



社會網絡分析

第12章 塊模型

布丁布丁吃布丁 2017.8.29

<http://blog.pulipuli.info>

為什麼要做社會網絡分析？

Anti-categorical Imperatives

反類屬律令論

傳統社會統計將個人視為獨立的單位，
以個體屬性解釋現象

社會網絡分析認為，行動者相互依賴，
而彼此的關係(紐帶)影響大於屬性

參考書目

PAJEK 蜘蛛：社會網絡分析技術

原名：

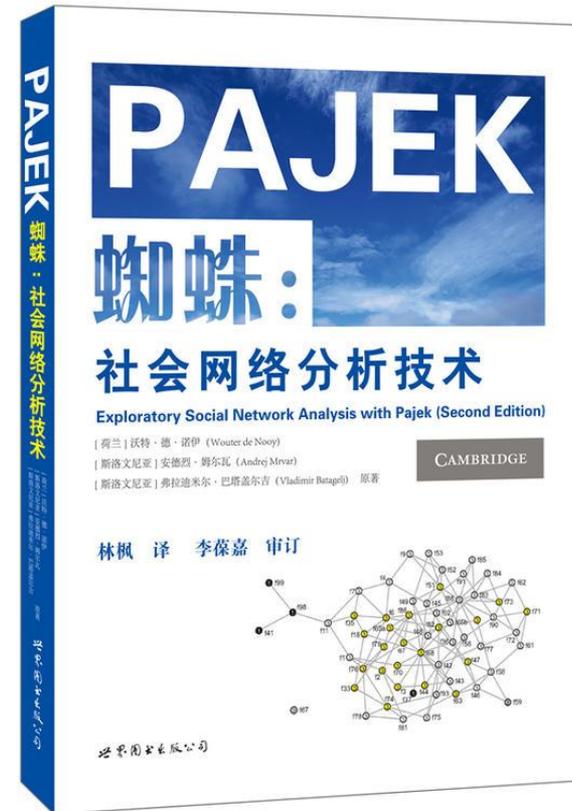
- Exploratory Social Network Analysis with Pajek (2nd Ed.)

作者：

- Wouter de Nooy
- Andrej Mrvar
- Vladimir Batagelj

翻譯者與編者

- 林楓 譯
- 李葆嘉 審訂



PAJEK 蜘蛛軟體 與 資料集

<http://l.pulipuli.info/17/pajek>



This repository Search

Pull requests Issues Marketplace Explore



pulipulichen / pajek-dataset

Unwatch 1

Star 0

Fork 0

Code

Issues 0

Pull requests 0

Projects 0

Wiki

Settings

Insights

保存「PAJEK 蜘蛛: 社會網絡分析技術」該書中會使用到的程式跟資料

Edit

Add topics

5 commits

1 branch

0 releases

1 contributor

MIT

Branch: master

New pull request

Create new file

Upload files

Find file

Clone or download

pulipulichen 0902 pajek

Latest commit 80628e6 5 minutes ago

chapter-12

0902 pajek

6 minutes ago

LICENSE

Initial commit

8 minutes ago

Pajek-book.7z

0902 上傳Ch12的資料

7 minutes ago

PAJEK 蜘蛛: 社會網絡分析技術 書籍目錄

Part 1. 基礎知識

1. 尋找社會結構
2. 屬性和關係

Part 2. 凝聚特性

3. 凝聚子群
4. 情感和友誼
5. 從屬關係

Part 3. 代理特性

6. 中心和周邊
7. 代理員和橋
8. 擴散

Part 4. 等級特性

9. 威望
10. 等級序列
11. 世系譜和引用關係

Part 5. 建模分析

- 12. 塊模型**
13. 隨機圖模型

塊模型



塊模型

章節目錄

第一節 簡介

第二節 矩陣和重排

第三節 角色集、位置集和對等性

第四節 塊建模

1. 塊模型
2. 塊建模
3. 正則對等性

本章概要

課後習題

課外作業

拓展閱讀

答案

塊模型簡介 (1/2)

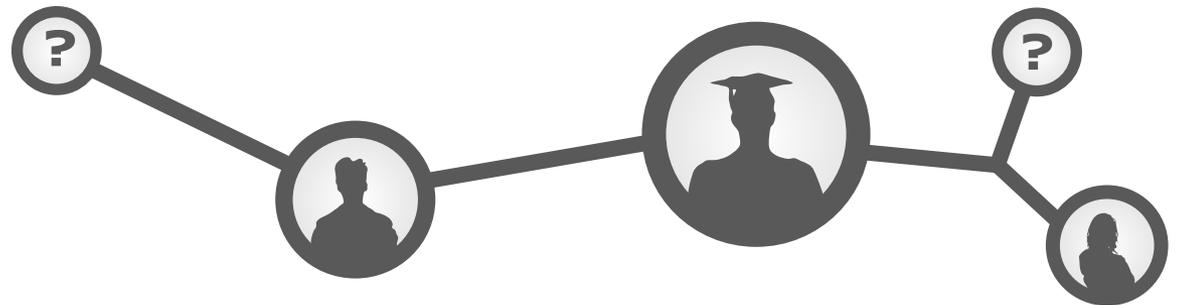
- 特性：
 - 能夠探查多種結構特性(凝聚性、中心性)的技術
 - 具備靈活性(rubost): 不容易受到幾條關係增減的影響
- 適用於小型高密度網路: 例如數位學習的討論區
- 理論基礎：
 - **對等性**: 在社會上所扮演的族群是否相同
 - **位置集**: 行動者(actor)在社會網絡中的位置, 跟實際上扮演的角色是否相同

塊模型簡介 (2/2)

- 分析做法：
 1. 以**鄰接矩陣**作為分析格式
 2. 塊建模將定點歸組於不同的雲集團(分組)
 3. 尋找雲集團之間的關係
- 分析目標：
 1. 凝聚性
 2. 核心—週邊結構
 3. 等級性

Section 2.

矩陣和重排



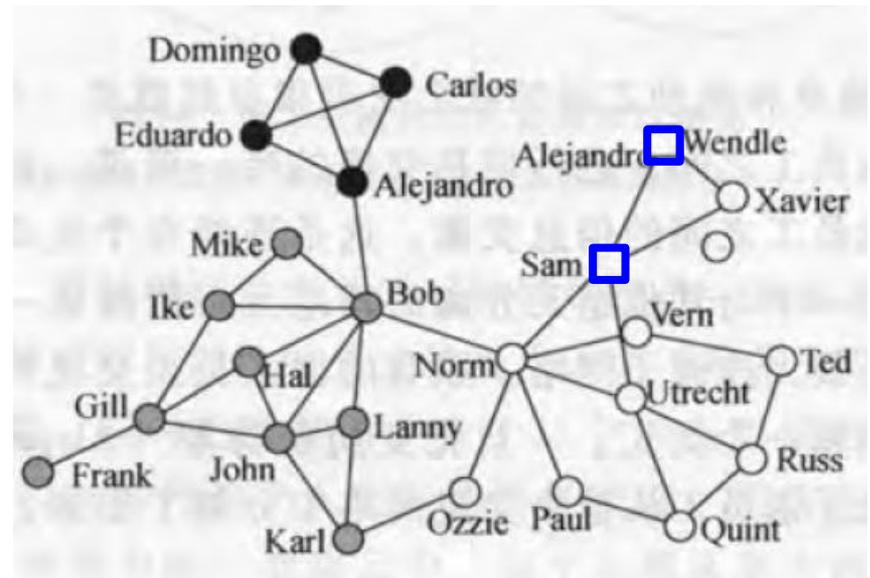
網絡圖

每個頂點(圓圈或方塊)表示一位行動者(actor)

- **對等性**：定點的顏色表示行動者的族群(使用語言)
- **位置集**：頂點的形狀表示行動者的角色：圓圈為工會成員，**方塊**為談判代表

罷工交流網絡圖

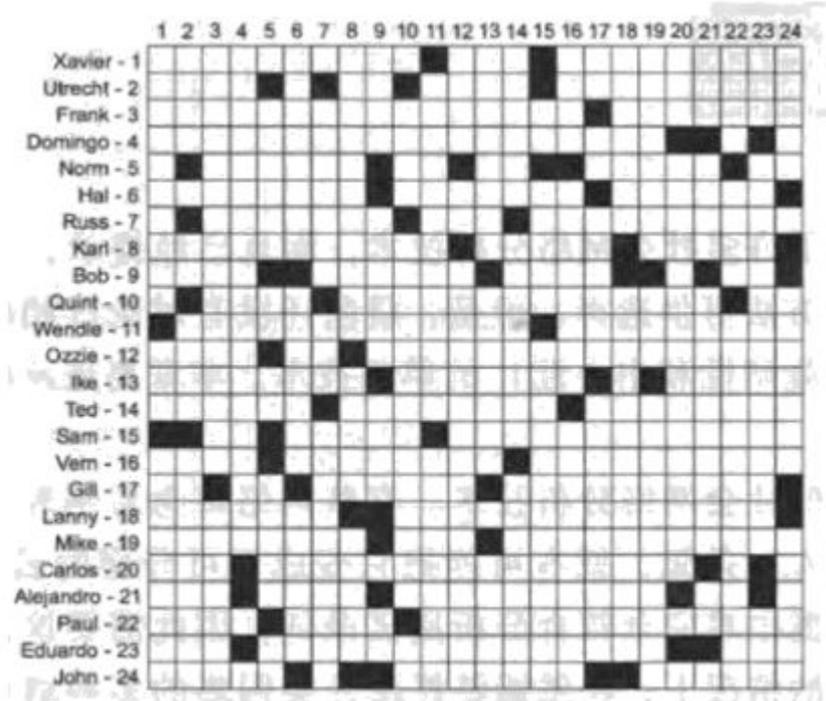
(第七章 代理員和橋)



