SNA 期末專案

王常在、劉于孺
指導教授：蘇維宗

1. 背景介紹

a. 背景

i. 連結預測（link prediction）在近年廣泛應用於各種網絡平台中，包括社交平台 Facebook 和 Instagram、影片串流平台 Netflix 和愛奇藝，以及電子商務平台蝦皮和淘寶等。許多企業都利用這項技術來提高使用者的黏著度，提升平台的效益。

ii. 透過連結預測，這些平台能夠分析使用者的行為模式、興趣和偏好，預測可能感興趣的內容或商品，並提供個性化的推薦服務，從而提升使用者體驗、提高留存率，並增強平台的競爭力。這些企業不斷進一步發展和改進連結預測技術，以更好地滿足使用者需求並提供更精準的個性化服務。

b. 目的

i. 本次研究將以 kaggle 上 netflix 的資料進行 link prediction 的學習並實際建立預測模型以應用。

c. 資料

i. combined_data_1.txt、combined_data_2.txt、combined_data_3.txt、combined_data_4.txt 四份 txt 文件皆以 → movie_id : user_id, ranking ,date 的形式表示。

ii. movie_titles.csv 包含了 movie_id、發行的年份以及電影名稱

iii. probe.txt 是提供在提交前進行測試的數據集

iv. qualifying.txt 是用在測試提交的預測數據集

d. 資料處理

i. Data Cleaning

這部分主要是將 c.資料 第 i 點 所說明到的 combined_data_1.txt、combined_data_2.txt、combined_data_3.txt、combined_data_4.txt 四份 txt 文件進行合併。接著篩選千分之一的資料筆數約 100000 筆作為本次研究用的資料。合併後 csv 檔的格式為：（以下為範例）

user_id	rating	date	movie_id
6	2	2005/12/4	14358
6	4	2005/1/12	6134
6	4	2005/10/26	5926
6	3	2004/11/10	6797
6	3	2005/12/4	3905

接著解決 c. 資料 第 ii 點提到的 movie_titles.csv 在讀取時會發生 'movie_title' 欄位被切斷導致欄位數量不一致而無法讀取的問題

（如下圖所示）

67	1997	Vampire Journals			
68	2004	Invader Zim			
69	2003	WWE: Armageddon 2003			
70	1999	Tai Chi: The 24 Forms			
71	1995	Maya Lin: A Strong Clear Vision			
72	1974	At Home Among Strangers	A Stranger Among His Own		
73	1954	Davy Crockett: 50th Anniversary Double Feature			
74	1999	6ixtynin9			
75	1997	Grind			
76	1952	I Love Lucy: Season 2			
77	1995	Congo			
78	1996	Jingle All the Way			

推測原因該位作者在製作該 csv 檔時，並未將直接使用逗號將欄位分隔，而未考慮到標題原有的逗號。

修改後的內容如下：(以下為範例)

67	1997	Vampire Journals
68	2004	Invader Zim
69	2003	WWE: Armageddon 2003
70	1999	Tai Chi: The 24 Forms
71	1995	Maya Lin: A Strong Clear Vision
72	1974	At Home Among Strangers A Stranger Among His Own
73	1954	Davy Crockett: 50th Anniversary Double Feature
74	1999	6ixtynin9
75	1997	Grind
76	1952	I Love Lucy: Season 2
77	1995	Congo
78	1996	Jingle All the Way

右側為此步驟之參考網址：Cleaning Netflix Data

2. 相關研究

a. Loading Graphs from CSV
https://pytorch-geometric.readthedocs.io/en/latest/notes/load_csv.html

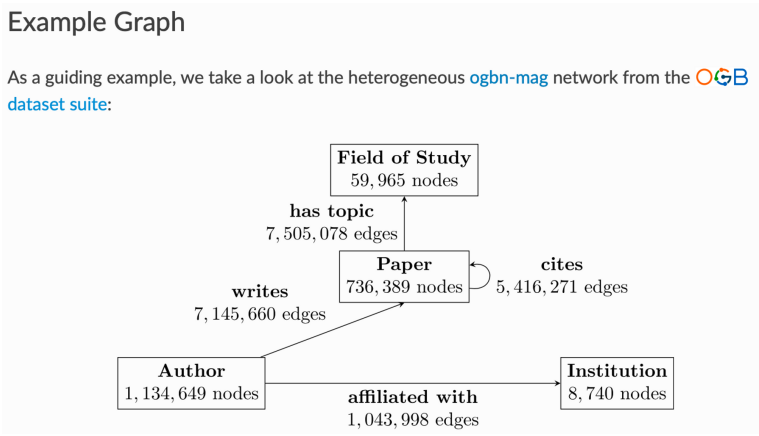
b. Link Prediction on Heterogeneous Graphs
https://medium.com/@pytorch_geometric/link-prediction-on-heterogeneous-graphs-with-pyg-6d5c29677c70
https://colab.research.google.com/drive/1r_FWLSFf9iL0OWeHeD31d_Opt031P1Nq?usp=sharing#scrollTo=Vi25Z7lFPPjc

c. SAGEConv
https://pytorch-geometric.readthedocs.io/en/latest/generated/torch_geometric.nn.conv.SAGEConv.html
https://arxiv.org/pdf/1706.02216.pdf

