Reasoning in Knowledge Graphs using Embeddings

Joint work with H. Ren, W. Hamilton, R. Ying, J. You, M. Zitnik, D. Jurafsky

Jure Leskovec
Knowledge Graphs
Knowledge Graphs

- Knowledge Graphs are **heterogenous** graphs
  - Multiple types of entities and relations exist
- Facts are represented as triples \((h, r, t)\)
  - (‘Alice’, ‘friend_with’, ‘Bob’)
  - (‘Paris’, ‘is_a’, ‘City’)
  - …
Traditional Tasks

Knowledge Graph Competition/Link Prediction

- Predict the missing head or tail for a given triple \((h, r, t)\)
- Example:

Barack Obama \textcolor{red}{\textbf{BornIn}} United States

Barack Obama \textcolor{red}{\textbf{Nationality}} American
Our work: Beyond Link Prediction

Our goal: Reason over the knowledge graph using complex multi-hop queries

- Logical queries: Subset of first-order logic with existential quantifier ($\exists$), conjunction ($\land$) and disjunction ($\lor$)

"Where did all Canadian citizens with Turing Award graduate?"

$$q = V_\? \cdot \exists V : \text{Win}(\text{TuringAward}, V) \land \text{Citizen}(\text{Canada}, V) \land \text{Graduate}(V, V_\?)$$

Jure Leskovec, Stanford University
"Where did Canadian citizens with Turing Award graduate?"

Knowledge Graph

Computation Graph

Each point corresponds to a set of entities
Why is it Hard?

- **Heterogeneity**: Lack of schema, or quite large schema (65K for DBpedia)
- **Noise** and incompleteness
- **Uncertainty**
- **Massive size**
- **Fast query time**
Why is it Hard?

**Key challenge:** Big graphs and queries can involve noisy and unobserved data!

Some links might be noisy or missing

**Problem:** Naïve link prediction and graph template matching are too expensive
Our Idea: **Query2Box**

Use representation learning to map a graph into a Euclidean space and learn to reason in that space.
Our Idea: **Query2Box**

**Idea:**

- 1) Embed nodes of the graph
- 2) For every logical operator learn a spatial operator

**So that:**

- 1) Take an arbitrary logical query. Decompose it into a set of logical operators (∃,∧,∨)
- 2) Apply a sequence of **spatial operators** to embed the query
- 3) Answers to the query are entities close to the embedding of the query
Our Idea: **Query2Box**

**Idea:**

1. Embed nodes of the graph

**Key insight:**

Represent query as a box. Operations (union, intersection) are well defined over boxes.

3. Answers to the query are entities close to the embedding of the query
Embedding Queries

**Query2Box embedding:**

Embed queries with hyper-rectangles (boxes): \( q = (\text{Cen}(q), \text{Off}(q)) \).

![Diagram showing embedding space with points Cambridge, Edinburgh, McGill, and Stanford]

[Probabilistic Embedding of Knowledge Graphs with Box Lattice Measures. Vilnis, et al., ACL 2018]
Embedding Queries

- Geometric Projection Operator
- Geometric Intersection Operator
Projection Operator

Geometric Projection Operator $\mathcal{P}$

- $\mathcal{P} : \text{Box} \times \text{Relation} \rightarrow \text{Box}$
Projection Operator: Example

Turing Award

Win

Pearl
Hinton
Bengio

Turing Award
Win

Pearl
Bengio
Hinton

Hinton
Edinburgh
Graduate
Cambridge
McGill

Graduate

McGill
Cambridge
Edinburgh

Bengio
Hinton
Geometric Intersection Operator $\mathcal{I}$

- $\mathcal{I} : \text{Box} \times \cdots \times \text{Box} \rightarrow \text{Box}$
  - The new center is a weighted average
  - The new offset shrinks
Intersection Operator: Example

Turing Award
Canada
Citizen

Pearl
Hinton
Bengio
Bieber
Trudeau

Turing Award
Canadian
Citizen

Win

Pearl
Hinton
Bengio
Bieber
Trudeau
“Where did Canadian citizens with Turing Award graduate?”

