Knowledge and Retrieval

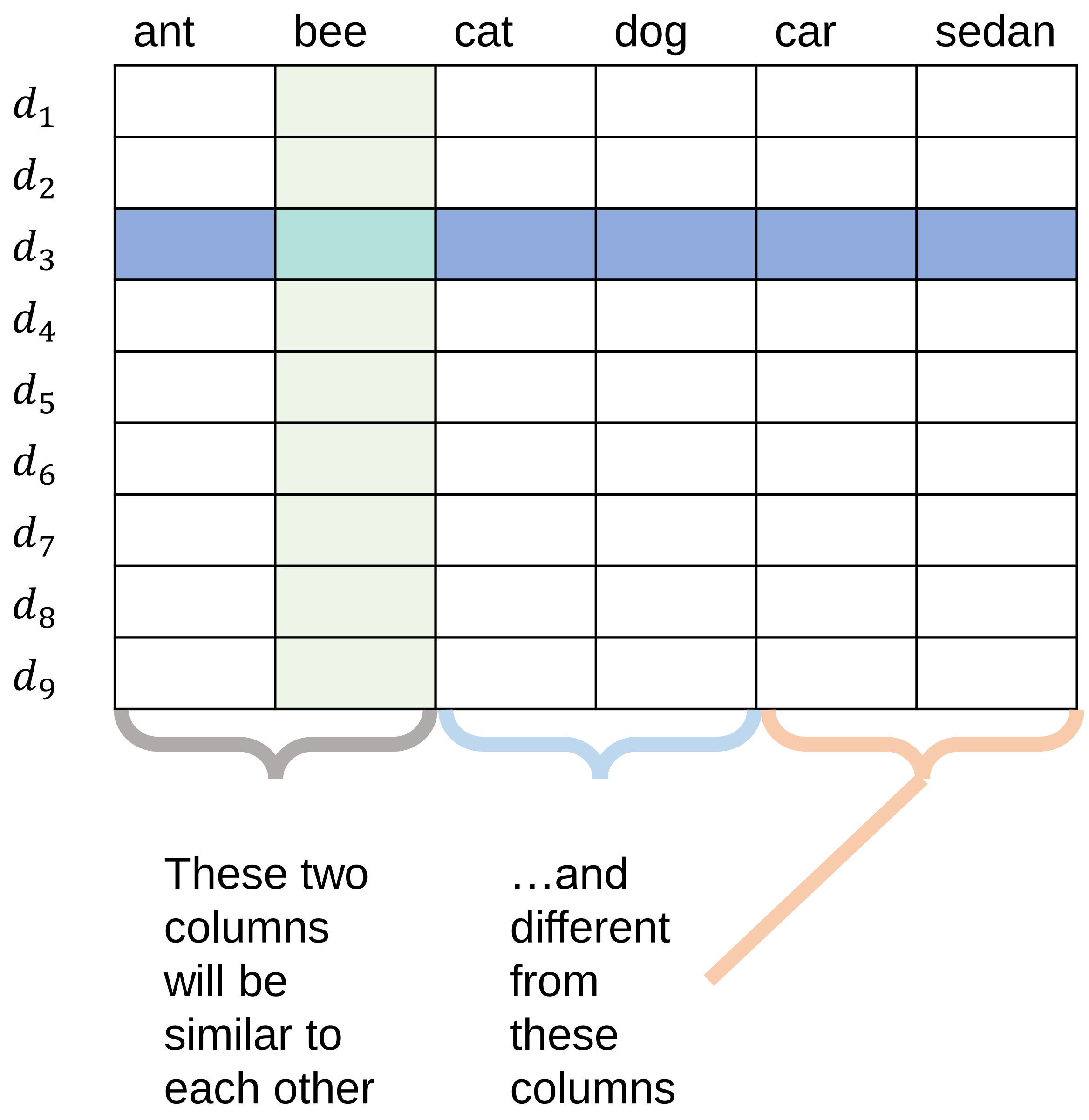
Multiplicative Models

From word to graph embeddings

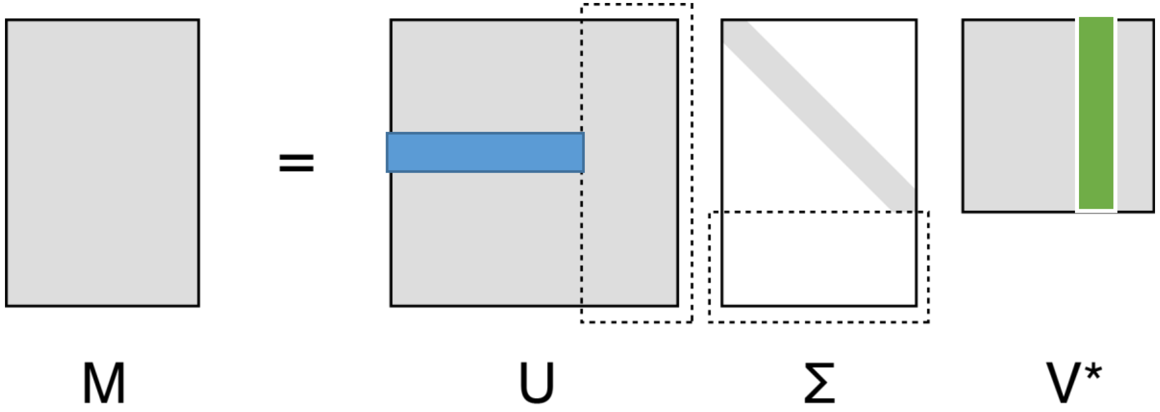
- Factor analysis known from 1960s
- Refined as Latent Semantic Analysis in 1988
- Used in NLP as early as in 1992
- Popularized by word2vec and GloVE in 2013
- We will review these briefly
- This will help understand an important family of KG embedding techniques

Document × term matrix M

- Row = document (ID)
- Column = term / word (ID)
- Element can be
  - Boolean (does term occur in document?)
  - Count (how many times?)
  - Some transformation (e.g., normalize doc length)
- Document contains words in its row
- Word contained by docs in its column
- The matrix is not random
  - Systematic similarities between row or column pairs
- Duality of doc and word similarity



Any matrix M can be factorized



- Written as $M[i,j] = \sum_k U[i,k]\, \Sigma[k,k]\, V[j,k]$
  - Or compactly as $M = U\Sigma V^\top$
- $\Sigma = \mathrm{diag}(\{\sigma_i\})$ with $\sigma_1 \geq \sigma_2 \geq \cdots \geq 0$
- Columns of U are unit vectors, perpendicular to each other; so are columns of V
- Truncated rows of U and V provide “embeddings” of docs and terms

