3 Network Science

registrazione 1315 (tutorial sui grafi)

Non tutti i tipi di dati hanno una forma tabellare, e quindi esulano dall'universo tidyverse. Vediamo 3 importanti esempi che vedremo nel seguito del corso.

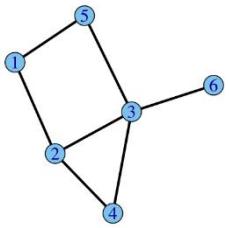
- **Reti**: (i grafi) possono essere rappresentate attraverso un DFrame per i nodi e DFrame per gli archi. Ma tipicamente è usata la *matrice di adiacenza*.
- **Dati gerarchici** : chiamati *dati semi-strutturati* cioè hanno una si una struttura non così rigida da essere incasselata in una tabella. Ad esempio, documento xml, dati html. I tag danno una semistruttura, ma non una tabella (non possiamo mettere una pagina html in una tabella).
- **Testo**: un libro non c'entra niente con una tabella, anche se vedremo un approccio che si basa sulla struttura del DFrame per fare l'analisi del testo.

Dovremo per cui analizzarli con un approccio a se stante rispetto il tidy.

Ripasso sui grafi, tratto da http://users.dimi.uniud.it/~massimo.franceschet/teaching/datascience/network/graphtheory.html

Teora dei grafi (o reti in network science)

Un *grafo* o una *rete*, sono sinonimi, è una collezione di **nodi** (o nodes o legami) ed **archi** (edges o legami). Tipicamente un grafo in matematica èdefinita da una matrice matematica, chiamata matrice di adiacenza. La matrice di adiacenza è una matrice *nxn* e conterrà nell *(i,j)-esima* posizione 1 se c'è un legame (arco) tra l'elemento i-esimo e j-esimo, 0 se non c'è.

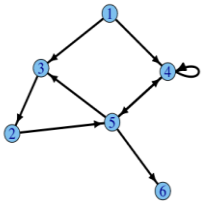


$$\begin{pmatrix} 0 & 1 & 0 & 0 & 1 & 0 \\ 1 & 0 & 1 & 1 & 0 & 0 \\ 0 & 1 & 0 & 1 & 1 & 1 \\ 0 & 1 & 1 & 0 & 0 & 0 \\ 1 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \end{pmatrix}$$

Grafi indiretti

I grafi indiretti sono l'esempio più semplice di grafi. I cerchi sono i nodi e gli archi sono le linee. Di seguito c'è la corrispondenza matrice di adiacenza, che qui è booleana. La matrice di adiacenza di un grafo indiretto è simmetrica, a causa della mancanza di direzione degli archi. Come ad esempio l'amicizia su Facebook che è simmetrica

Grafo diretto



$$\begin{pmatrix} 0 & 0 & 1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 1 & 0 \\ 0 & 0 & 1 & 1 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix}$$

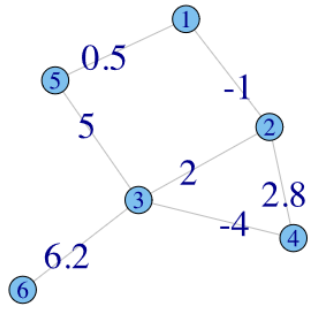
Ad esempio, il follow su Twitter. A differenza dei grafi diretti, nei grafi indiretti ci può essere un **cappio**, cioè una relazione che parte da un nodo ed arrivano allo stesso nodo. La corrispondente matrice di adiacenza non è simmetrica. Sulla diagonale stanno le relazioni che partono ed arrivano sullo steso elemento

Grafi semplici e grafi multipli

Grafi semplici: se c'è al più un solo arco fra 2 nodi. Sono quelli visti finora.
Grafi multipli o **multigrafi**: I grafi multipli sono quelli in cui ci sono molti archi fra coppie di nodi. Ad esempio, le strade fra 2 città, ci possono essere più strade che portano da una città all'altra. In forma tabellare i grafi multipli vengono rappresentati non limitando la cella ai due valori 1 e 0, ma si mette il numero di archi fra i nodi. Ciò ricorda molto i grafi pesati.

Grafi pesati

I grafi pesati sono grafi in cui su ogni arco è definito un numero reale spesso, ma non sempre, positivo detto **peso**. Ad esempio, il peso potrebbe indicare la quantità di informazione che passa mediamente per un router, oppure la durata dell'amicizia fra due persone. Un esempio di peso negativo è quello nelle reti sociali in cui un soggetto può avere un certo grado di amicizia (positivo) o un certo grado di inimicizia (negativo). La rappresentazione avviene tramite matrice di adiacenza definita con celle che hanno come numero il peso o 0. Attenzione che un arco di peso 0 corrisponde all'assenza di relazione.



