Knowledge Graph Embeddings: Recent Advances

William Wang

UC SANTA BARBARA

Joint work with Liwei Cai

CIPS Summer School 2018

Outline

- Related Work
- KBGAN: Algorithm
- Experiments
- Conclusion

Related Work

- **Embedding-based method**
  - RESCAL, Nickel et al, 2011
  - TransE, Bordes et al, 2013
  - Neural Tensor Network, Socher et al, 2013
  - TransR/CTransR, Lin et al, 2015
  - Complex Embeddings, Trouillon et al, 2016

Embedding methods allow us to compare, and find similar entities in the vector space.

UCSB

RESCAL (Nickel et al., 2011)

- Tensor factorization on the
  - (head)entity-(tail)entity-relation tensor.

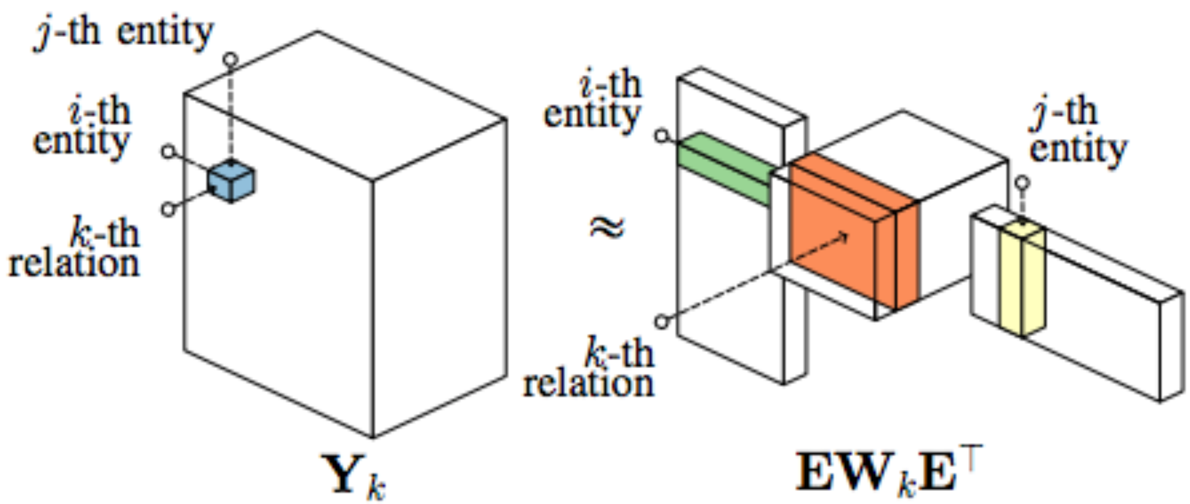


Fig. 4. RESCAL as a tensor factorization of the adjacency tensor **Y**.

UCSB

TransE (Bordes et al., 2013)

