關鍵詞語網路

Sung-Chien Lin

2018 年 9 月 30 日

課程簡介

課程內容

- 運用斷詞結果，統計每一文件中出現的詞語
- 根據詞語出現的文件情形(tfi*df)判斷關鍵詞語
- 根據關鍵詞語共同出現的文件(Jaccard Similarity, Correlation Coefficient)判斷詞語之間的相關性
- 將關鍵詞語與它們之間的相關性視覺化，表示成網路圖
- 利用關鍵詞語網路的結構，找出詞語形成的集群，發現可能的主題

學習目標

- 能夠說明共同出現的詞語所代表的意義。
- 能夠利用 Jaccard Similarity 或是 Correlation Coefficient 等相關性計算方式，計算詞語之間因為共現關係產生的相關程度。
- 能夠將詞語之間的相關性畫成網路圖，進行視覺化
- 能夠利用關鍵詞語網路的結構，找出詞語形成的集群

詞語之間的相關性

- 某一對詞語共同出現在同一文件的情形。如果有某一對詞語共同出現在文件中，它們很有可能彼此相關。
- 文本處理時常利用詞語之間的共現訊息估算它們之間的相關性。常見的相關性評估指標有：
  - 共同出現的文件數：共同出現文件的數量(d_{xy})愈多，兩個詞語是否愈相關？
  - Jaccard Similarity：($d_{xy}/(d_x+d_y-d_{xy})$)
  - Dice Similarity：($2d_{xy}/(d_x+d_y)$)
  - Correlation Coefficient：
  - 註：d 代表文件的總數. d_{xy} 為詞語 x 和 y 一起出現的文件數量，d_x 和 d_y 分別代表詞語 x 和 y 出現的文件數量

透過分析詞語之間的相關性找出文件集合內的重要主題

- 如果有一群詞語因為彼此之間具有共現現象而有高相關性的特質，該詞語群可能代表某一個主題
- 例如：一個新聞事件

網路圖

- 以節點和其間的連結線表示對象與其間的關係
  - 節點：詞語
  - 連結線：詞語之間的共現關係(相關性)
  - 節點間的距離愈接近，可能代表詞語之間愈相關
- 在圖形上聚集成群的詞語，可能是一個主題
  - 利用網路的結構，找出詞語的集群