(salto a Paths and cycles)

Cammino

Un **cammino** su un grafo è una sequenza di nodi connessi da archi. *Cammino non semplice* è un cammino con nodi ripetuti. *Cammino semplice*: cammino con nodi non ripetuti. Ad esempio, (1,4,4,6) è un cammino non semplice.

Ciclo

Un **ciclo** è un cammino con nodo partenza = nodo arrivo. Ad esempio, (2,5,3,2) che è un cammino diverso da (5,3,2,5). L'ordine è importante nelle sequenze, e quindi nei cammini e nei cicli.

Cammino minimo

Cammino minimo su un grafo non pesato è un cammino con il minor numero di archi tra 2 nodi.
Cammino minimo su un grafo pesato è la somma dei pesi sugli archi del cammino.
Mentre la **lunghezza** di un cammino è il numero di archi, in un grafo pesato il peso del cammino è la somma dei pesi sugli archi del cammino, quindi si dirà cammino minimo se il cammino è di peso minimo. Non è detto che un cammino minimo esista. E nemmeno che sia unico.

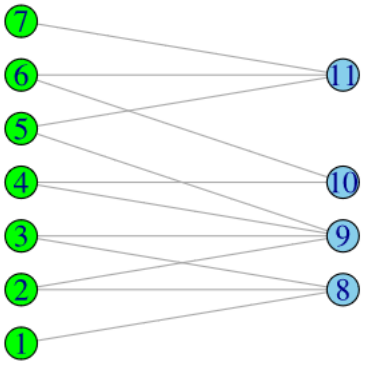
Grafi connessi e fortemente connessi

Un grafo indiretto è **connesso** se esiste un'unica **componente** connessa, ossia se per ogni coppia di nodi esiste un cammino che li unisce (detta altrimenti, se ogni coppia di nodi è raggiungibile da un cammino).
Un grafo diretto si dice **fortemente connesso** (ocio) se per ogni coppia di nodi (i,j) esiste un cammino da i a j ed un altro cammino da j a i. Per esempio, un grafo diretto ciclico.

Componenti connesse e fortemente connesse

Si chiamano **componenti connesse** il massimo insieme dei nodi che sono connessi in un grafo. Detto altrimenti, sono i sottoinsiemi massimali di nodi che sono connessi nel grafo. I sottoinsiemi minimali che sono fortemente connessi sono detti **componenti fortemente connessi**. Un grafo è connesso se ha un'unica componente fortemente connessa.

Salto Bipartite graphs and projections



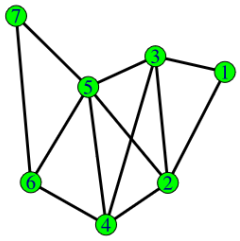
Grafo bipartito

I grafi bipartiti sono dei grafi per i quali si possono suddividere i nodi in due gruppi, tali che gli archi vanno solo da nodi di un tipo a quelli dell'altro. Ad esempio, attori a sinistra e film a destra.

I grafi bipartiti sono degli esempi di grafi indiretti, per i quali solitamente si usa un'altra matrice che è chiamata **matrice di incidenza**: se i due gruppi hanno cardinali n e k, la matrice avrà ampiezza n x k In questo modo le righe saranno gli elementi di un gruppo e le colonne dell'altro, e si usa la medesima idea di porre 1 o 0 nelle celle se esiste o meno la relazione tra due elementi.

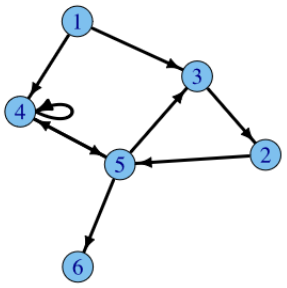
	1	2	3	4	5	6	7
8	1	1	1	0	0	0	0
9	0	1	1	1	1	0	0
10	0	0	0	1	0	1	0
11	0	0	0	0	1	1	1

Grado di un nodo



Il grado di un nodo nel caso di grafo indiretto è il numero di **nodi vicini**, cioè quelli che sono raggiungibili con un arco (un solo passo). Ad esempio, in figura a lato il grado del nodo 3 è 4. Il grado di un nodo è ottenibile sommando la corrispondente riga della matrice di adiacenza (oppure la colonna essendo simmetrica).