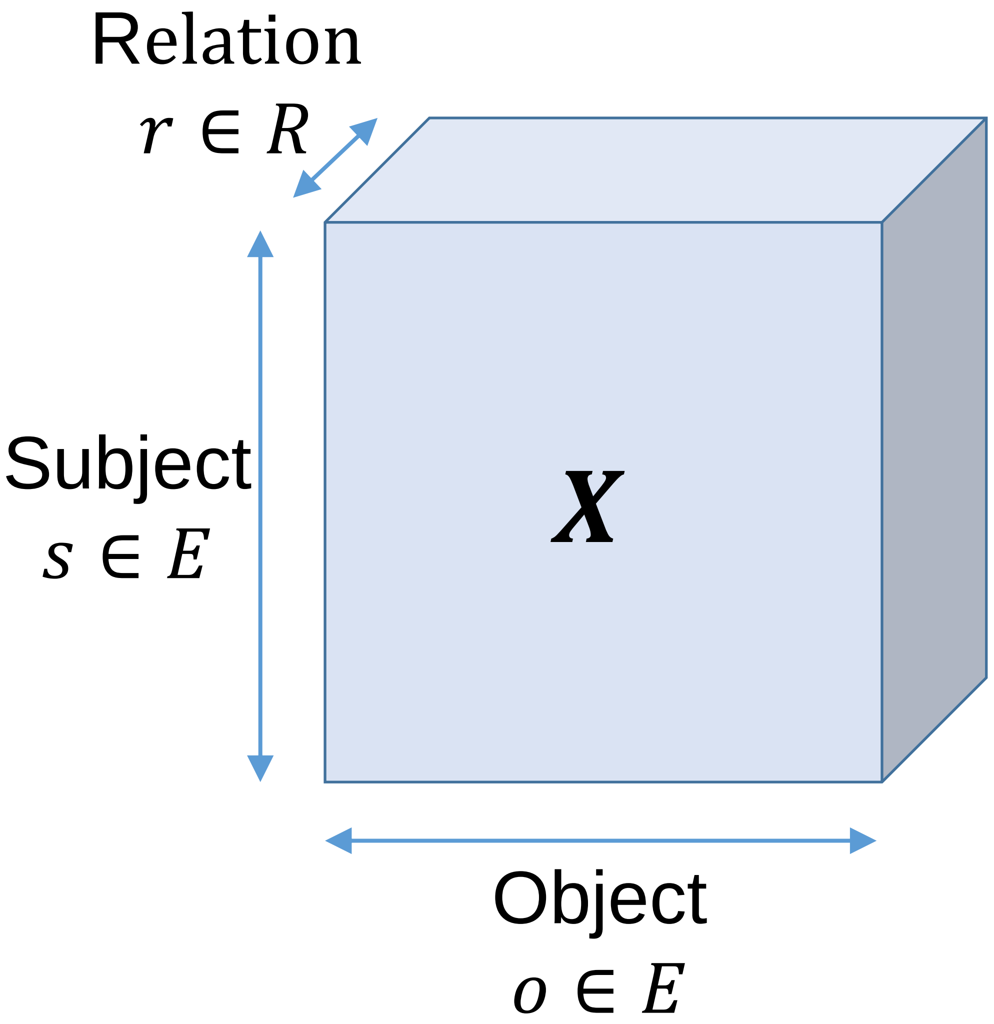
Similarity between word vectors

- Suppose M is the doc (rows) x terms (cols) matrix
- What is $(M^\top M)[i,j]$?
  - Number of docs that contain both words i and j
  - $V[i,*] \cdot V[j,*]$ gives a similarity score between words i and j

	d_1	d_2	d_3	d_4	d_5
ant					
bee					
cat					
dog					
car					
sedan					

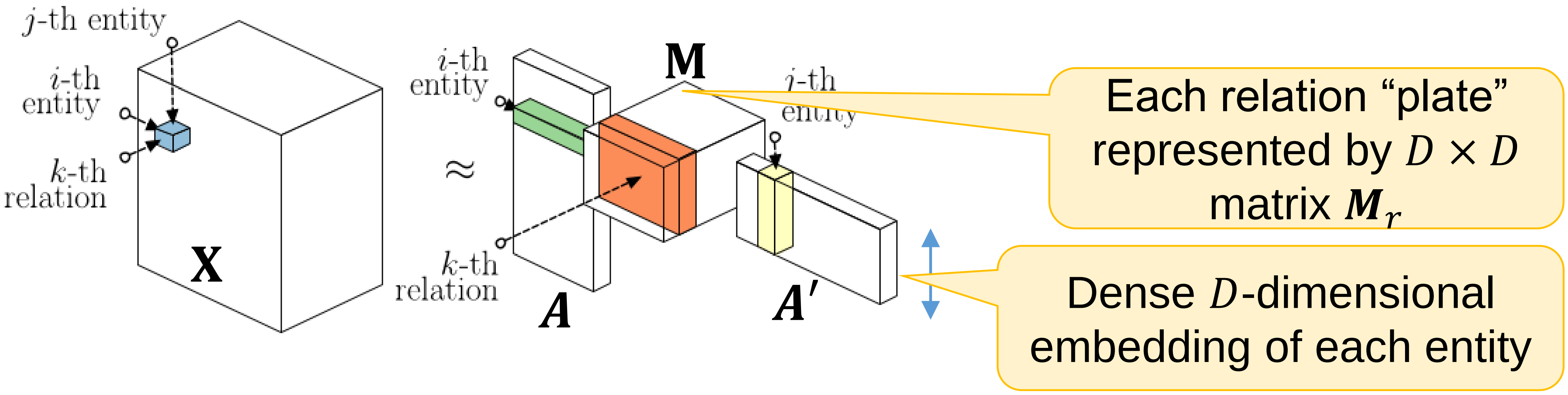
	ant	bee	cat	dog	car	sedan
d_1						
d_2						
d_3						
d_4						
d_5						

2-axis doc-term matrix to 3-axis KG tensor



- Three axes one each for subject, relation, object
- $E \times R \times E$ bits
- Can represent any KG
- $\boldsymbol{X}$ is extremely sparse
- Suspected low rank (rows/cols/"fibers" approx. linearly dependent)

Factoring the KG tensor



- $\boldsymbol{A}$ is the entity embedding matrix, used for both subject and object positions (can also separate roles)
- Objective: $\min_{\boldsymbol{A},\mathbf{M}} \|\mathbf{X} - \mathbf{M} \times_1 \boldsymbol{A} \times_2 \boldsymbol{A}\|_F^2$
- Regularizers: $+ \blacksquare \|\boldsymbol{A}\|_F^2 + \blacksquare \|\mathbf{M}\|_F^2$