本次課程需先安裝套件

- 網路分析套件 igraph
- 請在 Console 上輸入

```
install.packages("igraph")
```

本次課程程式

```
# 載入套件
library(tidyverse)
library(jiebaR)
library(igraph)

# 讀入聯合報五月一日到十日的要聞新聞資料
news <- data.frame()

for (i in 1:10) {
  date <- sprintf("2018_05_%02d", i)

  news.df <- read.csv(file=paste0("udn_", date,".csv"),
                      fileEncoding="UTF-8", stringsAsFactors=FALSE) %>%
    mutate(date=date)
  news <- rbind(news, news.df)
}

#########
# 找出新聞中重要的詞語及其共同出現的新聞數量
# 利用 tfidf 找出重要詞語

# 設定 jieba 斷詞器
mp.seg <- worker(type="mp", user="./udn_2018_09.dict", bylines=TRUE)

# 將新聞內容斷詞
news <- news %>%
  filter(!is.na(text)) %>%                  # 保留有內容的新聞資料
  mutate(id=row_number()) %>%               # 每則新聞加上編號(id)
  mutate(word=segment(text, mp.seg)) %>% # 斷詞
```

```
  select(id, word) %>%                          # 選擇新聞編號和斷詞結果
  unnest(word) %>%                              # 將斷詞結果展開
  filter(grepl("\\p{Han}+", word, perl=TRUE)) %>% # 保留至少一個中文字的
詞語
  filter(nchar(word)>1)

# 計算每個詞語 idf(inverse document frequency)
word.idf <- news %>%
  mutate(total.doc = n_distinct(id)) %>%  # 統計文件總數
  group_by(word, total.doc) %>%           # 統計詞語出現文件數
  summarise(doc=n_distinct(id)) %>%
  mutate(idf=log(total.doc/doc)) %>%      # 計算 idf
  ungroup()

# 計算每個詞語在各個新聞中的出現次數與頻率(tf)
word.news <- news %>%
  group_by(word, id) %>%    # 計算每個詞語在各個新聞中的出現次數
  summarise(c=n()) %>%
  ungroup() %>%
  group_by(id) %>%          # 計算各新聞的詞語出現次數總和(sum(c))
  mutate(tf=c/sum(c)) %>%   # 計算詞語的出現頻率(tf)
  ungroup()

# 計算 tf*idf
word.news <- word.news %>%
  left_join(word.idf, by="word") %>%      # 連結 idf 資料
  mutate(tfidf=tf*idf)                    # 計算

# 計算詞語在所有新聞的 tfidf 總和，做為詞語的重要性，並且選出前 100 個重要詞語
keyword <- word.news %>%
  filter(tfidf>0.04) %>%                # 去除新聞中 tfidf 過低的詞語
```

```
  group_by(word) %>%                    # 計算詞語在所有新聞的 tfidf 總和
  summarise(sum.tfidf=sum(tfidf)) %>%
  top_n(100, sum.tfidf) %>%             # 選出前 100 個重要詞語
  arrange(desc(sum.tfidf))

# 找出每則新聞中出現的重要詞語
keyword.news <- word.news %>%
  filter(tfidf>0.04) %>%                # 去除新聞中 tfidf 過低的詞語
  select(id, word) %>%
  semi_join(keyword, by="word") %>%  # 以重要詞語比對每個新聞內容
  arrange(id)

# 統計重要詞語出現的新聞數量
kw_docs <- keyword.news %>%
  group_by(word) %>%                  # 統計重要詞語出現的文件數
  summarise(c=n_distinct(id)) %>%
  ungroup() %>%
  arrange(desc(c))

# 計算重要詞語共同出現的新聞數量
kw_codocs <- keyword.news %>%
  inner_join(keyword.news, by=c("id")) %>% # 找出每一則新聞共同出現的重要詞語
  group_by(word.x, word.y) %>%                # 統計每一對重要詞語共同出現的新聞數量
  summarise(dxy=n()) %>%
  arrange(desc(dxy)) %>%                      # 按照共同出現的新聞數排序
  ungroup() %>%
  filter(word.x != word.y)                    # 刪除相同的詞語

# 加上重要詞語出現的新聞數量
```

```r