La nozione si triplica nel caso di grafi diretti:



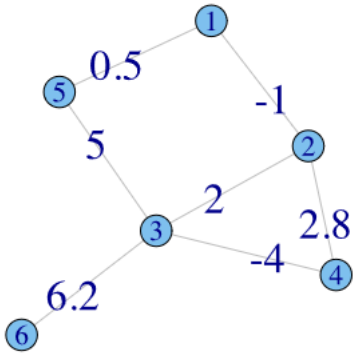
- **Grado uscente**: il numero di nodi (ocio) che posso raggiungere da un nodo. Numero dei successori. Nella tabella si somma la **riga**, *sulla riga vedo i nodi che raggiungo*. In figura a fianco: grado uscente di 4 è 2.
- **Grado entrante**: il numero di archi (ocio) che mi portano al nodo. Numero dei predecessori. Nella tabella si somma la **colonna**, *sulla colonna vedo i nodi che mi raggiungono*. In figura a fianco: grado entrante di 4 è 2.
- **Grado totale**: la somma di grado uscente ed entrante.

Nb. Il cappio è sia grado entrante che grado uscente.

Grado pesato di un nodo

Dato un grafo pesato, il **grado pesato** di un nodo è somma dei pesi degli archi che collegano quel nodo. In figura a lato: il grado pesato di 1 è -0.5.

Nel caso di grafi diretti avremo un grado uscente pesato e un grado entrante pesato.



30 \4\ 2018
Reg 326

Arriviamo così alle reti (tutto in 4 ore circa)

Visione Ted talks Manuel Lima

Abbiamo visto fino ad ora dati che possono essere archiviati in forma tabellare, ma ci sono dati che "sfuggono" a questa griglia: testo, dati semi-strutturati e grafi/reti.

http://users.dimi.uniud.it/~massimo.franceschet/ds/r4ds/syllabus/make/madrid/madrid.html

Network Science

- `learn` Network Science in R - A Tidy Approach
- The igraph package
  - `glance` igraph
  - `learn` Getting started with igraph html + Rmd
- The ggraph package
  - `glance` ggraph
  - `learn` Getting started with ggraph html + Rmd
- The tidygraph package
  - `glance` tidygraph
  - `learn` Getting started with tidygraph
  -

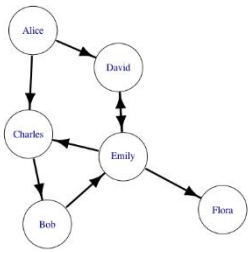
Nel mondo reale possiamo modellare del modo reale in 5 categorie
1. Le reti tecnologiche: internet
2. Le reti sociali
4. Reti di informazione: come il web
5. Reti di citazioni fra articoli scientifici o brevetti
6. Reti biologiche: interazione proteina-proteina, reti alimentari

Si modellano i dati come insieme di punti collegati fra di loro. Quello che conta è la relazione fra i nodi, il punto senza la relazione non è molto significativo.
Tramite il pacchetto ***igraph*** di R analizziamo le reti. E' il corrispettivo di dplyr nel senso che così come dplyr permette di analizzare dataframe, igraph permette di analizzare reti e quindi calcolare misure di centralità, cammini minimi, calcolare gruppi di comunità, distribuzione del grado dei nodi,etc.
Tramite il pacchetto ***ggraph*** di r visualizziamo le reti. E' il corrispettivo di ggplot. Aggiungiamo delle geometrie per archi e nodi e visualizzare le reti.
Non vedremo il pacchetto *tidygraph*, perché siamo solo all'inizio.

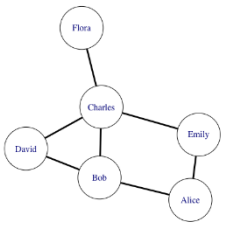
Network science in R - A tidy approach

Le entità sono i nodi, e le relazioni sono gli archi. Eulero ha cominciato ad usarli (7 ponti di Colinsberg)

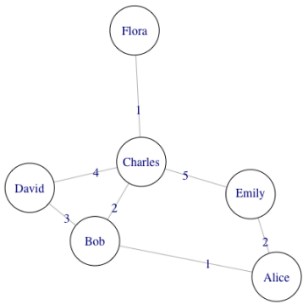
Rete diretta
Se hanno le frecce. Sono le più semplici.



Rete indiretta
Gli archi non hanno direzione

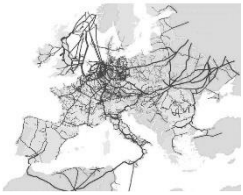


Reti pesate e non pesate
Anche le reti indirette possono essere pesate. Nelle reti pesate gli archi hanno associato un numero reale che ne quantifica la forza della relazione, se è negativo allora la relazione è inversa. Possiamo anche avere pesi nulli, ovvero assenza di relazione.
Definiremo una metrica per misurare la relazione: ad esempio la quantità di bit scambiati fra server, quantità di commenti ai post di un altro. I pesi nulli significano l'assenza di relazione (come indice di correlazione in statistica).