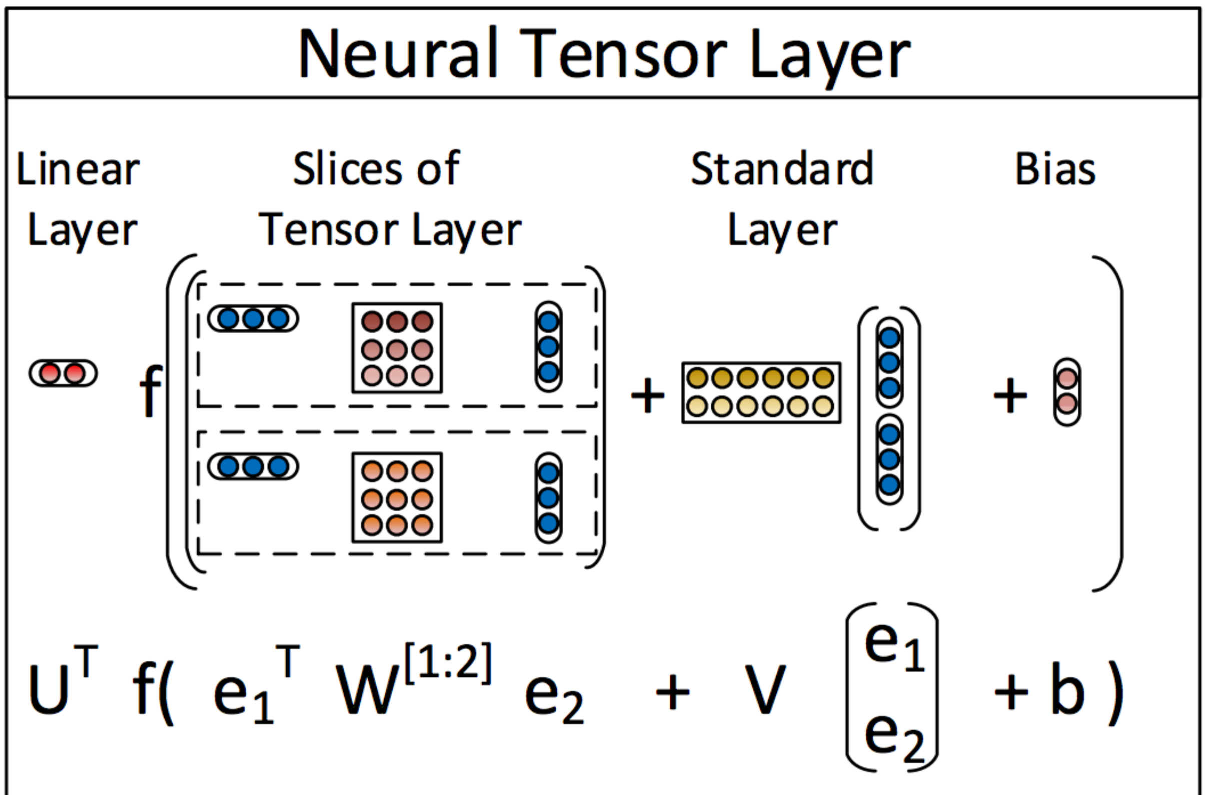
- Assumption: in the vector space, when adding the relation to the head entity, we should get close to the target tail entity.
- Margin based loss function:
  - Minimize the distance between (h+l) and t.
  - Maximize the distance between (h+l) to a randomly sampled tail t' (negative example).

$$\mathcal{L} = \sum_{(h,\ell,t)\in S} \sum_{(h',\ell,t')\in S'_{(h,\ell,t)}} \left[\gamma + d(\boldsymbol{h} + \boldsymbol{\ell}, \boldsymbol{t}) - d(\boldsymbol{h'} + \boldsymbol{\ell}, \boldsymbol{t'})\right]_+$$

UCSB

Neural Tensor Networks (Socher et al., 2013)

- Model the bilinear interaction between entity pairs with tensors.