3. 學習過程

a. 經過資料處理後，檔案為格式 csv 檔，因此要從 csv 檔中讀取 graph。我們參考了 Loading Graphs from CSV 進行操作，建立了兩種類型的 node ，分別為 movie, user，其中我們又將 movie 的 title 進行了 encoding 作為 movie node 的特徵，而 user 則沒有特徵。
b. edge 的部分則只有一個種類，即 rates（ (user)-[rates]->(movie) ）。
c. 因為本次專案之資料集會生成兩個類別節點，分別是使用者和電影，以及一種類型的邊來表示使用者如何評價電影的關係。因此使用了 Pytorch Geometric 提供的

torch_geometric.data.HeteroData 這個模組建構 Heterogeneous Graph。（以下為範例圖）



d. Heterogeneous Graph Data 如下：

```
HeteroData(
  user={ node_id=[524] },
  movie={
    node_id=[17770],
    x=[17770, 384]
  },
  (user, rates, movie)={ edge_index=[2, 100154] },
  (movie, rev_rates, user)={ edge_index=[2, 100154] }
)
```

包含 524 個 user node、17770 個 movie node、100154 個 edge。而最後一行的 edge 是 (user, rates, movie) 的反向關係，目的是為了確保 GNN 可以雙向的傳遞訊息（message passing in both direction）

e. 我們將 edge set 拆分成 80% training data，10% validation data，以及 10% testing data。training data 中只使用 70% 進行訊息傳遞（message passing）。在 validation data 中生成正負樣本比例為 2:1 的負樣本（正樣本表示有連結，副樣本反之）。

```
Training data:
==============
HeteroData(
  user={ node_id=[524] },
  movie={
    node_id=[17770],
    x=[17770, 384]
  },
  (user, rates, movie)={
    edge_index=[2, 56087],
    edge_label=[24037],
    edge_label_index=[2, 24037]
  },
  (movie, rev_rates, user)={ edge_index=[2, 56087] }
)

Validation data:
================
HeteroData(
  user={ node_id=[524] },
  movie={
    node_id=[17770],
    x=[17770, 384]
  },
  (user, rates, movie)={
    edge_index=[2, 80124],
    edge_label=[30045],
    edge_label_index=[2, 30045]
  },
  (movie, rev_rates, user)={ edge_index=[2, 80124] }
)
```

f. 用 *torch_geometric.loader.LinkNeighborLoader* 讀取 train data，用於載入 graph data 的 neighbor information。它用於處理大規模 graph data，在訓練過程中動態生成每個 node 的 sub graph，以減少計算和記憶體損耗。

g. 使用 *torch_geometric* 建立一個 Heterogeneous Link-level GNN。（以下為模型的結構）

```
Model(
  (movie_lin): Linear(in_features=384, out_features=64, bias=True)
  (user_emb): Embedding(524, 64)
  (movie_emb): Embedding(17770, 64)
  (gnn): GraphModule(
    (conv1): ModuleDict(
      (user__rates__movie): SAGEConv(64, 64, aggr=mean)
      (movie__rev_rates__user): SAGEConv(64, 64, aggr=mean)
    )
    (conv2): ModuleDict(
      (user__rates__movie): SAGEConv(64, 64, aggr=mean)
      (movie__rev_rates__user): SAGEConv(64, 64, aggr=mean)
    )
  )
  (classifier): Classifier()
)
```

其中 Classifier 用於對 graph 中的 edge 進行預測或分類，並利用 source node 和 target node 的 embedding vector 之間的相關性進行預測。

4. 實驗過程及結果

a. 訓練 5 個回合，損失函數採用 torch.nn.functional.binary_cross_entropy_with_logits 最低結果為 0.2052

```
Device: 'cpu'
100%|██████████| 188/188 [00:10<00:00, 18.73it/s]
Epoch: 001, Loss: 0.2363
100%|██████████| 188/188 [00:11<00:00, 16.86it/s]
Epoch: 002, Loss: 0.2263
100%|██████████| 188/188 [00:13<00:00, 13.59it/s]
Epoch: 003, Loss: 0.2178
100%|██████████| 188/188 [00:08<00:00, 21.37it/s]
Epoch: 004, Loss: 0.2129
100%|██████████| 188/188 [00:10<00:00, 17.10it/s]Epoch: 005, Loss: 0.2052
```

b. 使用 validation data 進行驗證，並採用 *sklearn.metrics.roc_auc_score* 作為評估指標，分數為 0.9480

```
100%|██████████| 79/79 [00:01<00:00, 45.57it/s]
Validation AUC: 0.9480
```

5. 結論

因為我們兩個的論文方向皆與 Graph 有關，而這次的專案讓我們有機會深入了解 GNN 以及 Graph 的應用。透過實際操作和對網路上各種資源的學習，我們不僅僅侷限於課堂上學習到的基本概念，更是擴展了我們的能力。我們學會了如何使用 GNN 進行 Graph 的分析和預測，並能夠利用這種技術應用於各種領域，如推薦系統、社交網路分析等。這次專案的完成不僅提升了我們的技能，也培養了我們獨立學習和研究的能力，這將對我們未來寫論文的過程以及職涯發展都有重要的影響。