L'unica caratteristica fondamentale che tutte le reti hanno in comune è che sono *webs without a spider*, non c'è un'unità centrale che coordina e controlla la rete. Il tutto viene in modo organizzato autonomamente, gli agenti si auto-organizzano. Ad esempio, posso costruire un sito web senza chiedere permesso a Roma, il web è decentralizzato e autorganizzato. Un altro esempio è lo stormo di uccelli che con 3 regole fa si che si formino le figure che vediamo nei cieli. Le regole sono: non sovrapporsi ai simili, non allontanarsi troppe e se i vicini virano vira anche lui.
La numerosità degli individui assieme al comportamento molto semplice degli agenti può creare una *proprietà emergente* visibile a livello globale. C'è un dunque un passaggio da micro a macro, interessante nelle reti complesse.

La network science può essere considerata come uno spin-off della data science. L'obiettivo è quello di analizzare e visualizzare i dati di rete. Un esempio di output è la rete di gasodotti europea.



3 rappresentazioni in corrispondenza
Una rete ha tre rappresentazioni tutte in corrispondenza:

- grafo
- matrice
- dataframe

Posso passare da Matrici a grafo, da grafo a dataframe, da daframe a matrice. Le tr diverse rappresentazioni sono comode in diversi problemi: per calcolare lo spettro del grafo è comoda la matrice del grafo, per calcolare una misura di centralità (pagerank) è meglio l'oggetto di igraph, per visualizzare il grafo con gpraph o fare un raggruppamento con dplyr è meglio la rappresentazione a DFrame.

Agenda:

- Capitolo 1: **misura di centralità dei nodi**: vedremo 2 modi per determinare quali sono i nodi importanti di una rete

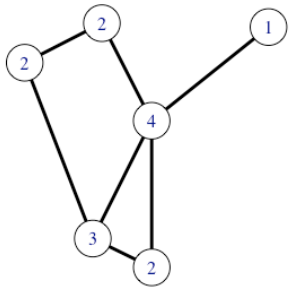
- Capitolo 2: **misura di centralità per gli archi**: vedremo quando le connessioni sono importanti. Vedremo la *teoria della forza dei legami deboli*. Ne vedremo una.
- Capitolo 3: **misura di similarità fra coppie di nodi**. Vedremo dei modi per stabilire quando 2 nodi sono simili, a seconda del pattern che hanno. Ne vedremo una.
- Capitolo 4: metodo per trovare gruppi di nodi simili basati sul **clustering gerarchico**

1) Misure centralità dei nodi

È una funzione che assegna un rating, un numero che denota l'importanza, ad ogni nodo. Avendo ogni nodo della rete un punteggio otteniamo un ranking, **una** classifica di importanza. Con centralità si intende l'importanza di ogni nodo.

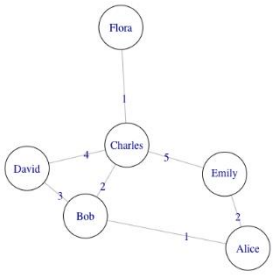
Ad esempio, Google usa un algoritmo chiamato *pagerank*, che è una misura di centralità ricorsiva, per ottenere i risultati in un certo ordine, per prime le pagine con centralità maggiore. Un altro esempio è la valutazione della ricerca dei **ricercatori** in termini di citazioni sui loro articoli. Ancora un altro esempio, su internet potremmo cercare di capire quali sono le macchine più importanti per salvaguardarle da un attacco terroristico o da una rottura di tubi.
La nozione di centralità non è univoca. Ogni misura da l'importanza a un determinato aspetto della rete, è bene non fissarsi su una singola misura ma considerarne tante. Ne vedremo solo 2.

Grado di un nodo (1 di 4)



La prima misura di centralità è il grado di un nodo. Un nodo è importante se ha molte connessioni. Ossia se ha molti archi. È una misura semplice e abbastanza affidabile, ad esempio nelle reti di citazioni si usa per contare le citazioni. Nel caso di reti dirette abbiamo due misure: grado uscente e grado entrante. Ad esempio, in figura il nodo 4 ha grado 4, il nodo 2 ha grado 2.

Nel caso di reti dirette abbiamo due misure: **grado uscente** e **grado entrante**. Dobbiamo quindi specificare il tipo di grado, anche se in realtà conta molto di più il nodo entrante, visto che è meno controllabile (giudizio della comunità è più importante rispetto a quello che uno può dare alla comunità).



Grado pesato (o forza)
Solo nelle reti pesate c'è un'altra misura di centralità chiamata grado pesato. Non si conta il numero di archi, ma si somma il peso degli archi. È la somma dei pesi degli archi incidenti il nodo in esame.

Come esempio di studio useremo la rete terroristica che ha organizzato gli attacchi a Madrid nel 2004. Ogni arco è etichettato con un numero intero da 1 a 4 a seconda del di quante relazioni sono soddisfatte (4 al massimo). Challenges 7: quali sono stati i terroristii più importanti nell'esplosione del treno di Madrid del 2011? file html e file Rmd.