網絡矩陣圖

- 將網絡圖以行跟列組成二位表格
 - 每個頂點表示行列的一個編號
 - Xavler是行跟列的1號
 - Norm是行跟列的5號
 - 行列相交的格子表示關係
 - 黑色(值=1): 存在紐帶
 - 白色(值=0): 不存在紐帶
- 紐帶(Bonding): 人和人之間的關係

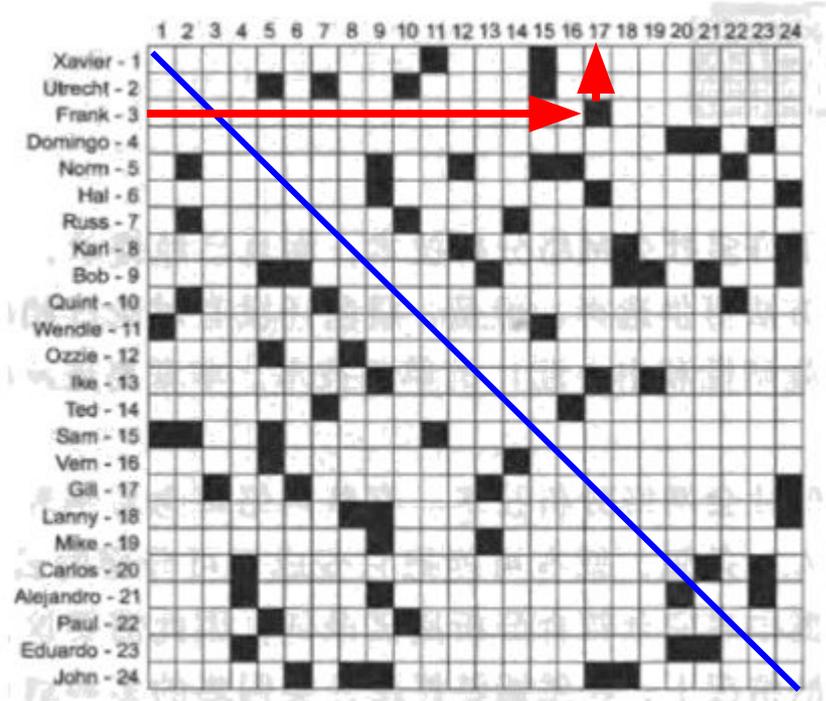
罷工交流網絡矩陣圖



網絡矩陣圖的方向

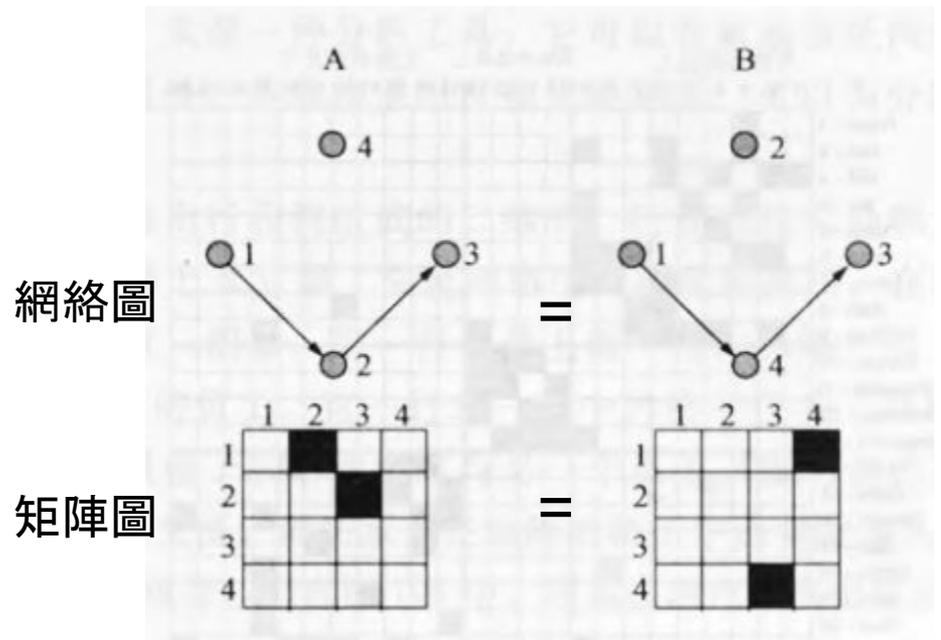
- 行表示發出者，列表示接收者
- 如果是無向網路：全部都由邊(edge)組成，則**矩陣對角線**會是對稱的

罷工交流網絡矩陣圖



網絡重排

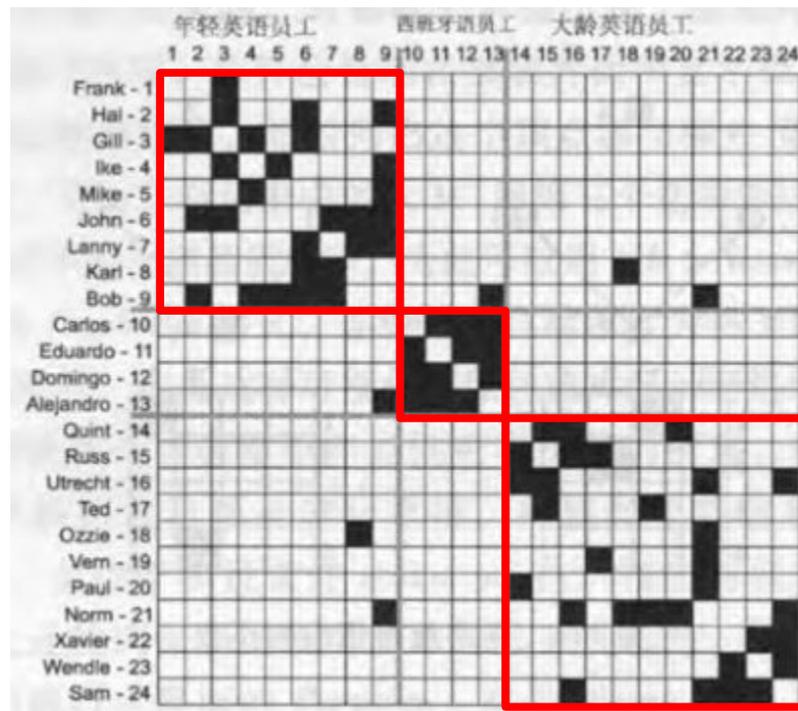
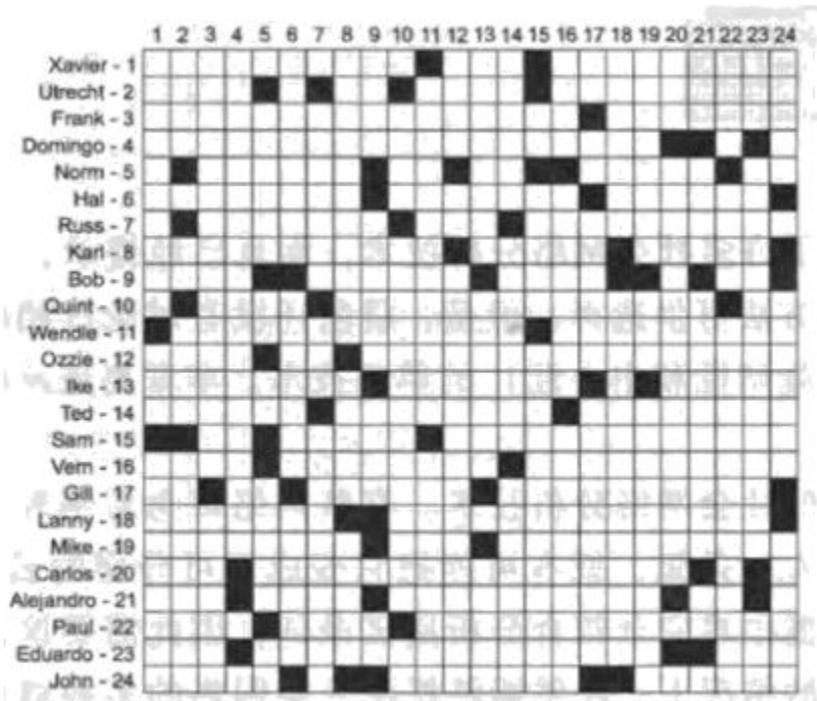
- 矩陣的網絡重排是指重新改變「編號」
- 右圖的A網絡跟B網絡雖然編號不一樣，但網絡結構相同，稱之為「同構」
- 但是看起來似乎對矩陣圖有很大的影響？



