Reti come grafi dalla struttura disomogenea

Introduzione

- Il materiale descritto in queste lezioni è contenuto
  - nel capitolo 3 del testo
  - nella dispensa Communities [lucidi 12-29]
- Fino ad ora abbiamo studiato grafi che modellano reti a livello statistico
  - e, infatti, abbiamo utilizzato pesantemente il calcolo delle probabilità
- In questo modo, ci siamo disinteressati delle *peculiarità dei singoli nodi*
- piuttosto, abbiamo studiato proprietà che valevano *mediamente* per i nodi
- Cominciamo con oggi una serie di lezioni in cui cambiamo sostanzialmente il punto di vista
- analizziamo la posizione dei singoli nodi all'interno della rete per studiare
  - se è possibile evidenziare particolari strutture nella rete
  - se tutti i nodi si trovano, grosso modo, nella stessa situazione all'interno della rete, oppure vi sono delle differenze osservabili fra nodo e nodo
  - e se i nodi possono trarre vantaggio (e di quale tipo) da queste differenze

L’esperimento di Granovetter

- Negli anni ‘60 il sociologo statunitense Mark Granovetter, nel preparare la sua tesi di dottorato, intervistò un gruppo di individui che avevano recentemente cambiato lavoro
- Granovetter fece loro una serie di domande volte a capire in che modo erano venuti a conoscenza della possibilità di ottenere l’impiego che avevano ottenuto
  - come avevano saputo dell’esistenza dell’azienda che poi li aveva assunti
  - come avevano saputo che quell’azienda cercava personale con le loro competenze
  - ecc.
- Il risultato fu che molti degli intervistati avevano avuto le informazioni che li avevano condotti alla loro occupazione attuale attraverso una comunicazione da parte di qualcuno che conoscevano
  - una sorta di passa parola
  - e questo può ben essere prevedibile
- Il fatto abbastanza inaspettato che emerse dalle sue interviste fu che spesso l’informazione era arrivata da qualcuno che conoscevano superficialmente – non dagli amici più stretti

L'esperimento di Granovetter

- ***Spesso l'informazione era arrivata da qualcuno che conoscevano superficialmente – non dagli amici più stretti***
- Sorprendentemente, perché uno si aspetta che i nostri più cari amici sono quelli che più si darebbero da fare per aiutarci nella ricerca di un lavoro
- Allora, l'esperimento di Granovetter sembra suggerire che
  - non è tanto la forza del legame quella che ha aiutato a trovare lavoro
  - **quanto il tipo di informazione che una relazione è capace di veicolare**
- In effetti, pensandoci bene, i nostri più cari amici frequentano, grosso modo, gli stessi posti che frequentiamo noi
  - se siamo così tanto amici vuol dire che abbiamo tante cose in comune e che ci piace trascorrere il tempo insieme
- e, così, noi e i nostri grandi amici abbiamo accesso alle stesse fonti di informazione
- invece, i nostri conoscenti (alla lontana) frequentano posti diversi da quelli che frequentiamo noi, cosicché hanno accesso a fonti di informazione diverse
  - provenienti da "zone" diverse della rete, alle quali non avremmo accesso senza il loro aiuto
- e, così, ampliano il nostro "raggio d'azione"…

Bridges e Local Bridges

- Pensandoci bene, i nostri più cari amici frequentano, grosso modo, gli stessi posti che frequentiamo noi – avendo accesso alle stesse nostre fonti di informazione
- invece, i nostri conoscenti frequentano posti diversi da quelli che frequentiamo noi, cosicché hanno accesso a fonti di informazione diverse
  - provenienti da "zone" diverse della rete, alle quali non avremmo accesso senza il loro aiuto
- Se consideriamo il grafo G=(V,E) che modella la rete, gli archi che "aumentano" le nostre informazioni sono quelli che ci collegano a regioni della rete *inaccessibili* ai nostri amici
- allora, definiamo un ***bridge*** come un arco la cui rimozione disconnette la rete
  - segue dall'esperimento di Granovetter che i bridge sono gli archi che hanno maggiore "valore informativo"
- D'altra parte, sappiamo che una rete contiene con buona probabilità componenti giganti densamente connesse
  - e, dunque, la presenza di bridge è poco probabile
- pertanto, rilassiamo la definizione: un arco è un *local bridge* se i suoi vicini non hanno vicini in comune: **(u,v) $\in$ E è un *local bridge* se N(u) $\cap$ N(v) = $\emptyset$**
  - dove N(u) è l'insieme dei nodi adiacenti al nodo u

