Modeling Relational Data with Graph Convolutional Networks

Presented by

Lee Guan Bo Ambrose

UNIVERSITY OF WATERLOO | DAVID R. CHERITON SCHOOL OF COMPUTER SCIENCE

SCHOOL OF COMPUTING & DATA SCIENCE
The University of Hong Kong

Modeling Relational Data with Graph Convolutional Networks

Michael Schlichtkrull*
University of Amsterdam
m.s.schlichtkrull@uva.nl

Thomas N. Kipf*
University of Amsterdam
t.n.kipf@uva.nl

Peter Bloem
VU Amsterdam
p.bloem@vu.nl

Rianne van den Berg
University of Amsterdam
r.vandenberg@uva.nl

Ivan Titov
University of Amsterdam
titov@uva.nl

Max Welling
University of Amsterdam, CIFAR†
m.welling@uva.nl

arXiv:1703.06103v4 [stat.ML] 26 Oct 2017 Cited by 6172

https://arxiv.org/pdf/1703.06103

(Schlichtkrull et al., 2018)

Outline

- Introduction
- Relational-GCN
- Entity classification
- Link Prediction
- Conclusion

Part 1: Introduction

Knowledge Base and Knowledge Graph

- Knowledge base stores factual information
  - In a form of triples (Subject, predicate, object)

- Represent it as knowledge graph
  - Node: Entity (subject or object); Edge: Relation

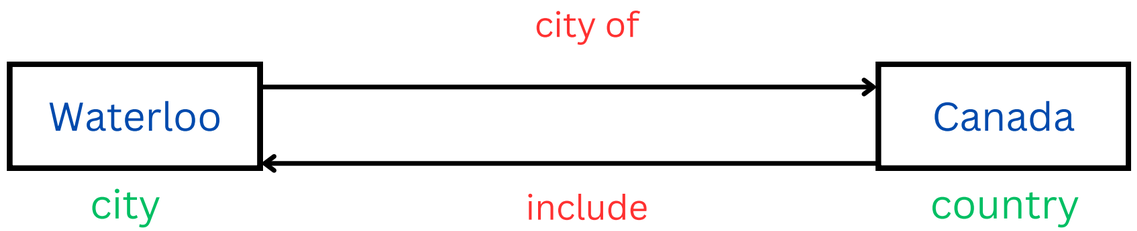
Subject	Predicate	Object
Waterloo	city of	Canada
Canada	include	Waterloo

More on Knowledge Graph

- Modeled as knowledge graph
  - Directed and labeled multigraph
  - Node-edge-node: Entity-relationship-entity (subject, predicate, object)

- Application: Question Answering, Information Retrieval

Subject	Predicate	Object
Waterloo	city of	Canada
Canada	include	Waterloo



What is the problem?

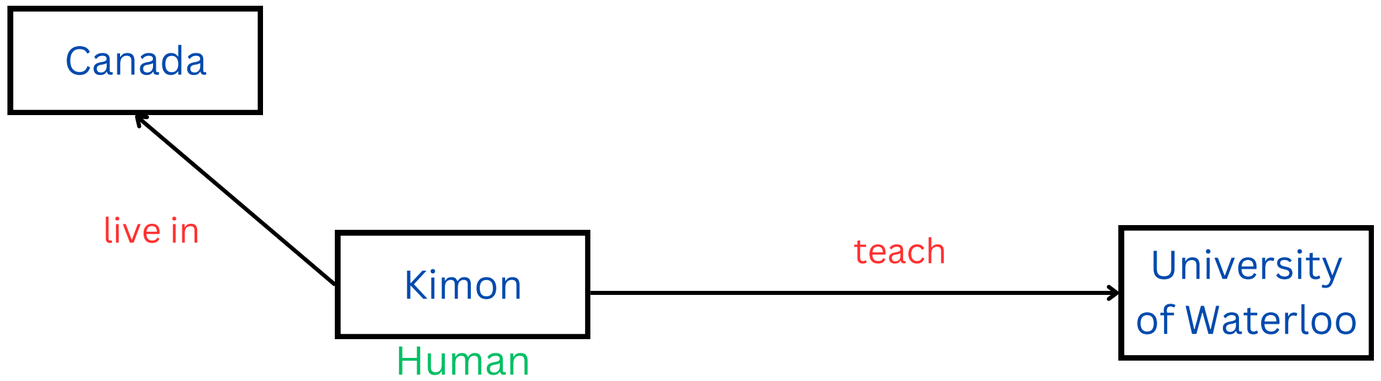
- Missing information in knowledge base
  - Exp: DBPedia, Wikidata or Yago
- Need for statistical relational learning
  - predicting missing information

Why is it important?

