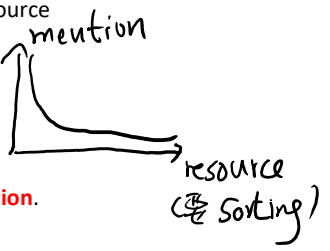
Evaluation (2018/1/3)

2018年1月3日　　上午 10:02

實驗方式

- 請先整理一下1997~2017涉及多少resources, articles(pmid), mentions, co-mentions.

- 在進行實驗前, 我們會隨機去挑 *L*個 resources, 並任兩個resources組成一個resource evaluation pairs. 這些resource evaluation pairs將會用來評估(比較)resource communities的優劣.
  - 請整理一份resource – mention的圖表 (如右圖), 好決定要挑多少resources產生evaluation pairs.
    我們理應挑較多mention的resources來產生要測試的resource evaluation pairs.



- 資料集: 目前鎖定1997~2017共20年的articles(pmid), 將其所含的mention/co-mention進行**10-fold cross validation**.

- 10-fold Cross Validation細節:
  - 每個fold要均勻包含1997~2017的資料, 這是因為不同年的pmid數差很多, 不均分資料恐會影響實驗結果
  - 先查出每個pmid對應的年份, 然後整理出每年的pmid 集合.
  - 再將該集合隨機分成10分 (注意每份要差不多大小), 這樣就可以將這20年的資料均分成10分

  - 每個fold的training:
    - 每次cross valiation挑一份出來做testing, 其他9份做 resource間的MI計算, 建network且detect communities.
    - 注意resource間的co-mention不要高估, 高估會影響MI計算, 只能算該cross valiation那9份內的co-mention次數, 不能把testing那份的co-mention抓進來算. mention也是不能高估
    - SLM的resolution parameter先設定為1, 且記錄該fold產生多少communities (比較對象LDA需要這項資訊), 我們後續還會跑其他parameter設定.

  - 每個fold的testing:
    - 依照該fold(上述)的community dection結果, 我們可將測試兩個效能指標: **Louvain Modularity** 與 **AU-ROC**

    - **AU-ROC** - 就是area under ROC (ROC可去看一下課程投影片 CH08 Page 28)
      我們首先將那*L*個resource evaluation pairs按照**它們再**testing data內的co-mention frequency, 由大到小排序.
      接著一筆一筆將每個resource evaluation pair標 1/0 標籤,
      1 表示該pair的resources屬於同一個community, 0 則不屬於同一個集團.
      這裡有個API可以用來計算AU-ROC
      http://scikit-learn.org/stable/modules/generated/sklearn.metrics.roc_auc_score.html
      (如何使用該API我們開會時說明)

    - **Louvain Modularity - 主要衡量communities是否夠扎實 (注意!! 這measure不會用到**resource evaluation pairs)
      https://en.wikipedia.org/wiki/Louvain_Modularity

      $$Q = \frac{1}{2m}\sum_{ij}\left[A_{ij} - \frac{k_i k_j}{2m}\right]\delta(c_i, c_j),$$

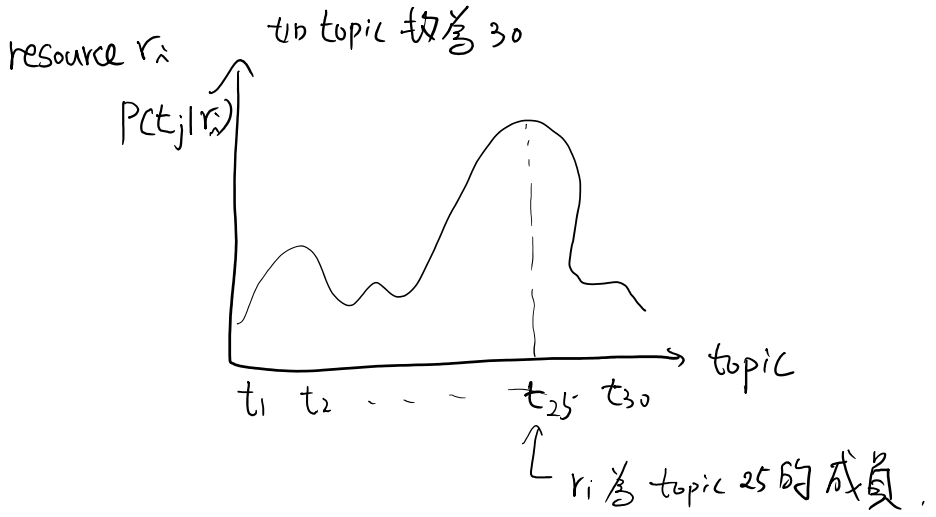
      where

      - A_{ij} represents the edge weight between nodes i and j;
      - k_i and k_j are the sum of the weights of the edges attached to nodes i and j, respectively;
      - m is the sum of all of the edge weights in the graph;
      - c_i and c_j are the communities of the nodes; and
      - δ is a simple delta function.

      $\delta(c_i,c_j)$ = 1 if the communities of nodes *i* and *j* are the same, otherwise, it is 0.
      計算L-modularity的過程如下:
      先將testing data內所有的resources建出network, 公式內的A_{ij}為resources *i* 與 *j*的mutual information, 注意!! 這個mutual information是用testing內的mentions/co-mentions算出來的, 不可以用到training data的資料.
      k_i為在這個用testing data建出來的網路內, 跟node (resource) *i* 有聯結的edges MI總和
      m為在這個用testing data建出來的網路內, 所有edges的MI總和
      簡單來說, L-modularity就是用testing data來建出一個network (MI為edge權重), 再看training data找出的communities是否能反映testing data建出的network連結緊密情況.

關於 LDA

- 由於LDA是分析文本, resource的description沒有年份資訊, 所以只能依照上述每個fold, SLM生出來的community 數量來將resource分成相同的clusters (communities).
  - 舉例來說若某個fold, SLM產生30個communities, 我們就以topic數30來跑LDA (文本就是所有resources的descriptions), LDA會產生兩組重要的distributions, 其中一組是每個文件(resource - description)對topic的機率分布 (如右圖), 我們可將每個resource分派給機率最大的topic (cluster or communities).
  - 同樣, 我們可以將這個clustering結果進行上述的testing, 並計算其AU-ROC還有L-modularity



工作:

- 趕緊整理好1997~2017的10 fold資料
- 這個時段的resource-mention圖表(好決定要挑多少resource產生 resource evaluation pairs)
- 了解怎麼跑LDA package