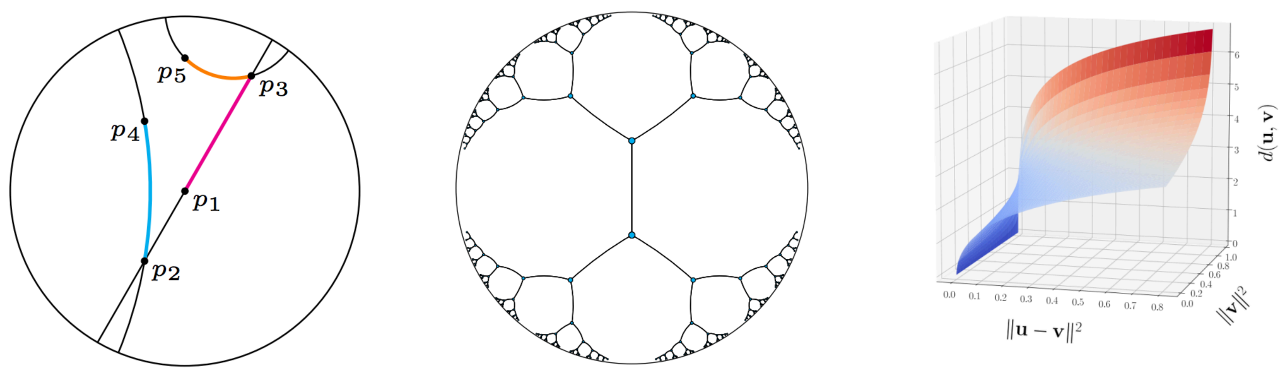


UCSB

Poincaré Embeddings (Nickel and Kiela, 2017)

- Idea: learn hierarchical KB representations by looking at hyperbolic space.

$$d(\boldsymbol{u}, \boldsymbol{v}) = \operatorname{arcosh}\left(1 + 2\frac{\|\boldsymbol{u} - \boldsymbol{v}\|^2}{(1 - \|\boldsymbol{u}\|^2)(1 - \|\boldsymbol{v}\|^2)}\right).$$



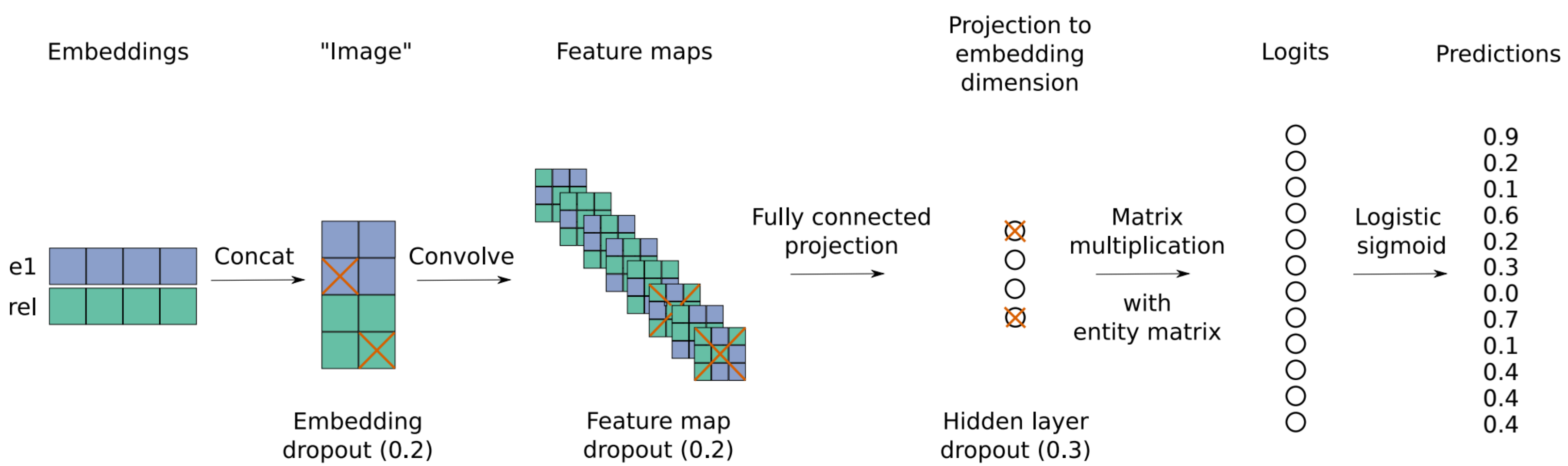
(a) Geodesics of the Poincaré disk (b) Embedding of a tree in $\mathcal{B}^2$ (c) Growth of Poincaré distance

Figure 1: (a) Due to the negative curvature of $\mathcal{B}$, the distance of points increases exponentially (relative to their Euclidean distance) the closer they are to the boundary. (c) Growth of the Poincaré distance $d(\boldsymbol{u}, \boldsymbol{v})$ relative to the Euclidean distance and the norm of $\boldsymbol{v}$ (for fixed $\|\boldsymbol{u}\| = 0.9$). (b) Embedding of a regular tree in $\mathcal{B}^2$ such that all connected nodes are spaced equally far apart (i.e., all black line segments have identical hyperbolic length).

