HW2 Community Detection

M11207509 王佑強

執行程式方法

1. 使用 jupyter(此檔案為 ipynb)
2. 安裝套件 `pandas`, `networkx`, `python-louvain`
3. 執行每個 code block
4. 生成 submission_jimmy.csv 檔再放到 kaggle 即可

演算法流程

1. **讀取訓練及測試資料**: `train.csv` , `test.csv`
2. **建構無向圖**: 根據 `train.csv` 使用 `networkx` 構建無向圖。
3. **community檢測**: 使用 Louvain 方法 (`community_louvain.best_partition`) 對圖進行community檢測，將每個節點分配到一個community。
4. **預測測試資料**: 對於 `test.csv` 中的每對節點，檢查它們是否屬於相同community，並生成預測結果。
5. **生成submission文件**: 將預測結果格式化並保存為 `submission_jimmy.csv` 文件，用於提交至 Kaggle。

預處理

- **去除重複的edge**: 在圖構建過程中去除重複的邊，以保證圖的準確性。
- **資料映射**: 使用dictionary將節點快速映射到其所屬的community，提升預測效率。

方法

- **Louvain 算法**: 一種基於模塊度最大化的community檢測算法。反覆地將節點分配到不同社區來最大化modularity
    - **Modularity**是在網絡科學和圖論中用來衡量網絡劃分成community強度的指標，比較community內部邊的密度與community之間邊的密度。

    - High Modularity：表示commutniy結構強，即同一社區內的節點連接密集，不同社區之間的節點連接稀疏。值通常在 0 到 1 之間，值越高表示社區定義越清晰。
    - **Low or Negative Modularity**：表示community劃分不比隨機分配顯著，這表明社區結構弱或不存在。
- Community是指節點之間連接密度比隨機分佈的邊預期連接密度更高的群體。
- **步驟**:
    1. **局部節點移動**: 將節點移動到臨近community以最大化modularity gain。
    2. **合併community**: 基於第一步的結果，合併小社區形成大社區。
    3. **網絡合成**: 形成新網絡，重複上述步驟直到模塊度不再增加。

6	M11207509_王佑強		0.7566	5	5h