網絡重排:效果

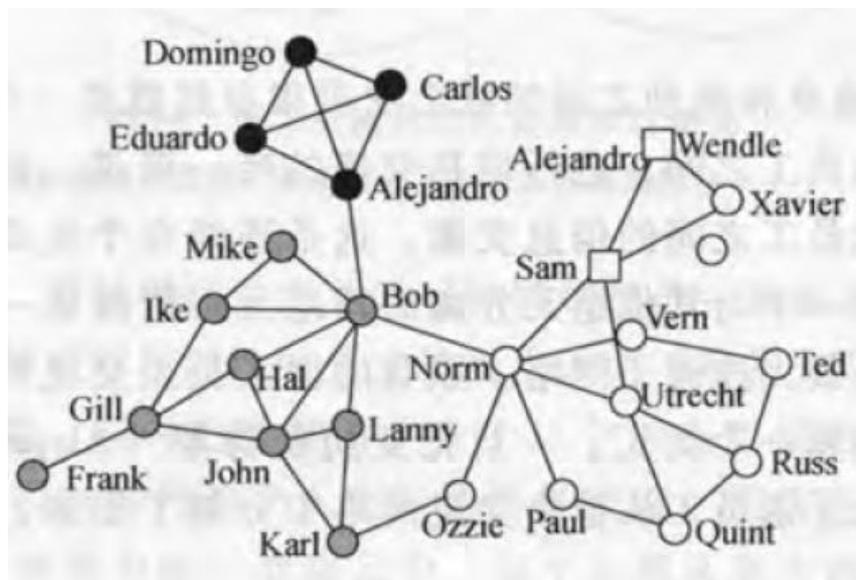
隨機編號的網絡

按照族群重排網絡

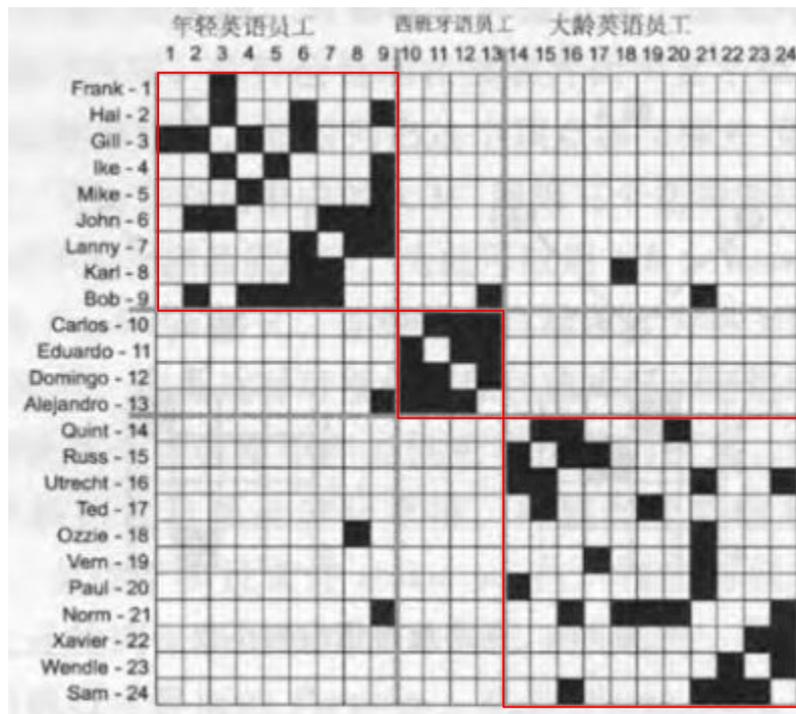


網絡圖跟網路重排的比較

網絡圖



按照族群重排網絡



操作：資料集

strike.paj

<http://l.pulipuli.info/16/pajek>

```
*Network Strike.net
*Vertices 24
  1 "Frank" 0.0410 0.6175 0.5000
ellipse
  2 "Gill" 0.1781 0.5504 0.5000
...

*Arcs
*Edges
  1 2 1
  2 3 1
...

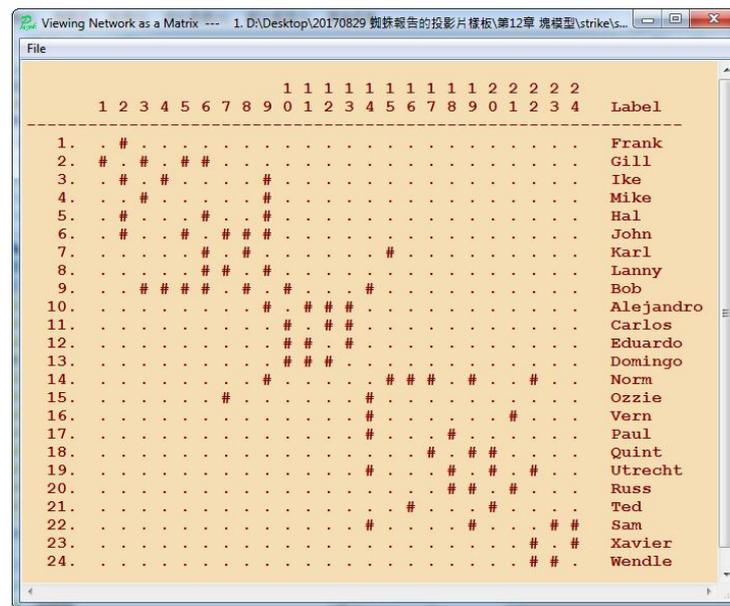
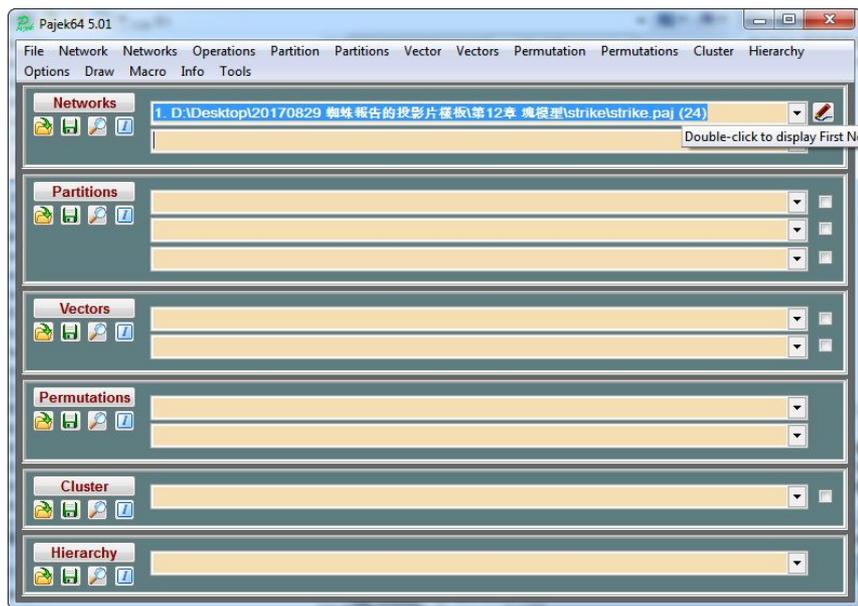
*Partition Strike_groups.clu
*Vertices 24
  2
  2
...
  2
  2
...
```

Strike_groups.clu

```
*Partition Strike_groups.clu
*Vertices 24
  2
  2
  ...
  3
  3
  3
  3
```

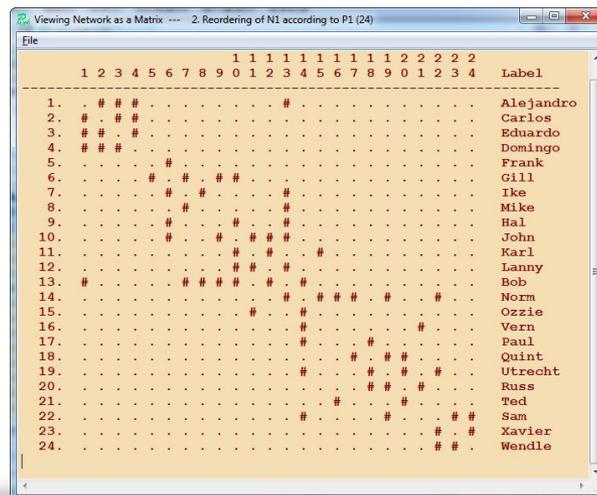
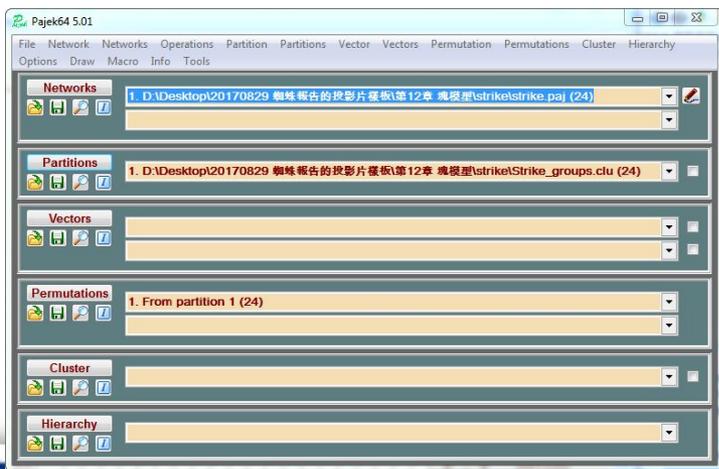
操作：顯示矩陣