Legami forti e deboli

- L'esperimento di Granovetter ci fornisce anche un altro tipo di indicazione
  - oltre all'individuazione del potere "informativo" di bridge e local bridge
- ***Spesso l'informazione era arrivata da qualcuno conosciuto solo superficialmente – non dagli amici più stretti***
- Allora: all'interno di una rete sociale possiamo riconoscere
  - relazioni forti
    - quelle che ci collegano agli amici stretti
  - e relazioni deboli
    - quelle che ci collegano ai semplici conoscenti
- Conseguentemente, possiamo modellare una rete di questo tipo mediante un grafo G in cui gli archi sono partizionati in due sottoinsiemi:
  - **archi forti** (***strang ties***) e **archi deboli** (***weak ties***)
- ossia, $G = (V, S \cup W)$ con $S \cap W = \emptyset$
- E, come ha mostrato l'esperimento di Granovetter, gli archi deboli veicolano informazioni alle quali non avremmo accesso tramite gli archi forti:

  e questa è la ***forza degli archi deboli***

Legami forti e deboli e chiusura triadica

- Pensandoci bene, i nostri più cari amici frequentano, grosso modo, gli stessi posti che frequentiamo noi
  - e, così, i nostri amici hanno buone probabilità di entrare in contatto fra loro
- Perciò se c'è una relazione di amicizia fra a e b, e una relazione di amicizia fra a e c
  - ossia, ***se (a,b) e (a,c) sono archi forti***
- ***allora è probabile che prima o poi si creerà anche l'arco (b,c)***
- ossia, si genererà una **chiusura triadica**
  - che abbiamo già incontrato parlando di Small World
- e, quindi, per descrivere queste reti abbiamo bisogno di grafi dinamici
  - ossia, di grafi che evolvono nel tempo
- Alla base della creazione di chiusure triadiche possiamo individuare ragioni di
  - *opportunità* di incontrarsi – come abbiamo già visto
  - *incentivo* a stringere una relazione – se voglio andare al cinema con il mio amico a e lui ci va sempre con il suo amico b, mi conviene diventare amico di b
  - *fiducia* – se mi fido di a che si fida di b, posso fidarmi anch'io di b

Stabilità e proprietà della chiusura triadica

- se (a,b) e (c,d) sono **archi forti** (***strang ties***) allora è probabile che prima o poi si creerà anche l'arco (b,c)
  - ossia, è probabile che si generi una **chiusura triadica**
- Ora, lasciamo che la rete evolva
  - che si formino via via triangoli – ossia, chiusure triadiche
- fino a raggiugere una configurazione *stabile*
  - in cui la rete non cambia più nel tempo
- Cosa possiamo dire della struttura del grafo G = (V, S ∪ W) corrispondente a una rete in una configurazione stabile?
  - Che, probabilmente, tutti i triangoli possibili si sono formati
- ossia, che tutte le coppie di nodi collegati da un arco forte a uno stesso nodo sono fra loro collegati

Stabilità e proprietà della chiusura triadica

- Abbiamo lasciato evolvere la ret fino a raggiugere una configurazione *stabile*
  - in cui la rete non cambia più nel tempo
- Possiamo pensare che, a questo punto, probabilmente, tutti i triangoli possibili si sono formati
- ossia, che tutte le coppie di nodi collegati da un arco forte a uno stesso nodo sono fra loro collegati

