SDML HW1 Task1

B04902105 戴培倫

October 13, 2018

Contents

1 Adjacency + Similarity

1.1 Cosine Similarity

計算兩個 adjacency matrix 的 cosine similarity

- one layer
  直接計算兩個 node neighbor 的相似度
  private: 0.613
  public: 0.615

- two layers
  計算 A 跟 B neighbor 的相似度
  計算距離 A 兩步的 neighbor 與 B 一步的 neighbor 的相似度
  計算距離 B 兩步的 neighbor 與 A 一步的 neighbor 的相似度
  計算距離 A 兩步的 neighbor 與 B 兩步的 neighbor 的相似度
  分別給不同的 weight，加起來。
  private: 0.616
  public: 0.618

1.2 Jaccard Coefficient

計算兩個 adjacency matrix 的 cosine similarity

- one layer
  直接計算兩個 node neighbor 的相似度
  private: 0.643
  public: 0.635

- two layers
  計算 A 跟 B neighbor 的相似度
  計算距離 A 兩步的 neighbor 與 B 一步的 neighbor 的相似度
  計算距離 B 兩步的 neighbor 與 A 一步的 neighbor 的相似度
  計算距離 A 兩步的 neighbor 與 B 兩步的 neighbor 的相似度
  分別給不同的 weight，加起來。
  private: 0.619
  public: 0.616

- confidence
  採用曾奕青的作法，把 graph 設定為 undirected，在跑 testing 時，先跑過一次，並設一個 confidence threshold，當 similarity 超過此 threshold 時，便假設此 edge 為 True，藉此增加 positive edge。對下一步的判斷有幫助。
  private: 0.660
  public: 0.662
  Note: 設為 *undirected graph* 的 *accuracy* 進步約 *5%*

2 Embeddings + Similarity

2.1 DeepWalk

計算兩個 embedding vector 的 cosine similarity
private: 0.590
public: 0.593

2.2 GCN

用 github 上 pytorch version 的 GCN，target 設為每個 node 的 degree，embedding dimension = 128，train 完之後抽出中間的 embedding layer 當作 node 的 feature vector，再計算 cosine similarity。
private: 0.501
public: 0.502

3 Negative Sampling

3.1 Two Steps

取所有 train.txt + test-seen.txt 的 node，將距離兩步的 node 都當作 negative edge(扣掉所有出現在 test-seen.txt 和 test.txt 的 edge)，約有 150 萬筆。

3.2 Three Steps

取所有 train.txt + test-seen.txt 的 node，將距離三步的 node 都當作 negative edge(扣掉所有出現在 test-seen.txt 和 test.txt 的 edge、並扣掉兩步的 edge)。

3.3 Topological sort + Random

將 graph 中隨機拿掉造成 cycle 的 edge（約 1000 多條）直到沒有 cycle 後，做 topological sort。隨機選 N 個 node，從這些 node 後面距離 5 步以內的點隨機挑一個當作 negative edge（沒出現在 training 和 testing data 中的）。

4 Embeddings + Classifier

4.1 DeepWalk + Linear SVC

無論取多少維度的 deepwalk embedding + [two steps | two and three steps | topological] negative sampling + linear SVC，傳上 kaggle 都不到 0.5 的 accuracy。

4.2 DeepWalk + Random Forest

deepwalk embedding + [two steps | two and three steps | topological] negative sampling + Random Forest。
private: 0.524
public: 0.520

4.3 DeepWalk + GCN + Random Forest

deepwalk embedding 拿來當作 GCN 的 feature，也就是用 Adjacency Matrix 和 deepwalk embedding 做 convolution，target 為 node degree。取出中間的 embedding + [two steps | two and three steps | topological] negative sampling + Random Forest。
private: 0.517
public: 0.519

4.4 PRUNE + XGBoost + Jaccard Coefficient

PRUNE embedding + [40%two steps + 40%three steps + 20%four steps + 20%topological] negative sampling + XGBoost Binary Logistic Classifier。
private: 0.631
public: 0.622

PRUNE embedding + [40%two steps + 40%three steps + 20%four steps + 20%topological] negative sampling + XGBoost Binary Hinge Classifier。
private: 0.609
public: 0.608

5 Ensemble

因為 single graph embedding model 都做得很不好，嘗試將幾個放在一起用 mean / voting 做 ensemble。
private: 0.644
public: 0.638

6 Final Score

最後嘗試就是用 Jaccard Coefficient + Confidence 的方式，連續做 3 5 次，把 confidence 高的當作 positive，再用個 threshold 判斷 positive or negative。
private: 0.662
public: 0.663

7 Reflection

在這個 task 上雖然花了很多時間，也做了不少嘗試，一直沒有做好的主要原因是我沒有建立一個良好的 validation system，沒有辦法及早發現問題出現的環節。
在作業剛開始的時候就用了 DeepWalk，看他的 code 感覺很容易，所以就自己寫了一個，用 Cosine Similarity 做分類可以達到約 0.60，因此覺得沒什麼問題，跟同學討論 DeepWalk 時也沒有想到我是用自己寫的，可能參數和細節方面有些不同，導致我都以為是在 DeepWalk 之後的部份出了問題，前幾天聽分享才發現原來 DeepWalk + Cosine Similarity 就可以達到 0.67 0.68，才知道在根本的地方就出現問題了。
之後便是嘗試不同的方法，但因為一開始的 DeepWalk Embedding 是爛掉的，之後的實驗就是一直嘗試一直失敗。等到我驚覺可能是 Embedding 的問題時，我才開始嘗試 PRUNE，雖然有比較好的結果，但次數已經不夠讓我可以調整參數了。