kw_codocs <- kw_codocs %>%
  left_join(kw_docs, by=c("word.x"="word")) %>%
  rename(dx=c) %>%
  left_join(kw_docs, by=c("word.y"="word")) %>%
  rename(dy=c)

#########
# 計算詞語共同出現的相關性
# 本次課程以Jaccard similarity 和Correlation Coefficient 兩種方式計算

# Jaccard similarity
jaccardSimilarity <- function(dx, dy, dxy) {
  dx <- as.numeric(dx)
  dy <- as.numeric(dy)
  dxy <- as.numeric(dxy)
  return(dxy/(dx+dy-dxy))
}

# 計算每對重要詞語的Jaccard Similarity
word_net.js <- kw_codocs %>%
  rowwise() %>%
  mutate(js=jaccardSimilarity(dx, dy, dxy)) %>%
  ungroup() %>%
  arrange(desc(js))

# 刪減較不重要共現資訊
word_net.js <- word_net.js %>%
  mutate(pr=percent_rank(js)) %>%           # 根據jc 進行百分比排序(由小到大)
  filter(pr>0.75) %>%                       # 保留後1/4
  select(from=word.x, to=word.y, weight=js)
```

```
########
# 將重要詞語與其共現資訊表示成網路圖

# 將資料轉成網路
wg.js <- graph_from_data_frame(word_net.js, directed=FALSE)

# 將節點之間的線合併
wg.js <- simplify(wg.js, edge.attr.comb = list("mean"))

# 根據節點分群的結果為各節點設定顏色
cl.js <- cluster_louvain(wg.js)
cl.js.mem <-  membership(cl.js)

# 計算各節點在圖形上的座標
coords.js <- layout_(wg.js, with_graphopt())

# 畫圖
png(file="graph_js.png", width=800, height=600)
plot(x=cl.js, y=wg.js, vertex.shape="none",
     vertex.label.cex=0.8, edge.lty="blank", layout=coords.js)
dev.off()

# 查看各分群(主題)內的詞語
cl.js.mem <- membership(cl.js)
for (i in seq(max(cl.js.mem))) {
  print(paste("Cluster", i))
  print(V(wg.js)$name[cl.js.mem==i])
}

# correlation coefficient
phiCoefficient <- function(d, dx, dy, dxy) {
```

```
  d <- as.numeric(d)
  dx <- as.numeric(dx)
  dy <- as.numeric(dy)
  dxy <- as.numeric(dxy)
  d.not.x <- d - dx
  d.not.y <- d - dy
  dx.not.y <- dx - dxy
  dy.not.x <- dy - dxy
  d.not.x.not.y <- d.not.y - dx.not.y
  return((dxy*d.not.x.not.y-dx.not.y*dy.not.x)/sqrt(dx*dy*d.not.x*d.no
t.y))
}

# 所有的關鍵詞語共出現在多少則新聞(d)
d <- keyword.news %>%
  distinct(id) %>%
  nrow()

# 計算每個詞語的Correlation Coefficient
word_net.cc <- kw_codocs %>%
  rowwise() %>%
  mutate(cc=phiCoefficient(d, dx, dy, dxy)) %>%
  ungroup()

# 刪減較不重要共現資訊的網路圖
word_net.cc <- word_net.cc %>%
  mutate(pr=percent_rank(cc)) %>%
  filter(pr>0.75) %>%
  select(from=word.x, to=word.y, weight=cc)

# 將資料轉成網路
wg.cc <- graph_from_data_frame(word_net.cc, directed=FALSE)
```

```r
# 將節點之間的線合併
wg.cc <- simplify(wg.cc, edge.attr.comb = list("mean"))

# 對節點分群
cl.cc <- cluster_louvain(wg.cc)

# 計算各節點在圖形上的座標
coords.cc <- layout_(wg.cc, with_graphopt())

# 畫圖
png(file="graph_cc.png", width=800, height=600)
plot(x=cl.cc, y=wg.cc, vertex.shape="none",
     vertex.label.cex=0.8, edge.lty="blank", layout=coords.cc)
dev.off()

cl.cc.mem <- membership(cl.cc)
for (i in seq(max(cl.cc.mem))) {
  print(paste("Cluster", i))
  print(V(wg.cc)$name[cl.cc.mem==i])
}
```