Usiamo i pacchetti dplyr, readr ggplot2, igraph ggraph, visNetwork.

```
g <- graph_from_data_frame(edges, directed = FALSE, vertices = nodes)
print(g)
IGRAPH c629d58 UNW- 64 243 --
+ attr: name (v/c), weight (e/n)
```

```
+ edges from c629d58 (vertex names):
```

'UNW' è acronimo di Undirected (indiretto) Weighted (pesato). Il grafo ha 64 e 243 archi. 'v' = vertici (o nodi) 'e'= edges (o archi). 'name ' è attributo di vertici di tipo 'c' carattere. 'weight' è un attributo di edges ti tipo 'n' numeric.

- `V(g)` restituisce l'insieme dei nodi del grafo --> lista terroristi
- `vcount(g)` restituire il numero dei nodi
- `V(g)` analogamente per gli archi
- `ecount(g)` analogamente per gli archi

Le proprietà sono attributi che posso assegnare al grafo, o ai singoli nodi e ai singoli archi.

- `g$name <- "Madrid network"` #assegno un nome all'attributo name del grado. "nome del grafo $ attributo <- nome da assegnare".
- `V(g)$id <- 1:vcount(g)` # assegno i numeri da 1 a 64
- `E(g)$weight` # vedo i pesi

Alcune visualizzazioni
Il grafo è unico, visualizzabile in molti modi.

- `set_graph_style()` # imposta lo stile a quello di grafo: rimuove gli inutili assi cartesiani
- `ggraph(g, layout = "with_kk") + geom_edge_link(aes(alpha = weight)) + geom_node_point()` # imposto layout basato su algoritmo Kamada-Kawai tale che nodi con più archi saranno più vicini nel grafico. C'è poi una geometrica che disegna gli archi, qui in proporzione al peso (più scuri). L'ultima geometria disegna i nodi come punti
- `+ geom_node_text(aes(label = id), repel=TRUE)` # aggiunge l'etichetta id al nodo. Repel = T non fa sovrapporre le etichette
- `ggraph(g, layout = "in_circle"` # dispone in cerchio i nodi
- `ggraph(g, layout = "grid "` # dispone i nodi sulla griglia

Registrazione 327

Closness

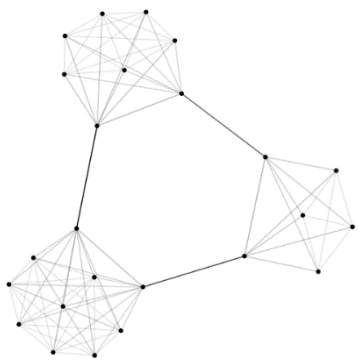
Misura quanto un nodo è vicino agli altri nodi. La closness di un nodo è il reciproco della distanza media di un nodo dagli altri. Dove con distanza media si intende la distanza geodetica, ossia il numero di passi che deve fare un nodo per raggiungere un altro nodo (lunghezza del cammino minimo).

Registrazione 328

2) Misura centralità per archi

Betweenness

È una misura di centralità che useremo sugli archi, ma che si può usare anche seui nodi. Misura il numero di cammini di altri nodi passano sull'arco. Tanti più cammini passano, tanto più importante sarà quell'arco. Tanta più forza avrà.



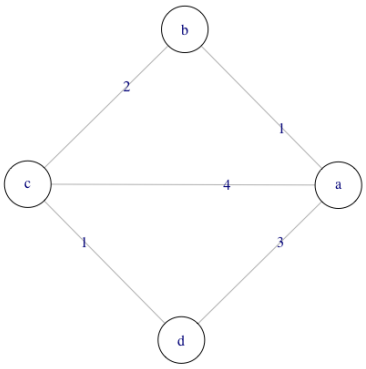
In figura: ci sono 3 archi (grassetto) che connetto le 3 comunità di nodi. Ogni cammino che connette due comunità deve passare per i 3 archi. Essi avendo tanti cammini in un certo senso controllano l'informazione. Se li rimuovessimo le 3 comunità sarebbero isolate.

È il numero di cammini minimi, che connettono un'altra coppia di nodi, che passano per un fissato arco. Conta quante volte un arco è attraversato da cammini minimi. Se ho ben capito: fissata una coppia di nodi considero la frazione di cammini minimi che passano su un arco fissato rispetto alla frazione di cammini minimi che collegano I due nodi.

Lo stesso concetto vale anche per I nodi. Invece di selezionare il numero di cammini minimi che passano per un arco, seleziono il numero di cammini minimi che passano per un nodo.