UCSB

ConvE (Dettmers et al, 2018)

- 1. Reshape the head and relation embeddings into “images”.
- 2. Use CNNs to learn convolutional feature maps.

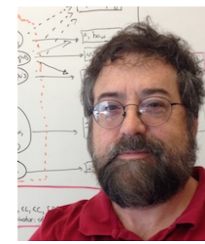


UCSB

It all started in 2013

Me: How did you get negative examples from knowledge graphs?

William Cohen: We did some samplings from the knowledge graph.



Me: OK... (🙄)

UCSB

Reality about Knowledge Bases

- Only positive facts are stored, and no negative examples are stored.
  - This makes sense, for efficiency considerations.
- But for machine learning (e.g., margin-based models)
  - We often need negative examples.

UCSB

Negative Sampling is Pervasive

- TransE (Bordes et al., 2013): Replace head/tail entity with a randomly sampled entity from KB to create a negative example.
- Margin-based loss function:
  - Positive Examples: Minimize the distance between (h+l) and t.
  - Negative Examples: Maximize the distance between (h+l) to a randomly sampled tail t' (negative example).

$$\mathcal{L} = \sum_{(h,\ell,t)\in S} \sum_{(h',\ell,t')\in S'_{(h,\ell,t)}} \left[\gamma + d(\boldsymbol{h}+\boldsymbol{\ell},\boldsymbol{t}) - d(\boldsymbol{h'}+\boldsymbol{\ell},\boldsymbol{t'})\right]_+$$

UCSB

Negative Sampling's Main Issue

- Main Issue for KB Embedding:
  - It often generates low-quality negative examples that do not help you learn good embedding models.

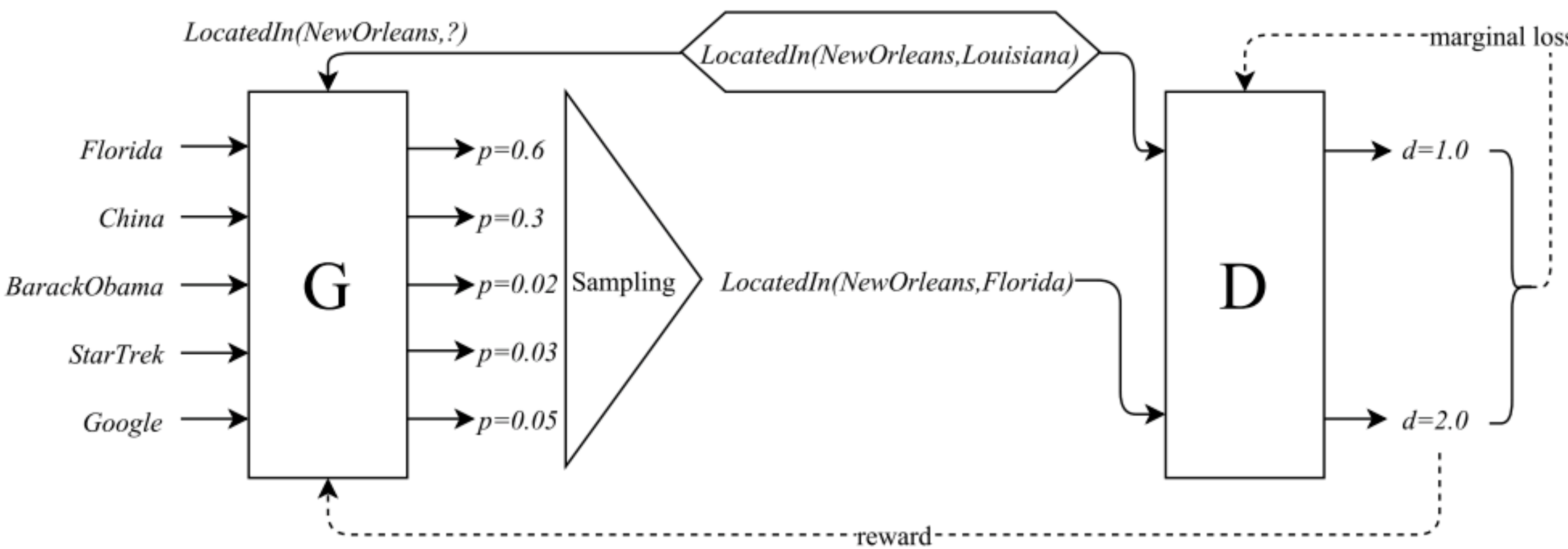
LocatedIn(NewOrleans, [redacted])

