Riassunto di network science

# Cos’è la network science?

La network science è del come diverse unità interagiscono tra di loro utilizzando le reti come mezzo. Esistono infatti dei metodi che, sfruttando proprio le reti, permettono di comprendere l’evoluzione di un dato fenomeno o di trarre conclusioni in base a esso.

## Cos’è una rete?

Le reti sono sistemi complessi composti da tanti elementi che interagiscono tra di loro, essi sono responsabili di un fenomeno collettivo che dobbiamo studiare la cui emergenza non è direttamente correlata a esso.

Questi sistemi possono essere:

* lineare o non lineari;
* eterogenei oppure omogenei.

Le reti sono ovunque:

* dalle reti sociali, in cui gli attori sono i singoli individui e gli archi le relazioni tra essi;
* nei sistemi di informazione come le pagine web connesse tramite link;
* l’interazione tra proteine;
* eccetera.

## Reti semplici

Possiamo rappresentare le reti attraverso dei grafi, le più semplici sono:

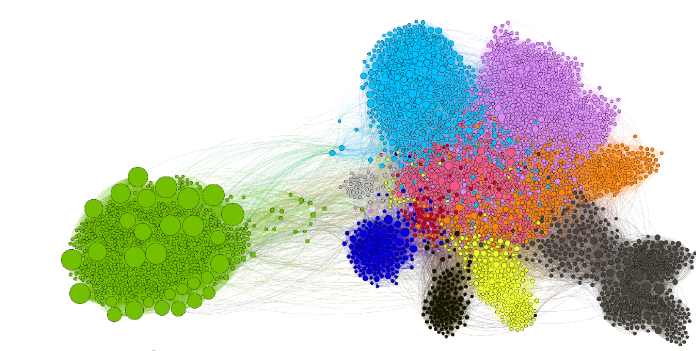
* le reti a stella, in cui i nodi sono connessi tutti a un altro nodo che funge da hub per la comunicazione;
* le reti ad anello, qui i nodi sono connessi in modo circolare;
* le reti a griglia, in cui i nodi sono disposti a griglia.

Abbiamo quindi bisogno di un linguaggio e un framework per descrivere queste reti.

## In generale…

Non esiste apparentemente un modo semplice per descrivere le reti (specie se complesse). Tuttavia possiamo individuare delle regolarità, infatti alcuni individui possono risultare più centrali rispetto ad altri, come possiamo vedere nella foto a destra.

E’ presente della casualità in tutto ciò?

Considerando internet come esempio (visibile nella foto a destra), possiamo notare che, nella sua caoticità, è presente dell’ordine. Infatti ci sono delle sottoreti che formano una struttura gerarchica, abbiamo quindi bisogno di un modo per descrivere queste regolarità.

La visualizzazione delle reti permette di capirne la complessità e la struttura, ma non molto riguardo alle motivazioni che ci stanno dietro. Inoltre è difficile riassumere l’intera rete in maniera semplice.

## Teoria delle reti (o dei grafi)

La teoria dei grafi ci viene in aiuto dal momento che, coi concetti di distanza, grado, cluster, eccetera, possiamo estrapolare più informazioni rispetto alla sola rappresentazione grafica.

La costruzione di queste reti può avvenire attraverso i dati, da qui gli approcci di tipo data-driven. Questi modelli permettono di fare previsioni ma solitamente non sono generalizzabili, ciò per via del fatto che ogni rete è differente.

## Livello individuale e di popolazione

Considerando un grande numero di persone densamente interconnesse, a volte queste rispondo in maniera complessa, ciò può essere osservabile solo col livello di popolazione, come ad esempio le idee virali che si espandono a macchia d’olio.

Esiste inoltre un livello individuale in cui un’informazione o una persona diventa più importante rispetto alle altre, ciò tuttavia a volte non accade.

## Dinamiche di rete

Dal momento che nelle grandi popolazione c’è l’emergenza di nuove idee, opinioni, eccetera, come possiamo stabilire queste pratiche sociali e come gli individui possono influenzare gli altri? A fronte di ciò esistono le cosiddette dinamiche di rete, le quali descrivono come la popolazione e gli individui. Gli individui infatti hanno la tendenza a conformarsi e perciò quella di adottare lo stesso comportamento dei suoi “vicini”, ciò prende il nome di comportamento a cascata.

# Perchè i grafi?

I grafi sono un modello matematico che aiuta a rappresentare sistemi complessi (e quindi anche le reti) attraverso dei nodi e degli archi. Oltre a ciò bisogna definire un linguaggio per capire gli elementi base di una rete, quindi definire una struttura e una comportamento.

### Definizioni di base

Un grafo è un insieme di nodi e archi;

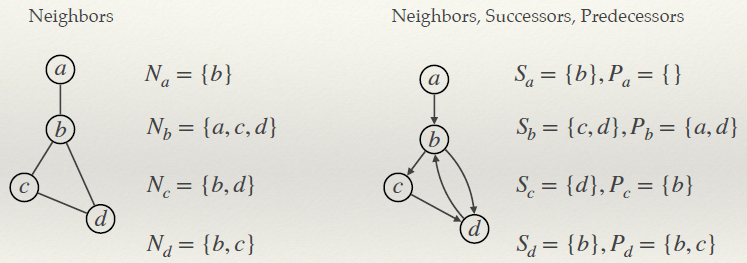
dove N è l’insieme di nodi e L quello di archi;

Un nodo si rappresenta con un numero intero, un arco invece con una coppia di interi;

Un grafo può essere diretto o indiretto, nel primo caso gli archi (i,j) e (j,i) sono archi differenti, nel secondo invece no;

Un grafo può anche essere pesato, in questo caso si rappresentano gli archi come terne di valori:

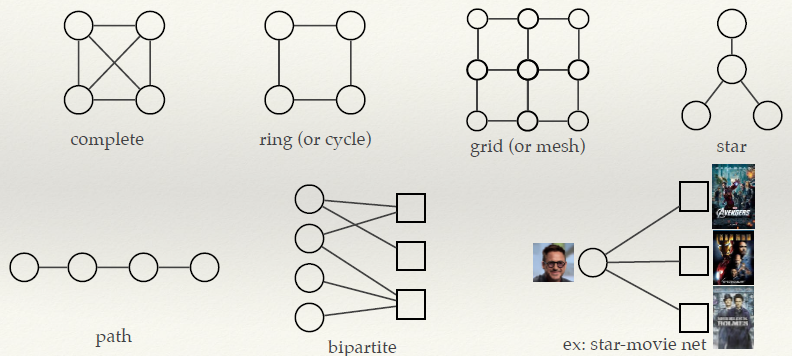
I vicini di un nodo n in un grafo sono tutti i nodi che hanno un arco verso n.

I successori di un nodo n sono tutti i nodi a cui punta n attraverso un arco.

I predecessori dello stesso nodo n sono invece tutti i nodi che puntano a n attraverso un arco.

Un grafo è completo quando tutte le coppie di nodi sono tra loro connesse attraverso un arco.

### Grafi semplici



### Densità e sparsità

Un grafo con N nodi può contenere al massimo N(N-1)/2 archi, contrassegnamo questo numero come Lmax. Come si misura la densità? In questo modo:

se d è molto minore di 1, il grafo è sparso. QUesto metodo non è tuttavia intuitivo, quindi utilizziamo le seguenti regole:

* Se L è nell’ordine di N, il grafo è sparso;
* Se L è nell’ordine di N^2, il grafo è denso.

### Sottoreti di un grafo

Una sottorete o sottografo è un sottoinsieme di nodi tra loro connessi, prende il nome di clique invece una sottorete completa.

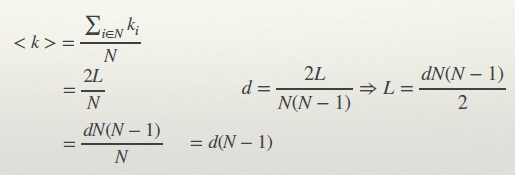
### Grado e forza

Il grado di un nodo è il numero di vicini a cui è connesso, o meglio in numero di archi che partono da esso. Nei grafi diretti possono distinguere il grado in:

* entrante, cioè il numero di archi che arrivano a un nodo n;
* uscenti, il numero di archi che partono da un nodo n.