1. 開啟Network: strike.paj
2. strike.paj雙擊，選1，開啟矩陣。
3. 這是尚未重排的矩陣



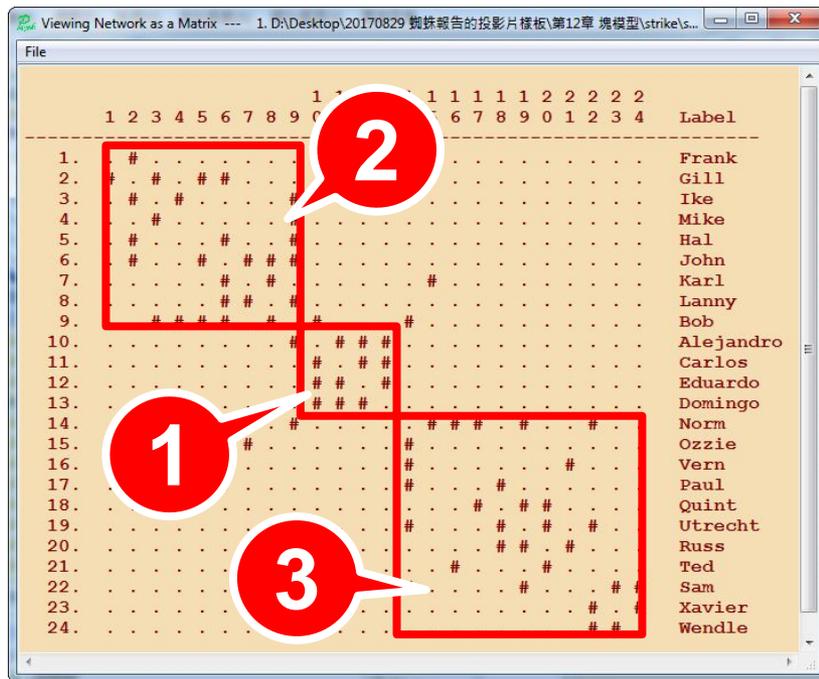
操作：按照分區重排網路

1. 開啟Partitions: Strike_groups.clu
2. Menu > Partion > Make Permutations (產生排列)
下面Permutations會多出排列結果
3. (Pajek舊版) Menu: Operations > Reorder > Network
(Pajek 5新版) Menu: Operations > Network +
Permutation (網路+排列) > Reorder Network
4. 再到strike.paj雙擊，選1，開啟矩陣。

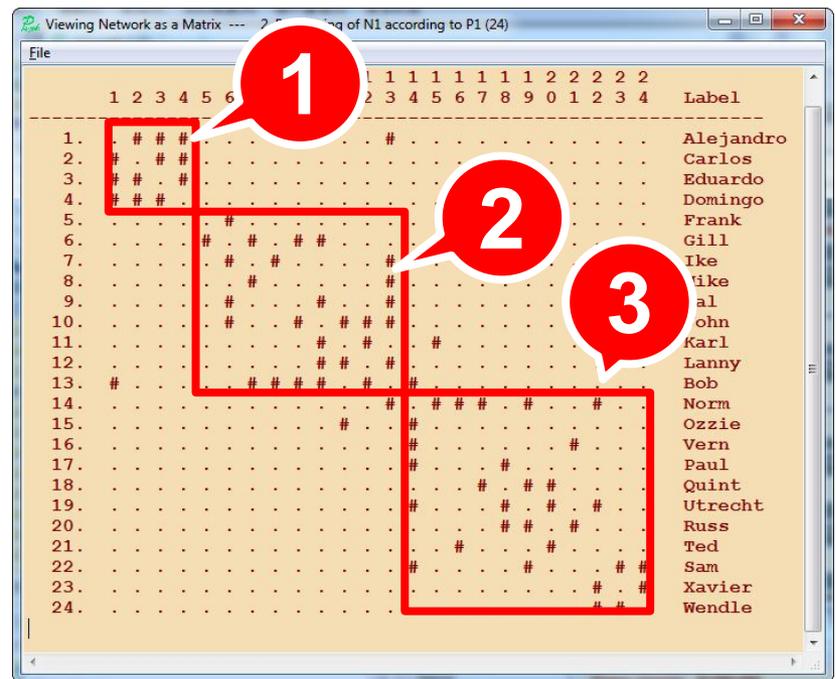


矩陣比較

原始矩陣



按照分區網絡重排矩陣



好像...差不多...

Section 3.

角色集、位置集 和對等性



建立模式=找尋社會關係的模式

名詞定義

- **位置**：特定的關係模式
- **對等性**：不同行為者但是有相似的關係模式 (相似的位置)
- **內等構類別**：將有對等性的行為者歸類為同一個類別 (指一群相似位置的行為者)
- **內等構位置**：內等構類別的位置

社會關係模式的例子

核心——週邊結構

模式說明

- 教師彼此之間互動，成為明顯的凝聚子群

凝聚子群=彼此緊密互動的行為者們

- 學生僅跟教師互動，不太跟學生之間互動，不成為凝聚子群
 - 但是學生們具備對等性

教師和學生間的紐帶

教師: i1, i2

學生: s1, s2, s3

		i1	i2	s1	s2	s3
核心	i1		■	■	■	■
	i2	■		■	■	■
周边	s1	■	■			
	s2	■	■	■		
	s3	■	■			

評估對等性的程度

對等性完全相同=結構對等性

比較兩個內等構類別

教師和學生間的紐帶

「教師」內等構類別

- 教師跟自環(自己對自己)、相互間的紐帶、與其他人的紐帶相同
- 教師可以彼此互換

「學生」內等構類別

- 結構對等性不完整

教師: i1, i2

學生: s1, s2, s3

	i1	i2	s1	s2	s3
核心 i1	1	1	1	1	1
核心 i2	1	1	1	1	1
周边 s1	1	1	1	1	0
周边 s2	1	1	1	1	1
周边 s3	1	1	0	1	1

結構對等性的量化數值

非相似值 Dissimilarity

各頂點之間的非相似值

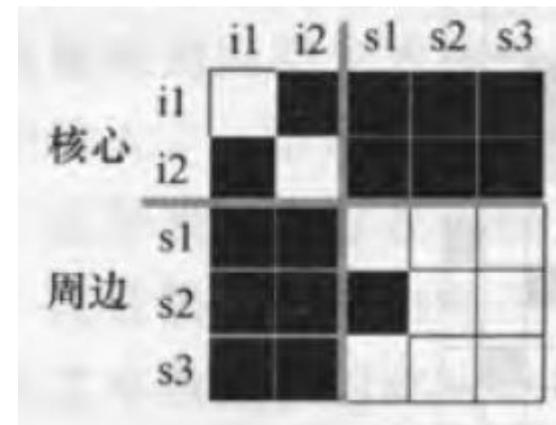
0~1, 數值越低越相似

	i1	i2	s1	s2	s3
i1	0.000	0.000	0.187	0.187	0.250
i2	0.000	0.000	0.187	0.187	0.250
s1	0.187	0.187	0.000	0.125	0.062
s2	0.187	0.187	0.125	0.000	0.062
s3	0.250	0.250	0.062	0.062	0.000

