✅

TransE: Traslating Embeddings for Modeling Multi-relational Data

Domain	Graph
tag	Graph learning, Low-dim vector space, Vector embedding
Conference / Journal	NeurIPs
Publish year	2013
정리 날짜	@2024년 1월 16일
AI summary	TransE is a vector embedding model for knowledge graphs that focuses on energy efficiency and minimal parameters. It represents entities and relationships in a low-dimensional vector space and uses a distance-based energy framework for training. The model shows good performance and expressivity, outperforming previous methods in relational learning. However, it struggles with representing 3-way dependencies and interactions.
AI key info	TransE, Vector embedding, low-dim vector space, Knowledge graph, Graph learning, NeurIPs, 2013

Summary

knowledge graph를 vector embedding으로 구현

Energy efficiency에 초점 맞춰 적은 파라미터로 구현

Background

Multi-relational data

- Heterogeneous relationships
  - 다양한 특성의 edge들로 구성
- different type of relation, entity로 구성되기 때문에, 모델링하는데 더 generic한 approach 필요

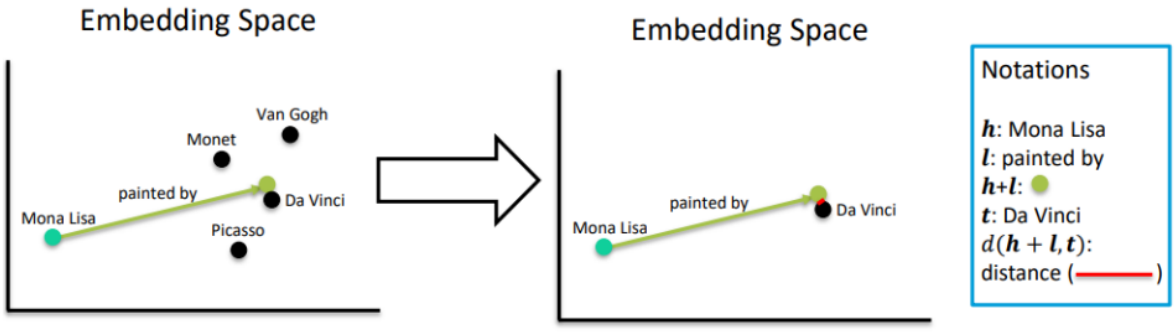
Knowledge graph

- Entity 간의 relation을 나타낸 그래프

(h, l, t) -> head, label, tail

Translation

- 어떤 entity를 다른 entity로 매핑하는 relationship vector
- Embedding space에 translation을 나타냄



Motivation

기존 knowledge graph methods

- latent attribute로부터 relational learning하는 framework

- 문제점
  - 모델의 expressivity, universality 중시
    - complexity 높음 → high cost, 이해 어려움
  - ovefitting: proper regularzation 어려움
  - underfitting: non-convex optimization problem

간단하고 좋은 성능의 모델 만들어보자

- multi-relational data를 low-dimensional vector space에 표현
  - triplet당 low-dimensional vector가 하나인 embedding을 만들어보자
    - parameter set 크기 줄일 수 있음
    - t가 head + translation vector로 표현됨
      - 계층구조 relationship
      - 1 to 1 relationship : different type of relationship
- knowledge base model를 적은 데이터로 효율적으로 완성하는 tool 제공
- 이렇게 하는 기존 energy-based work들 성능 낫밷이더라

Methodology

vector embedding space

- t가 h+l로 표현되어, h+l과 가장 가까운 neighbor = t

$$\boldsymbol{h} + \boldsymbol{\ell} \approx \boldsymbol{t} \text{ when } (h, \ell, t) \text{ holds}$$

- energy-based framework에 기반
  - distance $d(h + l, t)$: energy of a triplet: dissimilarity

Corrupted triplet

- training triplet에서 head나 tail을 랜덤하게 바꿈

$$S'_{(h,\ell,t)} = \left\{(h',\ell,t)|h' \in E\right\} \cup \left\{(h,\ell,t')|t' \in E\right\}$$

- 기존 triplet의 distance와 비교하여 loss function 구하기 위함

Loss function

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

$[x]_+$ denotes the positive part of x, $\gamma > 0$ is a margin hyperparameter

- corrupted triplet보다 energy 적은 training triplet을 학습
- SGD로 학습

Training algorithm

Algorithm 1 Learning TransE

input Training set $S = \{(h,\ell,t)\}$, entities and rel. sets E and L, margin γ, embeddings dim. k.

initialize $\boldsymbol{\ell} \leftarrow$ uniform$(-\frac{6}{\sqrt{k}}, \frac{6}{\sqrt{k}})$ for each $\ell \in L$
  $\boldsymbol{\ell} \leftarrow \boldsymbol{\ell} / \|\boldsymbol{\ell}\|$ for each $\ell \in L$
  $\mathbf{e} \leftarrow$ uniform$(-\frac{6}{\sqrt{k}}, \frac{6}{\sqrt{k}})$ for each entity $e \in E$
loop
  $\mathbf{e} \leftarrow \mathbf{e} / \|\mathbf{e}\|$ for each entity $e \in E$
  $S_{batch} \leftarrow$ sample(S, b) // sample a minibatch of size b
  $T_{batch} \leftarrow \emptyset$ // initialize the set of pairs of triplets
  for $(h,\ell,t) \in S_{batch}$ **do**
    $(h',\ell,t') \leftarrow$ sample$(S'_{(h,\ell,t)})$ // sample a corrupted triplet
    $T_{batch} \leftarrow T_{batch} \cup \left\{\left((h,\ell,t),(h',\ell,t')\right)\right\}$
  end for
  Update embeddings w.r.t. $\sum_{\left((h,\ell,t),(h',\ell,t')\right)\in T_{batch}} \nabla\left[\gamma + d(\boldsymbol{h}+\boldsymbol{\ell},\boldsymbol{t}) - d(\boldsymbol{h'}+\boldsymbol{\ell},\boldsymbol{t'})\right]_+$
end loop

- 시작
  - 각 entity와 relationship의 embedding 초기화
- Loop

    - triplet batch를 random sampling하여
    - 각 triplet당 random corrupt를 만들어
    - 이 각 non-corrupt, corrupt간 energy 차이로 학습 진행

Conclusion

Knowledge base를 minimal parameter로 학습

- energy efficient하지만, 좋은 성능
    - Expressivity가 SE에 비해 낫지만 성능은 더 좋음
        - relational propertie에 더 직접적으로 접근하기 때문에
        - Embedding model에서 최적화가 어렵기 때문에
    - 더 expressible하다는 게 더 나은 성능이 아니라 underfitting에 가까움
- Scailable

단점

- 2 way interaction을 표현하기 어려움
    - 1-to-many, many-to-many 같은 3-way dependency가 큰 경우 거의 실패함

Questions