La forza è di un nodo i è la somma di tutti i pesi degli archi a cui è connesso, anche qui possiamo distinguerla in entrante e uscente nei grafi diretti.

Il grado medio indica il numero medio di archi che può avere un nodo in un grado, questo numero è connesso alla densità, infatti:



## Reti multilivello e temporali

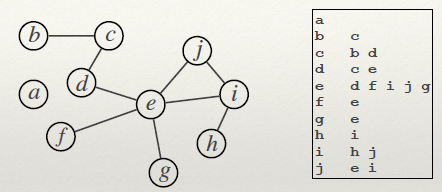
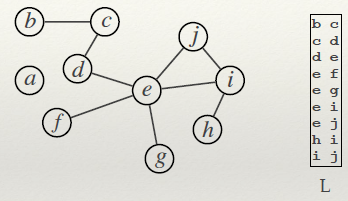
Una rete può avere più livelli, ognuna con i suoi nodi e archi. Questi livelli sono connessi attraverso appositi archi detti infralivello. Se due insiemi di nodi di due differenti livelli sono identici, la rete è una multiplex.

Le reti temporali sono multiplex in cui ogni livello rappresenta un istante temporale differente.

Possiamo considerare le reti multilivello come reti di reti, qui l’interazione tra nodi di livelli differenti è data proprio dagli archi infralivello. Questo tipo di reti non sono esenti da problemi, infatti una loro vulnerabilità imprevedibile sono i fallimenti a cascata.

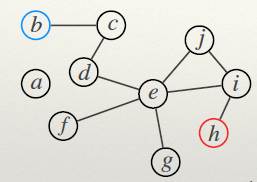
## Rappresentazione

La rappresentazione di un grafo può essere:

* una matrice di adiacenza, in cui si utilizza una matrice NxN le cui celle rappresentano la presenza o meno di un arco. Questa rappresentazione risulta essere particolarmente adatta per i grafi densi, per quelli sparsi invece è inefficiente in memoria (occupa N^2 di spazio e la maggiorparte delle celle è “vuota”);
* una liste di incidenza, qui utilizziamo un array di liste lungo N, ogni cella contiene tutti i vicini del nodo i nel grafo. Al costo di un tempo maggiore per le operazioni, questa rappresentazione si adatta bene anche per i grafi sparsi.
* una liste di archi.

## Cammini e connettività

Un cammino in un grafo è un sottografo in cui ogni nodo è connesso al successivo attraverso un arco, esso può essere:

* semplice quando non ci sono nodi ripetuti all’interno;
* ciclo se invece è così.

Un cammino presenta una lunghezza, dettata dal numero di nodi da cui è formato.

Di tutti i cammini ci interessa quello minimo, cioè il sottografo avente la lunghezza o il peso più basso.

Per fare un esempio, considerando il grafo a destra, il cammino minimo da b a h passa per c, d, e ed i.

E’ possibile anche passare per j per arrivare a h, tuttavia questo cammino non è minimo dato che la lunghezza è maggiore rispetto al precedente.

Attraverso la nozione del cammino, possiamo definire:

* la lunghezza media dei cammini:
* il diametro del grafo, cioè la lunghezza del cammino massimo.

Un grafo è connesso se esiste almeno un cammino per ogni coppia di nodi, altrimenti è disconnesso.

Una componente connessa è un sottografo connesso, in un grafo la più grande di queste è la componente gigante. La componente connessa più piccola possibile prende il nome di singleton.

Nei grafi diretti, una componente è fortemente connessa se,per ogni coppia di nodi all’interno di essa, esiste un cammino orientato.

La componente entrante di una componente fortemente connessa S è un sottoinsieme di nodi che può raggiungere S.

La componente uscente per S è sempre un insieme di nodi che può essere raggiunto a partire da S.

## Distanza e ricerca

La distanza è data dalla lunghezza del cammino minimo di un grafo, come possiamo calcolarla? Utilizzando le strategie di ricerca.

### Breadth First Search (BFS)

La strategia di ricerca BFS permette di calcolare le distanze minime a partire da un nodo verso tutti gli altri. che ragionamento utilizza?

* prima visita tutti i vicini di un nodo;
* dopodiché visita tutti i vicini dei vicini, eccetera.

Come possiamo vedere nella foto a destra, la visita BFS forma un albero, in ogni livello troviamo tutti i vicini dei nodi del livello precedente.

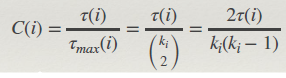
La complessità di questo algoritmo è O(N^2), questa lentezza è data proprio dal ragionamento dell’algoritmo.

Ma a parte le strategie di ricerca, il cammino minimo è davvero minimo? Ciò dipende dalla dimensione del grafo ma, analizzando la lunghezza media (APL) e la dimensione della rete, possiamo vedere una certe correlazione tra esse.

Possiamo quindi concludere che la lunghezza media è corta se cresce lentamente rispetto alla dimensione del grafo. Precisamente, questa crescita è logaritmica.

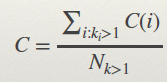
## Coefficienti di clustering

Il coefficiente di clustering di un nodo è la frazione di paia di vicini che sono tra di loro connessi, lo possiamo calcolare nel seguente modo:



Nella formula τ(i) indica il numero di triangoli in cui i è un nodo vertice.

Il coefficiente di clustering di una rete invece è la media dei coefficienti dei nodi:



# Omofilia

L’omofilia è il principio per cui una persona tende a essere simile ai propri amici, ciò li rende statisticamente significanti come un campione casuale della popolazione.

Come possiamo misurare l’omofilia? Considerando un grafo da colorare con due colori:

* assegnamo un colore random a ogni nodo;
* contiamo gli archi “di due colori”;
* calcoliamo le frazioni dei nodi colorati.
* confrontiamo i valori.

Consideriamo la foto a destra:

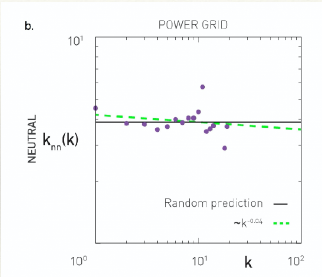
* ci sono 18 archi in tutto, 5 di questi “cambiano colore”, quindi consideriamo 5/18 come valore;
* calcoliamo le frazioni dei nodi colorati, su 9 nodi in tutto 6 sono bianchi e 3 sono rosa quindi la percentuale dei primi è 6/9 e 3/9 quella dei secondi;
* calcoliamo le frazioni degli archi bicolore, cioè il prodotto tra la frazione dei bianchi e dei rosa moltiplicato per due. Il risultato è 4/9.

Confrontando la frazione effettiva con quella calcolata, notiamo che 5/18 è minore di 4/9, ciò significa che è presente omofilia!

## Grado di correlazione

Nelle reti assortative, è presente una struttura perisphera con degli hub al centro, in quelle dissortative invece ci sono tanti hub distribuiti, formando delle piccole reti a stella tra loro sconnesse.

Come misuriamo l’assortatività?

* prima calcoliamo la media knn(i) dei gradi dei vicini di un nodo i;
* calcoliamo la media dei knn(i) per tutti i nodi i di grado k.

Nelle reti casuali, calcoliamo knn(k) come <k^2>/<k>.

knn(k) indica la media dei gradi dei vicini di tutti i nodi di grado k.

L’assortatività può essere:

* positiva se knn(k) cresce insieme a k in modo proporzionale. Nel grafo gli hub tendo a stare tutti al centro in questo caso (Collaborazione scientifica);
* neutrale se knn(k) né cresce né decresce rispetto a k. Di conseguenza, i nodi sono equamente distribuiti (Power grid);
* negativa se knn(k) decresce rispetto a K, in questo caso si parla di rete metabolica in cui c’è un solo hub al centro e tutti gli altri nodi connessi a esso.

## Meccanismi di omofilia

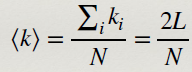
I meccanismi naturali di omofilia sono:

* la selezione, cioè il come nodi simili diventano connessi;
* l’influenza sociale, che essenzialmente è il contrario.

