Project3
Knowledge Graph-Based Recommender System

Outline

- Introduction
- Task Description
- Reference

Outline

- Introduction
- Task Description
- Reference

Recommender System

- Recommender systems infer the preferences of users and recommend relevant information to users.

- Recommender systems recommend items to users according to a score function.
  - Score function predicts how likely the user would be interested in the item.
  - Getting from historical user-item interaction records or additional side information.

Recommender System

- Based on what can we recommend an item (to a user)?
  - The items that the user interacted with.
  - The users who have interacted with similar.
  - The feature of users and items. → Additional side information, like the Knowledge Graph

} Historical user-item interaction records

Recommender System

- Typical methods of Recommender System:
  - Collaborative Filtering method: Recommend according to **historical user-item interaction records**.
  - Content-based method: Recommend according to **item features**.
  - Hybrid method: Recommend according to **both** historical user-item interaction records and user/item features.

KG-based Recommender System

- Knowledge Graph-based (KG-based) Recommender System is a kind of hybrid method:
  - Use historical user-item interaction records.
  - Use a Knowledge Graph as the additional side information.
  - The knowledge graph is always used to describe the items.

KG-based Recommender System

Knowledge Graph (KG) is an abstract representation of knowledge in the real world.

- From the data structure view：
  - KG is a directed heterogeneous graph, entities are the vertices, relationships are the edges
- From the content view:
  - KG is a structured representation of knowledge.
  - It can provide some features about the users/items in the recommender system.