Dependency Graph

Turing Award → Win → Graduate
Canada → Citizen → \( V \) → \( V' \)
“Where did Canadian citizens with Turing Award graduate?”

Dependency Graph

Computation Graph

Embedding Process

Each point corresponds to a set of entities
“Where did Canadian citizens with Turing Award graduate?”

Dependency Graph

Computation Graph

Embedding Process

Each point corresponds to a set of entities
Example

"Where did Canadian citizens with Turing Award graduate?"

Dependency Graph

Computation Graph

Embedding Process

Each point corresponds to a set of entities
“Where did Canadian citizens with Turing Award graduate?”

**Dependency Graph**

- Turing Award
- Win
- Graduate
- Citizen
- Canada

**Computation Graph**

- Turing Award
- Projection
- Intersection
- Projection
- Citizen
- Canada

**Embedding Process**

- Turing Award
- Win
- Graduate
- Intersection
- Projection
- Citizen
- Canada
- McGill
- Edinburgh
- Hinton
- Bengio
- Bieber
- Trudeau

Each point corresponds to a set of entities
How to Handle Disjunction

So far we can handle Conjunctive queries.

Can we learn a geometric disjunction operator?

- Theorem (paraphrased): For a KG with $M$ nodes, we need embedding dimension of $M$ to handle disjunction.
Any query with AND and OR can be transformed into equivalent Disjunctive Normal Form (disjunction of conjunctive queries).
Disjunctions: Solution

Given an arbitrary AND-OR query

1) Transform it into an DNF
2) Answer each conjunctive query
3) Overall answer is the union of conjunctive query answers
Benefits of Query2Box

Scalability and efficiency:
- Any query can be reduced to a couple of matrix operations and a single k-nearest neighbor search

Generality:
- We can answer any query (even those we have never seen before)

Robustness to noise:
- Graph can contain missing and noisy relationships
Query2Box: Model Training

Training examples: Queries on the graph

- **Positives:** Path with a known answer
- **Negatives:** Random nodes of the correct answer type
- **Goal:** Find embeddings and operators so that queries give correct answers
Experimental Setup

We essentially learn to “memorize” the answers to queries

- We embed entities so that our geometric operators give correct answers

Questions:

- Does our method generalize to new unseen queries?
- Does our method generalize to new query structures?
- Can method handle missing relations?
Experimental Setup

Training on an incomplete graph:

- Test queries are **not** answerable in the training graph
  - Every test query has at least one missing edge
- Method has to (implicitly) input missing edges
  - **Note:** Query template matching would have accuracy of random guessing
艰

**KG and Query Statistics**

- **Freebase**: FB15K, FB15K-237

<table>
<thead>
<tr>
<th>Dataset</th>
<th>Entities</th>
<th>Relations</th>
<th>Training Edges</th>
<th>Validation Edges</th>
<th>Test Edges</th>
<th>Total Edges</th>
</tr>
</thead>
<tbody>
<tr>
<td>FB15k</td>
<td>14,951</td>
<td>1,345</td>
<td>483,142</td>
<td>50,000</td>
<td>59,071</td>
<td>592,213</td>
</tr>
<tr>
<td>FB15k-237</td>
<td>14,505</td>
<td>237</td>
<td>272,115</td>
<td>17,526</td>
<td>20,438</td>
<td>310,079</td>
</tr>
</tbody>
</table>

- **Queries:**

  - **Training Conjunctive Queries**: 1p, 2p, 3p, 2i, 3i
  - **Unseen Conjunctive Queries**: ip, pi
  - **Union Queries**: 2u, up
Experiments Freebase

Observations:
- On “training” queries: +20% H@3
- On new conjunctive query structures: +15%
- On disjunctive queries: +36%
FB15k: Embedding Space

"List male instrumentalists who play string instruments"
“List male instrumentalists who play string instruments”
FB15k: Embedding Space

“List male instrumentalists who play string instruments”