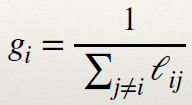
# Misure delle reti

Nella realtà, le reti sono eterogenee, quindi alcuni nodi sono più al centro rispetto ad altri. Come possiamo calcolare questa centralità? Utilizzando tre misure:

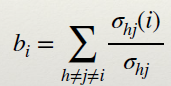
* il grado, cioè il numero di vicini di un nodo, esso si indica con ki. Un nodo di grado alto viene chiamato hub. E’ inoltre possibile calcolare il grado medio:



* la vicinanza gi, cioè quanto un nodo è più vicino ad altri nodi, il che lo rende anche più centrale. Questa misura si calcola come segue:



* la prossimità bi, quanto un nodo è attraversato da cammini:

Nella formula, σ(h,j) indica il numero di cammini da h a j mentre σ(h,j,i) è il numero di cammini da h a j che passa per i. I nodi hub sono soliti avere un’alta prossimità, ciò tuttavia non è una regola, infatti esistono nodi con alta prossimità che non sono hub;

* La prossimità degli archi è la frazione di cammini tra tutti i possibili nodi che passa attraverso un arco, il calcolo infatti è lo stesso del punto precedente.

## Distribuzione della centralità

Nelle reti piccole ha senso chiedersi quali nodi e archi siano i più importanti, ma in quelle grosse? No, o meglio non focalizzandoci su ogni nodo e arco. La soluzione infatti è l’utilizzo di approcci statistici su classi di nodi e archi aventi proprietà simili.

Considerando nk come il numero di nodi aventi grado k, definiamo la frequenza fk come segue:

Per valori grandi di N, la frequenza fk diventa la probabilità pk che un nodo abbia grado k.

In questi contesti ritornano utili:

* la distribuzione cumulativa, cioè la probabilità che una variabile sia maggiore o uguale a un numero x dato;
* la scala logaritmica per plottare grandi range di valori in poco spazio.

## Distribuzioni del grado

Il parametro di eterogeneità κ indica quanto è diffusa la distribuzione, esso si calcola come segue:

Se la maggior parte dei gradi tende allo stesso valore, allora κ tende a 1. In generale, se la distribuzione è molto eterogenea, κ è molto maggiore di 1.

## Paradosso dell’amicizia

Il paradosso dell’amicizio indica che, considerando il grafo a destra:

* Scegliendo nodi random, Tom ha la stessa probabilità di essere scelto rispetto agli altri;
* Scegliendo invece archi random, Tom ha più probabilità di essere scelto rispetto agli altri, questo perché Tom è il nodo che ha più archi connessi a esso.

Calcolando il grado medio di un nodo e quello di un suo vicino, otteniamo rispettivamente 2.29 e 2.83, quindi il vicino ha in media più vicini, e ciò è un paradosso.

Più il numero di hub è grande, più questo effetto sarà presente.

## Mondi ultra piccoli

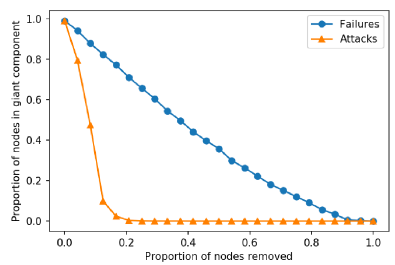
Nei casi reali, ci sono reati con molti cammini minimi che passano attraverso gli hub. Per fare un esempio, nei trasporti aerei potrebbe non esistere un volo diretto per andare da una città all’altra, tuttavia possono esserci degli scali.

La proprietà del mondo piccolo è tipica delle reti di nostro interesse, se inoltre sono presenti molti hub, i cammini sono ultra piccoli.

## Robustezza

Un sistema è robusto se i fallimenti non ne influenzano il funzionamento, come la definiamo nelle reti? Rimuoviamo nodi e archi e dopo verifichiamo che succede nella struttura. La connettività diventa quindi un punto chiave in questo contesto.

Come si testa la robustezza?

* si plotta la dimensione S della componente connessa più grande come una funzione dei nodi rimossi;
* supponendo una rete inizialmente connessa, abbiamo una sola componente connessa e S = 1;
* più tolgo nodi e archi, più questa rete si rompe progressivamente in componenti, quindi S cresce.

Come togliamo i nodi?

* l’approccio random failure sceglie un nodo casuale e lo toglie, tutti i nodi hanno la stessa probabilità di essere scelti. In questo modo rimuoviamo sempre una frazione f di nodi, la decrescita quindi è più o meno lineare;
* l’approccio attack dà priorità ai nodi di grado alto. In questo modo la decrescita è più rapida dato che appunto rimuoviamo i nodi di maggior grado.

## Core decomposition

Un core è una parte densa della rete, in cui i nodi hanno un grado molto alto. La core decomposition è quindi una procedura per identificare le parti più dense della rete rimuovendo progressivamente i nodi di grado più basso. Se rimuoviamo tutti nodi con grado k-1 o più basso, la porzione rimanente è chiamata k-core.

L’algoritmo di core decomposition è ricorsivo e funziona come segue:

* si inizia ponendo k uguale a 0;
* ricorsivamente si rimuovono tutti i nodi di grado k, fino a quando non ce n’è nessuno;
* l’insieme dei nodi rimosso è il k-th shell mentre i rimanenti sono il k+1-core;
* Se non ci sono più nodi, l’algoritmo termina, altrimenti si incrementa k di 1.

La core decomposition aiuta a visualizzare meglio le grandi reti, infatti possiamo vederlo come un pruning dei nodi di grado basso per vedere solamente le parti più dense.

## Località topologica

Nei casi reali, la maggior parte delle reti sono grafi diretti, quindi con archi orientati. Nel caso del web infatti, internet rappresenta un enorme grafo in cui i vari siti web sono tutte componenti connesse, ogni nodo infatti è una pagina web e gli archi sono gli ipertesti.

Per omofilia, le pagine web per un dato argomento tendono a linkarsi tra di loro, quindi possiamo pensare a un concetto di similarità basato proprio su questo. Chiamiamo topic drift quel concetto per cui, a partire da una pagina iniziale, la similarità decresce ogni volta che passiamo da un nodo all’altro. Gli web crawler fanno una ricerca BFS per trovare la località topologica, infatti più una pagina risulta lontana, più è dissimile dalla pagina di partenza.

Un altro metodo per calcolare la similarità è il coseno in spazio vettoriale, infatti:

* usiamo una dimensione per ogni termine;
* rimuoviamo o diamo un minor peso ai termini più comuni e con meno significato;
* usiamo il deep learning per incorporare spazi euclidei con poche dimensioni.

## Page ranking

Possiamo usare il page ranking come misura di centralità, inoltre è utile per calcolare l’importanza o il prestigio di una pagina web. I motori di ricerca hanno bisogno del ranking in modo da rendere equa la rilevanza dei contenuti.

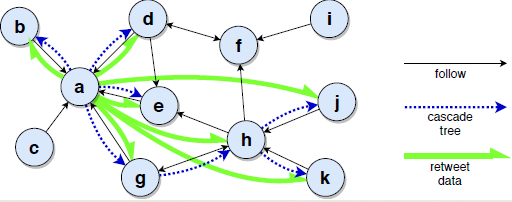
Page ranking e in-degree hanno una distribuzione molto simile, in ogni caso si può dare importanza ai link di pagine più importanti rimuovendo quelli meno importanti o che risultano spam.

## Diffusione delle informazioni

Tante informazioni oggi vengono trasmesse attraverso i meme, infatti questi contengono idee, immagini, hashtag e link. I meme inoltre possono mutare e dar vita ad altri meme, di conseguenza nascono nuove informazioni.

Prendendo come esempio Twitter, come possiamo tracciare la diffusione di un meme?

* con la rete di retweet, infatti questo contengono un link da retweettato e retweettante;
* le menzioni, qui i link vengono utilizzati per indicare chi replica o chi viene menzionato.

I tweet inoltre posseggono un timestamp, di conseguenza è possibile aggregarli in reti temporali.

Considerando la rete di Twitter, l’albero di diffusione è una sottorete formata da tutti gli utenti che ricevono un dato tweet, da qui possiamo ricavare la rete di retweet come una sottorete a stella di tweet che linkano quello originale.

