1. 爬蟲資料:

{"postdate": "2016-08-01 13:25\u00a0\u00a0#1", "author": 1, "topic": "\u5bcc\u90a6\u4e59\u5f0f\u4fdd\u96aa\u7d30\u9805\u4fdd\u984d\u54ea\u9805\u8a72\u66f4\u63db\u6216\u65b0\u589e\u7684\u55ce", "post\_id": "f=264&t=4878261", "usr\_name": "Bou\u5e03", "usr\_id": "userinfo.php?id=3025887", "my\_url": "http://www.mobile01.com/topicdetail.php?f=264&t=4878261"}

資料為mobile01論壇之發文回文資料，共有toyota、BMW等數個廠牌之資料集，每個約有數萬筆資料，資料時間2006~2016年，每筆資料包含:

1. 發表時間
2. 是否為本篇貼文作者(1代表作者、0代表回文者)
3. 主題
4. 貼文ID
5. 使用者名稱
6. 使用者ID(取=號後面數字即可，範例之id為3025887)
7. URL
8. 程式功能需求

主要目的是希望透過建構社群網路架構找出具備高度影響力的意見領袖使用者

步驟如下:

1. 建立社群網路:

是一個社群網路圖，V和E分別是頂點和邊的集合： ，，而W則為邊之權重。一個點表示使用者∈U，而一個邊則表示兩點間的關係，在一篇文章中有回應文章的關係，則一個邊存在於E之中。

以資料來舉例，相同post\_id下，所有author為0的使用者和author為1的使用者皆會建立一條邊。

而每一條邊的會有權重，其計算方式如下:

考慮兩種相似性，行為相似性和結構相似性。行為相似性中兩個點之行為分別為和，其行為相似性公式為:

B為使用者之行為向量，是將其發文時間依照2小時為一個區間做紀錄，如:0:00~2:00為t1、2:00~4:00為t2……22:00~24:00為t12，例:A共有三筆時間分別為0:00、1:00、23:05之資料，擇其行為向量為{2,0,0,0,0,0,0,0,0,0,0,1}

行為相似度，即為兩使用者的行為向量取cos相似度



另一個是結構相似性，adj(v)為使用者的相鄰節點，結構相似性計算方式為兩使用者的相鄰節點交集數/兩節點之相鄰節點數成績之根號值



而邊的權重為， 值預設為0.5

1. 社區結構偵測

將建好的社群網路圖進行使用者社區分區，建立之演算法如下:

|  |
| --- |
| **Algorithm : *Community\_Detection* (***G*, *k***)** |
| **Input:** *G*(*V, E*, *W*): a social network,  *k*: user-specified number  **Output:** *Cs*: set of significant communities   1. *C* ← ∅; *Cs* ← ∅; 2. *C* ={*c*1, *c*2,..., *c|V|*} ← seteach *u*∈V as a community in *C*; 3. **while** *true* **do** // community detection 4. **for each** *ci*∈ *C***do** 5. *adj*(*ci*) ← collect all communities connected to *ci*; 6. **for each** *cj*∈ *adj*(*ci*)**do** 7. **if** max\_edge(*ci*) = max\_edge(*cj*) 8. *ci* ← merge *ci* and *cj*; 9. *C* ← *C−cj*; 10. **if** Δ*Q* < 0 // modularity gain in Definition 4 11. break; 12. **for each** *ci*∈ *C***do** 13. *ni*← count the number of nodes in *ci*; 14. **if** *ni* > *|V| / k* // significant community extraction 15. *Cs* ← *Cs* ∪ *ci*; 16. **output** *Cs*; |

需要輸入讓使用者輸入一個k值，以做為顯著社區的篩選值。之後將每個節點視為一個社區，並且如果這兩個節點之間的權重在其周圍邊緣中彼此之間的權重最大，則將每對節點分組到社區中。例如，給定兩個節點a和b，如果邊（a，b）在連接到a的所有邊中最大，並且在連接到b的所有邊中也是最大的，我們將a和b合併到一個社區中。接下來，我們將每個新創建的社區作為一個節點來處理，並且該過程一直持續到達到終止條件ΔQC→C’ = Q(C’) − Q(C) < 0。

終止條件為 Modularity Gain定義為：給定一個社會網絡"G(V,E,W)" 和社區結構C = {c1，c2，...，cp}，Modularity Gain計算方式:



ISi = Σu,v∈ci w(u, v)是群集ci中節點權重之總和，DSi = Σ u∈ci,v∈V ，w(u, v)是群集ci和群體外其他節點權重總和，TS = Σ u, v∈V w(u, v)是網路中任意兩點間權重之總和。

總節點/K=>社區保留門檻值，K為演算法說明裡設定的

1. K-means分群

|  |  |
| --- | --- |
| attribute | Description |
| ***article\_num*** | Total number of articles published by a user |
| ***replied\_by\_prob*** | The probability of a user’s articles replied by other users |
| ***reply\_prob*** | The probability that a user replies other users |
| ***deg\_centrality*** | The number of a user’s neighbor nodes |
| ***bet\_centrality*** | The number of the shortest path pass through a user |

基於五個特徵:貼文數(回文數+發文數)、文章被回應之比率(有被回文之發文數/總發文數)、回應其他文章之比率(回文數/總貼文數)、相鄰節點數、最短路徑經過數

1. 在每個顯著社區中進行K-means分群，最後產出分群結果和每個群的平均五個特徵分數，給定一個使用者輸入意見領袖個數n，則顯著社區會按照社區大小分配此n值，然後依照分配到的數目k，從每個社區中依分數高低之K-means分群

依序輸出使用者ID，直到滿足配額。

1. 特別需求
2. 被回文數最多的前n個使用者ID
3. 上述產出之n個使用者ID
4. 考慮時間因素的n個使用者ID

考慮時間因素版本之說明:

主要程式需求如上述，但要追加考慮時間因素版本，與上述之不同處為編之權重計算，權重計算時需考慮時間，2006年之回文權重0.1、2007年0.2、20080.3依此類推至20161.1，其餘不變。

1. 評估意見傳遞之功能

計算n個使用者被其他使用者回應之ID個數，每個ID只計算一次，例如:ID:1570之使用者皆有回應在n個中3位使用者的發文，但1570只能算1次。