Factoring, continued

- Alternating substeps
  - Fix $\mathbf{M}$ and improve $\boldsymbol{A}$ via gradient descent
  - Fix $\boldsymbol{A}$ and improve $\mathbf{M}$ via gradient descent
- Local optimum, slower than matrix factorization
- Two popular (further) approximations
  - Convert to matrix factorization and apply matrix factorization
  - Restrict the relation tensor $\mathbf{M}$ to simpler forms

Reshape to matrix factorization

- Each row corresponds to a (s, o) pair
- Each column corresponds to a relation r
- Matrix SVD gives embedding for each entity pair

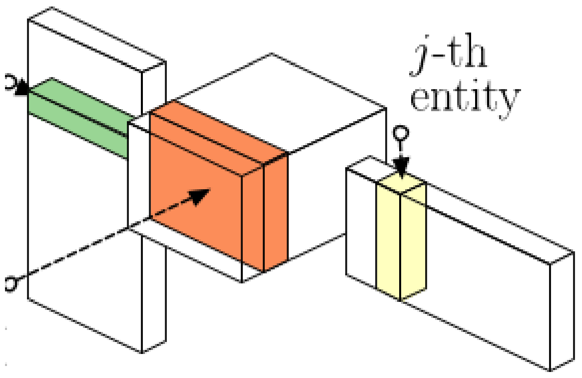
(subject, object)	**Born-in**	**Capital-of**	**Citizen-of**
(Obama, Honolulu)	✓		
(Honolulu, Hawaii)		✓	
(Bangkok, Thailand)		✓	
(Einstein, Ulm)	✓		
(Einstein, USA)			✓
(Obama, USA)			✓

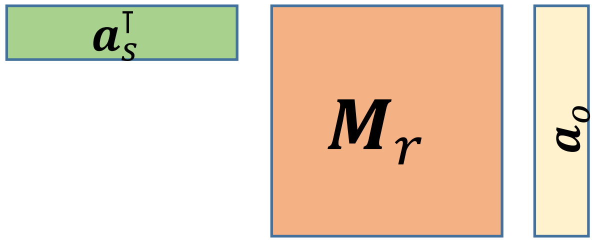
Simplify relation tensor $\mathbf{M}$

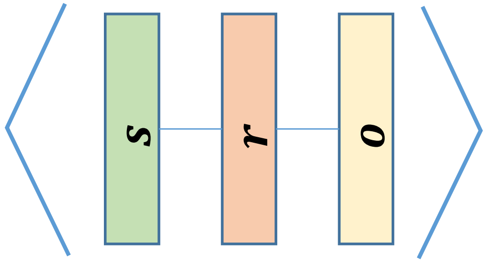
- $\mathbf{X}[s,r,o] \approx \boldsymbol{a}_s^\intercal \mathbf{M}_r \boldsymbol{a}_o$
- If $\mathbf{M}_r$ is constrained to be diagonal, then we can simplify to

$$\mathbf{X}[s,r,o] \approx \sum_d \boldsymbol{s}_d \boldsymbol{r}_d \boldsymbol{o}_d$$

- Shorthand: $\langle \boldsymbol{s}, \boldsymbol{r}, \boldsymbol{o} \rangle$
- Called DistMult
- Faster to train than general $\mathbf{M}$
- Clearly, cannot handle asymmetry and anti-symmetry







Has DistMult been stress-tested?

"The good performances of DistMult on notoriously nonsymmetric datasets such as FB15K or WN18 are surprising. First, let us note that for the symmetricity to become an issue, one would have to evaluate queries $(s, r, ?)$ while also trying to answer correctly to queries of the form $(?, r, s)$ for a non-symmetric predicate r. In FB15K, those type of problematic queries make up only 4% of the test set and thus, have a small impact."

—Lacroix et al.