- User cannot get access to the full set of correct information

Motivation

- Given “Kimon teaches at University of Waterloo”
- We know that
  - => Kimon is a human (node classification)
  - => Kimon lives in Canada (link prediction)



A glimpse on the paper

- Use graph neural network to model relational data
- Techniques for parameter sharing and to enforce sparsity constraint
- Show the impact of encoder in improving the performance of facctorization model

Part 2: Relational Graph Convolutional Network (R-GCN)

Notations of knowledge graph

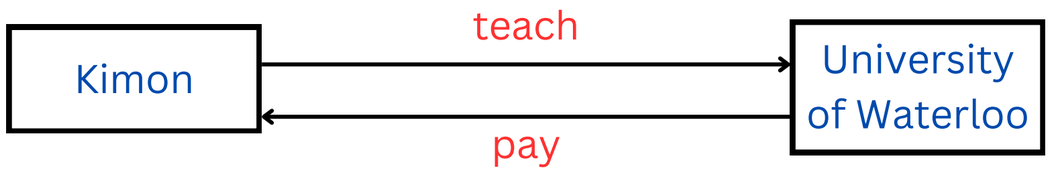
Directed and labeled multi-graphs: $G = (\mathcal{V}, \mathcal{E}, \mathcal{R})$

Node (Entities): $v_i \in \mathcal{V}$

Relation type: $r \in \mathcal{R}$

Labeled edges (relations): $(v_i, r, v_j) \in \mathcal{E}$

v ∈ {Kimon, University of Waterloo}
r ∈ {teach, pay}
e ∈ {(Kimon, teach, University of Waterloo), (university of waterloo, pay, Kimon)}



Relational-Graph Convolutional Network

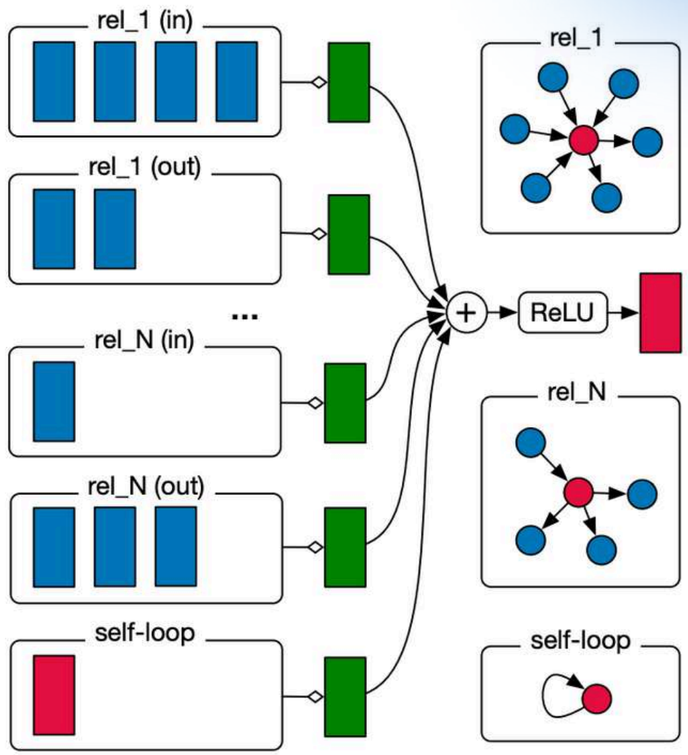
Standard Message Passing:

$$h_i^{(l+1)} = \sigma\left(\sum_{m \in \mathcal{M}_i} g_m(h_i^{(l)}, h_j^{(l)})\right)$$

set of incoming messages

Relational-Graph Convolutional Network:

$$h_i^{(l+1)} = \sigma\left(\sum_{r \in \mathcal{R}} \sum_{j \in \mathcal{N}_i^r} \frac{1}{c_{i,r}} W_r^{(l)} h_j^{(l)} + W_0^{(l)} h_i^{(l)}\right)$$

