GCN-VAE for Knowledge Graph Completion

Karen Yang

Stanford University

Motivation

- Knowledge graphs are useful abstraction for relational data but often have missing links.
- Most methods failed to capture multiple relation semantics when a point vector presentation.
- Few studies have applied generative models for representation learning on knowledge graphs.

Problem Formulation

- Input preprocessing

set of triplets { (head entity, relation type, tail entity) }

Input Knowledge Graph: $G = (E, A)$
Entity attribute matrix: $E \in \mathbb{R}^{n \times d_e}$
Relation-specific Adjacency List: $A = \{A_\tau, \forall \tau \in R\}$

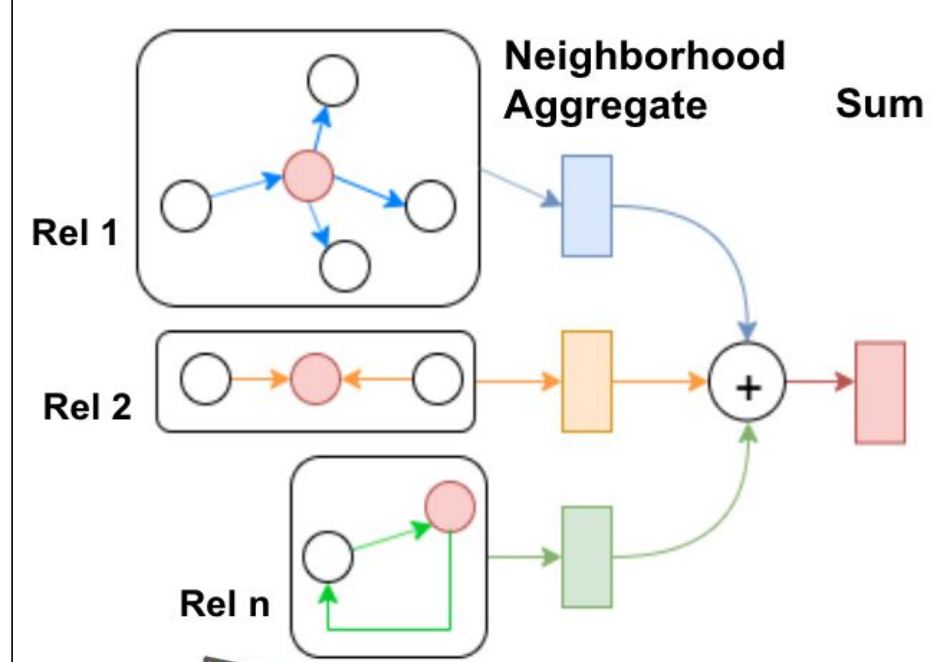
- Only 50% of edges in input graph are shown to model.
- Latent z: distribution of entity embedding given input graph
- Output: given relation τ, predict likelihood of triplets, with learned relation embedding.

Experiments

- **Link Prediction**
  - Predict whether a relation exists between a pair of entities.
  - Use negative sampling to add a balanced set of positive and negative (false) triplets for training.
- **Dataset**

Datasets	#entities	#relations	#triplets	#train triplets	#test triplets
FB15K237	14541	237	272115	251649	20466
WN18RR	40943	11	40943	37809	3134

Model Overview



- **Encoder** - A multi-relational extension of graph convolution network to encode local neighborhood info..
- **Latent** - Entity embedding's distribution
- **Decoder** - Conditioned on **relation**, output likelihood scores for possible triplets