KG-based Recommender System

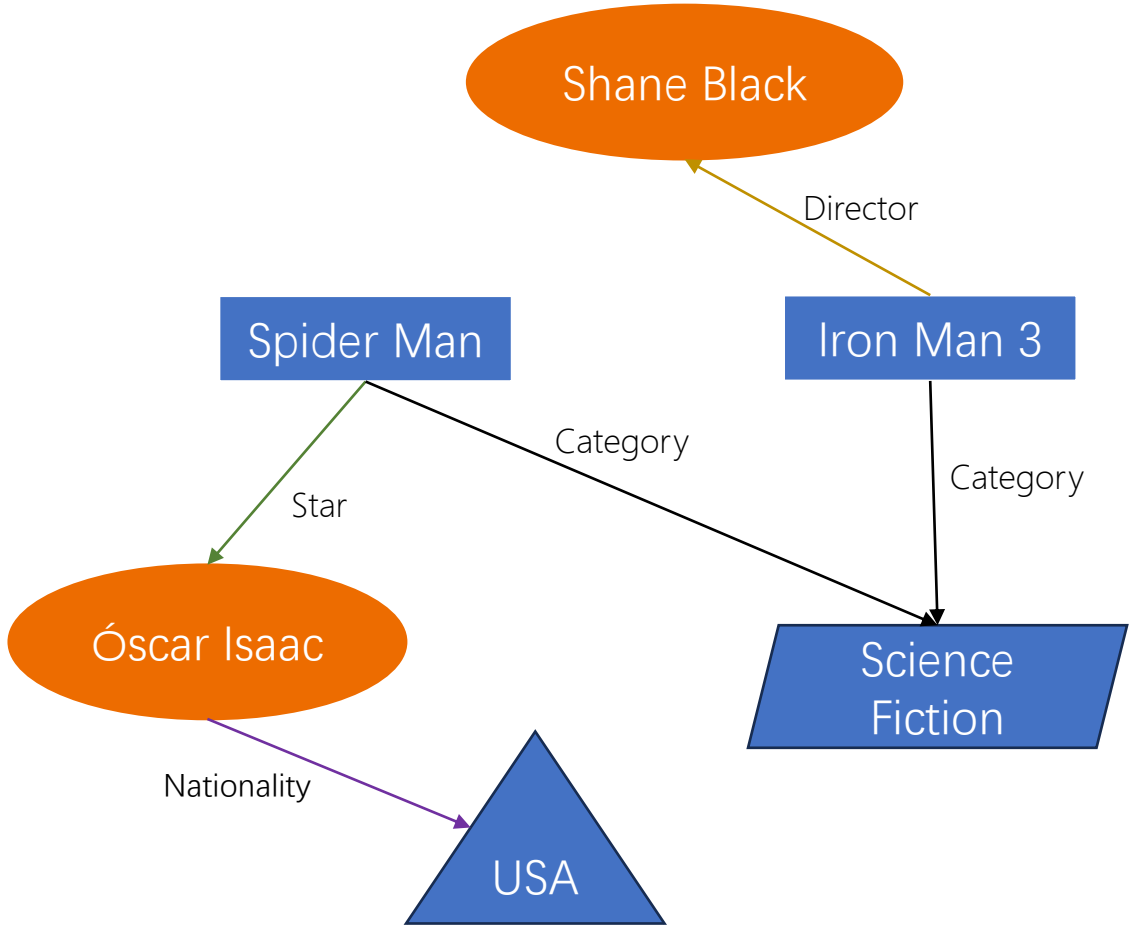
Knowledge Graph

- It records the facts by describing the **relations** between **entities**.
- A relation in the KG is always represented as a relation triple:
  - Head entity: the subject of this relation
  - Relation type: the category of this relation
  - Tail entity: the object of this relation

A simple example of the KG

Head Entity	Relation Type	Tail Entity
Spider Man	Star	Óscar Isaac
Iron Man 3	Category	Science Fiction
Óscar Isaac	Nationality	USA
Spider Man	Star	Daniel Kaluuya
Spider Man	Category	Science Fiction
Iron Man 3	Director	Shane Black

The relations in the KG



KG-based Recommender System

- The key to constructing a KG-based Recommender System:
  - Design a score function $f(u, w)$.
  - The score function is essentially a model trained with data.

How to get the score function from data?

- If we only have historical user-item interaction records:
  - Use Collaborative Filtering method
  - Represent the user/item by correlation or matrix factorization (Lecture PPT)
- If we have additional side information:
  - Use hybrid method
  - Represent the user/item by d-dimension embedding vector (Like the representation in the matrix factorization)

How to get the score function from data?

- If we have additional side information:
  - Represent the user/item by d-dimension embedding vector
  - For the embedding vector of each user/item, **it should contain the information from additional side information**.
  - Calculate the score of the combination of user and item by their embedding vector:
    - Inner product of vectors
    - MLP
  - For each combination of user and item, the score function should satisfy:
    - If the user has interacted with the item (positive) in the historical data, its score should be **high**.
    - Else, its score should be **low**.