A SimplE extension

- To address asymmetry or anti-symmetry, each entity e has *two* associated vectors $\overrightarrow{\mathrm{sub}}(e)$ and $\overrightarrow{\mathrm{obj}}(e)$, depending on its role in a fact
- For each relation r, introduce inverse relation r^{-1} unless explicitly provided
- Corresponding vectors $\overrightarrow{\mathrm{rel}}(r)$ and $\overrightarrow{\mathrm{rel}}(r^{-1})$
- Define the score as $\frac{1}{2}\left(\left\langle\overrightarrow{\mathrm{sub}}(s), \overrightarrow{\mathrm{rel}}(r), \overrightarrow{\mathrm{obj}}(o)\right\rangle + \left\langle\overrightarrow{\mathrm{sub}}(o), \overrightarrow{\mathrm{rel}}(r^{-1}), \overrightarrow{\mathrm{obj}}(s)\right\rangle\right)$
- Can fully express any KG with a(nti)symmetric rels

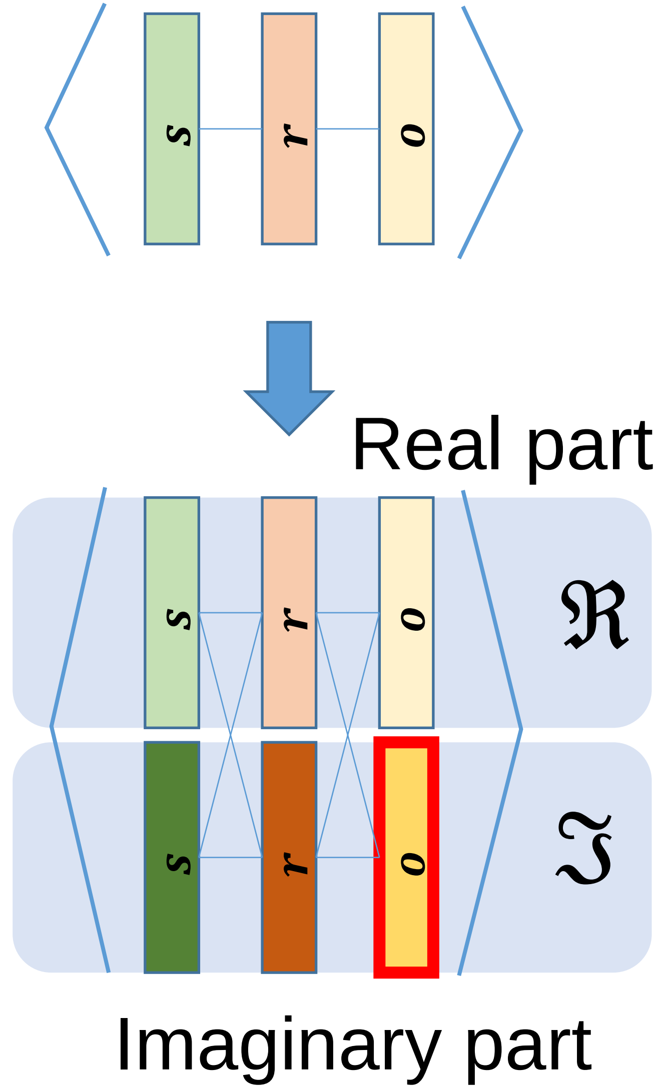
ComplEx extension of DistMult

- Change embeddings from $\boldsymbol{s}, \boldsymbol{r}, \boldsymbol{o} \in \mathbb{R}^D$ to $\boldsymbol{s}, \boldsymbol{r}, \boldsymbol{o} \in \mathbb{C}^D$

Real part of

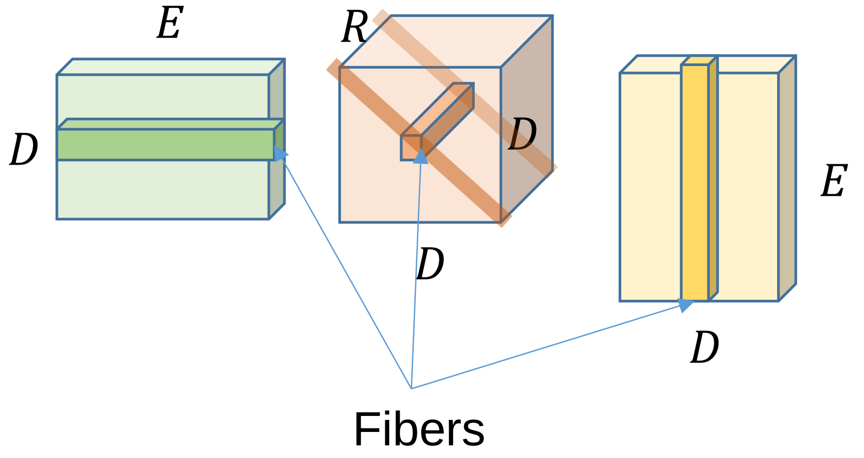
- Change f from $\langle \boldsymbol{s}, \boldsymbol{r}, \boldsymbol{o} \rangle$ to $\Re\langle \boldsymbol{s}, \boldsymbol{r}, \boldsymbol{o}^\star \rangle$
- Here $\boldsymbol{o}^\star$ is the complex conjugate of $\boldsymbol{o}$
  - $(a + jb)^\star = a - jb$
- If a relation is symmetric, set $\Im(\boldsymbol{r}) = \boldsymbol{0}$, reducing to DistMult
- If a relation is anti-symmetric, set $\Re(\boldsymbol{r}) = \boldsymbol{0}$: ensures