- Formalizziamo questo concetto:
- Sia G = (V, S ∪ W); un nodo u soddisfa la ***proprietà della chiusura triadica forte*** (Strong Triadic Closure Property, in breve STCP) se, per ogni coppia di archi forti incidenti su u, i loro estremi sono collegati da un arco:
  - **$u \in V$ soddisfa la STCP se $\forall (u,v) \in S\ \forall (u,z) \in S\ [\,(v,z) \in S \cup W\,]$**
- G soddisfa la STCP se tutti i suoi nodi la soddisfano

STCP e Local Bridges

- **Teorema**: sia G = (V, S ∪ W); se un nodo u ∈ V soddisfa la STCP e se esistono due nodi distinti x e z tali che (u,x) ∈ S e (u,z) è un local bridge, allora (u,z) ∈ W
  - Dimostrazione. Poiché (u,z) è un local bridge
    - allora N(u) ∩ N(z) = ∅;
  - se fosse (u,z) ∈ S, allora, poiché (u,x) ∈ S e u soddisfa la STCP, dovrebbe essere (x,z) ∈ S ∪ W,
    - ossia, sarebbe x ∈ N(u) ∩ N(z)
    - ossia N(u) ∩ N(z) ≠ ∅ - una contraddizione. QED
- Questo teorema mostra come *una proprietà globale (essere un local bridge) si riflette in una proprietà locale (essere un weak tie)*
  - essere weak tie è proprietà locale: un nodo sa da solo se un suo vicino è un amico o un conoscente
  - essere local bridge è una proprietà globale: un nodo deve chiedere a tutti i suoi vicini se qualcuno di loro conosce un suo conoscente per sapere se è un local bridge
- E, in effetti, i weak ties dell'esperimento di Granovetter, in quanto portatori di nuova informazione, erano anche local bridges

Legami forti, chiusura triadica e clusters

- se (a,b) e (c,d) sono **archi forti** (***strang ties***) allora è probabile che prima o poi si creerà anche l'arco (b,c)
  - ossia, è probabile che si generi una **chiusura triadica**
  - che abbiamo già incontrato parlando di Small World
- E, naturalmente, una chiusura triadica dopo l'altra, si formerà nella rete ***un gruppo di nodi fortemente coeso***
  - con un elevato grado di interconnessione fra i nodi che lo compongono
- e, a questo gruppo coeso, diamo il nome di ***cluster*** *o* ***comunità***

- Sì, ma quando è che possiamo dire che un gruppo di nodi è *abbastanza* coeso per poterlo definire una comunità?

Comunità e coefficiente di clustering

- Sì, ma quando è che possiamo dire che un gruppo di nodi è *abbastanza* coeso per poterlo definire una comunità?
- Intanto, per misurare il grado di coesione di un nodo u all'interno di un gruppo di nodi è stato definito il ***coefficiente di clustering* c(u)** come il rapporto fra il numero di relazioni fra vicini di u rispetto a tutte le coppie possibili di vicini di u:

$$\mathbf{c(u)} = \frac{|\{\,(\mathbf{x},\mathbf{y}) \in \mathbf{E}\colon \mathbf{x} \in \mathbf{N}(\mathbf{u}) \wedge \mathbf{y} \in \mathbf{N}(\mathbf{u})\,\}|}{\frac{|\mathbf{N}(\mathbf{u})|\,[\,|\mathbf{N}(\mathbf{u})| - \mathbf{1}\,]}{\mathbf{2}}}$$

- Informalmente, il coefficiente di clustering misura quanto un nodo è "ben inserito" all'interno della rete costituita dai suoi vicini
  - un nodo con coefficiente di clustering basso è "male" inserito nel gruppo dei suoi amici – si trova, in quel gruppo, in una posizione periferica
  - un nodo con coefficiente di clustering elevato è "molto" inserito – possiamo dire che è un elemento *centrale* nel gruppo dei suoi amici
- infatti, il coefficiente di clustering è un ***indice di centralità*** di un nodo

Comunità e coefficiente di clustering

- Per misurare il grado di coesione di un nodo u all'interno di un gruppo di nodi è stato definito il ***coefficiente di clustering* c(u)** come il rapporto fra il numero di relazioni fra vicini di u rispetto a tutte le coppie possibili di vicini di u:

$$c(u) = \frac{|\{ (x,y) \in E: x \in N(u) \wedge y \in N(u) \}|}{\frac{|N(u)|\,[\,|N(u)| - 1\,]}{2}}$$