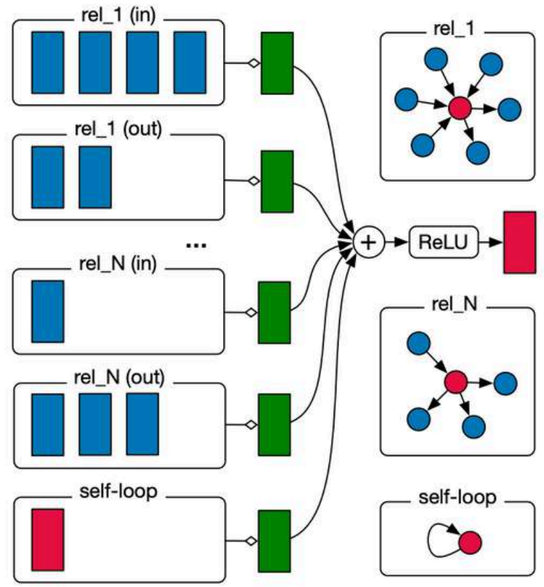


Relational-Graph Convolutional Network

$$h_i^{(l+1)} = \sigma\left(\sum_{r\in\mathcal{R}}\sum_{j\in\mathcal{N}_i^r}\frac{1}{c_{i,r}}W_r^{(l)}h_j^{(l)} + W_0^{(l)}h_i^{(l)}\right)$$

set of relation

set of neighbour of node i under relation r



h: node embeddings/ representation
W: relation-specific transformation
c: normalisation constant (|N| or learnable parameter)

Problem: Easy to overfit to rare relation

Regularisation

Solution 1: Basis Decomposition

All $W_r^{(l)}$ share the same basis

$$W_r^{(l)} = \sum_{b=1}^{B} a_{rb}^{(l)} V_b^{(l)}$$

$$h_i^{(l+1)} = \sigma\left(\sum_{r\in\mathcal{R}}\sum_{j\in\mathcal{N}_i^r}\frac{1}{c_{i,r}} W_r^{(l)} h_j^{(l)} + W_0^{(l)} h_i^{(l)}\right)$$

- W is the linear combination of basis V
- a is the only parameter depends on relation r

Benefit:

1. Reduce overfitting on rare relations
2. Reduce parameter size by parameter sharing

Regularisation

Solution 2: Block Decomposition

$$W_r^{(l)} = \bigoplus_{b=1}^{B} Q_{br}^{(l)}$$

$$W_r^{(l)} = \begin{bmatrix} Q_{1r}^{(l)} & 0 & 0 \\ 0 & Q_{2r}^{(l)} & 0 \\ 0 & 0 & Q_{3r}^{(l)} \end{bmatrix}$$

(Example)

- Sparsity constraint on W
  - must be block-diagonal matrices

Benefit:

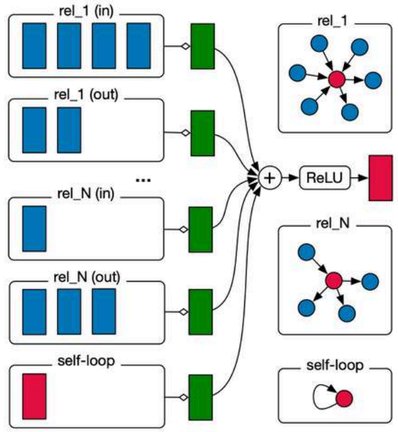
1. Reduce parameter size by parameter sharing

Summary

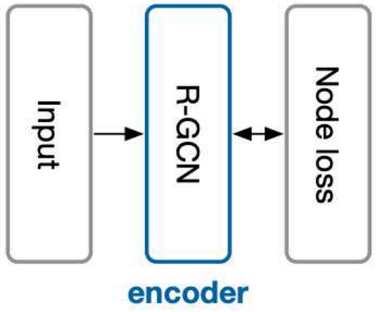
Input of first layer:

- Node features or
- One-hot vector

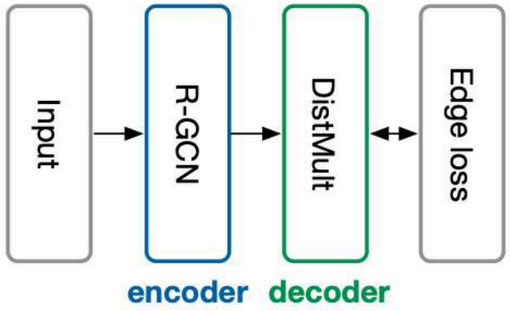
At each layer: $h_i^{(l+1)} = \sigma\left(\sum_{r\in\mathcal{R}}\sum_{j\in\mathcal{N}_i^r}\frac{1}{c_{i,r}}W_r^{(l)}h_j^{(l)} + W_0^{(l)}h_i^{(l)}\right)$