$$\Re\langle \boldsymbol{s}, \boldsymbol{r}, \boldsymbol{o}^\star \rangle = -\Re\langle \boldsymbol{o}, \boldsymbol{r}, \boldsymbol{s}^\star \rangle$$



ComplEx, other notation

- Can also write as
  $\mathbf{X} \approx \Re \sum_d \boldsymbol{a}_{\cdot,d} \otimes \mathbf{M}_{\cdot,d,d} \otimes \boldsymbol{a}^{\star}_{\cdot,d}$
- $\boldsymbol{a}_{\cdot,d} \in \mathbb{C}^E, \mathbf{M}_{\cdot,d,d} \in \mathbb{C}^R$
- Thus $\mathbf{X} \in \mathbb{R}^{E\times R\times E}$
- Each rhs term is a rank-1 “fiber”
- Expresses $\mathbf{X}$ as sum of D rank-1 tensors
- Thus, this is a rank $\leq D$ decomposition of $\mathbf{X}$



Further ComplEx-ity

- Like SimplE, use $\overrightarrow{\mathrm{sub}}(e)$ and $\overrightarrow{\mathrm{obj}}(e)$ for entities, $\overrightarrow{\mathrm{rel}}(r)$ and $\overrightarrow{\mathrm{rel}}(r^{-1})$ for relations and inverses
- Except these are complex vectors
- Regularize a nuclear p-norm
- We call this version Complex-N3

Sample KG completion [performance]

- ComplEx-V2 uses large/full negative sampling
- Almost as good as Complex-N3
- Both generally better than others

	FB15k-237			WN18RR		
Method	**MRR**	**HITS@1**	**HITS@10**	**MRR**	**HITS@1**	**HITS@10**
SimplE	0.23	0.15	0.40	0.42	0.40	0.46
SIMPLE-V2	0.34	0.25	0.53	0.46	0.43	0.52
Complex	0.31	0.22	0.51	0.42	0.40	0.47
COMPLEX-V2	0.35	0.26	0.54	0.47	**0.46**	0.53
Complex-N3	**0.37**	**0.27**	**0.56**	**0.49**	0.44	**0.58**

	FB15k			WN18			YAGO3-10		
Method	**MRR**	**HITS@1**	**HITS@10**	**MRR**	**HITS@1**	**HITS@10**	**MRR**	**HITS@1**	**HITS@10**
SimplE	0.73	0.65	0.86	0.95	0.94	0.95	—-	—-	—-
SIMPLE-V2	0.85	0.82	**0.91**	**0.95**	**0.96**	0.95	0.56	0.49	0.69
Complex	0.81	0.75	**0.91**	0.94	0.93	0.95	0.51	0.40	0.63
COMPLEX-V2	**0.86**	**0.83**	**0.91**	**0.95**	0.95	**0.96**	**0.58**	**0.50**	**0.71**
Complex-N3	**0.86**	**0.83**	**0.91**	**0.95**	0.94	**0.96**	**0.58**	**0.50**	**0.71**

WN18→WN18RR and FB15k→FB15k-237

- WN18 KG was extracted from WordNet
  - 18 relations and 40943 entities
  - Many test triples (s, r, o) had corresponding (o, r^{-1}, s) in train fold
  - Removing them gave 93003 triples over 40943 entities and 11 relation types
  - Harder benchmark, less leakage
- FB15k to FB15k-237: similar story
  - 592213 triples, 14951 entities, 1345 relations $\longrightarrow$ 310079 triples, 14505 entities, 237 relations

Summary so far

- Definition of KG: entity, relation, fact triple
- KG completion problem
- Triple scoring function
- Vector translation and rotation models
- Matrix and tensor factorization models