- Informalmente, il coefficiente di clustering misura quanto un nodo è "ben inserito" all'interno della rete costituita dai suoi vicini
- Sia C un sottoinsieme di nodi: definiamo per ogni nodo u in C il ***coefficiente di clustering di u relativo a C*** come $c_C(u) = \frac{|\{ (x,y) \in E: x \in N(u) \cap C \;\wedge\; y \in N(u) \cap C\}|}{\frac{|N(u) \cap C|\,[\,|N(u) \cap C| - 1\,]}{2}}$
- Se tutti i nodi contenuti in C hanno coefficiente di clustering relativo a C *elevato*, possiamo ben pensare che C sia una comunità
  - sì, ma ***quanto*** *elevato* deve essere il coefficiente di clustering relativo a C dei nodi in C per poter definire C una comunità?
  - E poi: sì, va bene, il coefficiente di clustering sembra un concetto promettente per definire una comunità, ma le comunità non potrebbero essere definite sulla base di altri concetti?

Comunità

- Ebbene: in effetti, le comunità sono state ampiamente studiate, e di definizioni di comunità ne sono state proposta a bizzeffe
  - e di queste ne vedremo qualcuna…

- In queste lezioni parleremo di:
  - cut-communities
  - web-communities
  - metodi agglomerativi e partitivi per l'individuazione di comunità
  - e descriveremo un metodo partitivo che porterà ad individuare comunità basate su un altro concetto di centralità
    - in particolare, sul concetto di betweenness che è un indice di centralità degli archi (invece che dei nodi)

Cut-communities

- Una cut-community per un grafo G=(V,E) è un sottoinsieme *proprio* e *non vuoto* C dei nodi di G che minimizza gli ***archi del taglio***,
  - ossia, gli archi che collegano nodi in C a nodi in V-C

- Formalmente, dato un grafo G = (V,E), **C ⊂ V è una cut-community** se **C** ≠ ∅ e

  $$|\{ (u,v): u \in C \wedge v \in V\text{-}C \}| = \min_{C' \subset V:\, C' \neq \emptyset} \left(|\{ (u,v): u \in C' \wedge v \in V\text{-}C' \}| \right)$$

- Dati un grafo G=(V,E) e una coppia di nodi s e t, un *taglio minimo rispetto alla coppia (s,t)* è un sottoinsieme *proprio* e *non vuoto* C ⊂ V dei nodi di G che contiene s e non contiene t, e che minimizza gli archi del taglio

- Calcolare il taglio minimo rispetto ad una data coppia di nodi è facile
  - ad esempio, si può utilizzare l'algoritmo di Ford-Fulkerson

Cut-communities

- L'algoritmo di Ford-Fulkerson calcola il taglio minimo che separa due dati nodi
  - ossia: dati due nodi s e t, l'algoritmo calcola un sottoinsieme C di nodi tale che: $s \in C$, $t \in V - C$ e

    $$|\{ (u,v): u \in C \wedge v \in V\text{-}C \}| = \min_{C' \subset V: s \in C' \wedge t \in V - C'} (|\{ (u,v): u \in C' \wedge v \in V\text{-}C' \}|)$$

