Ng, Moses Charikar, and Chris Re for teaching CS 229, as well as the rest of the course staff for running it so smoothly. We found the course very interesting and educational and appreciated the opportunity to work on ProdKG. We would also like to thank Fred Sala and Adva Wolf for their help and insight while working on ProdKG.