$$p_\theta((z, \tau, z') \in G | z, \tau)$$

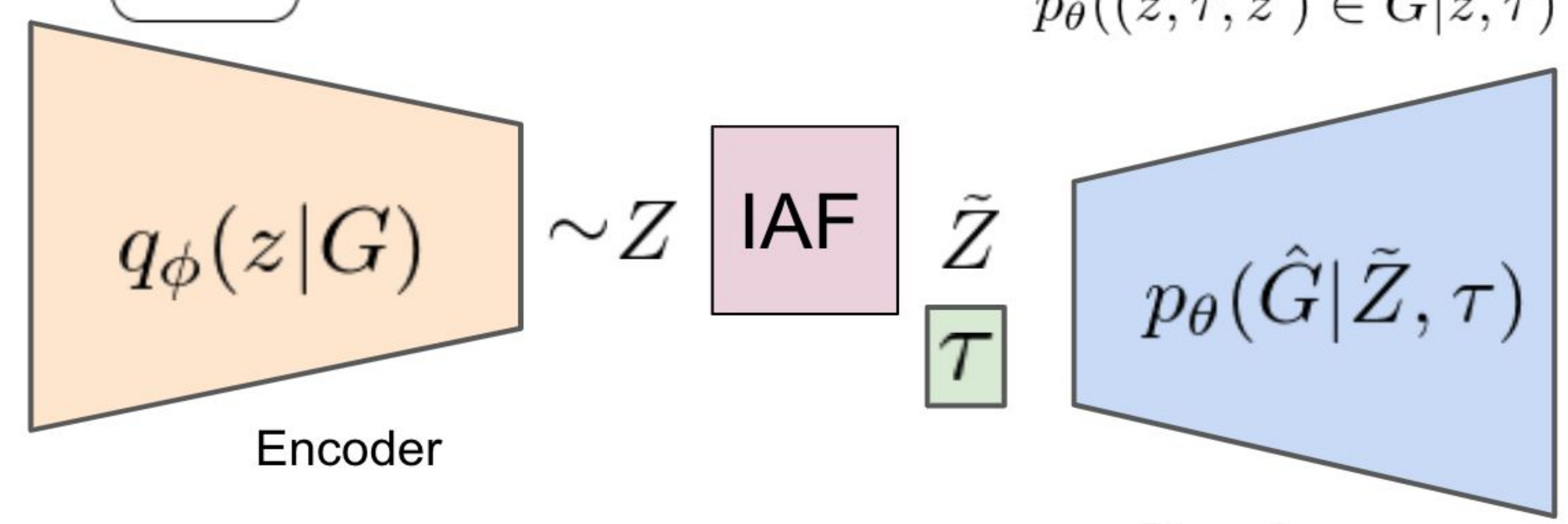


Figure1. GCN-VAE Model Architecture

Methods

- Relational Graph Convolution Network as Encoder

$$H^{(l)} = \sigma(B_l + \sum_{\tau \in R} A_\tau H^{(l-1)} W_l)$$

- Bilinear diagonal model as decoder: $\langle \mathbf{r}, \mathbf{h}, \mathbf{t} \rangle$ (dot product)
- Objective:

$$\mathcal{L}(\phi, \theta; G) = -ELBO(\phi, \theta; G) = \mathbb{E}_{q_\phi(z|G)}[-\log p_\theta(G|z, \tau)] + \mathcal{D}_{KL}[q_\phi(z|G)||p(z)]$$

- Mixture of Gaussians as prior:

$$p_\theta(\mathbf{z}) = \sum_{i=1}^{k} \frac{1}{k} \mathcal{N}(\mathbf{z}|\mu_i, \mathrm{diag}(\sigma_i^2))$$

- Inverse Autoregressive Flow to transform posterior:

$$z_i = \frac{x_i - \mu_i(\mathbf{x}_{1:i-1})}{\sigma_i(\mathbf{x}_{1:i-1})} = -\frac{\mu_i(\mathbf{x}_{1:i-1})}{\sigma_i(\mathbf{x}_{1:i-1})} + x_i \odot \frac{1}{\sigma_i(\mathbf{x}_{1:i-1})}$$

- More Informative Latent Codes With Maximum Mean Discrepancy (MMD)

$$\mathbb{E}_{p(z),p(z')}[k(z, z')] + \mathbb{E}_{q(z),q(z')}[k(z, z')] - 2\mathbb{E}_{p(z),q(z')}[k(z, z')]$$

Results

Model	FB15k-237 MRR Raw	Hits @ 1	Hits @ 3	Hits @ 10
TransE	.144	.147	.263	.398
DistMult	.241	.155	.263	.419
RotateE	.338	.241	**.375**	.533
R-GCN	.248	.153	.258	.417
TransG(generative)	.304	.182	.298	.471
GCN-VAE(k=1)	.279	.189	.300	.465
GCN-GMVAE(k=10)	.281	.191	.300	.471
GCN-GMVAE-IAF	**.343**	**.243**	**.375**	**.548**

Table 1. Model comparison on Mean reciprocal rank (MRR) and Hits@m on variants of GCN-VAE and other common methods.