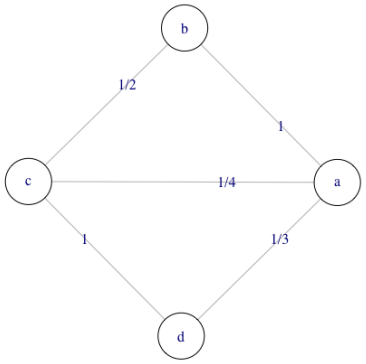
Betweenness in reti pesate

Il peso di un cammino è la sommatoria dei pesi degli archi del cammino. Il cammino minimo fra due nodi è il cammino con peso minimo. Ad esempio, in figura a fianco il cammino minimo fra a e c è (a,b,c) con peso 3. Nota che non ha il numero minimo di archi e ciò dà problemi di incoerenza. Perciò vd poi.



Prima di calcolare la betweenness in archi pesati dobbiamo invertire il peso degli archi. Così archi con peso basso sono preferiti rispetto ad archi con peso alto, nell'ottica di una minimizzazione\ Betweenness. (index1)

In sintesi: dopo l'inversione dei pesi, archi con peso basso corrispondono a nodi molto vicini e che sono preferiti rispetto ad archi di peso alto che corrispondono a nodi distanti.



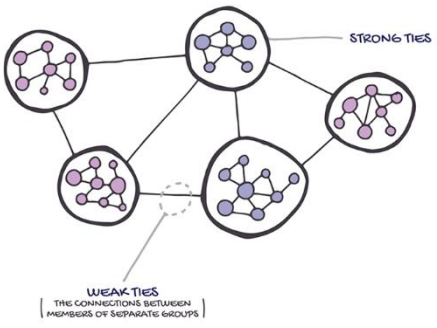
La misura di centralità appena introdotta permette di verificare la teoria sociologica Teoria sociologica de *la forza dei legami deboli* di Martin Granovetter (anni '70).

Ties = archi=legami.

Granovetter dice che se prendiamo una rete sociale si formano delle comunità molto coese connesse da archi sporadici.

Gli archi dentro le comunità sono chiamati archi forti (*strong ties*).

Gli archi fra le comunità sono chiamati archi deboli (*weak ties*), che sono relazioni fra membri di comunità diverse.



Da un punto di vista sociologico un **legame debole** è una relazione fra membri di comunità diverse. I legami deboli non hanno bisogno di grosso accudimento per rimanere vive. Ad esempio, un amico di penna americano. Un **legame forte** è una relazione fra persone che condividono tante esperienze, e necessitano per rimanere attive di molta cura. Ad esempio, i legami fra amici intimi o compagni di studi. Questi due legami hanno un'importanza molto differente. La ricerca dimostrò che i legami forti tendono a generare idee dominanti e stagnanti, viceversa i legami deboli tendono a favorire la diversità delle idee e un modo diverso di pensare. Il sociologo, con molta sorpresa, dimostrò in un articolo che i legami più importanti sono i legami deboli, in quanto danno la possibilità di accedere a mondi nuovi e modi di pensare diversi da quelli della propria comunità. Un esempio è dato dal fatto che l'amico di penna ha un intero mondo: di amici, relazioni, fidanzata, lavoro.

Ma la teoria vale anche per la rete terroristica del nostro esempio? Definiamo legame debole quel legame con un peso pari a 1. Verifichiamo se i legami che hanno una connessine peso pari a 1 sono importanti o frequenti rispetto agli altri legami.

Continua da registrazione 329
Vai su Rmd

Calcoliamo la betweenness

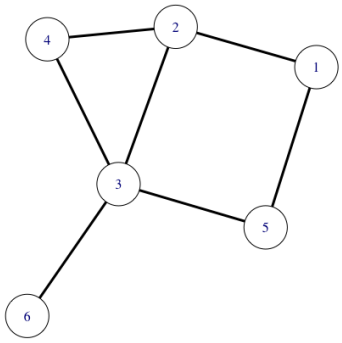
- `dist_weight <- 1 / E(g)$weight` # imposta l
- `btw <- edge_betweenness(g, weights = dist_weight)` # con la funzione edge_betweenness calcoliamo la betweenness, pesata attraverso weights. Lo fa solo sui cammini minimi

- `ggraph(g, layout = "with_kk") + geom_edge_link(aes(filter = weakness), alpha = 0.5) +`
- `geom_node_point()` # posso usare l'estetica filter per filtrare i legami deboli, infatti weakness è booleana T se il legame è debole.

3) Misura di similarità fra coppie di nodi

Quando due nodo sono simili fra loro? Lo scopo è capire se 2 nodi sono simili. Trovare nodi simili è un problema interessante, ad esempio è utile trovare pagine web simili, Amazon cerca spesso clienti simili. Il problema della similarità è un problema binario, abbiamo due nodi e vogliamo capire quanto siano simili fra di loro. Diremo che due nodi sono simili se il pattern di connessione è simile. Dati due nodi S e T e un terzo nodo X, possiamo fare considerazioni di questo genere: se sono entrambi in relazione con X allora c'è più similarità, se S è in relazione con X e T no allora la similarità sarà maggiore.