預備工作

準備工作目錄與檔案

- 在 rCourse 下，建立工作目錄 09
- 將 08 的新聞資料以及詞典檔複製到 09 下

設定工作目錄

- 首先開啟新的 Script
- 在 Script 上，設定工作目錄

```
setwd("rCourse/09")
```

載入套件

- 在 Script 上輸入

```
library(tidyverse)
library(jiebaR)
library(igraph)
```

讀入聯合報九月一日到十五日的生活新聞資料

- 在 Script 上輸入

```
news <- data.frame()


for (i in 1:15) {
  date <- sprintf("2018_09_%02d", i)


  news.df <- read.csv(file=paste0("udn_", date,".csv"),
                      fileEncoding="UTF-8", stringsAsFactors=FALSE) %>%
    mutate(date=date)
  news <- rbind(news, news.df)
}
```

設定 jieba 斷詞器

- 在 Script 上輸入

```
mp.seg <- worker(type="mp", user="./udn_2018_09.dict", bylines=TRUE)
```

- user="./udn_2018_03.dict" → 將斷詞的詞典加上未知詞偵測的結果
- bylines=TRUE → 斷詞時將輸入的 vector 上每一個文字字串分別輸出

將新聞內容斷詞

- 在 Script 上輸入

```
news <- news %>%
  filter(!is.na(text)) %>%                    # 保留有內容的新聞資料
  mutate(id=row_number()) %>%                 # 每則新聞加上編號(id)
  mutate(word=segment(text, mp.seg)) %>% # 斷詞

  mutate(word=segment(paste(title, text, sep="。"), mp.seg))) %>%
  select(id, word) %>%                        # 選擇新聞編號和斷詞結果
  unnest(word)                                # 將斷詞結果展開
```

- 可在 Console 上輸入 View(news)，檢視 news 上的資料

id	word
1	麥當勞
1	之
1	亂
1	1
1	0
1	3
1	0
1	之後

保留至少一個中文字的詞語

- \\p{Han}中文字
- \\p{Han}+至少一個中文字
- 在 Script 上輸入

```
news <- news %>%
  filter(grepl("\\p{Han}+", word, perl=TRUE)) %>%
  filter(nchar(word)>1)
```

計算每個詞語 idf(inverse document frequency)

- 在 Script 上輸入

```
word.idf <- news %>%
  mutate(total.doc = n_distinct(id)) %>%  # 統計文件總數
  group_by(word, total.doc) %>%           # 統計詞語出現文件數
  summarise(doc=n_distinct(id)) %>%
  mutate(idf=log(total.doc/doc)) %>%      # 計算 idf
  ungroup()
```

計算每個詞語在各個新聞中的出現次數與頻率(tf)

- 在 Script 上輸入

```
word.news <- news %>%
  count(word, id) %>%    # 計算每個詞語在各個新聞中的出現次數
  rename(c=n) %>%
  group_by(id) %>%          # 計算各新聞的詞語出現次數總和(sum(c))
  mutate(tf=c/sum(c)) %>%   # 計算詞語的出現頻率(tf)
  ungroup()
```

計算 tf*idf

- 在 Script 上輸入

```
word.news <- word.news %>%
  left_join(word.idf, by="word") %>%      # 連結 idf 資料
  mutate(tfidf=tf*idf)                    # 計算
```

測試 tf*idf 的過濾值

```
wn <- data.frame()
for (th in seq(0, 0.1, 0.01)) {
  wn <-  rbind(wn, (word.news %>%
    filter(tfidf>th) %>%
    summarise(doc=n_distinct(id), word=n_distinct(word)) %>%
    mutate(th=th)))
}
```

- 請在 Console 上輸入 View(wn)，查看各種閾值篩選後的剩餘詞語與新聞資料數

doc	word	th
1650	30991	0.00
1650	29616	0.01
1650	27618	0.02
1650	21774	0.03
1650	15654	0.04
1650	12038	0.05
1648	9652	0.06
1642	7894	0.07

設定詞語篩選閾值

```
tfidf.th <- 0.05
```

計算詞語在所有新聞的 tfidf 總和，做為詞語的重要性，並且選出前 100 個重要詞語

- 在 Script 上輸入

```
keyword <- word.news %>%
  filter(tfidf>tfidf.th) %>%                    # 去除新聞中 tfidf 過低的詞語
  group_by(word) %>%                            # 計算詞語在所有新聞的 tfidf 總和
  summarise(sum.tfidf=sum(tfidf)) %>%
  top_n(100, sum.tfidf) %>%                     # 選出前 100 個重要詞語
  arrange(desc(sum.tfidf))
```

- 可在 Console 上輸入 View(keyword)，檢視篩選出的前 100 個關鍵詞語

word	sum.tfidf
中獎號碼	29.273012
期開獎	16.106221
台彩	15.157279
中獎	14.372980
雙贏	13.544410
頭獎	12.877562
開獎	11.659712
注中獎	10.880660