## Eco camere

La rete di retweet può essere rappresentata anche come una eco camera, in essi è possibile comprendere come un determinato topic influenza gli utenti. Con gli archi invece è possibile indicare la credibilità di un mezzo.

## Viralità

Ci sono tanti modi per misurare la viralità di un meme:

* il numero di utenti esposti;
* la profondità dell’albero di diffusione;
* la frazione di utenti che retwittano rispetto al numero totale di esposti;

Una cosa strana riguarda l’errore dell’informazione, questo infatti risulta più virale di una news sullo stesso argomento.

## Influenza

Anche per l’influenza di un account esistono tanti modi per misurarla:

* il numero di follower, traducibile come in-degree;
* il numero di utenti esposti,l’out-degree;
* il numero di retweet, la forza di uscita;
* la frazione di retweet ai follower;
* Il targeting di bot;

## Bot social

I bot sociali sono account controllati da programmi per manipolare la diffusione delle rete, alcune manovre includono:

* i finti follower;
* i finti retweet, la cosiddetta amplificazione;
* l’astrosurf, cioè un’apparenza di viralità organica;

Tutti i social media e gli utenti sono vulnerabili ai bot.

## Reti di co-occorrenza

Le reti di co-occorrenza sono particolari reti in cui possiamo aggregare più nodi aventi dei vicini comuni. L’operazione di proiezione infatti serve proprio ad aggregare questi nodi tra loro.

## Cocitazione e coreferenza

Le reti di cocitazione e di coreferenza sono particolari reti in cui si fa riferimento ad un altro documento:

* Le due citazioni condividono un predecessore, quindi hanno lo stesso “padre” o “antenato”;
* I due riferimenti condividono un successore, un “figlio” o un “discendente”.

## Social tagging

La folcosomia è un insieme di triple (utente,risorsa,tag) che indica dove un utente assegna un tag a una risorsa. Ciò apre alle cosiddette reti tripartite, in cui ogni arco connette tre nodi di tre insiemi distinti.

Possiamo proiettare i tag in base ai vicini comuni:

* prima si proietta sulla rete bipartita pesata;
* dopodiché si proietta sulla rete pesata, possiamo infatti computare i pesi con il coseno.

## Eterogeneità dei pesi

E’ possibile allineare i pesi in molteplici ordini di magnitudine. Inoltre possiamo comparare la distribuzione della forza interna con il page ranking, qui è presente correlazione ma è molto debole. Da ciò deduciamo che il page ranking non si adatta molto bene al traffico web, poiché i salti casuali causano molte discrepanze.

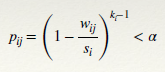
## Filtraggio dei link

Ci possono essere dei link di peso molto basso che però rendono la rete troppo densa, quindi difficile da visualizzare o analizzare.

Nel caso in cui i pesi bassi indichino una bassa importanza, possiamo filtrarli e analizzare una rete più sparsa. L’approccio più semplice è utilizzare un threshold, tuttavia esistono anche metodi migliori.

Come si imposta il threshold?

* si usa la backbone, cioè si settano differenti threshold a ogni nodo, quindi possiamo rimuovere i link meno significativi in base al nodo a cui è connesso;
* possiamo inoltre comparare le probabilità dei pesi dei link in base al grado e alla forza di un nodo e calcolare il threeshold con la seguente formula:



# Reti casuali

Una rete casuale è una particolare rete in cui i link dei nodi vengono posizionati in modo randomico. Precisamente si genera un numero random r tra 0 e 1 e, se questo è minore di un threshold p dato, si genera l’arco, altrimenti no. Questo ragionamento si applica per ogni coppia di nodi.

Cosa possiamo dire del threshold p? questo valore indica la probabilità di generazione di un arco tra due nodi i e j, infatti:

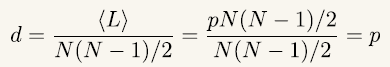
* p uguale a 0 indica che non ci sono archi nel grafo;
* p uguale a 1 invece indica che il grafo è completo.

Che succede quando aggiungiamo archi al grafo? La dimensione della componente connessa più grande cresce.

C’è un modo per stimare il numero di archi di un grafo in base a p? Ragionando come per il lancio di una moneta, abbiamo che, su un numero t di tentativi, otterremmo un numero h di teste. Quindi, dato che la probabilità per ottenere una testa è p, possiamo calcolare h (e il numero di archi di conseguenza):

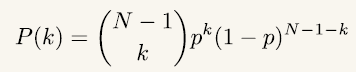
E per la densità e il grado medio?







La distribuzione del grado k nella rete è binomiale e si calcola come segue:



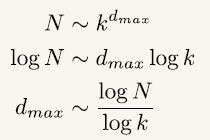
## Proprietà del mondo piccolo nelle reti causali

Quanti nodi troviamo mediamente dopo d step? Sono k al primo step, k(k-1) al secondo, eccetera.

Possiamo quindi riassumere tutto con la seguente formula:



E invece quanti step sono necessari per coprire l’intera rete? Ciò dipende dal numero di nodi N e dal grado k. Per via della precedente formula, c’è una correlazione tra N e k^d, in questo caso però d indica la distanza massima raggiungibile:



Per il coefficiente di clustering invece usiamo la formula:



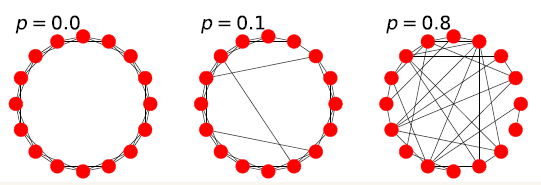
## Reti del mondo piccolo

L’idea è quella di costruire una rete con le proprietà del mondo piccolo e un alto coefficiente di clustering, come facciamo? Interpoliamo tra un lattice regolare e una rete casuale.

L’alto clustering però può causare cammini troppo lunghi, come risolviamo? Aggiungendo degli archi che fungano da scorciatoia.

## Modello di Watts-Strogatz

Il modello di Watts-Strogatz è un particolare modello in cui i nodi sono disposti ad anello e hanno un grado k. Con probabilità (p) si riconnettono i nodi randomicamente.



Un p piccolo indica che solo pochi link vengono riconnessi, di conseguenza il coefficiente di clustering medio tende a rimanere lo stesso.

Con p uguale a 1, tutti i nodi vengono riconnessi, di conseguenza si ottiene una rete causale.

Dal momento che la maggior parte dei nodi presenta lo stesso grado, non ci sono nodi hub in questo tipo di modelli. Questo modello fallisce nel replicare la distribuzione del grado che è osservabile in molti casi reali.

## Modello di configurazione

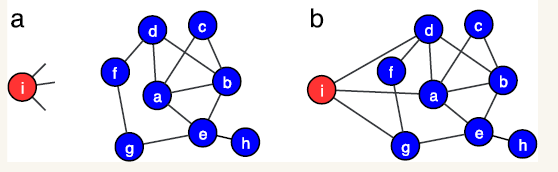
Il modello di configurazione prevede di costruire reti in base a una data distribuzione del grado:

* si genera una lista di N numeri, ognuno indica il grado di ogni singolo nodo;
* in base alla lista, si generano tanti archi che rispettano la distribuzione.

Per preservare la randomizzazione del grado, si generano differenti versioni della rete data una stessa sequenza, come mai?

* Ciò è utile se una proprietà specifica è determinata proprio dalla distribuzione del grado;
* se la proprietà viene mantenuta, la distribuzione è ciò che influenza maggiormente la costruzione;
* se invece non è così, i responsabili sono altri fattori.

## Modelli preferenziali

Nei casi reali, le reti sono dinamiche, quindi il numero di nodi all’interno continua a variare così come il numero di archi, come possiamo introdurre questa dinamicità?

Quando un nuovo nodo viene aggiunto alla rete, si aggiungono tanti archi quant’è il suo grado, per farlo si seguono date regole che dipendono dal contesto.

La prassi è quella di connettere i nuovi nodi a quelli con grado maggiore, nel web ad esempio si tende infatti a scegliere le pagine più popolari.