Più sono simili le righe della matrice di adiacenza più simili sono i nodi (di riga). La similarità può essere misurata tramite il coefficiente di correlazione di Pearson [-1,1]. Come



	1	2	3	4	5	6
1	0	1	0	0	1	0
2	1	0	1	1	0	0
3	0	1	0	1	1	1
4	0	1	1	0	0	0
5	1	0	1	0	0	0
6	0	0	1	0	0	0

Diremo che 2 nodi sono simili se essi hanno lo stesso pattern di connessione di nodi vicini. Guardo alle righe per vedere se 2 nodi sono simili. Ma non sommo! Faccio indice di pearson. Diremo che 2 nodi sono simili se hanno coeff di correlazione alto, (in posizione 1,1 = 1,2 etc) vicino ad uno. Sono invarianti se coefficiente tende a zero.

- `A <- as_adjacency_matrix(g, attr = "weight", sparse = FALSE, names = FALSE)` # passaggio da grafo a matrice di adiacenza: uso la funzione as_adjacency_matrix che come oggetto vuole un grafo, attributo il pes, sparse dice se voglio una matrice sparsa
- `rowSums(A)` # fa somma righe, quindi calcola la forza\ grado pesato.

- `B = A > 0, rowSums(B)` # se non voglio il grado trasformo la matrice in booleani e poi sommo le righe, ottenendo il grado semplice
- `S <- cor(A), diag(S) = 0` # colcolate le correlazioni metto a 0 la diagonale (è 1)

Poiché la massima similarità è raggiunta da nodi con grado molto basso, allora filtro gli archi al di sora di un grado minimo

- `We` # (dovrei scrivere altri comandi, visto che li chiede nel compito scritto)

La matrice di adiacenza non è simmetrica per gradi diretti, lo è per grafi indiretti.
La matrice di adiacenza per grafi pesati riporta il numero del peso.

Vai su datacamp
Registrazione 330

(ripasso lungo per assenti)

07/05/2018
registrazione 337

Misure di similarità:

- Correlazione fra coppie: misura correlazione fra coppie di riga e colonna (eliminare?)

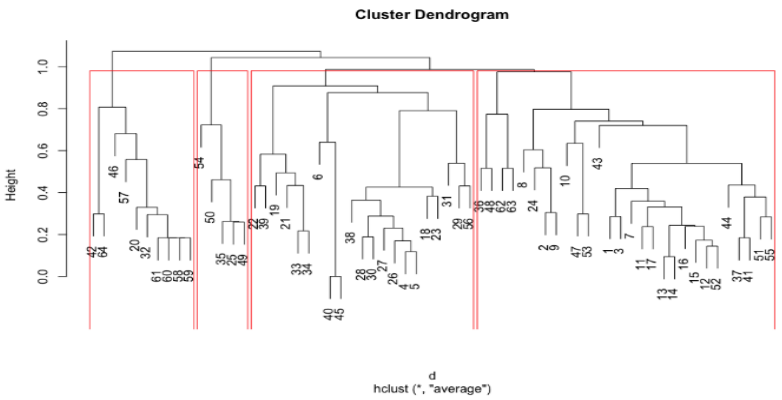
4) Clustering gerarchico

Metodo che utilizza come distanza fra due nodi l'inverso della similarità. Infatti, se 2 nodi sono simili hanno una similarità elevata, allora la loro distanza è molto bassa. E viceversa. Come misura di similarità usiamo quella calcolata al punto 3.

Faccio tante classi quanti sono i nodi, poi ad ogni passo unisco due gruppi alla volta. Calcolo la distanza minima fra tutti i nodi, che diventa il criterio per unire i gruppi in uno step. Unisco i gruppi la cui distanza è ?maggiore? della distanza minima. La distanza minima cambierà aumentando, ad ogni turno. Avrò tante soluzioni. Otterremo così un *dendrogramma* dove i nodi dell'albero (ocio) sono le unioni dei gruppi e le foglie sono i nodi del grafo iniziale. Non sapendo quanti gruppi è meglio avere, devo decidere a che altezza fra 0 e 1. potare l'albero (cut): taglio tracciando linea orizzontale, ogni volta che incrocia un alinea verticale è determinato un gruppo. Nell'esempio sono 4 gruppi. L'analista scelta la soluzione intermedia fra n e 1 gruppi.

La distanza è il complemento a 1 della similarità.

C'è un ultimo problema da affrontare. Non sappiamo cosa sia la similarità fra gruppi.