For node classification task:
Softmax Layer

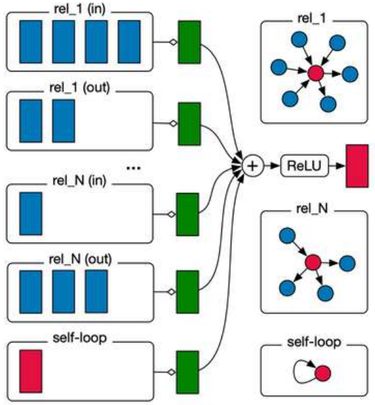


For link prediction task:
Apply a decoder/ scoring function

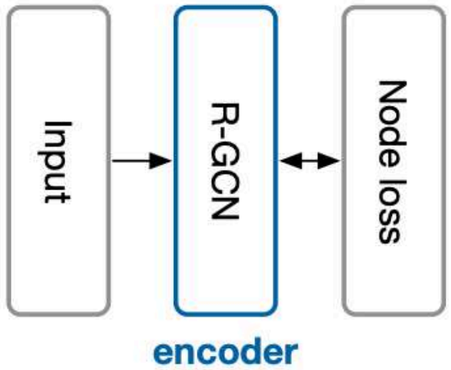


Part 3: Entity Classfication

Architecture & Loss Function



R-GCN layer



$$\mathcal{L} = -\sum_{i \in \mathcal{Y}} \sum_{k=1}^{K} t_{ik} \ln h_{ik}^{(L)}$$

(Loss function)

Y: Set of node indices that have label
k: index of output entry

Datasets for evaluation

Dataset	AIFB	MUTAG	BGS	AM
Entities	8,285	23,644	333,845	1,666,764
Relations	45	23	103	133
Edges	29,043	74,227	916,199	5,988,321
Labeled	176	340	146	1,000
Classes	4	2	2	11

(Converted into Resource Description Framework Format)

Baseline Models for comparison

- RDF2Vec embeddings
- Weisfeiler-Lehman kernels (WL)
- Hand-designed feature extractors (Feat)

Results

Model	AIFB	MUTAG	BGS	AM
Feat	55.55	77.94	72.41	66.66
WL	80.55	**80.88**	86.20	87.37
RDF2Vec	88.88	67.20	**87.24**	88.33
R-GCN	**95.83**	73.23	83.10	**89.29**

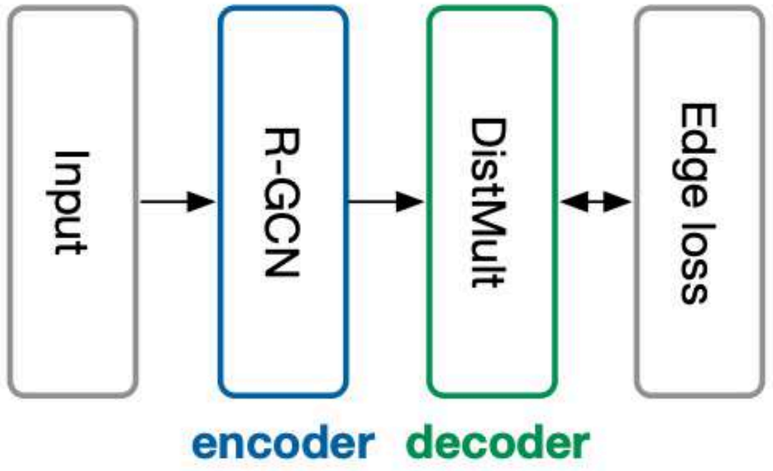
$$h_i^{(l+1)} = \sigma\left(\sum_{r\in\mathcal{R}}\sum_{j\in\mathcal{N}_i^r}\frac{1}{c_{i,r}}W_r^{(l)}h_j^{(l)} + W_0^{(l)}h_i^{(l)}\right)$$

$a_{ij,r}$

- R-GCN achieved competitive results
- To get better result, replace normalisation constant with data-dependent attention weight

Part 4: Link Prediction

Architecture of R-GCN



Encoder: Stack of GCN layers
Decoder: Scoring function (mapping an edge to a score)

Scoring Function

- To predict whether there is an edge (subject, relation, object), the author use scoring function to give a score

Dismult function: $f(s, r, o) = e_s^T R_r e_o$

Embeddings of subject

Diagonal matrix of relation r (learnable)