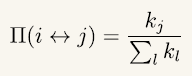
Il nuovo modello di cui abbiamo bisogno deve quindi saper crescere e deve preferire i nodi più “popolari” come prima scelta.

Esiste un modello che permette ciò?

## Il modello di Barabàsi-Albert

Il modello di Barabàsi-Albert è un modello che permette di rispondere alla precedente domanda! Come funziona?

* si parte da un gruppo iniziale di nodi m0, solitamente completo;
* A ogni step si aggiunge un nuovo nodo i e si collega con m archi a un sottoinsieme di m0;
* come si scelgono gli archi? in base a una probabilità proporzionale al grado del nodo di arrivo, quindi immaginando la possibile aggiunta di un arco da i a j:



* Si continua così fino a quando non si raggiungono N nodi.

In questo modello, i nodi più popolari lo diventano ancora di più a causa proprio delle preferenze, proprio in questo modo vengono generati i nodi hub. I nodi hub si riferiscono principalmente ai nodi più vecchi, ciò è vantaggioso dato che permette di trasferire l’informazione più in fretta.

## Comunità

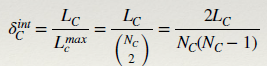
Una community equivale a un cluster della rete, una componente fortemente connessa. Le community sono utili per indicare come i nodi siano più propensi ad andare verso un certo valore anziché verso un altro. Un esempio sono le elezioni politiche, le persone (i nodi) si allineano con le loro preferenze di partito, tutte insieme formano il cluster.

Perché studiamo le community?

* scoprire l’organizzazione della rete;
* identificare le feature dei nodi;
* classificare i nodi in base alla loro posizione nella community;
* trovare archi perduti.

Che variabili ci sono nelle comunità?

* il grado di un nodo (interno ed esterno);
* il grado della community, cioè il numero di archi verso nodi esterni alla community;
* la densità dei link, il rapporto tra il numero di link della community con quello totale che la stessa community può contenere:



Oltre a ciò esistono anche i concetti di:

* l’alta coesione, cioè quando ci sono molti archi interni alla community;
* l’alta separazione, in questo caso nella community ci sono nodi connessi tra loro con pochi archi.

Si parla quindi di comunità forte quando il grado interno di ogni nodo è maggiore rispetto a quello esterno.

Una community invece è debole quando la somma di tutti i gradi interni di ogni nodo è maggiore rispetto alla somma dei gradi esterni.

In base a queste ultime due definizioni, possiamo concludere che una comunità forte è anche debole mentre non è vero il contrario.

Non ha comunque senso confrontare una comunità con il resto della rete, perciò utilizziamo delle definizioni più stringenti:

* una comunità è forte quando presenta molti nodi che hanno nella direzione interna alla community stessa anziché in altre;
* una community è invece debole quando la somma dei gradi interni dei nodi eccede il numero totale di vicini che i nodi hanno in altre community.

## Partizioni

Il numero di partizioni di n oggetti è dettato dal numero di Bell Bn, questo numero cresce più velocemente dell’esponenziale rispetto a n. Ciò non ha senso dato che non ha senso cercare le community in questo modo, bisogna trovare un modo più triviale per farlo.

## Overlapping community

Le overlapping community sono comunità particolari che condividono dei nodi con altre comunità. Un divisione di una rete in overlapping community è chiamata coperta, il numero di possibili coperte è molto più alto del numero di partizioni, questo perché due comunità possono intersecarsi in molti modi.

## Community gerarchiche

Le community gerarchiche sono particolari comunità che ne contengono altrettante all’interno, qui il partizionamento diventa importante dato che un buon algoritmo di clustering dovrebbe rilevare tutte le comunity.

## Problemi dei cluster

Il graph partitioning è il problema utile per trovare le regioni più dense in una rete. Dividere una rete in partizioni di una data grandezza non è comunque una cosa semplice da fare, quel che si può fare è dividere i link tra gruppi.

Il numero di cluster e la dimensione di questi potrebbe non essere conosciuto, in questi casi che si fa?

Un’idea è quella di dividere i nodi in due gruppi di egual misura:

* dati N nodi, si dividono in due gruppi A e B rispettivamente di dimensione Na e Nb;
* per ogni arco (i,j) per cui i è nel gruppo A mentre j è in B, si calcola quanto varia il cut size tra la partizione corrente e quella con l’arco (j,i);
* la coppia (i,j) con decremento più grande viene invertita e si bloccano i nodi i e j;
* si ripetono questi step finché tutti i nodi non saranno bloccati.

Questo è l’algoritmo di Kernighan-Lin, tuttavia esso presenta un problema: la scelta della partizione iniziale influenza pesantemente il risultato finale dato che da ciò dipende la grandezza del cut size, come si potrebbe ovviare al problema? Si creano più partizioni random e si prende quella col cut size minore come punto di partenza.

L'algoritmo è greedy e a causa di ciò si rischia di ottenere un ottimo locale. Il problema può essere mitigato, ma non risolto, scambiando i nodi che incrementano il cut size.

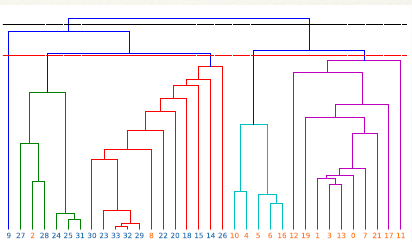
Per questo motivo questo algoritmo viene utilizzato in post-processing per migliorare le partizioni di altri metodi.

Il partitioning presenta comunque dei limiti, infatti i cluster devono essere ben separati, ma non necessitano un’alta densità di archi. Quindi, in generale, i cluster ottenuti col graph partitioning non sono community. Inoltre il numero di cluster deve essere già noto e dato in input.

## Data clustering

Nel clustering gerarchico, il concetto principale è la similarità, la quale si calcola come segue:

Possiamo inoltre definire la similarità tra gruppi di nodi S(A,B) come segue:

* per il linkaggio singoli si prende la similarità massima per ogni coppia di nodi;
* per il linkaggio completo si segue lo stesso ragionamento ma prendendo il minimo;
* per il linkaggio medio, si fa la media delle similarità.

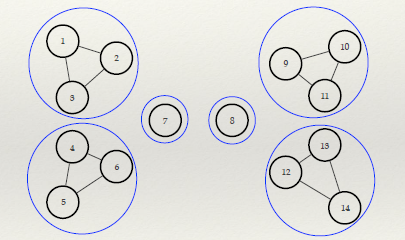
Per il clustering gerarchico esistono due approcci:

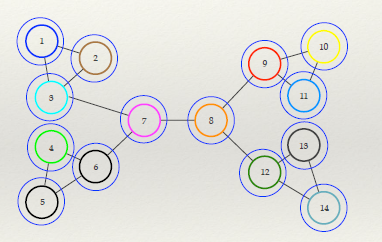
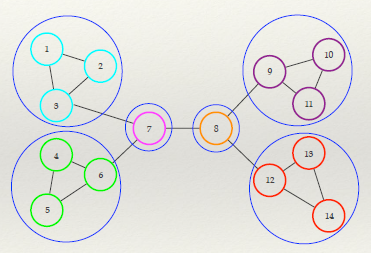
* l’approccio agglomerativo genera le partizioni iterativamente unendo gruppi di nodi più simili;
* l’approccio divisorio segue lo stesso ragionamento ma dividendo un gruppo dove i nodi sono meno simili.

In entrambi gli approcci otteniamo un dendrogramma, un albero che indica quali cluster vengono uniti tra loro e a quale step.

Il partitioning corrisponde alla linea rossa orizzontale nella foto a destra, quindi si seleziona fino a che livello si vuole unire/dividere.

# Rilevamento dei cluster

Il metodo generale per rilevare le community è divisivo, precisamente è top down e consiste nella rimozione degli archi ricoprenti della rete.

Un’altro approccio generale è quello agglomerativo, esso è bottom up e consiste nell’unione di gruppi di nodi.

## Approccio divisivo

Quale arco rimuoviamo per prima? Tra tutti gli archi ricoprenti della rete, togliamo quelli meno robusti che sono quelli che connettono tra loro più community. Possiamo trovarli calcolando la betweenness di ogni arco; più questo valore è alto, più è probabile che l’arco venga rimosso.