教師和學生間的紐帶

教師: i1, i2

學生: s1, s2, s3



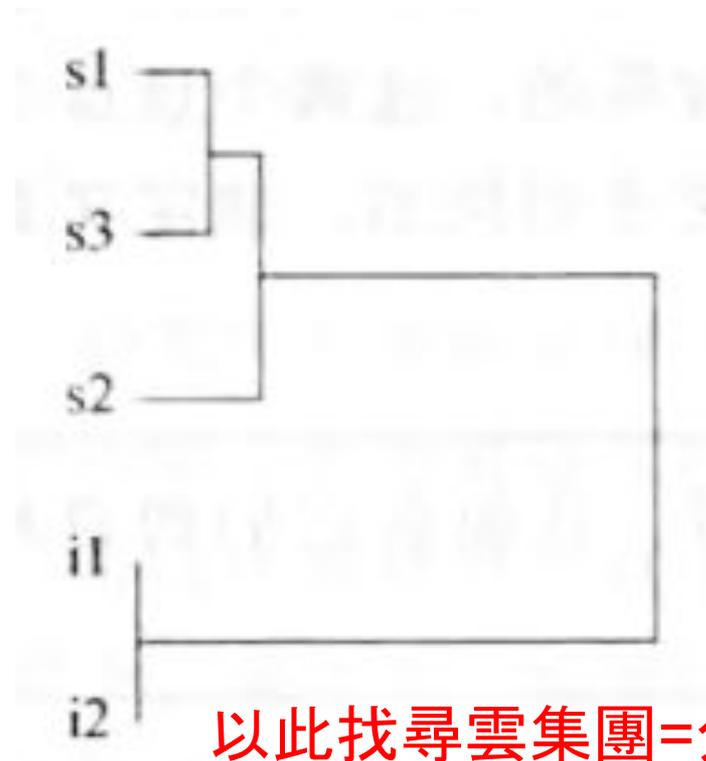
層次聚類演算法

各頂點之間的非相似值

0~1, 數值越低越相似

	i1	i2	s1	s2	s3
i1	0.000	0.000	0.187	0.187	0.250
i2	0.000	0.000	0.187	0.187	0.250
s1	0.187	0.187	0.000	0.125	0.062
s2	0.187	0.187	0.125	0.000	0.062
s3	0.250	0.250	0.062	0.062	0.000

層次聚類演算法樹狀圖



操作：資料集 <http://l.pulipuli.info/16/pajek>

world_trade.paj

- 世界金屬貿易網絡
- 80個國家
- 分為四個體系
 - 核心
 - 強半邊緣國家
 - 弱半邊緣國家
 - 邊緣國家

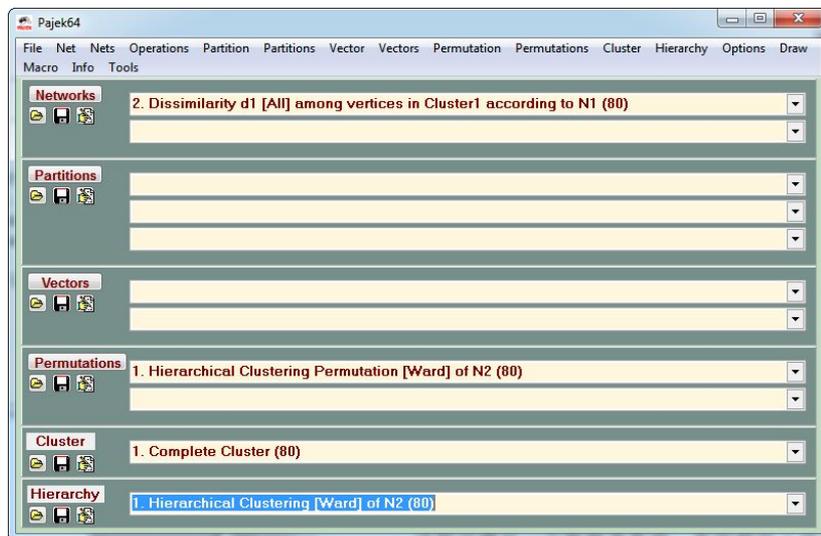


操作：產生層次聚類演算法樹狀圖 (1/2)

1. 開啟Networks: world_trade.paj
2. Menu > Cluster > Create Complete Cluster
3. Menu > Operations > Dissimilarity > Network based > d1 > All
4. 儲存eps文件
5. 雙擊產生的Hierarchy, 顯示層級文件編輯介面

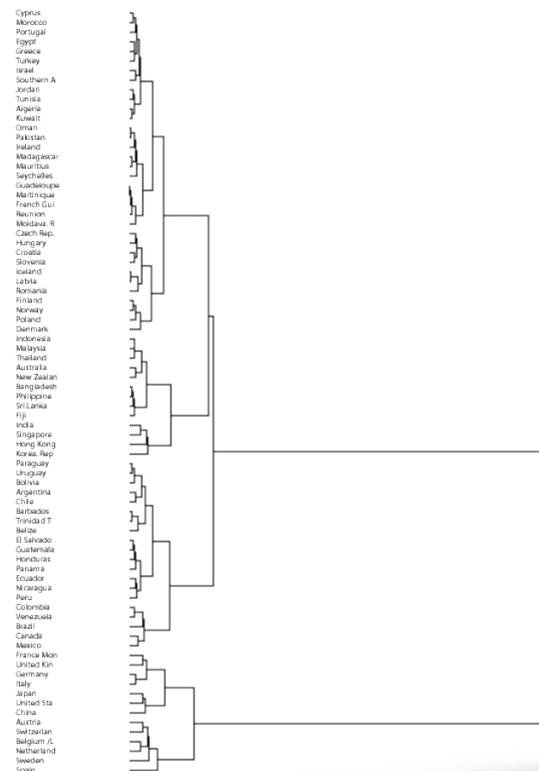
操作：產生層次聚類演算法樹狀圖 (2/2)

操作畫面

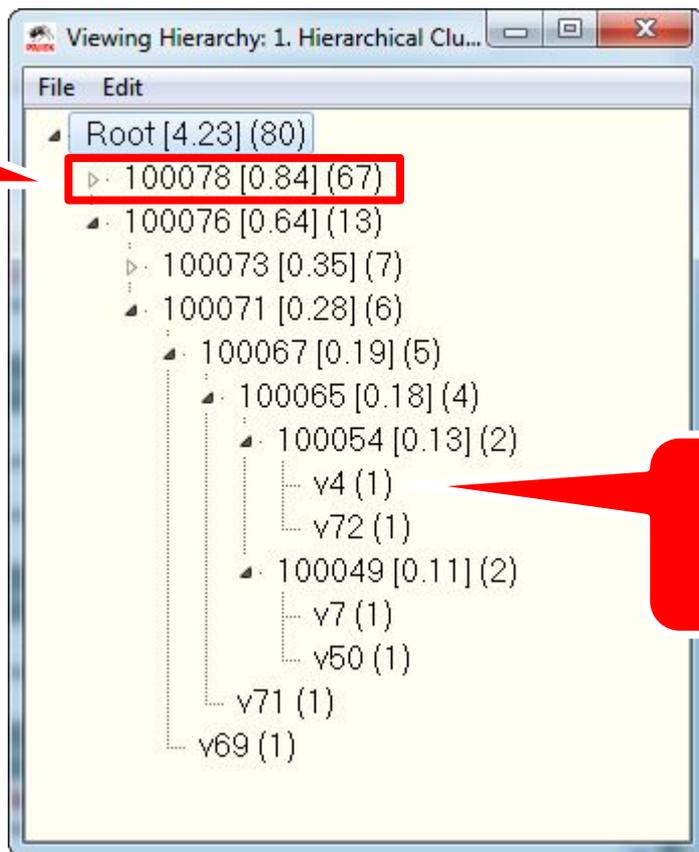


層次聚類演算法樹狀圖

Pajek - Ward [0.00,4.23]



操作：閱讀層級文件編輯器

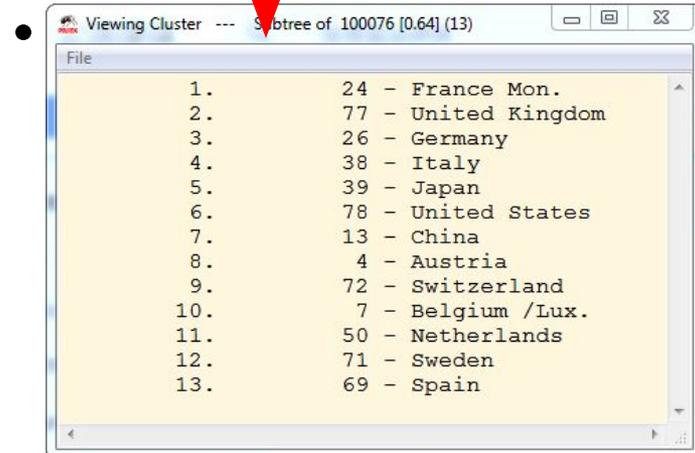
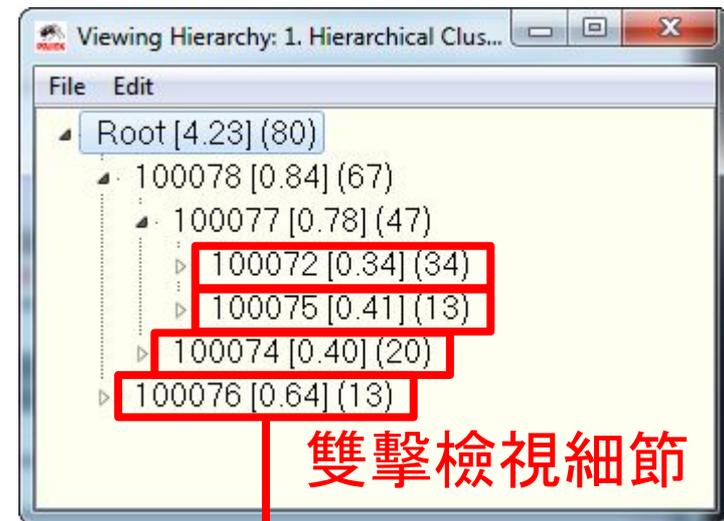


