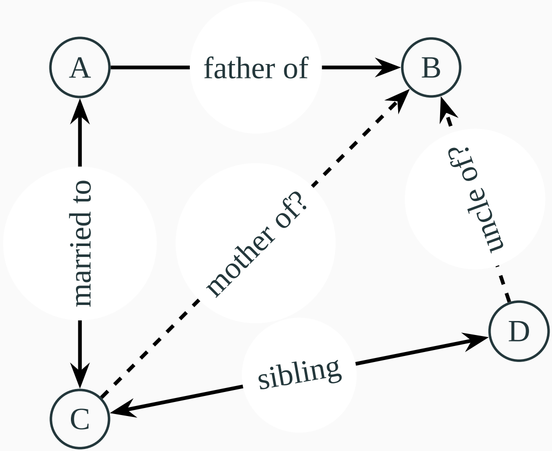
TuckER: Tensor Factorization for Knowledge Graph Completion

Ivana Balažević, Carl Allen and Timothy Hospedales

November 7, 2019

CDT in Data Science, School of Informatics, University of Edinburgh
2019 Conference on Empirical Methods in Natural Language Processing (EMNLP)

Task: Link Prediction on Knowledge Graphs



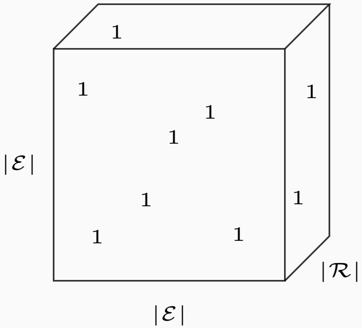
Entities $\mathcal{E} = \{A, B, C, D\}$

Relations $\mathcal{R} = \{\textit{married to}, \textit{father of}, \textit{uncle of}, ...\}$

Knowledge Graph $\mathcal{G} = \{(A, \textit{father of}, B), (A, \textit{married to}, C), ...\}$

Background: Binary Tensor Representation

An alternative view of a knowledge graph: sparse **binary adjacency tensor representation** of known facts.



Background: Score Function

Subject and object entities e_s, e_o are **vectors** (embeddings), shared across all relations; relations r are (non-)linear **transformations**.

A **score function** $\phi : \mathcal{E} \times \mathcal{R} \times \mathcal{E} \to \mathbb{R}$ is typically learned to assign a score $s = \phi(e_s, r, e_o)$ to each triple.

Score functions are defined by:

- **form of relation representation** (e.g. matrix multiplication, vector addition); and
- **proximity measure** (e.g. dot product, Euclidean distance).

Background: Score Function

Type	$\phi(e_s, r, e_o)$	Models
Linear	$\mathbf{e}_s^\top \mathbf{W}_r \mathbf{e}_o = \langle \mathbf{e}_s^{(r)}, \mathbf{e}_o \rangle$	RESCAL (Nickel et al., 2011) DistMult (Yang et al., 2015) ComplEx (Trouillon et al., 2016) SimplE (Kazemi and Poole, 2018) **TuckER** (ours)
Translational	$-\lVert \mathbf{e}_s \mathbf{W}_r^s + \mathbf{r} - \mathbf{e}_o \mathbf{W}_r^o \rVert_p = -\lVert \mathbf{e}_s^{(r)} + \mathbf{r} - \mathbf{e}_o^{(r)} \rVert_p$	TransE (Bordes et al., 2013) STransE (Nguyen et al., 2016))
Non-linear	$f_r(\mathbf{e}_s, \mathbf{e}_o)$	ConvE (Dettmers et al., 2018) HypER (Balažević et al., 2019)

Table 1: Types of score functions.

Background: Linear Models

All linear models **factorize** the binary adjacency tensor (subject to σ).

Model	$\phi(e_s, r, e_o)$	Notes
RESCAL	$\mathbf{e}_s^\top \mathbf{W}_r \mathbf{e}_o$	prone to overfitting
DistMult	$\mathbf{e}_s^\top \mathrm{diag}(\mathbf{w}_r) \mathbf{e}_o$	cannot model asymmetric relations
ComplEx	$\mathrm{Re}(\mathbf{e}_s^\top \mathrm{diag}(\mathbf{w}_r) \overline{\mathbf{e}}_o)$	extends DistMult to complex domain
SimplE	$\frac{1}{2}(\mathbf{h}_{e_s}^\top \mathrm{diag}(\mathbf{w}_r) \mathbf{t}_{e_o} + \mathbf{h}_{e_o}^\top \mathrm{diag}(\mathbf{w}_r^{-1}) \mathbf{t}_{e_s})$	extends DistMult with inverse relations
TuckER	$\mathcal{W} \times_1 \mathbf{e}_s \times_2 \mathbf{w}_r \times_3 \mathbf{e}_o$	enables multi-task learning

Table 2: Linear models.

