TRƯỜNG ĐẠI HỌC KHOA HỌC TỰ NHIÊN, ĐHQG-HCM

HOMEWORK 04

Học kỳ 3 – Năm học 2022-2023

MÃ LƯU TRỮ *(do phòng KT-ĐBCL ghi)*	

Tên học phần:	**Khai thác Dữ liệu Đồ thị**	Mã HP:	**CSC17103**
Thời gian làm bài:	**07 ngày**	Ngày nộp:	**10/08/2023**

HOMEWORK 04

LINK PREDICTION & GRAPH EMBEDDING

Họ tên sinh viên: Võ Văn Hoàng **MSSV**: 20127028

BÀI TRÌNH BÀY

1. Knowledge Graph:

- A knowledge graph is a structured representation of knowledge that captures relationships between entities in a particular domain. It is a way to organize and connect information to make it more accessible, useful, and meaningful for machines and humans alike. In a knowledge graph, information is represented as nodes and edges connecting these nodes. Here are some key characteristics of a knowledge graph:
  - ✓ Nodes (Entities): Entities are the individual pieces of information or concepts represented in the graph. These can be anything from people, places, organizations, products, to abstract concepts, events, and more.
  - ✓ Edges (Relationships): Edges represent the connections or relationships between entities. These relationships describe how entities are related or linked to each other. For example, "born in," "works for," "is married to," etc.

2. Link Prediction

- Link prediction in knowledge graphs is a technique used to predict missing or potential relationships (links) between entities in the graph. Since knowledge graphs often represent a subset of the real-world knowledge and relationships, they are rarely complete. The process of link prediction involves using the existing information in the knowledge graph to predict new edges (relationships) between nodes (entities) that are not explicitly present but are likely to exist. This can be valuable for various reasons, such as:

(Bài làm gồm 6 trang)

- ✓ Knowledge Completion: Link prediction helps in filling the gaps in the knowledge graph, thereby making it more comprehensive and informative.
- ✓ Recommendation Systems: Link prediction can help suggest potential connections between users and items to make personalized recommendations.
- ✓ Identifying Missing Data: Link prediction can be used to identify entities with potential missing attributes or properties based on their relationships with other.

3. Graph Embedding

- Graph embedding, also known as network embedding or graph representation learning, is a technique that aims to learn low-dimensional vector representations (embeddings) of nodes, edges, or subgraphs in a graph. The goal of graph embedding is to transform the graph's complex structure into a continuous vector space while preserving essential information and capturing meaningful relationships between nodes. Graph embeddings have become popular because they enable the application of machine learning and data mining techniques on graph-structured data, allowing algorithms designed for continuous vector spaces to be used for tasks on graphs. By representing nodes as vectors in a lowerdimensional space, graph embeddings facilitate efficient and scalable processing of graph.

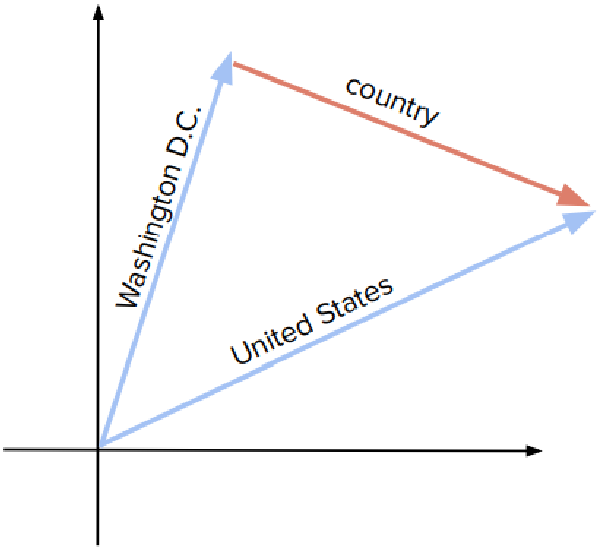
4. TransE

a. Learn and present the TransE model the way you understand it.

- TransE is a translation-based approach for embedding knowledge graphs. In TransE, relationships are represented as translations in the embedding space: if (h, r, t) holds, then the embedding of the tail entity t should be close to the embedding of the head entity h plus some vector that depends on the relationship r. The fundamental idea underlying TransE is to represent entities and relations as points in a vector space, with the distance between an entity's representation and a relation's head being equal to that distance, with the score function formula: $d(h + r, t)$.
- TransE is trained using a loss function with a margin with the formula: $\sum_{(h,r,t)\in S}\sum_{(h',r,t')\notin S}[\gamma + d(h + r, t) - d(h' + r, t')]$. With γ is a margin hyper-parameter, $d(h + r, t)$ is the score of the true triple while $d(h' + r, t')$ is the score of the true triple.
- In other words, if the entity is the head of the relation, the model is trained to minimize the distance between the representations of the entity and the relation, and

if the entity is not the head of the connection, the model is taught to maximize the distance between the representations of the entity and the relation.

- It has been demonstrated that TransE performs well on a range of knowledge graph tasks, including link prediction, entity categorization, and question answering.
- Here are some of TransE's salient characteristics: it has been demonstrated to work well for a range of knowledge graph activities and is a straightforward, effective model that is also basic and easy to understand.
- The following are some TransE's drawbacks: it can be sensitive to the selection of hyperparameters; it is not as expressive as some other knowledge graph embedding models.
- Examples:



Example of how TransE represents and models the interactions between entities and relationships in vector space.

b. Suppose we consider a simpler loss function for the TransE model as follows:

$$\mathcal{L}_{\mathbf{simple}} = \sum_{(\mathbf{h},\mathbf{r},\mathbf{t})\in \mathbf{S}} \mathbf{d}(\mathbf{h}+\mathbf{r},\mathbf{t})$$

Whether the entity and relational embeddings become better than the original version after minimizing the loss function to zero. Give an example to support that statement.