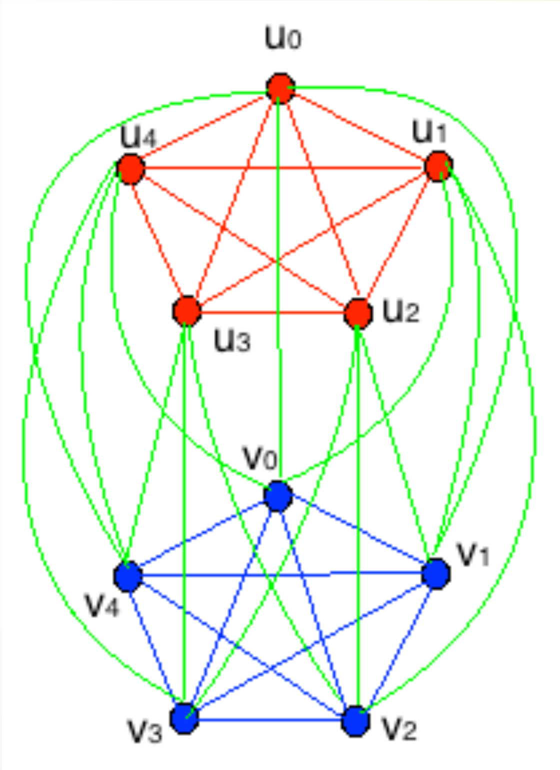
- Quindi, per calcolare una cut-community di un grafo possiamo procedere così:
  - per ogni coppia di nodi distinti $s,t \in V$: calcola l'insieme $C_{s,t}$ tale che $s \in C_{s,t}$, $t \in V - C_{s,t}$ e che minimizza il taglio
  - la cut-community cercata è il sottoinsieme $C_{x,y}$ tale che

    $$|\{ (u,v): u \in C_{x,y} \wedge v \in V\text{-} C_{x,y} \}| = \min_{s,t \in V: s \neq t} (|\{ (u,v): u \in C_{s,t} \wedge v \in V\text{-} C_{s,t} \}|)$$

- Ma tutti questi algoritmi non permettono di "controllare" i due insiemi che costituiscono il taglio. Cioè…

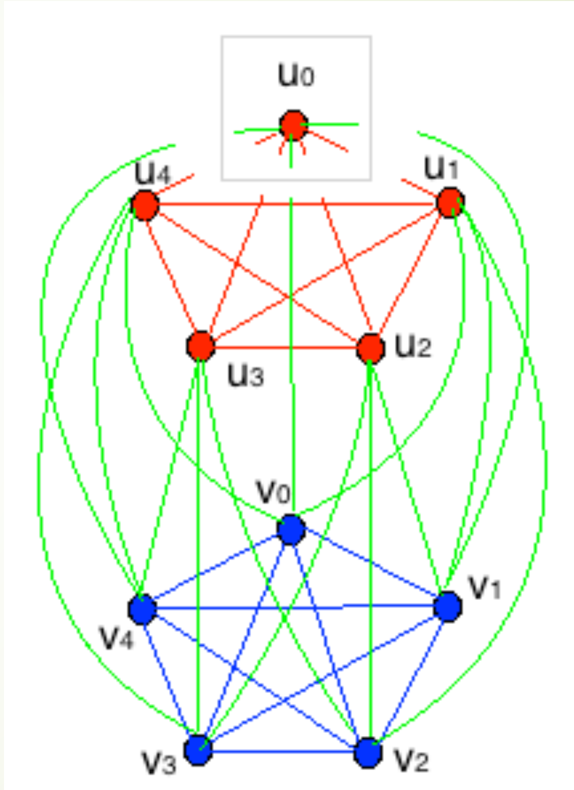
Cut-communities

- **Esempio**: Il grafo G è l'unione di due clique
  - A sui nodi u_0, u_1, u_2, u_3, u_4 (in rosso in figura)
  - B sui nodi v_0, v_1, v_2, v_3, v_4 (in rosso in figura)
- ed è completato dagli archi che collegano A e B (verdi in figura): il nodo u_i è adiacente a v_{i-1}, v_i e v_{i+1} (somme e differenze modulo 5)
  - ad esempio, u_0 è adiacente a v_4, v_0 e v_1
- Ogni nodo nel grafo ha grado 7



Cut-communities

- **Esempio**: Il grafo G è l'unione di due clique
  - A sui nodi u_0, u_1, u_2, u_3, u_4 (in rosso in figura)
  - B sui nodi v_0, v_1, v_2, v_3, v_4 (in rosso in figura)
- ed è completato dagli archi che collegano A e B (verdi in figura): il nodo u_i è adiacente a v_{i-1}, v_i e v_{i+1}
- I tagli minimi in questo grafo sono tutti e soli quelli che isolano i singoli nodi
  - che tagliano 7 archi ciascuno
- Quindi, una cut-community è un singolo nodo
- e non è una cosa molto ragionevole…