Spoiler: all models above are a special case of TuckER!

TuckER: Tensor Factorization for Knowledge Graph Completion

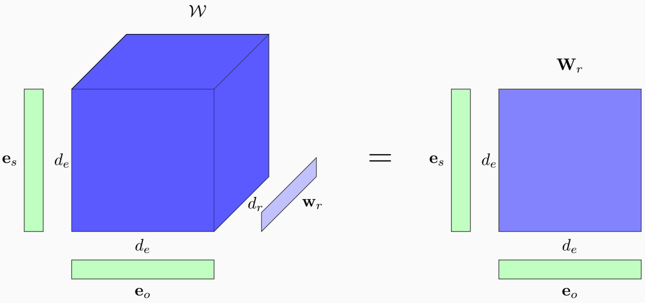
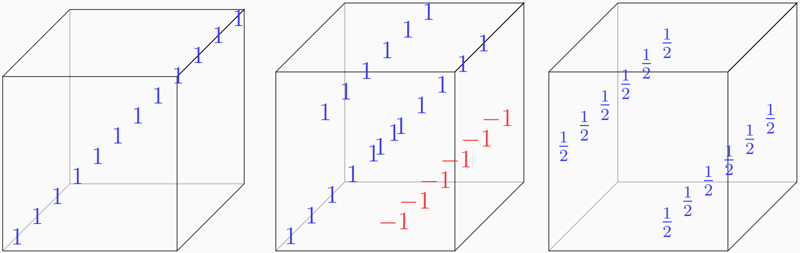


Figure 1: Visualization of the TuckER architecture.

$$\phi(e_s, r, e_o) = \mathcal{W} \times_1 \mathbf{e}_s \times_2 \mathbf{w}_r \times_3 \mathbf{e}_o \rightarrow \textbf{multi-task learning}$$

Intuition: Rather than learning distinct relation-specific matrices $\mathbf{W}_r$, the core tensor $\mathcal{W}$ contains a shared pool of "prototype" relation matrices, which are linearly combined using the parameters in each relation embedding $\mathbf{w}_r$.

Previous Models as Special Cases of TuckER



(a) DistMult **(b)** ComplEx **(c)** SimplE

Figure 2: Constraints imposed on the values of core tensor $\mathcal{W} \in \mathbb{R}^{d_e \times d_e \times d_e}$ for DistMult and $\mathcal{W} \in \mathbb{R}^{2d_e \times 2d_e \times 2d_e}$ for ComplEx and SimplE.

Results on WN18RR and FB15k-237

		WN18RR				FB15k-237			
	Linear	MRR	H@10	H@3	H@1	MRR	H@10	H@3	H@1
DistMult (Yang et al., 2015)	yes	.430	.490	.440	.390	.241	.419	.263	.155
ComplEx (Trouillon et al., 2016)	yes	.440	.510	.460	.410	.247	.428	.275	.158
Neural LP (Yang et al., 2017)	no	–	–	–	–	.250	.408	–	–
R-GCN (Schlichtkrull et al., 2018)	no	–	–	–	–	.248	.417	.264	.151
MINERVA (Das et al., 2018)	no	–	–	–	–	–	.456	–	–
ConvE (Dettmers et al., 2018)	no	.430	.520	.440	.400	.325	.501	.356	.237
HypER (Balažević et al., 2019)	no	.465	.522	.477	.436	.341	.520	.376	.252
M-Walk (Shen et al., 2018)	no	.437	–	.445	.414	–	–	–	–
TuckER (ours)	yes	**.470**	**.526**	**.482**	**.443**	**.358**	**.544**	**.394**	**.266**

Table 3: Link prediction results on WN18RR and FB15k-237.