找出每則新聞中出現的重要詞語

- semi_join 比對兩個 data frame，只保留第一個 data frame 出現在第二個的資料
- 在 Script 上輸入

```
keyword.news <- word.news %>%
  filter(tfidf>tfidf.th) %>%                # 去除新聞中 tfidf 過低的詞語
  select(id, word) %>%
  semi_join(keyword, by="word") %>%  # 以重要詞語比對每個新聞內容
  arrange(id)
```

統計重要詞語出現的新聞數量

```
kw_docs <- keyword.news %>%
  group_by(word) %>%                  # 統計重要詞語出現的文件數
  summarise(c=n_distinct(id)) %>%
  ungroup() %>%
  arrange(desc(c))
```

- 可在 Console 上輸入 View(kw_doc)，檢視關鍵詞語出現的文件數

word	c
台彩	68
中獎	67
公布	66
中獎號碼	63
今晚	47
台中	47
實際	47
開獎	46

計算重要詞語共同出現的新聞數量

- inner_join 比對兩個 data frame，連接兩個 data frame 出現的資料
- 在 Script 上輸入

```
kw_codocs <- keyword.news %>%
  inner_join(keyword.news, by=c("id")) %>% # 找出每一則新聞共同出現的重要詞語
  count(word.x, word.y) %>%                # 統計每一對重要詞語共同出現的新聞數量
  rename(dxy=n) %>%
  filter(word.x != word.y)                 # 刪除相同的詞語
```

- 可在 Console 上輸入 View(kw_codoc)，檢視關鍵詞語共同出現的文件數

word.x	word.y	dxy
中獎	台彩	66
台彩	中獎	66
中獎	中獎號碼	62
中獎號碼	中獎	62
中獎號碼	台彩	62
台彩	中獎號碼	62
中獎	公布	61
中獎號碼	公布	61

加上重要詞語出現的新聞數量

```
kw_codocs <- kw_codocs %>%
  left_join(kw_docs, by=c("word.x"="word")) %>%
  rename(dx=c) %>%
  left_join(kw_docs, by=c("word.y"="word")) %>%
  rename(dy=c)
```

計算詞語共同出現的相關性

Jaccard similarity

- dxy/(dx+dy-dxy)

```
jaccardSimilarity <- function(dx, dy, dxy) {
  dx <- as.numeric(dx)
  dy <- as.numeric(dy)
  dxy <- as.numeric(dxy)
  return(dxy/(dx+dy-dxy))
}
```

計算每對重要詞語的 Jaccard Similarity

```
word_net.js <- kw_codocs %>%
  rowwise() %>%
  mutate(js=jaccardSimilarity(dx, dy, dxy)) %>%
  ungroup() %>%
  arrange(desc(js))
```

刪減較不重要共現資訊

- 刪減 Jaccard Similarity 在後 75%的共現資訊

```
word_net.js <- word_net.js %>%
  mutate(pr=percent_rank(js)) %>%          # 根據 jc 進行百分比排序(由小到大)
  filter(pr>0.75) %>%                      # 保留後 1/4
  select(from=word.x, to=word.y, weight=js)
```

將資料轉成網路

```
wg.js <- graph_from_data_frame(word_net.js, directed=FALSE)
```

- 可在 Console 上輸入 summary(wg.js)檢視結果

```
> summary(wg.js)
IGRAPH 24fb62f UNW- 75 486 --
+ attr: name (v/c), weight (e/n)
```

- 75 個節點，486 條線

將節點之間的線合併

- 重複計算了詞語之間的相關性

```
wg.js <- simplify(wg.js, edge.attr.comb = list("mean"))
```

- 可在 Console 上輸入 summary(wg.js)檢視結果

```
> summary(wg.js)
IGRAPH e994c58 UNW- 75 243 --
+ attr: name (v/c), weight (e/n)
```