Similarità fra gruppi

- **Single-linkage**: la similarità fra tra due gruppi è definita come il massimo delle similarità tra i nodi di diversi gruppi.
- **Complete-linkage**: similarità fra gruppi è definita come il minimo delle similarità fra i gruppi di nodi
- **Average-linkage**: dove la similarità tra due gruppi è la media delle similarità tra i nodi di diversi gruppi (più frequente)

Il metodo di clustering gerarchico è il seguente:

- Valutare le misure di similarità per tutte le coppie di nodi
- Assegnare ciascun nodo a un gruppo a sé stante.
- Trova la coppia di gruppi con la similarità più elevata (ossia distanza minore) e uniscili in un unico gruppo.
- Calcola la somiglianza tra il nuovo gruppo composito e tutti gli altri.
- Ripetere i passaggi 3 e 4 fino a quando tutti i nodi sono stati uniti in un singolo gruppo.

L'algoritmo non sveglie il numero k di gruppi, è scelto dall'analista.

- D <- 1-S, d <- as.dist(D) # calcolo matrice di distanza come complemento a 1 della similarità
- cc <- hclust(d, method = "average"), plot(cc) # hclust calcola il clustering gerarchico e visualizzo
- cls <- cutree(cc, k = 4) # taglio il dendrogramma t.c. formi 4 gruppi
- nodes <- mutate(nodes, cluster = cls) # aggiungo poi tale classificazione al DFframe, che conterrà per ogni nodo la sua appartenza ad uno dei 4 gruppi.
- filter(nodes, cluster == 1) %>% select(name)# posso filtrare al solito con dplyr i nodi
- V(g)$cluster <- nodes$cluster # aggiungo informazione alla rete attraverso l'attributo cluster dei nodi
- ggraph(g, layout = "with_kk") +
  geom_edge_link(aes(alpha = weight), show.legend=FALSE) +
  geom_node_point(aes(color = factor(cluster))) +
  labs(color = "cluster") # nella geometria geom_node_point indico il colore

Visualizzazione interattiva delle reti con Visnetwork

Permette di zoomare, selezionare e spostare i nodi: che fungono da elastico, c'è una forza di repulsioni sui nodi e di attrazione sugli archi.

VisNetwork è un pacchetto eccellente per realizzare bellissime visualizzazioni interattive delle reti. Posso visualizzare una rete personalizzata o una rete creata con il pacchetto igraph. Posso anche dare alla rete diversi layout. Ancora più importante, le reti che disegni sono vivi! Ciò significa che puoi interagire con i nodi e i bordi o con l'intero grafico. Ad esempio, puoi fare clic e evidenziare un nodo, spostarti intorno a un nodo o spostare e zoomare l'intera rete. Godrai il piacere dell'interazione durante gli esercizi. Posso selezionare un nodo tramite il suo identificatore, nel nostro caso il nome dei terroristi. Supponiamo che tu conosca il nome del terrorista e desideri localizzarlo sulla rete, insieme ai suoi vicini. È possibile suddividere i nodi in gruppi, ad esempio, in cluster di nodi simili, come abbiamo fatto nella nostra applicazione, e evidenziare tutti i nodi in un particolare gruppo.

Esercizio in classe sui delfini

Reg 339

http://users.dimi.uniud.it/~massimo.franceschet/ds/r4ds/syllabus/make/dolphin/dolphin.html

David Lusseau, ricercatore presso l'Università di Aberdeen, osservò il gruppo di delfini di Doubtful Sound. Ogni volta che una scuola di delfini si incontrava nel fiordo tra il 1995 e il 2001, ogni membro adulto della

scuola veniva fotografato e identificato da segni naturali sulla pinna dorsale. Questa informazione è stata utilizzata per determinare quanto spesso due individui sono stati visti insieme. Leggi la storia completa. download

Csv con nome e sesso delfini
Csv con legami fra delfini

Domande:

- Quali sono i delfini più socievoli?
- I delfini femmine sono più socievoli dei delfini maschi?
- Qual è il tipo di rapporto tra i delfini (amicizia o sessuale)?

Reg 340

Soluzione:

- Quali sono i delfini più sociali? Bastava calcolare il grado: sono Grin sn4, Topless, scabs, trigger.
- I delfini femmine sono più socievoli dei delfini maschi? C'è una differenza ma non significativa
- Qual è il tipo di rapporto tra i delfini (amicizia o sessuale)? Più amicale che sessuale.

Comandi soluzione

- r # (forse dovrei inserirne alcuni)

Assortatitività: cerchio = femmina, quadrato = maschio L R
1 cerchio -- quadrato 3
1 cerchio --- quadrato 2
poi
usare correlazione, oppure contare i rapporti MM + FF vs MF
Comodo usare vettori booleani:
SexU, sexF, sexM