- Applying the loss function $\mathcal{L}_{\text{simple}}$ is not expected to provide embeddings that are superior to those produced by the original TransE formulation. $\mathcal{L}_{\text{simple}}$'s primary flaw is that it cannot tell the difference between right and erroneous triplets when training.

Regardless of whether a triplet is a legitimate knowledge graph fact or not, it only aims to reduce the distance between embeddings for all triplets in the training set.As a result, the model can end up learning embeddings that incorrectly reflect semantic relations.

- Take into account these true facts: Ha Noi is a city in VietNam, Ottawa is a city in Japan.
- We have false information: (Ha Noi is located in Japan). Even if it is untrue, minimizing $\mathcal{L}_{\text{simple}}$ would cause the embeddings of (Ottawa, located, Viet Nam) to be near together. On the other hand, the original loss function of TransE takes into account the distinction between right and wrong triplets. According to the semantic meaning of the items and relations, the model is guided by this information.In conclusion, $\mathcal{L}_{\text{simple}}$ is overly simplistic because, unlike TransE's original function, which takes into account both right and false facts, it may lead to the model learning representations that do not accurately reflect the real relations in the knowledge graph.

c. **Suppose we consider a simpler loss function for the TransE model as follows:**

$$\mathcal{L}_{\text{no margin}} = \sum_{(\mathbf{h},\mathbf{r},\mathbf{t})\in S} \sum_{(\mathbf{h}',\mathbf{r},\mathbf{t}')\notin S} \max\left[\mathbf{0}, \mathbf{d}(\mathbf{h} + \mathbf{r}, \mathbf{t}) - \mathbf{d}(\mathbf{h}' + \mathbf{r}, \mathbf{t}')\right]$$

- **Whether the entity and relational embeddings become better than the original version after minimizing the loss function to zero. Give an example to support that statement.**
- Comparing entity and relational embeddings using the suggested $\mathcal{L}_{\text{no margin}}$ loss function to those utilizing the original TransE model does not guarantee superiority. Because the margin term has been eliminated, $\mathcal{L}_{\text{no margin}}$ differs significantly from the original TransE loss function.
- In the absence of a margin, the loss function merely penalizes wrong triplet predictions that are further apart from the true triplet; it does not encourage the true triplet to group together and maintain a margin of separation from false triplet predictions.
- As a result, rather of tightly grouping the accurate triplets, embeddings may be created that minimize loss by dispersing all predictions.
- Take X, Y, Z, and r, for instance, where (X, r, Y) is accurate but (X, r, Z) is erroneous. Using only $\mathcal{L}_{\text{no margin}}$, the model might discover embeddings in which:
  - ✓ d(X+R, Y) = 1
  - ✓ d(X+R, Z) = 2

- Although the loss is reduced, it is difficult to distinguish between correct and erroneous triplets. With the margin in the original TransE, the model would be motivated to discover:
  - ✓ d(X+R, Y) = 1
  - ✓ d(X+R, Z) > 1 + γ
- In conclusion, compared to the original TransE formulation, merely decreasing $\mathcal{L}_{\text{no margin}}$ to zero does not provide superior embeddings. When learning superior representations, the margin is crucial.

d. Do entities embedding in TransE need to be normalized to the same length? Why ?

- Entities embedding in TransE must be normalized to the same length in order to prevent being pushed toward having big embeddings, which would make it impossible for the model to discriminate between positive and negative triplets.

e. Give an example of a simple graph for which no perfect embedding exists, i.e., no embedding perfectly satisfies h + r = t for all (h, r, t) ∈ S and h' + r ≠ t' for (h', r, t') ∉ S, for any choice of entity embeddings (e for e ∈ E) and relationship embeddings (r for r ∈ R). Explain why this graph has no perfect embedding in this system.

- An example of a simple graph for which no perfect embedding exists:
  G = (V, E) where V = {h, r, t} (set of entities in the graph), E = {(h, r, t)} (set of edges in the graph).
- A single triple (h, r, t) may be seen in this graph. According to this triple, h and t are connected by r. We would obtain h + r = t for any ideal embedding of this network. The fact that h, r, and t are all separate things prevents this from being achievable. There isn't a perfect embedding of this graph as a result. For a more thorough explanation of why this graph lacks a perfect embedding, see below:

✓ The embedding h + r needs to match t. But as h, r, and t are all different entities, so must their embeddings. Consequently, it is not feasible for h + r to equal t.

✓ This would suggest that h and t are the same thing if h + r were equal to t. However, this is not conceivable because h and t are unmistakably separate entities.

Therefore, the graph G cannot be perfectly embedded.

- This illustration demonstrates that it is not always easy to locate the ideal embedding for a straightforward graph. No embedding will ever be able to meet all of the restrictions if the graph is intrinsically contradictory, which can happen in some situations.

5. Tài liệu tham khảo:

[1]: Translating Embeddings for Modeling Multi-relational Data: *https://proceedings.neurips.cc/paper_files/paper/2013/file/1cecc7a77928ca8133fa24680a88d2f9-Paper.pdf*

[2]: Evaluation of the impact of controlled language on neural machine translation compared to other MT architectures:

https://www.researchgate.net/publication/333518477_Evaluation_of_the_impact_of_controlled_language_on_neural_machine_translation_compared_to_other_MT_architectures

[3]: Knowledge Graph Embeddings and Explainable AI:

https://www.researchgate.net/publication/341068807_Knowledge_Graph_Embeddings_and_Explainable_AI/link/5ecd3859299bf1c67d1d8837/download

Hết