Cosa possiamo dire della betweenness?

* è inversamente correlata all’overlapping tra vicini, coefficienti di clustering ed endpoint;
* è collegata alle nozioni di traffico o flusso.

Quindi per rilevare i cluster, procediamo con l’algoritmo di Girvan-Newman:

* si calcola la betweenness di tutti gli archi;
* si prende l’arco con betweenness più alta e si rimuove;
* se una data condizione viene soddisfatta, si restituiscono i cluster, altrimenti si riprende dal primo punto.

Cosa possiamo osservare in questo algoritmo?

* non c’è bisogno di trovare archi ricoprenti;
* la betweenness è un egregia approssimazione;
* Ricalcolare le betweenness è necessario dato che ogni volta non si considera un’arco, ciò però rende l’algoritmo lento;
* Nelle reti con comunità forti, si possono ottenere community disconnesse molto velocemente, in questo caso non c’è bisogno di ricalcolare i valori dato che si considera l’ultimo arco rimosso. Ciò permette di abbattere la complessità del problema.

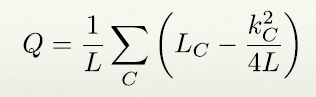
## Modularità

La modularità è una valutazione delle community rispetto a una baseline. Per ogni community, la modularità calcola la differenza tra il numero di nodi interni e il valore atteso di questo numero.

Se la rete è casuale, la modularità di ogni partizione si suppone sia bassa dato che il numero di archi interni al cluster dovrebbe essere vicino al valore atteso. Se invece questo numero è più grande, è improbabile che sia una concentrazione di archi interni.

Il test della modularità segue lo stesso principio del test dell’omofilia, con dovute differenze:

* abbiamo più di due gruppi;
* abbiamo bisogno di fare più confronti con una preservazione del grado.

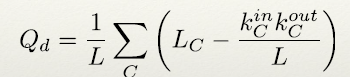


Come mai l’ultima divisione è fatta in questo modo? Dato che gli archi sono casuali (e al massimo possono essercene kC nella comunità C), la probabilità di selezionare uno degli estremi dell’arco è kC/2L, dove 2L è il numero totale di archi nella rete. E la probabilità di scegliere due estremi? Il quadrato.

Che proprietà ha la modularità?

* Q è minore di uno per ogni partizione di ogni rete;
* Q a 0 per le partizioni che comprendono l’intera rete;
* Q può essere negativo, succede quando ci sono N partizioni formate da un solo nodo;
* Per la maggior parte delle reti, Q non è sempre tra 0 e 1.

Esiste un’estensione della probabilità per le reti dirette:



Ottimizzare la modularità è un problema difficile, quindi ci accontentiamo delle approssimazioni.

L’algoritmo di Newman è un metodo greedy per ottenere la modularità:

* Si parte con N partizioni tutti di un nodo;
* Si uniscono le coppie di nodi incrementano molto la modularità;
* Si continua così finchè tutti i nodi non sono nella stessa comunità;
* Si prende la partizione con la modularità maggiore.

Questo algoritmo ha però dei limiti:

* Anche se tenta di massimizzare a ogni step, si rischia comunque di ottenere soluzioni subottimali;
* c’è la tendenza a generare partizioni sbilanciate, ciò rallenta il metodo;

## Algoritmo di Louvain

* All’inizio ogni nodo appartiene a una comunità differente;
* si cambia la community di ogni nodo mettendo in quella con la modularità più alta. Si continua così finchè ci sono nodi che aumentano questo valore;
* Si trasforma la rete in una super rete pesata: ogni nodo diventa un super nodo, ogni arco ottiene un peso che dipende dalla sua posizione nella rete;
* La procedura si ferma quando non ci sono cluster nella partizione corrente che incrementano la modularità.

Questo algoritmo è greedy come il precedente e prova a trovare la partizione migliore a ogni livello di agglomerazione, quella finale però dipende dall’ordine di visita dei nodi. L'algoritmo è comunque molto veloce dato che, dopo una prima iterazione, le trasformazioni successive sono in grado di ridurre la rete molto velocemente e di solito genera una manciata di partizioni. Quindi può rilevare le comunità su reti con milioni di nodi.

## Limiti della modularità

La modularità tende a essere più grande dove ci sono reti grandi, di conseguenza non si può utilizzare per confrontare la qualità delle partizioni. Inoltre la modularità massima:

* è alta nella reti causali non strutturate;
* non corrisponde necessariamente alla miglior partizione.

Riguardo ai limiti della risoluzione, l’ideale sarebbe ottenere una partizione naturale, cioè quella in cui le community sono clique, ma in un caso generale? Una soluzione è fare tuning sulla risoluzione mettendo il parametro nella formula, ciò però ha due problemi:

* è computazionalmente intensivo;
* il criterio è necessario per decidere quale risoluzione è più adatta alla rete.

## Limiti del clustering gerarchico

* i risultati dipendono dalla similarità e dal criterio per calcolarla;
* Questo approccio è tipicamente lento, specie con reti con milioni di nodi.

## Propagazione delle label

Sappiamo che i vicini di un nodo tendono a rimanere nella stessa comunità, possiamo sfruttare ciò? Si, utilizziamo l’algoritmo di propagazione delle label:

* all’inizio si assegna ogni nodo a una comunità differente, ogni con una label diversa;
* si visitano tutti i nodi casualmente, ognuno prenderà la label che occorre di più tra i suoi vicini. Se invece non c’è una maggioranza, ne prende una random.
* l’algoritmo si ferma se ogni nodo ha ottenuto la label maggiore dei vicini, altrimenti si ripete il secondo step.

Le comunità saranno quindi quei sottoinsiemi di nodi che posseggono una specifica label.

Che limiti ha questo algoritmo?

* la soluzione fornita non è unica, infatti dipende dall’ordine in cui i nodi vengono visitati;
* le differenti partizioni incontrate lungo il processo possono essere rotte in vari modi proprio a causa della randomicità. A causa proprio di queste instabilità, le partizioni trovate in questo modo tendono a essere simili, per un risultato più robusto bisognerebbe combinare le soluzioni di esecuzioni differenti.
* L’algoritmo non dà altre informazioni relative al numero e le dimensioni della community;
* non ha parametri;
* è semplice da implementare ed è molto veloce;
* è inoltre possibile utilizzare le label conosciute come seme per ottenere la partizione iniziale.

## Modello stocastico a blocchi

L’assunzione di base di questo approccio è che la rete viene generata utilizzando dei modelli, ognuno caratterizzato da uno o più parametri. Con quali valori il modello può produrre una rete vicina a quella di nostro interesse?

Focalizzandosi sulla struttura, abbiamo bisogno di generare reti con delle community!

Il principio quindi è quello di dividere i nodi in gruppi, la probabilità che due nodi siano connessi è determinata proprio dal gruppo in cui finiscono.

Dati q gruppi e gi label del gruppo per il nodo i, la probabilità di formare un arco tra i e j è la seguente:



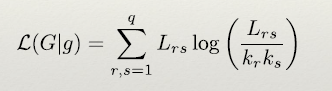
Memorizziamo tutte le probabilità in una matrice qxq, in essa l’elemento kl è il link con probabilità pkl. La diagonale indica le probabilità per cui l’arco rimanga all’interno di un gruppo di nodi k.

In base ai parametri, abbiamo diverse situazioni:

* la struttura a comunità la otteniamo quando le probabilità delle celle sulla diagonale sono maggiori rispetto al resto, qui i link sono probabilmente all’interno dei blocchi;
* la struttura a multipartita è quella duale alla precedente: i link sono quindi tra due blocchi;
* la struttura a periferia si ottiene quando q è uguale a 2 e p11>>p12>>p12, di conseguenza i nodi sono relativamente ben connessi tra loro quasi come un set periferico di nodi che interagiscono molto poco;
* se tutte le probabilità sono uguali, si ha una classica rete casuale in cui non è presente una struttura a gruppi.

Come si trova un modello stocastico per una rete?