Web-communities

- Una web-community è un sottoinsieme dei nodi di un grafo ciascuno dei quali ha più vicini fra i nodi del sottoinsieme che fra quelli esterni ad esso

- Formalmente, dato un grafo G = (V,E), **$C \subset V$ è una (strong) web-community** se

  $$C \neq \emptyset \quad \text{e} \quad \forall\, u \in C: \ |N(u) \cap C| > |N(u) - C| = |N(u) \cap (V\text{-}C)|$$

  - o, equivalentemente, poiché $|N(u) \cap C| + |N(u) - C| = |N(u)|$, $\frac{|N(u) \cap C|}{|N(u)|} > \frac{1}{2}$

- Analogamente, dato un grafo G = (V,E), **$C \subset V$ è una (weak) web-community** se

  $$C \neq \emptyset \quad \text{e} \quad \forall\, u \in C: \ |N(u) \cap C| \geq |N(u) - C|$$

  - o, equivalentemente, $\frac{|N(u) \cap C|}{|N(u)|} \geq \frac{1}{2}$

- Definizione semplice e intuitivamente ragionevole!
  - Che può essere generalizzata richiedendo $\frac{|N(u) \cap C|}{|N(u)|} > \alpha$ ($\geq \alpha$), per qualche $\alpha \in [0,1]$

Cut- e weak Web-communities

- Le definizioni di cut-community e web-community non sono del tutto scorrelate
- **Teorema**: sia $G = (V,E)$ un grafo; se $C \subset V$ è una cut-community per G
  tale che $|C| > 1$
  allora C è una weak web-community per G
- Dimostrazione. Supponiamo per assurdo che C non sia una weak web-community
  - allora, esiste un nodo $u \in C$ tale che $|N(u) \cap C| < |N(u) - C|$
  - Poiché $|C| > 1$, allora esiste in C un nodo v distinto da u: ossia, $C - \{u\} \neq \emptyset$
  - inoltre: $|\{ (x,y) \in E: x \in C - \{u\} \wedge y \in (V - C) \cup \{u\} \}|$ = considero C' = C – {u} e calcolo il taglio indotto da C'

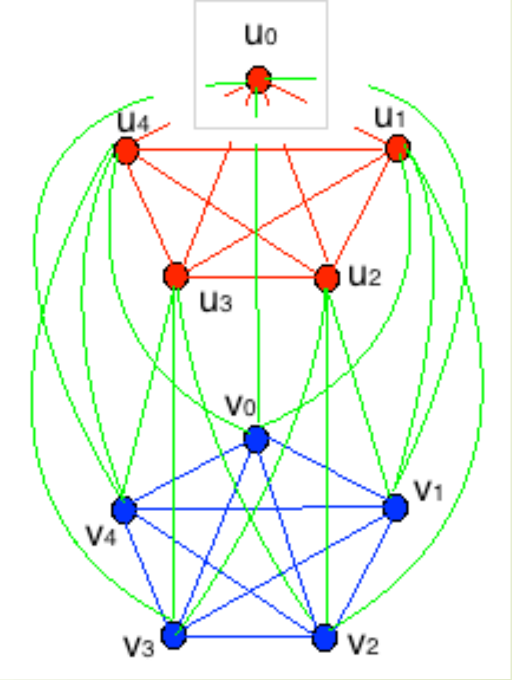
    $= |\{ (x,y) \in E: x \in C \wedge y \in V - C \}| - |\{ (u,z): z \in N(u) - C\}| + |\{ (u,z): z \in N(u) \cap C\}|$

    $= |\{ (x,y) \in E: x \in C \wedge y \in V - C \}| - |N(u) - C| + |N(u) \cap C|$

    $< |\{ (x,y) \in E: x \in C \wedge y \in V - C \}|$
  - e dunque: $C' = C - \{u\}$ è un sottoinsieme proprio e non vuoto di V e il numero di archi del taglio indotto da C' è minore del numero di archi del taglio indotto da C
- così contraddicendo l'ipotesi che C è una cut-community QED

