Knowledge and Retrieval

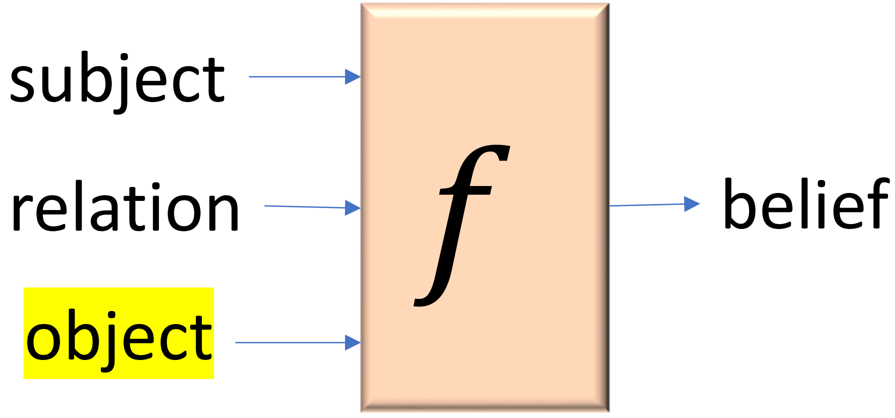
Knowledge Graph Completion and Evaluation

KG completion

- Incomplete KG provided
- Learning system fits embedding representations of entities and relations
  - Based on KG topology alone
  - Supplemented by text aliases
- Apply to KG and infer missing fact triples

Knowledge graph completion (KGC)

- Represent each entity *e* and relation *r* as continuous (geometric) artifacts $\vec{e}, \vec{r}$
  - Point, vector, displacement, projection, rotation, …
- Design a scoring function f for the belief in a (subject, relation, object) triple
- Train entity and relation reps using known triples
- Infer unknown triples (aka “link prediction”)

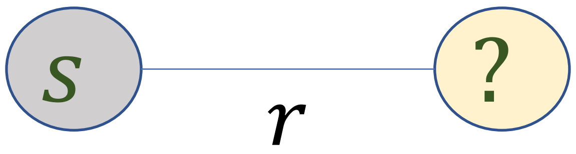


- TransE, HtransE, DistMult, HolE, ComplEx, RotatE, …

Initial notation

- Entity e, entity embedding $\vec{e}$ or $\boldsymbol{e}$
- Entity can be subject s or object o
- With corresponding embeddings $\boldsymbol{s}, \boldsymbol{o}$
- Similarly, relation r has representation $\boldsymbol{r}$
- These embeddings may be vectors, matrices or tensors with real or complex elements
- $f(s, r, o) \in \mathbb{R}$ is a raw confidence score
- $(s, r, o) \in \mathrm{KG}$ is a "positive" triple/fact, $(s', r', o') \notin \mathrm{KG}$ is a "negative" triple/fact

From score to probability



- Softmax distribution

$$\Pr(o|s,r) = \frac{\exp(f(s,r,o))}{\sum_{o'} \exp(f(s,r,o'))}$$

- Similarly, $\Pr(s|r,o) = \frac{\exp(f(s,r,o))}{\sum_{s'} \exp(f(s',r,o))}$
- (Somehow, sampling r is relatively rare)
- For $(s,r,o) \in \mathrm{KG}$, want large $\Pr(o|s,r) \approx 1$ and large $\Pr(s|r,o) \approx 1$

Probability to loss function

- A common objective to maximize is

$$\sum_{(s,r,o)\in \mathrm{KG}} \log \Pr(s|r,o) + \sum_{(s,r,o)\in \mathrm{KG}} \log \Pr(o|s,r)$$

- Loss to minimize is negative of above
- Indirectly encourages small $f(s',r',o')$ for negative triples $\langle s',r',o'\rangle$
- Probabilities involve sums over *all* entities in KG (expensive)

Negative sampling

- Take positive fact (s, r, o) and replace s with a randomly sampled entity s'
- Resulting fact (s', r, o) unlikely to be positive
- Similarly sample o' to get (s, r, o')
- "Local closed world" assumption
  - Not really valid, given KG is (very) incomplete
- Helps replace full sum in denominator with sampled estimates

Uniform negative sampling

- Let $A = \sum_{o \in E} \exp(f(s, r, o))$ for fixed s, r
- Sample K out of E entities uniformly at random
- $\mathbb{E}[\sum_{o \in K} \exp(f(s, r, o))] = \frac{K}{E} A$
  - High variance, need fairly large K (thousands)
  - Under-sampling known to degrade accuracy
- In $\frac{\exp(f(s,r,o))}{\frac{E}{K} \sum_{o' \in K} \exp(f(s,r,o'))}$, must include o into denom by force, may further bias estimate

Discriminative training (à la SVM)

- For each positive fact (s, r, o) and each negative fact (s', r', o'), we want

$$f(s, r, o) \geq \text{margin} + f(s', r', o')$$

- Turn into hinge/ReLU loss

$$\max\{0, \text{margin} + f(s'_k, r, o'_k) - f(s, r, o)\}$$

- If there are E entities and R relations
  - Number of possible facts is E^2R
  - Of which a small fraction is positive
- Infeasible number of constraints/loss terms
- Unnecessary for predicting one missing field in a fact triple, e.g., $(s, r, ?)$ or $(?, r, o)$

Will use E, $|E|$ and R, $|R|$ interchangeably

Sampling for discriminative training

- For each positive fact (s, r, o) in batch, sample K (presumed) negative facts (s'_k, r, o'_k)
- Accumulate to batch loss the average pair loss $\frac{1}{K}\sum_k \max\{0, \text{margin} + f(s'_k, r, o'_k) - f(s, r, o)\}$
- If true f is $\gg$ false f, loss term = 0
- Pairwise hinge/ReLU loss

Score polarity

- Thus far we have assumed
  - Large score $f(s, r, o)$ makes (s, r, o) more likely
  - Small score $f(s, r, o)$ makes (s, r, o) less likely
- Some models work with the opposite polarity
- Probability and losses can usually be adjusted without much trouble

Testing embedding and score quality

- KG completion task
  - KG sampled into train, dev, test folds
  - Given test queries $(s, r, ?)$ and $(?, r, o)$, trained system must provide ranked list for blanks
  - Mean (reciprocal) rank, hits@K
- WN18: 41k entities (WordNet synsets), 18 relation types (hypernymy, synonymy, . . .), folds 141k/5k/5k
- FB15k: 15k Freebase entities, 1345 relation types, folds 483k/50k/59k
- WN18RR, FB15k-237, YAGO, …
- Other tasks (alignment, analogy, QA, …)

Filtered evaluation

- Query $\langle s, r, ? \rangle$ with ranked system response list $(o_1, o_2, o_3, o_4, o_5, o_6, o_7, o_8, \ldots)$
- Suppose o_6, o_8 are test fold gold answers
- MRR = 1/6, MAP = ½(1/6+2/8)
- Suppose o_2 was a train fold gold answer
- Then MRR = 1/5, MAP = ½(1/5+2/7) is fairer
- A simple but important eval convention