下一層 > 雲集團編號(無意義) [相似性] (頂點個數)

頂點編號 (頂點個數)

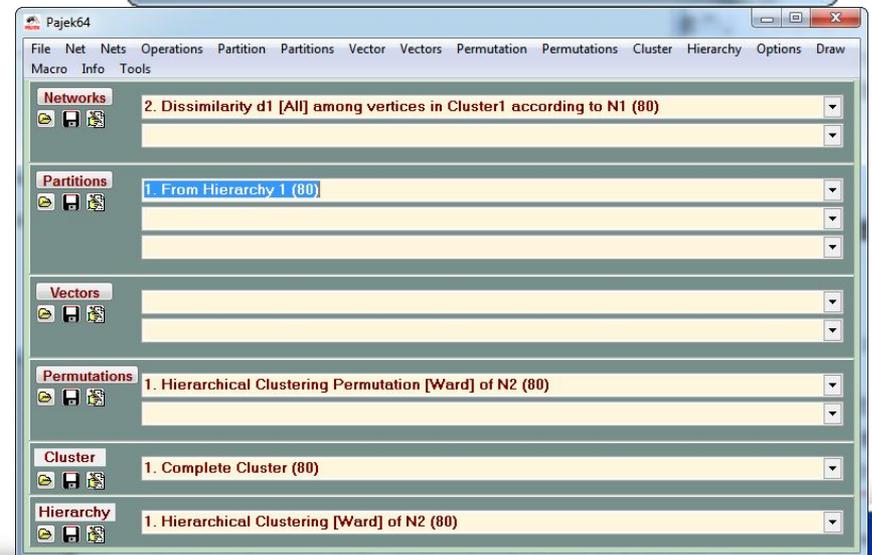
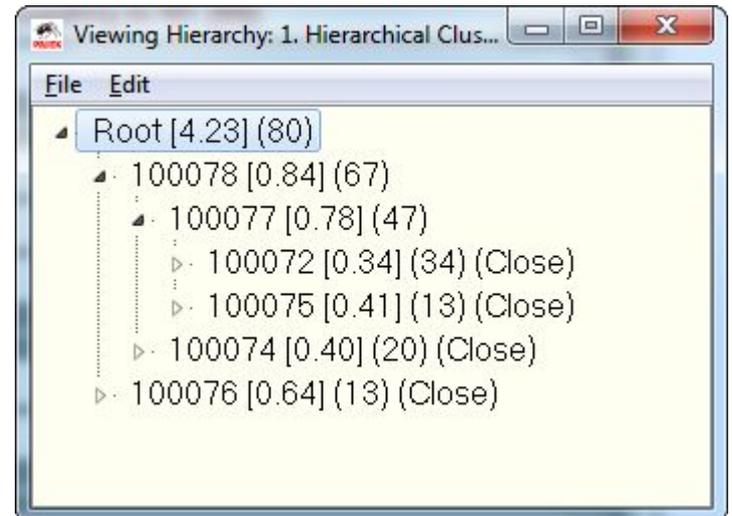
操作：如何決定要找幾群？

1. **決定數量**：世界體系分區文件分為四群，因此我們也找**四個**內等構類別
2. 由非相似值較高的雲集團分裂：
 - a. Root [4.23] 分裂，得到2群
 - b. 100078 [0.84] 分類，得到4群
3. 結果分為4個類別
34, 13, 20, 13



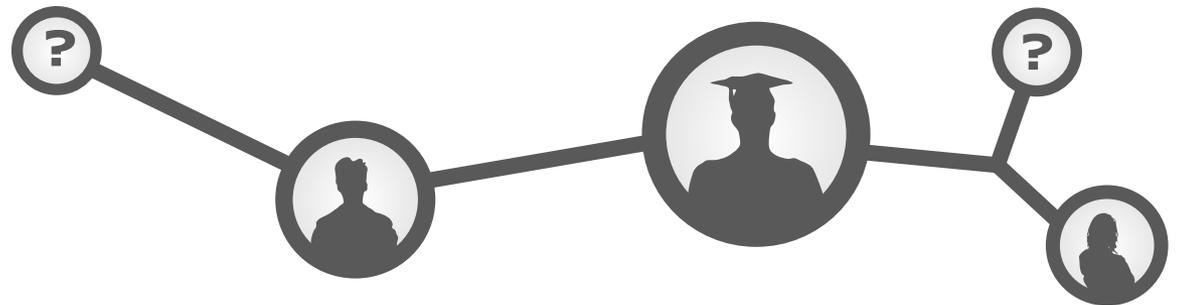
操作：建立分群文件

1. 選擇不分割的雲集團，按 Ctrl+t, 標示(Close)
2. 主介面 Menu > Hierarchy > Make Partition
3. 產生了新的分區文件，可以用於跟原本的世界體系分區文件比較



Section 4-1.

塊模型



名詞定義

塊 Block

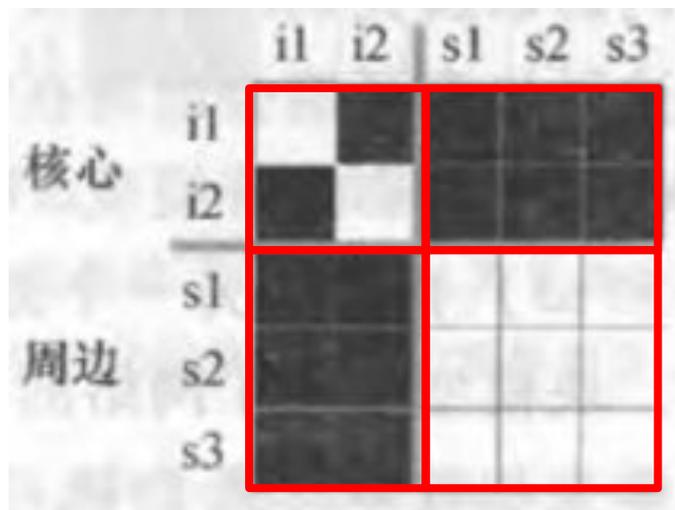
- 指鄰接矩陣中行列交叉區域的格子
- 標示類別與自身的關係
- 或與其他類別的關係
- 可用來簡單地描述整體網絡結構

		i1	i2	s1	s2	s3
核心	i1	白	黑	黑	黑	黑
	i2	黑	白	黑	黑	黑
周边	s1	黑	黑	白	白	白
	s2	黑	黑	白	白	白
	s3	黑	黑	白	白	白

塊的分類

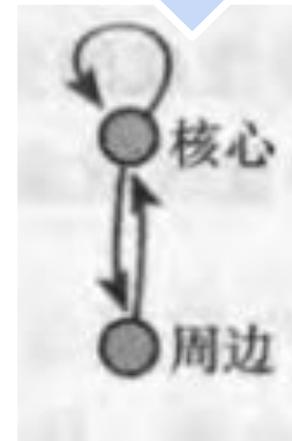
全型 com / 無型 -(null)

鄰接矩陣



縮攏矩陣=影矩陣

	核心	周边
核心	全型 com	com 全型
周边	全型 com	-(null)无型



塊模型

核心——週邊結構

	核心	週邊
核心	全型 com	com 全型
週邊	全型 com	-(null)无型

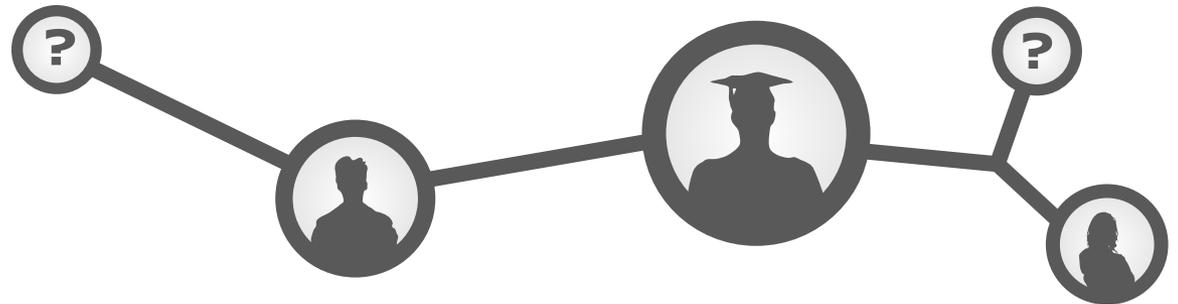
- 核心對核心的塊：全型
- 核心對週邊的塊：全型
- 週邊對週邊的塊：無型

- 塊模型能夠清楚描述整體網絡結構
- 說明每個頂點在這個結構中的所屬位置(類別)

Section 4-2.

