Social Media Analytics #HW2

**Community Detection**

M11007601 周沂潔

1. 說明如何執行程式(並附上程式碼檔案)

程式撰寫在**Google Colab**中，

匯入 (1) train.csv (2) test.csv 後，

可以直接執行，匯出一個community\_detection.csv file。

1. 簡介你所使用的程式架構及演算法流程(如果有進行前處理也請解釋原因)
2. Strategy to Solve the Community Detection

使用Community-louvain 的Library來實作。

* Community.best\_partition(graph, partition=None, weight=’weight’, resolution=1.0)

是使用Louvain Heurictices方法劃分的獲得最高modularity的community detection演算法

* Louvain Algorithm

Community內的edge weight和Community之間的edge weight的比例，edge weight越大則modularity越大，community內的node更緊密，劃分的質量越好。

A picture containing text, watch

Description automatically generated

M為Graph的edge數量，2m為總degree數。

Ａ為兩點i j相連時為1，否為0。

 為i j在同一community時為1，否為0。

* Louvain Flow Chart

Chart, radar chart

Description automatically generated

Fig 1. Louvain Flow Chart

1. 每個node都先分配一個單獨的community，再去嘗試與自己連接的節點歸入對方的community，計算加入不同鄰居community之後的modularity gain，最後將自己歸到modularity gain最大的community中。

A picture containing diagram

Description automatically generated

1. 將每個community變成super node，supernode與supernode之間的weight就是兩個community之間的edge數量。supernode自己的weight就是community內degree的數量
2. 回到 (1) 不斷的迭代，每次網路中的初始community數量會變小，因此迭代的速度會更快。
3. 程式架構及演算法流程

Import libraries

Text

Description automatically generated

讀取測試資料集

Text

Description automatically generated

將沒有連線到的點記錄下來

Text

Description automatically generated

將沒有連線到的node與最大值的node+10連在一起（與其他community區隔開來）

Text

Description automatically generated

將連線好的independentNode和原本的node append後做sorting後會出trainEdge.csv。



Text

Description automatically generated

使用networkx建立graph，並用community\_louvain.best\_partition做detection。

Text

Description automatically generated

最後以test.csv匯出預測結果

Text

Description automatically generated

Text

Description automatically generated