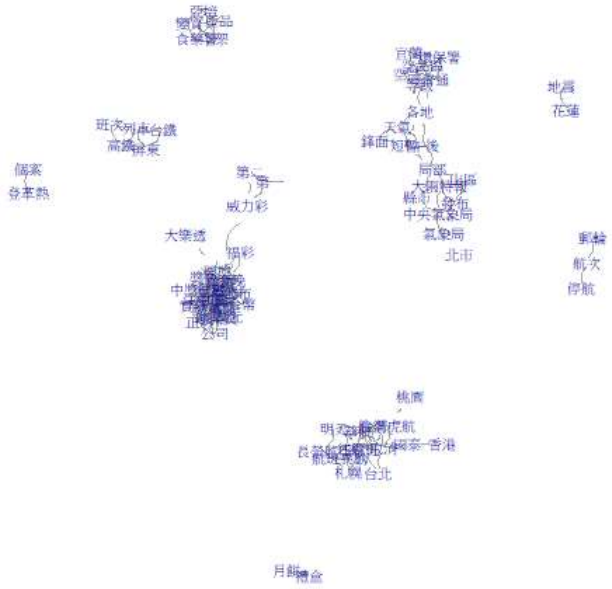
- 75 個節點，243 條線

計算各節點在圖形上的座標

```
coords.js <- layout_(wg.js, with_graphopt())
```

畫圖

```
png(file="graph_js.png", width=800, height=600)
plot(wg.js, vertex.shape="none",
     vertex.label.cex=0.8, edge.curved=TRUE, layout=coords.js)
dev.off()
```



對節點分群

```
cl.js <- cluster_louvain(wg.js)
```

- 可在 Console 上輸入 print(cl.js)檢視結果

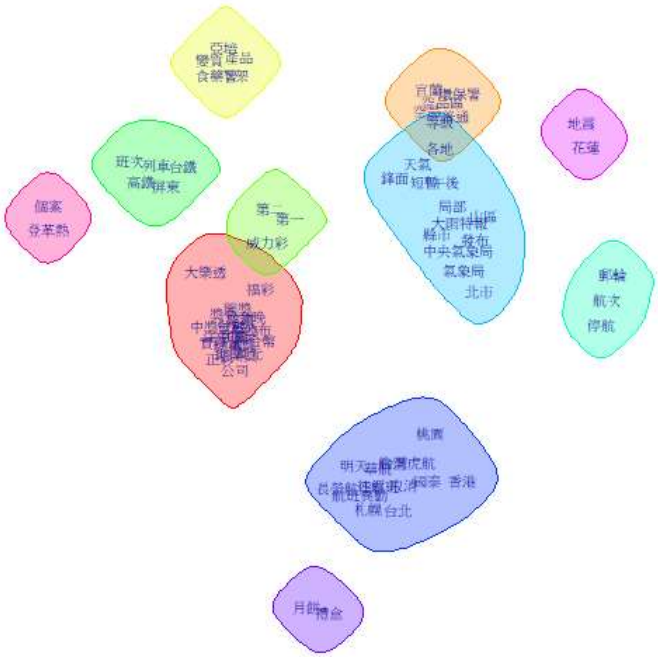
```
> print(cl.js)
IGRAPH clustering multi level, groups: 11, mod: 0.56
+ groups:
 $`1`
  [1] "中獎"   "台彩"   "中獎號碼" "結果"   "貳獎"   "公布"
  [7] "期開獎" "實際"   "頭獎"   "今晚"   "開獎"   "注中獎"
 [13] "新台幣" "萬元"   "雙贏"   "正彩"   "公司"   "獎金"
 [19] "福彩"   "大樂透"

 $`2`
 [1] "空品區" "普通"   "空品"   "等級"   "宜蘭"   "環保署"

 $`3`
 + ... omitted several groups/vertices
```