如何產生塊模型？

塊建模 Blockmodeling



塊建模的步驟

1. 指定類別數量: 如果假設網絡結構是核心——週邊結構, 那就是兩種類別 (位置)
2. 允許出現的塊模型種類(塊類型): 如果是結構對等性
 - a. 僅允許出現全型塊和無型塊
 - b. 希望沿著對角線出現一個全型塊(核心)和一個無型塊(週邊)
3. 由電腦計算出最佳化分區文件, 完成塊模型
 - a. 啟發式演算法: 嘗試找尋最低誤差分數的分區方式

誤差分數

- 對於理想的「核心——週邊結構」來說
- 因為s1-s2有多餘的紐帶
- 因此誤差分數為1

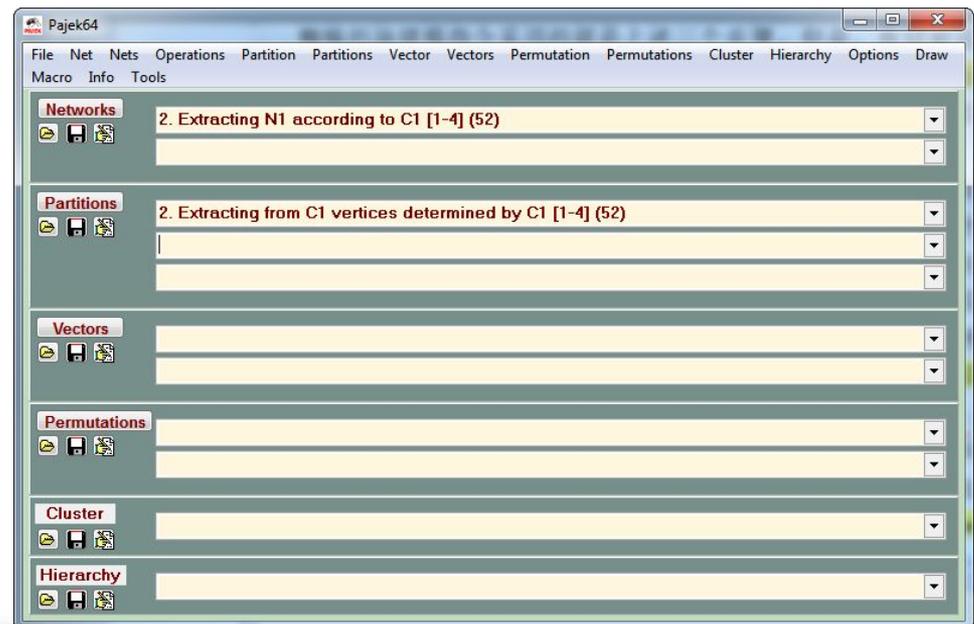
- 誤差分數越低，此結構越符合理想矩陣

		i1	i2	s1	s2	s3
核心	i1	■	■	■	■	■
	i2	■	■	■	■	■
週邊	s1	■	■	□	□	□
	s2	■	■	□	□	□
	s3	■	■	□	□	□

操作：篩選出有明確分區的頂點

<http://l.pulipuli.info/16/pajek>

1. 開啟Networks: world_trade.paj
2. 開啟Partitions: world_trade.clu
3. Menu: Operations > Extract from Network > Partition
4. Select clusters: 1-4 (表示僅取出4個明確的分區，不考慮9999998的遺失值)
5. 頂點被縮減成52個了



操作：用分區文件進行塊建模

1. 勾取選項：Menu: Operations > Blockmodeling中的
 - Restricted Options (限制條件)
 - Short Report (簡短報告)
2. Menu: Operations > Blockmodeling > Optimize Partition
3. 1-Mode BlockModeling: Structural Equivalence (結構對等性): Run
4. 得到最終結果：誤差分數339
 - 1核心類別對其他類別輸出貿易
 - 2 3 4其他國家沒有出口此類產品 (其實有, 只是被視為誤差分數)

Final Image Matrix:

	1	2	3	4
1	com	com	com	com
2	-	-	-	-
3	-	-	-	-
4	-	-	-	-

Final Error Matrix:

	1	2	3	4
1	7	28	32	39
2	36	50	55	28
3	6	13	17	14
4	0	0	1	13

Final error = 339.000

操作：隨機塊建模

1. Menu: Operations > Blockmodeling
> Random Start
2. 1-Mode BlockModeling
 - a. Structural Equivalence
 - b. 2 clusters > 4 clusters
 - c. RUN
3. 得到最終結果：誤差分數281
 - a. 3類別對其他國家輸出 (主要出口)
 - b. 2類別對1~3國家輸出 (次要出口)
 - c. 3,4類別為進口國

Final Image Matrix:

	1	2	3	4
1	-	-	-	-
2	com	com	com	-
3	com	com	com	com
4	-	-	-	-

Final Error Matrix:

	1	2	3	4
1	41	12	10	28
2	39	6	8	25
3	6	2	0	11
4	30	3	9	51

Final error = 281.000

Section 4-3.

更泛用的塊建模

正則對等性的塊建模



Structural Equivalence

Regular Equivalence

結構對等性 \Rightarrow 正則對等性

- 結構對等性：僅看位置，比較嚴苛
 - 位於不同位置的行為者就會被歸入不同組
 - 不同大學任教的教師之間則無法分析，因為他們教導的學生各是不同的行為者
- 正則對等性：看所連的頂點是否屬於同一類別(分區文件)
 - 正則塊：在每一行和每一列都至少含有一條弧 (Arc, 有向邊)
 - 正則塊 包含 全型塊
 - 正則對等性 包含 結構對等性

加入正則塊的塊建模

鄰接矩陣

学生会	主席	部长1	部长2	部长3	部长4	部长5	部长6	部长7	顾问1	顾问2	顾问3
学生会主席								X			
部长1	X										
部长2		X									
部长3	X		X								
部长4				X							
部长5					X						
部长6	X						X		X		
部长7										X	
顾问1											X
顾问2											
顾问3									X		

影矩阵跟误差矩阵

	1	2	3
1	-(null)	-(null)	-(null)
2	com	reg	-(null)
3	-(null)	reg	-(null)

	1	2	3
1	0	1	0
2	2	0	3
3	0	0	2

等級結構

結構說明

1. 學生會主席
2. 部長
3. 顧問

等級結構=低階往高階互動

- 2->1
- 2->2
- 3->2
- 1不跟其他人互動

影矩陣

		影矩陣		
		1	2	3
1		-(null)	-(null)	-(null)
2		com	reg	-(null)
3		-(null)	reg	-(null)

Generalized Blockmodeling

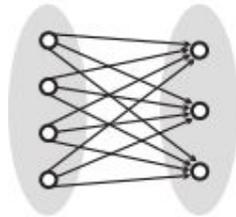
廣義塊建模

- 由研究者自行指定塊類型，以更好地反映網絡特徵
 - 部長對主席：全型塊 (每位部長都該對主席互動)
 - 部長之間：正則塊 (每位部長都應該跟至少一位部長互動)
 - 部長對顧問：無型塊 (部長不需對顧問互動)
- 塊類型的組合：
 - 核心—週邊塊模型
 - 等級塊模型
- 研究者使用塊模型來驗證假設
 - 如果塊模型找出的分區文件，有別於行動者本身的特徵
 - 則表示此塊模型可揭示行動者們隱含的社會角色集

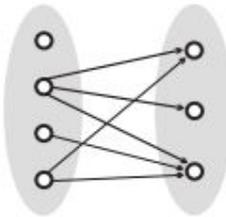
塊類型

0.null
1.complete
2.row-dominant
3.col-dominant
4.regular
5.row-regular
6.col-regular
7.row-function
8.col-function
9.density
10.do not care
11.non-null
12.symmetric
Penalty 1