* si massimizza la likelihood per una data partizione della rete, il modello infatti deve sapere riprodurre il piazzamento di archi tra nodi. La miglior partizione corrisponde alla likelihood più alta;
* Dal momento che il modello stocastico standard non descrive bene la rete reale (l’eterogeneità viene ignorata), si aggiunge una correzione utilizzando i gradi dei nodi della rete.



Nella formula sopra, Lrs è il numero di link tra il gruppo r e il gruppo s mentre kr p il grado del gruppo r.

### Algoritmo

* si parte con una partizione random con q cluster;
* si muove un nodo da un gruppo all’altro, precisamente si seleziona quello con maggior likelihood. Ogni nodo può essere mosso una sola volta;
* Quando tutti i nodi sono stati mossi, si ispeziona la partizione e si selezione quella con la likelihood più alta;
* L’algoritmo termina quando la likelihood non cresce più dopo due iterazioni consecutive.

### Limiti

E’ necessario fornire il numero di cluster in input, il quale è solitamente sconosciuto, esistono comunque tecniche per stimarlo.

Inoltre l'algoritmo di massimizzazione è greedy e rischia di bloccarsi in soluzione subottimali. Esistono metodi per iniziare con tanti stati randomici e scegliendo quello con likelihood più alta.

# Dinamicità delle reti

La diffusione di informazioni errate è un esempio di dinamicità nelle reti, i nodi infatti portano feature modificabili o copiabili dai loro vicini.

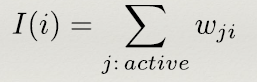
La diffusione delle informazioni parte coi seguenti standard:

* un certo gruppo di nodi detti influencer viene attivato all’inizio;
* ogni nodo inattivo si attiva seguendo certe regole, queste ultime dipendono dalla presenza o meno di nodi attivi tra i vicini;
* il risultato è un'influenza di attivazioni a cascata dei nodi della rete.

## Modelli a threshold

Il principio di questi modelli è quello di attivare un nodo solamente se l’influenza dei vicini supera un valore dato, il threshold.

Nelle varianti lineari, l’influenza di un nodo consiste nella somma dei contributi di ogni vicino, rappresentabile come il peso di un arco.

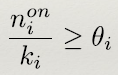


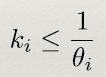
La condizione di attivazione è un predicato che indica se l’influenza è maggiore del threshold:



L’attivazione degli stati può avvenire in due modi:

* in modo asincrono, cioè che l’attivazione di un nodo dipende dalla configurazione corrente dei suoi vicini;
* in modo sincrono, in tal caso l’attivazione dipende dalla configurazione precedente.

Esistono inoltre delle varianti frazionarie, in questo caso i principii sono gli stessi ma cambia la funzione di attivazione, che è appunto una frazione:



E’ bene precisare che se la rete è sparsa, l’attivazione a cascata viene triggerata in base alla struttura. Possono quindi esserci dei nodi vulnerabili, i quali vengono attivati solamente da un singolo vicino. Come si fa fronte a ciò? Cambiando la funzione di attivazione:

Gli hubs nei modelli frazionari diventano dei veri e propri influencer, ciò tuttavia non ne testa la bontà.

La posizione dei nodi attivi è importante dal momento che è possibile attivare più o meno nodi in base a ciò, infatti è meglio essere nel cuore anziché nella periferia per permettere ciò.

Allo stesso modo la struttura a community è importante dal momento che l’informazione si espande più velocemente all’interno di esse anziché all’esterno, non è infatti semplice influenzare una community diversa dalla propria.

A fronte di ciò, è necessario rivelare i principali influencer e controllare l’attivazione a cascata, controllare l’evoluzione di quest’ultima risulta infatti importante dal momento che, anche se piccole, queste man mano possono crescere selezionando i nodi attentamente.

## Modello a cascata indipendente

Questo modello si basa sulla singole interazioni tra nodi, ogni nodo ha infatti una propria influenza e la utilizza per attivare o meno gli altri.

Un nodo i può convincere un vicino inattivo j con una probabilità pij. Tutti i nodi attivi vengono considerati in sequenza, se j viene attivato, c’è una sola possibilità di attivare i suoi vicini.

Rispetto ai precedenti modelli, il modello indipendente a cascata si focalizza sui nodi attivi ed utilizza un sistema di attivazione probabilistico (cosa che lo rende difficile da controllare).

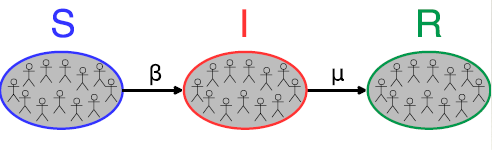
## Modelli epidemici

I modelli epidemici dividono la popolazioni in tre stadi:

* suscettibili: individui che non hanno contratto il virus;
* infetti: individui che hanno attualmente il virus;
* guariti: individui che sono guariti dal virus e non possono più riprenderlo.

Un suscettibile può diventare infetto con una probabilità b, il famoso infection rate. Il recovery rate invece è la probabilità u di un infetto di diventare guarito.

Questo è il modello SIR, il modello SIS invece ragionano allo stesso modo ma in questo caso i guariti possono nuovamente ammalarsi.

Come simuliamo il SIS?

* si prende una rete e si setta una frazione di nodi come infetti, l’altra come suscettibili;
* Se un nodo i è suscettibile, si controllano i vicini, se qualcuno di questi è infetto, i lo diventa con una probabilità i;
* Se invece i è infetto, diventa suscettibile con probabilità u.

L’algoritmo del SIR è praticamente uguale a SIS, l’unica differenza è che un nodo diventa guarito dopo essere stato infetto.

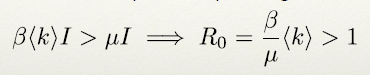
Sia in SIR, sia in SIS, possiamo notare tre caratteristiche:

* una fase iniziale dove con sono poche persone infetta e quindi il contagio è lento e irregolare;
* una fase di ramp up in cui il numero di infetti cresce e i suscettibili decrescono in modo esponenziale;
* una fase stazionaria in cui la crescita rallenta fino a diventare costante.

## Mixing omogeneo

L’ipotesi del mixing omogeneo è che tutti gli individui siano tra loro in contatto. Tutti gli individui presentano lo stesso comportamento e solo una piccola percentuale è importante per la dinamicità del modello. Per una popolazione piccola ciò si può attuare, per quelle grandi invece no dato che gli individui possono contagiare solo quelli “vicini”.

Per permettere ciò, introduciamo il grado nel calcolo:

* ogni nodo ha approssimativamente un grado pari a <k>;
* All’inizio poche persone sono infette, ogni infetto può trasmettere la malattia a <k> persone con una probabilità b a ogni interazione, il numero atteso di persone infette è quindi b\*<k>.
* Se ci sono I persone infette, b\*<k>\*I è il numero di nuovi infetti mentre u\*I è il numero di guariti.

Possiamo implementare SIS e SIR utilizzando un modello a threshold in questo modo:

R0 indica il numero base di riproduzione, se è minore di 1 allora l’outbreak finirà nel giro di poco tempo, altrimenti la situazione sarà pandemica!

Dal momento che il contatto reale non è omogeneo, cosa possiamo fare? Inseriamo una reazione: se l’infection rate è basso, il processo può infettare l’hub, questi nodi sono importanti dato che cambiano drasticamente lo scenario, infatti può contagiare moltissimi individui, inclusi altri hub.

Il contenimento della malattia dovrebbe ambire alla vaccinazione e l’isolamento di individui con più contatti. Possiamo identificare questi nodi prendendo l’endpoint di archi casuali,ciò cambia la probabilità di contagiare un hub.

## Diffusione di rumor

La diffusione dei rumor utilizza come modello una variante del SIR, l’idea di base è quella di trovare individui che diffondano il rumor non consapevoli di esso, se non così infatti c’è una perdita di interesse.

Gli individui in questo sistema possono essere:

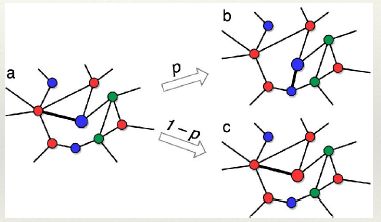
* Ignoranti: persone che non sanno niente;
* diffusori: persone che diffondono il rumor;
* soffocatori: persone che impediscono la diffusione del rumor.