畫圖

```
png(file="graph_jc.png", width=800, height=600)
plot(x=cl.js, y=wg.js, vertex.shape="none",
     vertex.label.cex=0.8, edge.lty="blank", layout=coords.js)
dev.off()
```



查看各分群(主題)內的詞語

```
cl.js.mem <- membership(cl.js)
for (i in seq(max(cl.js.mem))) {
  print(paste("Cluster", i))
  print(V(wg.js)$name[cl.js.mem==i])
}
```

Cluster 1

中獎、台彩、中獎號碼、結果、貳獎、公布、期開獎、實際、頭獎、今晚、開獎、注中獎、新台幣、萬元、雙贏、正彩、公司、獎金、福彩、大樂透

Cluster 2

空品區、普通、空品、等級、宜蘭、環保署

Cluster 3

亞培、變質、產品、食藥署、下架

Cluster 4

威力彩、第二、第一

Cluster 5

班次、高鐵、列車、屏東、台鐵

Cluster 6

航次、停航、郵輪

Cluster 7

大雨特報、縣市、局部、短暫、午後、中央氣象局、發布、各地、天氣、山區、鋒面、氣象局、北市

Cluster 8

取消、航班、往返、國泰、台灣虎航、華航、航空、長榮航空、航班異動、札幌、桃園、台北、明天、香港

Cluster 9

月餅、禮盒

Cluster 10

地震、花蓮

Cluster 11

個案、登革熱

correlation coefficient

所有的關鍵詞語共出現在多少則新聞(d)

```
d <- keyword.news %>%
  distinct(id) %>%
  nrow()
```

correlation coefficient

```
phiCoefficient <- function(d, dx, dy, dxy) {
  d <- as.numeric(d)
  dx <- as.numeric(dx)
  dy <- as.numeric(dy)
  dxy <- as.numeric(dxy)
  d.not.x <- d - dx
  d.not.y <- d - dy
  dx.not.y <- dx - dxy
  dy.not.x <- dy - dxy
  d.not.x.not.y <- d.not.y - dx.not.y
  return((dxy*d.not.x.not.y-dx.not.y*dy.not.x)/sqrt(dx*dy*d.not.x*d.no
t.y))
}
```

計算每個詞語的 Correlation Coefficient

```
word_net.cc <- kw_codocs %>%
  rowwise() %>%
  mutate(cc=phiCoefficient(d, dx, dy, dxy)) %>%
  ungroup()
```

刪減較不重要共現資訊的網路圖

- 刪減 Correlation Coefficient 在後 75%的共現資訊

```
word_net.cc <- word_net.cc %>%
  mutate(pr=percent_rank(cc)) %>%
  filter(pr>0.75) %>%
  select(from=word.x, to=word.y, weight=cc)
```

將資料轉成網路

```
wg.cc <- graph_from_data_frame(word_net.cc, directed=FALSE)
```

將節點之間的線合併

- 重複計算了詞語之間的相關性

```
wg.cc <- simplify(wg.cc, edge.attr.comb = list("mean"))
```

計算各節點在圖形上的座標

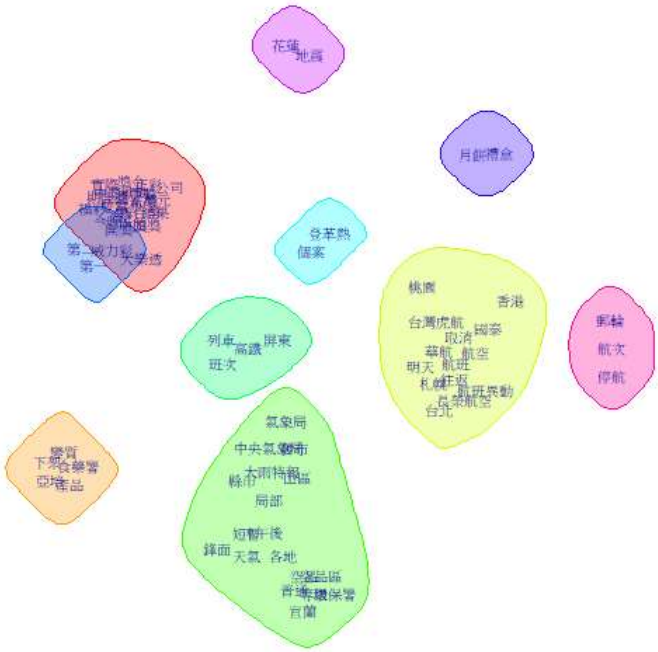
```
coords.cc <- layout_(wg.cc, with_graphopt())
```