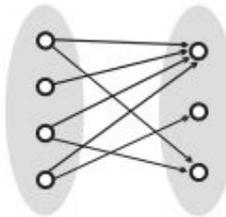
complete



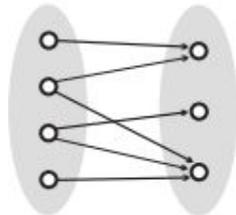
row-dominant



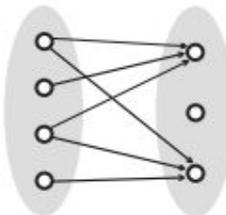
col-dominant



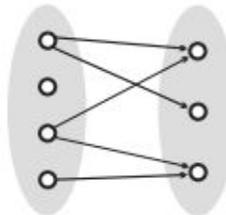
regular



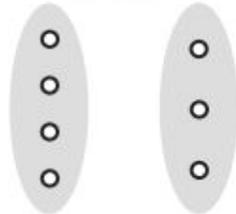
row-regular



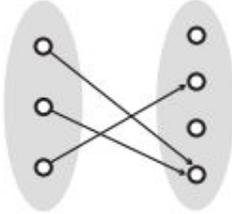
col-regular



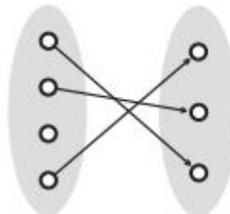
null



row-functional



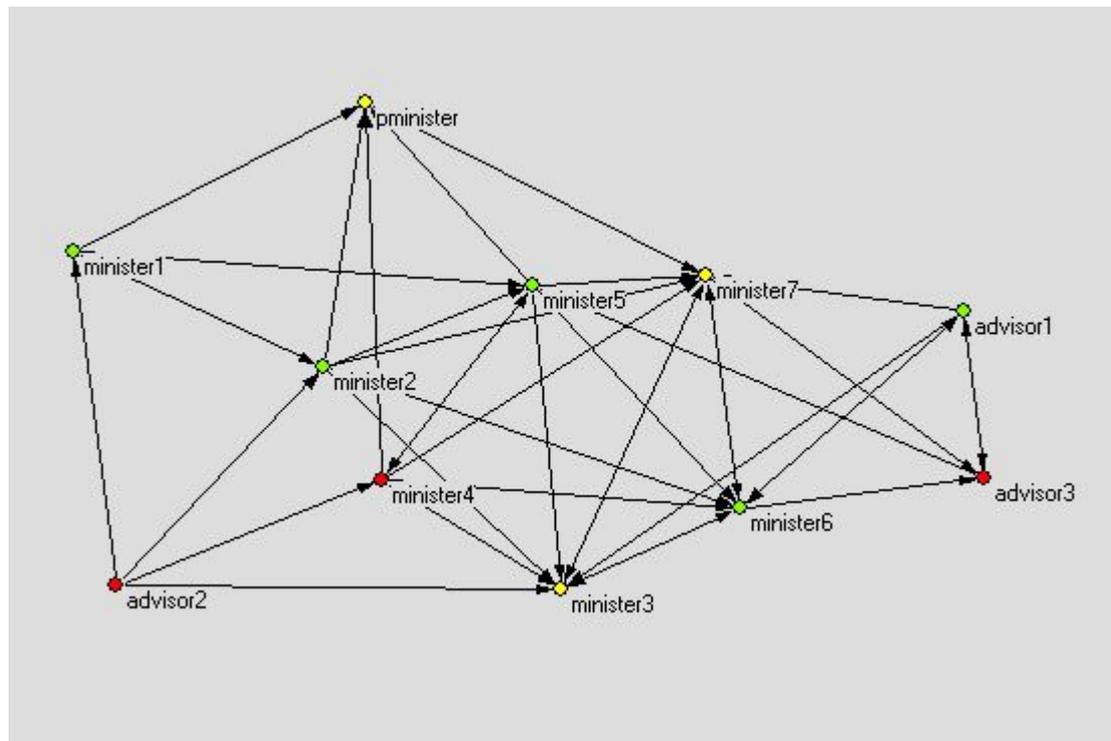
col-functional



操作：資料集

<http://l.pulipuli.info/16/pajek>

- 學生交流網絡 student_government.paj



操作：正則塊建模

1. 開啟Networks:
student_government.paj
2. Menu: Operations >
Blockmodeling > Random Start
3. 1-Mode BlockModeling
 - a. Regular Equivalence
 - b. 2 clusters > 3 clusters
 - c. 100 repetitions > 1000 repetitions
 - d. RUN (每次跑結果都不一樣)

Final Image Matrix (for the first obtained solution):

	1	2	3
1	-	com	-
2	-	-	reg
3	-	-	reg

Final Error Matrix (for the first obtained solution):

	1	2	3
1	0	1	1
2	0	2	0
3	0	3	0

Final error = 7.000 (6 solutions)

比較正則塊建模跟理論模型

正則塊建模

Final Image Matrix (for the first obtained solution):

	1	2	3
1	-	com	-
2	-	-	reg
3	-	-	reg

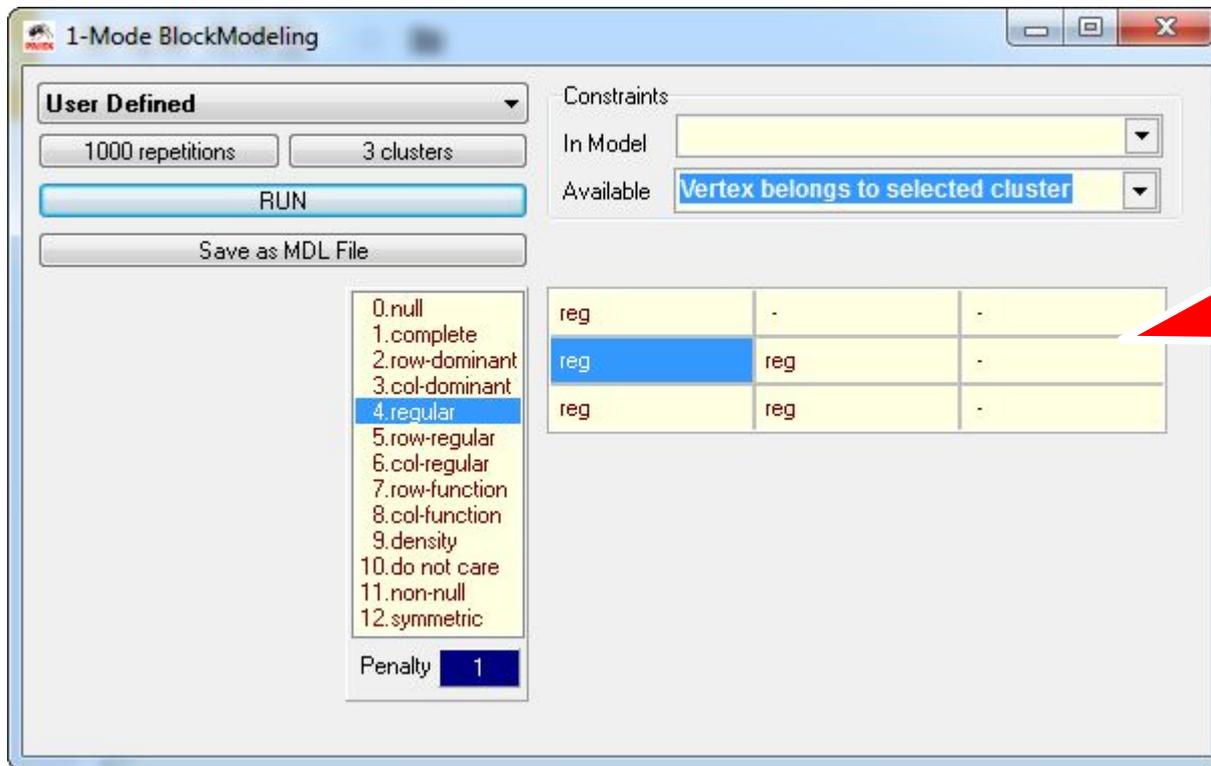
理論模型

影矩阵

	1	2	3
1	-(null)	-(null)	-(null)
2	com	reg	-(null)
3	-(null)	reg	-(null)

操作：廣義塊建模 (1/2)

只要改成「User Defined」即可



指定影矩陣中的塊類型 (按Shift多選)

操作：廣義塊建模 (2/2)

Final Image Matrix (for the first obtained solution):

	1	2	3
1	reg	-	-
2	reg	reg	-
3	reg	reg	-

Final Error Matrix (for the first obtained solution):

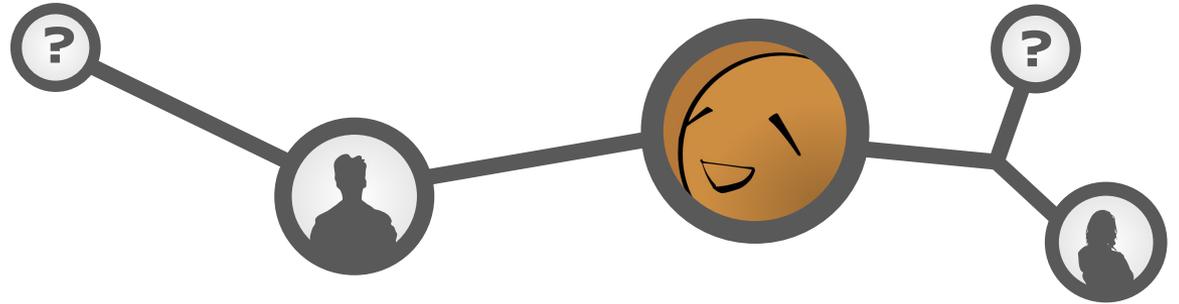
	1	2	3
1	0	3	0
2	0	0	3
3	0	0	1

Final error = 7.000 (2 solutions)

查看分區文件

1. 回到Pajek主介面，雙擊Partitions中BlockModel [User Defined]的文件
2. 得到分區文件
3. 3位行為者
有隱含的角色集
(兩位部長下剋上)

1.	2	-	minister1
2.	1	-	pminister
3.	2	-	minister2
4.	1	-	minister3
5.	3	-	minister4
6.	2	-	minister5
7.	2	-	minister6
8.	1	-	minister7
9.	3	-	advisor1
10.	3	-	advisor2
11.	2	-	advisor3



Thank you
for your attention

布丁布丁吃什麼？

<http://blog.pulipuli.info/>