TPR: 100%
FPR: 0%

# of string instruments: 10
"List male instrumentalists who play string instruments"
“List male instrumentalists who play string instruments”
FB15k: Embedding Space

“List male instrumentalists who play string instruments”
“List male instrumentalists who play string instruments”

# of answers: 396

TPR: 99.4%
FPR: 0.01%
Query2Box: Summary

- **Query2Box:**
  - Embed the query as a box
  - Logical operations become spatial operations

- **Composability of queries:**
  - Generalize well to unseen, extrapolated queries
  - Explicitly training for composability is important

- Instance vs. multi-hop generalization
Conclusion

- Box embeddings for answering logical queries on Knowledge graphs
- Handle union and intersection
- Generalize well to unseen, extrapolated queries
- Future work: Handle negation, other geometric model
Open Graph Benchmark

- On-going effort for large-scale realistic benchmark datasets for graph ML.

Webpage: [https://ogb.stanford.edu/](https://ogb.stanford.edu/)
Github: [https://github.com/snap-stanford/ogb](https://github.com/snap-stanford/ogb)
Why a New Benchmark?

1) **Current focus is on small graphs or small sets of graphs from just a handful of domains:**
   - Datasets are too small:
     - MUTAG graph classification is just 188 graphs
     - Hard to reliably and rigorously evaluate algorithms

2) **Lack of common benchmark datasets for comparing different methods:**
   - Every paper design its own, custom train/test splits
   - Performance across papers is not comparable

3) **Dataset splits follow conventional random splits:**
   - Unrealistic for real-world applications
   - Accuracies are over-optimistic under conventional splits
ML with Graphs

To properly track progress and identify issues with current approaches it is critical for our community to...

...develop diverse, challenging, and realistic benchmark datasets for machine learning on graphs
The Open Graph Benchmark

OGB is a set of benchmarks for graph ML:

1. Ready-to-use datasets for key tasks on graphs:
   - Node classification, link prediction, graph classification
2. Common codebase to load, construct & represent graphs:
   - Popular deep frameworks, e.g., DGL, PyTorch Geometric
3. Common codebase with performance metrics for fast model evaluation and comparison:
   - Meaningful data splits focusing on generalization
OGB Datasets are Diverse

![Diagram showing the diversity of OGB datasets across different scales and tasks.](image-url)
Open Graph Benchmark

https://ogb.stanford.edu
ogb@cs.stanford.edu

Core development team
W. Hu, M. Fey, M. Zitnik, Y. Dong, H. Ren, B. Liu, M. Catasta, J. Leskovec

Steering committee
Regina Barzilay, Peter Battaglia, Yoshua Bengio, Michael Bronstein, Stephan Günnemann, Will Hamilton, Tommi Jaakkola, Stefanie Jegelka, Maximilian Nickel, Chris Re, Le Song, Jian Tang, Max Welling, Rich Zemel
PhD Students

Alexandra Porter
Camilo Ruiz
Claire Donnat
Emma Pierson
Weihua Hu
Jiaxuan You
Bowen Liu
Hongyu Ren
Rex Ying

Post-Doctoral Fellows

Baharan Mirzasoleiman
Marinka Zitnik
Michele Catasta
Pan Li
Shantao Li

Research Staff

Maria Brbic
Adrijan Bradaschia
Rok Sosic

Industry Partnerships

Funding

NIH
IARPA
DARPA
NSF

Collaborators

Dan Jurafsky, Linguistics, Stanford University
David Grusky, Sociology, Stanford University
Stephen Boyd, Electrical Engineering, Stanford University
David Gleich, Computer Science, Purdue University
VS Subrahmanian, Computer Science, University of Maryland
Marinka Zitnik, Medicine, Harvard University
Russ Altman, Medicine, Stanford University
Jochen Profit, Medicine, Stanford University
Eric Horvitz, Microsoft Research
Jon Kleinberg, Computer Science, Cornell University
Sendhil Mullainathan, Economics, Harvard University
Scott Delp, Bioengineering, Stanford University
James Zou, Medicine, Stanford University

Jure Leskovec, Stanford University
References