Influence of Multi-task Learning

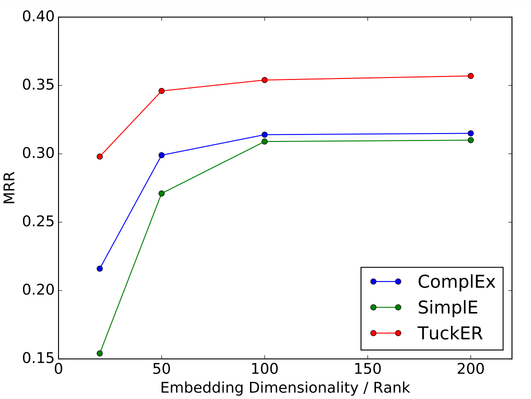


Figure 3: MRR for different embeddings dimensionalities on FB15k-237.

Conclusion

- TuckER is a **straightforward**, yet highly expressive **linear model** for link prediction on knowledge graphs.
- TuckER achieves **state-of-the-art** link prediction results, mainly due to **multi-task learning** between relations.
- TuckER's number of parameters increases **linearly** with the number of entities or relations in a knowledge graph.
- Previous linear state-of-the-art models (RESCAL, DistMult, ComplEx, SimplE) are all special cases of TuckER.

`https://github.com/ibalazevic/TuckER`

`ibalazevic`

Thanks!

If you are interested:

- **Multi-relational Poincaré Graph Embeddings**
  Ivana Balažević, Carl Allen, Timothy Hospedales
  Advances in Neural Information Processing Systems, 2019.
- **What the Vec? Towards Probabilistically Grounded Embeddings**
  Carl Allen, Ivana Balažević, Timothy Hospedales
  Advances in Neural Information Processing Systems, 2019.
- **On Understanding Knowledge Graph Representation**
  Carl Allen*, Ivana Balažević*, Timothy Hospedales
  arXiv preprint arXiv:1909.11611, 2019.

Any questions?

References

Ivana Balažević, Carl Allen, and Timothy M Hospedales. Hypernetwork Knowledge Graph Embeddings. In *ICANN*, 2019.

Antoine Bordes, Nicolas Usunier, Alberto Garcia-Duran, Jason Weston, and Oksana Yakhnenko. Translating Embeddings for Modeling Multi-relational Data. In *NeurIPS*, 2013.

Rajarshi Das, Shehzaad Dhuliawala, Manzil Zaheer, Luke Vilnis, Ishan Durugkar, Akshay Krishnamurthy, Alex Smola, and Andrew McCallum. Go for a Walk and Arrive at the Answer: Reasoning over Paths in Knowledge Bases Using Reinforcement Learning. In *ICLR*, 2018.

Tim Dettmers, Pasquale Minervini, Pontus Stenetorp, and Sebastian Riedel. Convolutional 2D Knowledge Graph Embeddings. In *AAAI*, 2018.

Seyed Mehran Kazemi and David Poole. SimplE Embedding for Link Prediction in Knowledge Graphs. In *NeurIPS*, 2018.

Dat Quoc Nguyen, Kairit Sirts, Lizhen Qu, and Mark Johnson. STransE: a Novel Embedding Model of Entities and Relationships in Knowledge Bases. In *NAACL-HLT*, 2016.

References ii

Maximilian Nickel, Volker Tresp, and Hans-Peter Kriegel. A Three-Way Model for Collective Learning on Multi-Relational Data. In *ICML*, 2011.

Michael Schlichtkrull, Thomas N Kipf, Peter Bloem, Rianne van den Berg, Ivan Titov, and Max Welling. Modeling Relational Data with Graph Convolutional Networks. In *ESWC*, 2018.

Yelong Shen, Jianshu Chen, Po-Sen Huang, Yuqing Guo, and Jianfeng Gao. M-Walk: Learning to Walk over Graphs using Monte Carlo Tree Search. In *NeurIPS*, 2018.

Théo Trouillon, Johannes Welbl, Sebastian Riedel, Éric Gaussier, and Guillaume Bouchard. Complex Embeddings for Simple Link Prediction. In *ICML*, 2016.

Bishan Yang, Wen-tau Yih, Xiaodong He, Jianfeng Gao, and Li Deng. Embedding Entities and Relations for Learning and Inference in Knowledge Bases. In *ICLR*, 2015.

Fan Yang, Zhilin Yang, and William W Cohen. Differentiable Learning of Logical Rules for Knowledge Base Reasoning. In *NeurIPS*, 2017.