LocatedIn(NewOrleans, Google)

UCSB

KBGAN: Learning to Generate High-Quality Negative Examples

Idea: use adversarial learning to iteratively learn better negative examples.



UCSB

KBGAN: Overview

- Both G and D are KG embedding models.

- Input:
  - Pre-trained generator G with score function $f_G(h, r, t)$.
  - Pre-trained discriminator D with score function $f_D(h, r, t)$.
- Adversarial Learning:
  - Use softmax to score and rank negative triples.
  - Update D with original positive examples and highly-ranked negative examples.
  - Pass the reward for policy gradient update for G.

- Output:
  - Adversarially trained KG embedding discriminator D.

UCSB

KBGAN: Adversarial Negative Training

For each positive triple from the minibatch:

1. Generator: Rank negative examples.

$$p_G(h', r, t' | h, r, t) = \frac{\exp f_G(h', r, t')}{\sum \exp f_G(h^*, r, t^*)}$$
$$(h^*, r, t^*) \in Neg(h, r, t)$$

2. Discriminator: Standard margin-based update.

$$L_D = \sum_{(h,r,t) \in \mathcal{T}} [f_D(h, r, t) - f_D(h', r, t') + \gamma]_+$$
$$(h', r, t') \sim p_G(h', r, t' | h, r, t) \quad (3)$$

UCSB

KBGAN: Adversarial Training (cont'd)

3. Compute the Reward for the Generator.

$$\mathrm{r} = -f_D(h', r, t').$$

4. Policy gradient update for the Generator.

$$G_G \longleftarrow G_G + (r - b)\nabla_{\theta_G} \log p_s;$$

The baseline b is total reward sum / mini-batch size.