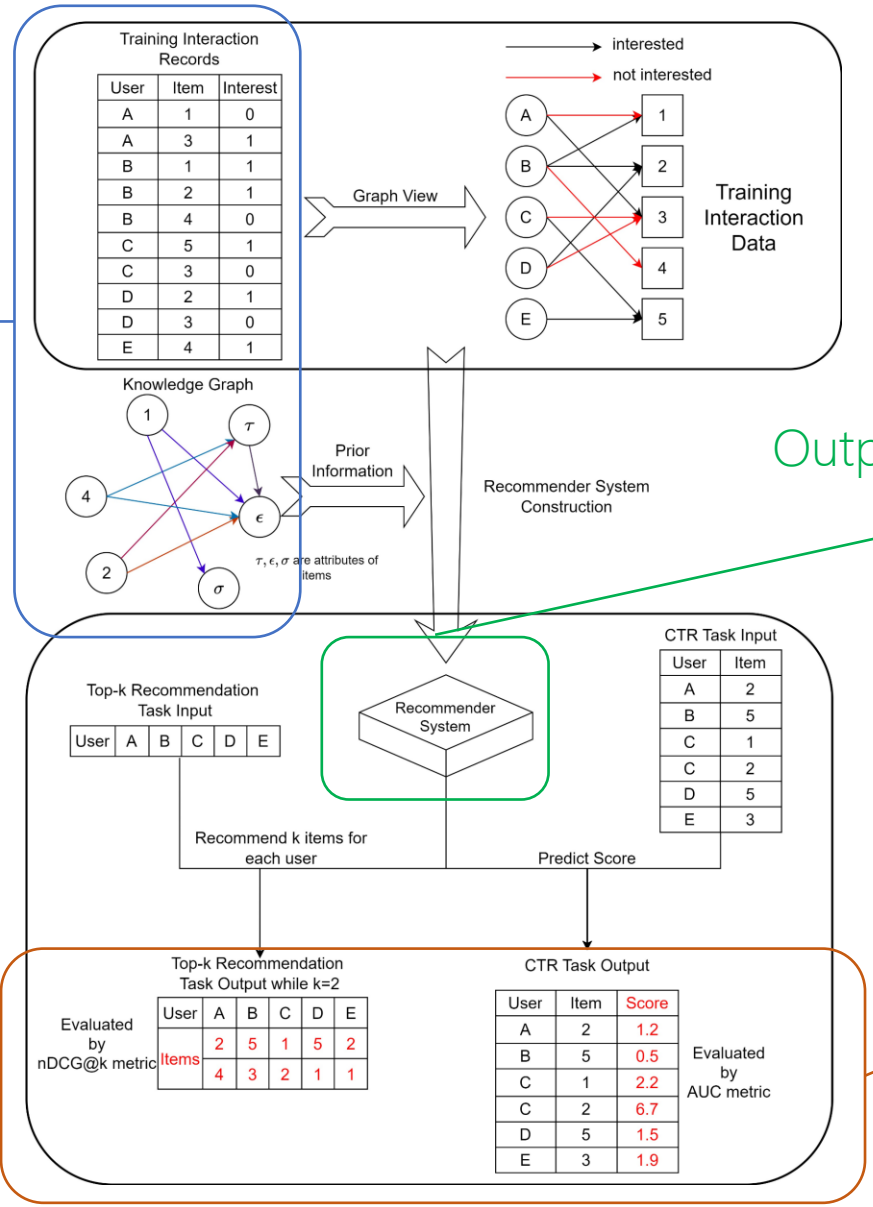
Outline

- Introduction
- Task Description
- Reference

Task Description

- Input:
  - Interaction record set Y_{train}
  - Knowledge graph $\mathcal{G} = (V, E)$
- Output:
  - A recommender system with a score function $f(u, w)$
- Two tasks in this project:
  - Maximum the $\mathbf{AUC}$ metric of your score function $f(u, w)$ on a test dataset Y_{test}
  - Maximum the $\mathbf{nDCG@5}$ metric of your score function $f(u, w)$ on a test dataset Y_{test}

Task Description



Evaluation process of the **AUC** metric and **nDCG@5** metric are shown in the project document.

Files Submission

- URL of the OJ platform is: https://lms.sustech.cloud/
- You need to submit a ZIP file that contains at least one file named `kgrs.py`
  - The `kgrs.py` should be placed in the root directory of your ZIP file.
  - The requirements of `kgrs.py` are shown in the project document.
- You can add other necessary files in your ZIP file.
- The total size of your ZIP file cannot exceed 1MB.

Evaluation Environment

- Max CPU thread num: 8
- Max Memory size: 8GB
- Time Limitation:
  - Initialization of your algorithm: 60 seconds.
  - Training process of your algorithm: 600 seconds.
  - Evaluation process of **AUC** metric: 30 seconds.
  - Evaluation process of **nDCG@5** metric: 60 seconds.

Score Rule

- Total score of this project is 15 points
- For **AUC** and **nDCG@5**, they each have 7.5 points

	Score = 5	Score = 7	Score = 7.5
AUC	≥ 0.6	≥ 0.65	≥ 0.7
nDCG@5	≥ 0.05	≥ 0.075	≥ 0.12

- A report is necessary! You will get 0 score in this project if you haven't submitted the report!

Outline

- Introduction
- Task Description
- Reference

Reference

[1] Q. Guo *et al.*, “A Survey on Knowledge Graph-Based Recommender Systems,” *IEEE Transactions on Knowledge and Data Engineering*, pp. 1–1, 2020, doi: 10/ghxwqg.

[2] Y. Zhang, Q. Ai, X. Chen, and P. Wang, “Learning over Knowledge-Base Embeddings for Recommendation.,” *arXiv*, vol. abs/1803.06540, 2018, [Online]. Available: http://arxiv.org/abs/1803.06540

[3] H. Wang, F. Zhang, M. Zhao, W. Li, X. Xie, and M. Guo, “Multi-Task Feature Learning for Knowledge Graph Enhanced Recommendation.,” in *The World Wide Web Conference*, ACM, 2019, pp. 2000–2010. doi: 10/ghcqwt.