Embeddings of object

Loss function

- Loss function:

$$\mathcal{L} = -\frac{1}{(1+\omega)|\hat{\mathcal{E}}|} \sum_{(s,r,o,y)\in\mathcal{T}} y \log l\big(f(s,r,o)\big) + (1-y)\log\big(1 - l\big(f(s,r,o)\big)\big)\,,$$

l is the logistic sigmoid function

$\mathcal{T}$ is the total set of real and corrupted triples

Training

- Create corrupted triples with negative sampling
- Adam optimizer with learning rate 0.01
- Regularise edge with dropout rate 0.2 for self-loops and 0.4 for other edges
- l2 regularisation to decoder with penalty 0.01

R-GCN+

$$f(s,r,t)_{\text{R-GCN+}}$$

$$= \alpha f(s,r,t)_{\text{R-GCN}} + (1-\alpha) f(s,r,t)_{\text{DistMult}}$$

where $\alpha = 0.4$

Dataset for evaluation

- WN18 (subset of WordNet containing lexical relations between words)
- FB15k (subset of relational database freebase)
- FB15k-237 (FB15k, but all inverse relations are removed to prevent memorization)

Dataset	WN18	FB15K	FB15k-237
Entities	40,943	14,951	14,541
Relations	18	1,345	237
Train edges	141,442	483,142	272,115
Val. edges	5,000	50,000	17,535
Test edges	5,000	59,071	20,466

Evaluation Metric

Mean reciprocal rank

$$MRR = \frac{1}{|Q|}\sum_{i=1}^{|Q|}\frac{1}{\text{rank}_i}$$

- Query 1 → Correct answer at **rank 1** → $\frac{1}{1} = 1.0$
- Query 2 → Correct answer at **rank 3** → $\frac{1}{3} = 0.333$
- Query 3 → Correct answer at **rank 2** → $\frac{1}{2} = 0.5$

$$MRR = \frac{1}{3}(1.0 + 0.333 + 0.5) = \frac{1.833}{3} = 0.611$$

Hit @ n

$$\text{Hits@N} = \frac{\text{Number of queries where correct answer appears in top-N}}{\text{Total number of queries}}$$

Prediction:

1. "Germany" ❌
2. "Switzerland" ❌
3. "Austria" ✅
4. "USA" ❌
5. "France" ❌

Baseline Models

Classical algorithms:

- CP
- TransW

SOTA models for FB15k and WN18:

- ComplWx
- HolE

Results

	FB15k					WN18				
	MRR		Hits @			MRR		Hits @		
Model	Raw	Filtered	1	3	10	Raw	Filtered	1	3	10
LinkFeat		0.779			0.804		0.938			0.939
DistMult	0.248	0.634	0.522	0.718	0.814	0.526	0.813	0.701	0.921	0.943
R-GCN	0.251	0.651	0.541	0.736	0.825	0.553	0.814	0.686	0.928	0.955
R-GCN+	**0.262**	**0.696**	**0.601**	**0.760**	**0.842**	0.561	0.819	0.697	0.929	**0.964**
CP*	0.152	0.326	0.219	0.376	0.532	0.075	0.058	0.049	0.080	0.125
TransE*	0.221	0.380	0.231	0.472	0.641	0.335	0.454	0.089	0.823	0.934
HolE**	0.232	0.524	0.402	0.613	0.739	**0.616**	0.938	0.930	**0.945**	0.949
ComplEx*	0.242	0.692	0.599	0.759	0.840	0.587	**0.941**	**0.936**	**0.945**	0.947

Results

	MRR		Hits @		
Model	Raw	Filtered	1	3	10
LinkFeat		0.063			0.079
DistMult	0.100	0.191	0.106	0.207	0.376
R-GCN	**0.158**	0.248	**0.153**	0.258	0.414
R-GCN+	0.156	**0.249**	0.151	**0.264**	**0.417**
CP	0.080	0.182	0.101	0.197	0.357
TransE	0.144	0.233	0.147	0.263	0.398
HolE	0.124	0.222	0.133	0.253	0.391
ComplEx	0.109	0.201	0.112	0.213	0.388

- Surpass the baseline by a margin of 29.8%

Part 5: Conclusion

Summary

- Demonstrated GNN achieved competitive results in recovering knowledge graph
- Proposed parameter sharing techniques for GNNs (basis & block decomposition)

Future work

- Experiment with more powerful GNNs architecture
- Is knowledge graph still useful at the LLM era? Can it ground the response of LLM?

THANK YOU FOR LISTENING!

Presented by

Lee Guan Bo Ambrose

BASc (Applied AI)

UNIVERSITY OF WATERLOO | DAVID R. CHERITON SCHOOL OF COMPUTER SCIENCE

SCHOOL OF COMPUTING & DATA SCIENCE
The University of Hong Kong