L’algoritmo visita ogni nodo a ogni iterazione (in modo sincrono o asincrono) seguendo un ordine casuale.

Quando un diffusore incontra un’ignorante, quest’ultimo diventa anch’esso un diffusore con una probabilità b. Se invece un diffusore incontra soffocatore, il primo diventa un soffocatore con una probabilità u. Sempre con probabilità u, due diffusori che si incontrano possono diventare soffocatori.

Che differenza presenta rispetto al classico modello SIR? La transizione tra I ed R non occorre spontaneamente ma dipende dall’interazione tra individui. Inoltre non è presente l’effetto del threshold, quindi il rumor può raggiungere un alto numero di persone se la probabilità b è bassa, sia in reti omogenee, sia in quelle eterogenee.

## Coevoluzione

La dinamica tradizionale ha dei limiti, infatti le reti sono fine e non è permessa la selezione, quindi non possiamo connettere i nodi simili e disconnettere gli altri. Cosa possiamo fare? La soluzione sono i modelli a coevoluzione, in essi i cambiamenti di opinioni possono indurre a cambiamenti nella rete.

Come funziona la dinamica precisamente?

* per ogni nodo i, selezioniamo un vicino j con un’opinione differente;
* con una probabilità p, rimuoviamo l’arco (i,j) e connettiamo i ad un altro nodo random (che non è suo vicino). Questa è la selezione;
* con una probabilità 1-p, il nodo i può cambiare opinione e prendere quella di j. Questa è l’influenza.

Sia selezione, sia influenza tendono a ridurre il numero di nodi vicini con differenti opinioni, la rete quindi raggiunge uno stato in cui si formano tanti sottogruppi disconnessi, ognuno con la sua opinione. Questo è uno scenario stabile in cui non ci sono più cambiamenti.

Quando la probabilità p è vicino a zero, l’influenza domani la selezione, quindi il sistema tenderà a omogeneizzare le opinioni. Con p vicino a uno invece, è la selezione a dominare e quindi la rete verrà suddivisa in piccole componenti connesse formati da nodi con la stessa opinione, questi nodi però è difficile che abbiano cambiato opinione, di solito tengono la stessa della configurazione iniziale.

Cosa succede se ci sono tante opinioni differenti? Partendo da un’unica grossa componente:

* con p piccolo, un’opinione diventa dominante e formerà una grossa community, le altre saranno in minoranza;
* con p grande invece, non ci saranno cambiamenti di opinione ma si formeranno tante community più o meno della stessa grandezza, una per ogni opinione.

C’è quindi un’enorme differenza di scenario in base al valore di p.

## Ricerca

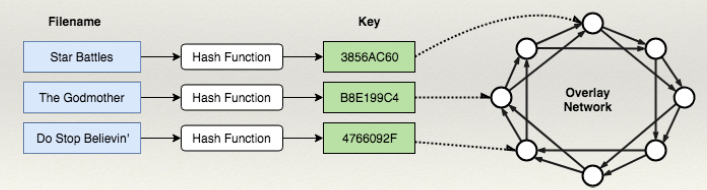
La ricerca in una rete ha bisogno di una strategia efficiente, precisamente si esplora la rete fino a quando non si raggiunge il nodo richiesto partendo da un origine.

La visita BFS è una strategia che permette di fare ciò, tuttavia è un approccio esaustivo e non va bene quando le reti presentano molti nodi. A fronte di ciò, la soluzione migliore è la ricerca locale!

## Reti peer to peer

Le reti peer to peer sono particolari reti in cui ogni computer comunica solamente con quelli direttamente connessi a esso. Come avviene la comunicazione? Ogni computer inoltra il messaggio a tutti i vicini fino a quando la destinazione non viene raggiunta.

I vantaggi del peer to peer riguardano l’assenza di un server centrale, quindi possiamo evitare il single point of failure o altre compromissioni sui singoli nodi.

Dall’altra parte, la locazione di un file è effettivamente sconosciuta, quindi bisogna chiedere ogni volta agli altri nodi usando delle query.

Dalle reti peer to peer siamo poi passati alle overlay network, una rete che utilizza una tabella hash distribuita su ogni nodo.

Quando bisogna memorizzare un file nella rete, viene generata una chiave univoca da esso attraverso la funzione hash, essa viene poi mappata su un nodo della rete. E per la ricerca del file? La ricerca utilizza la chiave e inoltra la query nella rete, fermandosi quando trova un nodo che contiene il file della chiave richiesta.

Ogni nodo ha quindi una tabella di routing contenente gli archi verso i vicini, essa viene utilizzata da un algoritmo di routing greedy per poter trovare il file col minor numero di passi possibile.

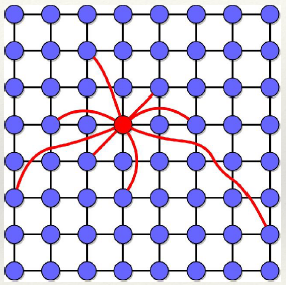
L’ultima ma non meno importante proprietà delle reti peer to peer: i nodi che entrano o abbandonano la rete non la affliggono interamente, infatti solamente i vicini vengono aggiornati.

## Ricerca locale nelle overlay network

Come avviene la ricerca locale in queste reti? Le informazioni di un nodo sono locali, quindi ognuno conosce il grado dei suoi vicini così come i file che contengono. Ogni nodo (non target) che riceve la query la inoltra al vicino con grado più grande, ciò indica che i nodi possono essere visitati più volte, è quindi opportuno marcare i nodi già visitati onde evitare loop.

Una rete è detta ricercabile quando è possibile cercare un nodo in un tempo ragionevole. Tutte le reti lo sono? Dipende, nel mondo piccolo ad esempio ci sono cammini molto corti, la bontà tuttavia sta nella ricerca fatta dai partecipanti, anche senza conoscere la struttura della rete. Questa strategia era basata sull’omofilia, dopo sono stati utilizzati metodi migliori.

## Ricercabilità geografica

Consideriamo una rete con nodi disposti a griglia, essa prende il nome di square grid ed è utile per incorporare le reti sociali nello spazio geografico. A questa rete aggiungiamo degli archi che fungono da scorciatoia. Per via del mondo piccolo, la probabilità degli archi decresce con la distanza geografica tra nodi. Ciò indica che le relazioni tra persone nella realtà avvengono quando sono geograficamente vicine.

E la ricerca? Ogni individuo conosce la posizione geografica dei suoi vicini, quindi sono in grado di indicare quale vicino è più vicino al target. A partire da una sorgente random, viene eseguito un algoritmo di ricerca greedy in cui ogni nodo viene passato al vicino più vicino al target. Da qui possiamo ricavare il delivery time, il numero di passaggi tra un nodo e l’altro che fa il messaggio.

Il delivery time diminuisce molto quando la probabilità della scorciatoia diminuisce al diminuire della distanza tra nodi.

Se la probabilità della scorciatoia diminuisce più rapidamente della distanza tra due nodi, ci saranno molti archi a lungo raggio che connettono nodi tra loro lontani.

Se invece diminuisce lentamente, ci saranno tanti piccoli cammini ma difficili da trovare.

## Ricercabilità topologica

La ricercabilità topologica utilizza gli attributi dei singoli nodi per riflettere l’omofilia nella rete e quindi facilitare la ricerca. I nodi della rete possono essere raggruppati in un ordine gerarchico basato proprio sugli attributi, ciò viene chiamato albero di distanza topologica. La radice di quest’albero rappresenta la categoria più generica e, a ogni livello si va più nello specifico, dividendo l’insieme iniziale in più sottogruppi più piccoli.

L’albero topologico permette di calcolare la distanza topologica tra due nodi, precisamente:

* se due nodi sono nello stesso sottoinsieme, la distanza è uno;
* altrimenti è il numero di livelli da risalire fino ad arrivare ad un antenato comune.

Ogni individuo può stimare la propria distanza da chiunque altro. Inoltre l’albero topologico cattura anche l’omofilia, quindi la probabilità di un arco decresce quando cresce la distanza topologica.