UCSB

Experimental Settings

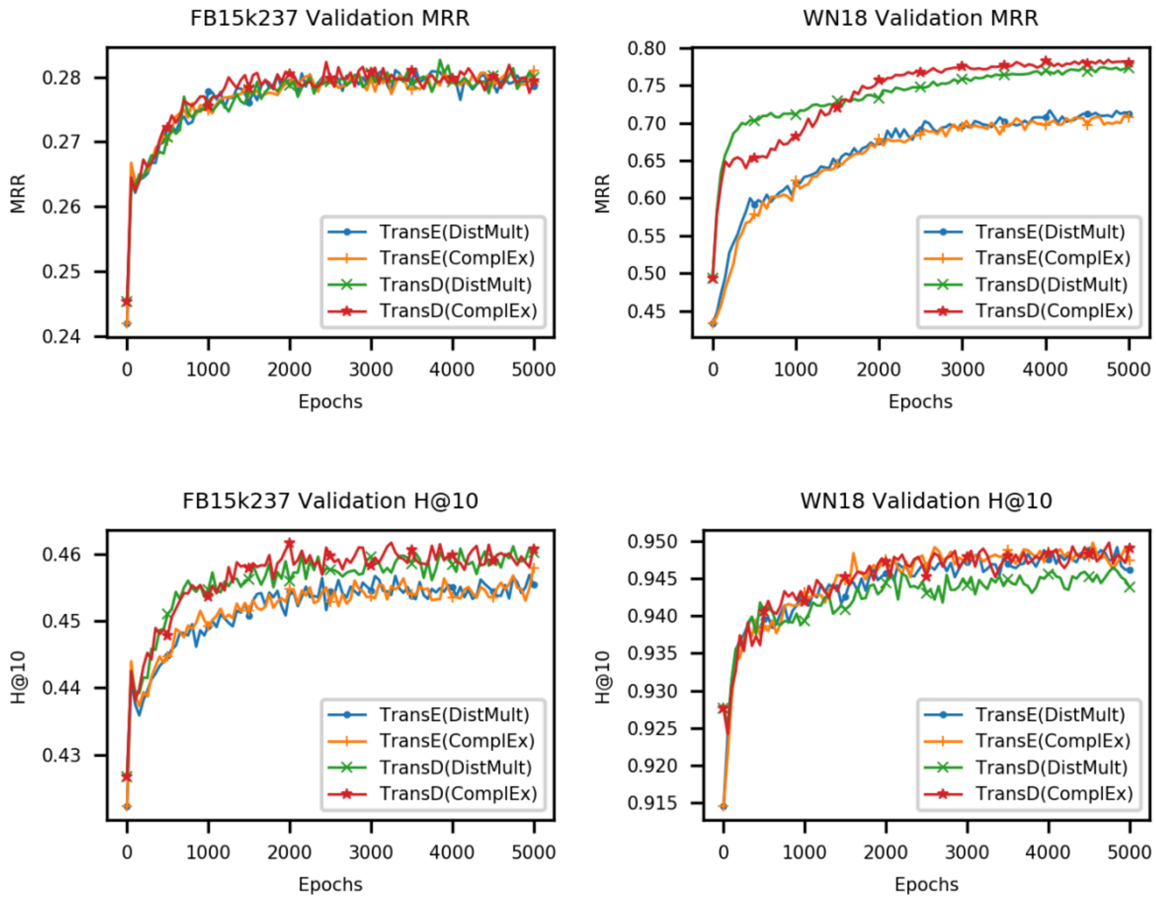
- Datasets: three standard KB completion datasets.
- Hyperparameters: documented in details in the paper.
- Metrics: Hits@10 and Mean Reciprocal Rank (MRR).

Dataset	**#r**	**#ent.**	**#train**	**#val**	**#test**
FB15k-237	237	14,541	272,115	17,535	20,466
WN18	18	40,943	141,442	5,000	5,000
WN18RR	11	40,943	86,835	3,034	3,134

Experimental Results

Method	FB15k-237 MRR	FB15k-237 H@10	WN18 MRR	WN18 H@10	WN18RR MRR	WN18RR H@10
TransE	-	42.8†	-	89.2	-	43.2†
TransD	-	45.3†	-	92.2	-	42.8†
DistMult	24.1‡	41.9‡	82.2	93.6	42.5‡	49.1‡
ComplEx	24.0‡	41.9‡	**94.1**	94.7	**44.4**‡	**50.7**‡
TransE (pre-trained)	24.2	42.2	43.3	91.5	18.6	45.9
KBGAN (TransE + DistMult)	27.4	45.0	71.0	<u>**94.9**</u>	21.3	<u>48.1</u>
KBGAN (TransE + ComplEx)	<u>**27.8**</u>	45.3	70.5	<u>**94.9**</u>	21.0	47.9
TransD (pre-trained)	24.5	42.7	49.4	92.8	19.2	46.5
KBGAN (TransD + DistMult)	<u>**27.8**</u>	<u>**45.8**</u>	77.2	94.8	21.4	47.2
KBGAN (TransD + ComplEx)	27.7	<u>**45.8**</u>	<u>77.9</u>	94.8	<u>21.5</u>	46.9

Convergence Analysis



Conclusion

- We propose an adversarial learning approach for generating high-quality negative examples.
- Our approach is model-agnostic, and it can be applied to various knowledge graph embedding models.
- Our work has shown improvements with various settings on two datasets.

Thank you!

- Code: https://github.com/cai-lw/KBGAN