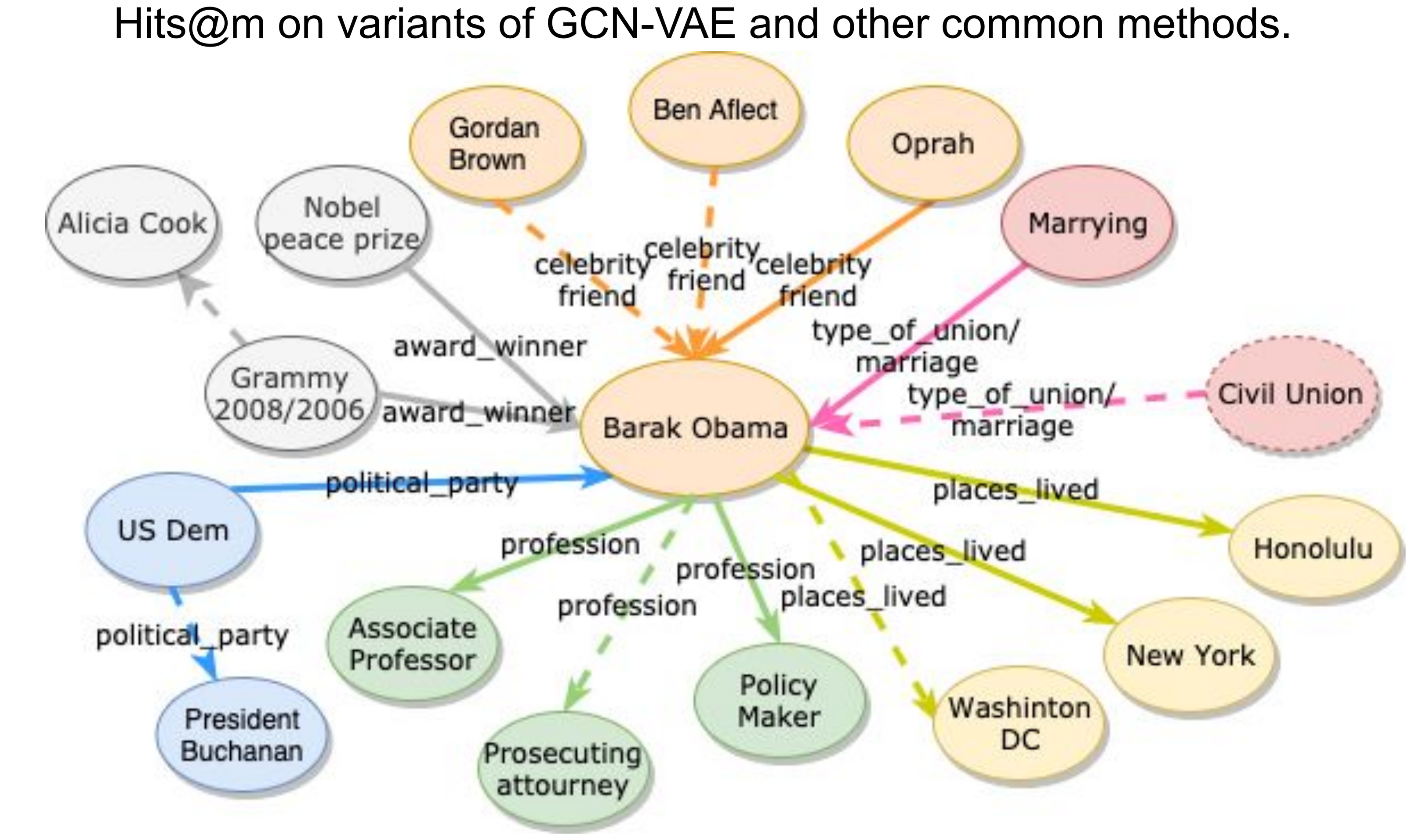


Figure 2. Generated neighborhood around entity "Barak Obama". Dotted lines are predicted links that were not in original dataset.

Conclusion

- Learned a powerful representation of knowledge graph's relational structure with VAE and relational graph convolutional network.
- Improve link prediction performance on SOTA baselines.

Reference

Aditya Grover, Aaron Zweig, and Stefano Ermon. Graphite: Iterative generative modeling of graphs. *arXiv preprint arXiv:1803.10459*, 2018.

Michael Schlichtkrull, Thomas N Kipf, Peter Bloem, Rianne Van Den Berg, Ivan Titov, and Max Welling. Modeling relational data with graph convolutional networks. In *European Semantic Web Conference*, pages 593–607. Springer, 2018.

Keyulu Xu, Weihua Hu, Jure Leskovec, and Stefanie Jegelka. How powerful are graph neural networks? *arXiv preprint arXiv:1810.00826*, 2018.