對節點分群

```
cl.cc <- cluster_louvain(wg.cc)
```

畫圖

```
png(file="graph_cc.png", width=800, height=600)
plot(x=cl.cc, y=wg.cc, vertex.shape="none",
     vertex.label.cex=0.8, edge.lty="blank", layout=coords.cc)
dev.off()
```



查看各分群(主題)內的詞語

```
cl.cc.mem <- membership(cl.cc)
for (i in seq(max(cl.cc.mem))) {
  print(paste("Cluster", i))
  print(V(wg.cc)$name[cl.cc.mem==i])
}
```

Cluster 1

中獎、台彩、中獎號碼、公布、開獎、實際、今晚、期開獎、頭獎、結果、貳獎、萬元、雙贏、注中獎、新台幣、獎金、正彩、公司、大樂透、福彩

Cluster 2

亞培、產品、變質、食藥署、下架

Cluster 3

往返、航班、取消、台北、航空、台灣虎航、明天、長榮航空、華航、航班異動、國泰、桃園、札幌、香港

Cluster 4

局部、短暫、午後、大雨特報、山區、中央氣象局、發布、天氣、各地、空品、空品區、普通、等級、縣市、宜蘭、氣象局、鋒面、環保署

Cluster 5

班次、高鐵、列車、屏東

Cluster 6

個案、登革熱

Cluster 7

威力彩、第一" "第二"

Cluster 8

月餅、禮盒

Cluster 9

地震、花蓮

Cluster 10

航次、停航、郵輪

本次課程小結

小結

- 本次課程從關鍵詞語之間的共現情形發現新聞的重要主題
- 運用關鍵詞語之間的共現情形做為它們之間的相關性
- 將關鍵詞語以及它們之間的共現情形表現成網路圖
- 圖形上彼此接近的節點形成的集群可能代表重要主題

小結

- 本次課程運用 Jaccard Similarity 以及 Correlation Coefficient 計算關鍵詞語之間的相關性
- 這兩種計算方法，基本上都僅考慮詞語是否出現在文件中的資訊，並未利用詞語在文件上的出現次數
- 對於出現次數及重要性較高的關鍵詞語，僅考慮是否出現在文件中的資訊便已經足夠
- 若是還需加上出現次數較少的詞語以及雜訊較多的情況，還可以考慮其他的估算方式

小結

- 本次課程將關鍵詞語表示成網路進行視覺化，並且根據網路結構對關鍵詞語進行集群
- 本次課程所採用的集群演算法是具有較高效率的 Louvain Clustering 社群偵測(Community Detection)演算法
- 也可參考 http://igraph.org/r/doc/communities.html 上的說明，採用其他社群偵測演算法
- 除了根據網路結構對關鍵詞語進行集群之外，也可對此網路進行中心性分析以及其他結構分析

延伸思考

1. 從詞語共現網路的觀察是否可以看到這 15 天的生活新聞著重在哪一方面？分群內的詞語愈多，可能表示麼樣的現象？分群內的詞語之間愈密集，又可能表示麼樣的現象？
2. 想想看你可以將詞語共現網路的相關技術應用本身工作或研究上的哪些方面？