Cut- e weak Web-communities

- Le definizioni di cut-community e web-community non sono del tutto scorrelate
  - una cut-community che contiene almeno 2 nodi è anche una web-community
- Ma abbiamo già osservato che nel calcolare una cut-community è difficile controllarne la cardinalità
- e gli algoritmi che calcolano cut-communities possono restituire comunità contenenti un solo nodo
  - come abbiamo visto nell'esempio
- e quindi poco significative
- e, inoltre, non è detto che una cut-community di un solo nodo sia anche una web-community
  - in figura, è mostrata la cut-community C = { u_0 }
  - ma l'insieme C = { u_0 } non è una web-community: u_0 ha, ovviamente, più vicini in V-C che in C !



Partizionare un grafo in comunità

- In effetti, più che individuare una singola comunità in un grafo quello che ci interessa è partizionare il grafo in comunità
- Le motivazioni per questo interesse sono molteplici
  - ad esempio, conoscere le comunità può aiutarci a capire come fluisce l'informazione nella rete (in linea con l'esperimento di Granovetter)
  - o come si diffondono idee, innovazioni, epidemie (sigh) in quella rete (e su questo torneremo a breve)
  - ma anche studiare una rete di grandissime dimensioni, riducendone la granularità (ossia, considerando le comunità come una sorta di macro-nodi e studiando il grafo dei macro-nodi)
- **Osservazione**: se C è una cut-community, allora anche V-C è una cut-community
  - perché V-C ≠ V , V-C ≠ ∅ e il taglio indotto da V-C è lo stesso di quello indotto da C
- Perciò, un algoritmo che calcola un taglio minimo individua una partizione di un grafo in due comunità
- Inoltre, se C è una cut-community con |C|>1 e |V-C|>1, allora 〈C,V-C〉 è una partizione del grafo in due weak web-communities
  - semplice generalizzazione del teorema (lasciata per esercizio)

Partizionare un grafo in comunità

- Un algoritmo che calcola un taglio minimo individua una cut-community in un grafo
- Inoltre, se C è una cut-community con |C|>1 e |V-C|>1, allora 〈C,V-C〉 è una partizione del grafo in due web-communities
  - quindi è possibile calcolare una partizione di un grafo G in due cut-communities in tempo polinomiale in |G|
  - ***Ma non possiamo garantire che sia |C|>1 e |V-C|>1***
- In effetti, calcolare una partizione di un grafo in due web-communities è un compito molto più complesso
- In effetti, mentre esiste sempre una partizione di un grafo in due cut-communities
  - perché un taglio minimo esiste sempre!
- non è detto che sia sempre possibile partizionare un grafo in due web-communities
- In effetti, decidere se un grafo è partizionabile in due web-communities è un problema NP-completo

Partizionare un grafo in due web-communities

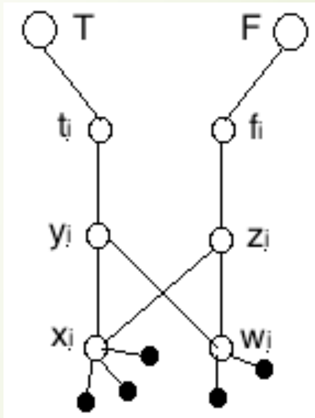
- Problema **Strong Web-Communities Partitioning (SWCP)**: dato un grafo $G=(V,E)$, decidere se esiste un sottoinsieme (proprio e non vuoto) C di V tale che C e V-C sono due strong web-communities
- **Teorema**: SWCP è NP-completo
  - Prima di dimostrare il teorema, abbiamo bisogno di un lemma
- **Lemma**: se $G=(V,E)$ è partizionabile in due strong web-communities
  e esistono $x, y, z \in V$ tali che $N(x) = \{y,z\}$ *ossia x ha grado 2 in G*
  allora per ogni $C \subset V$ tale che C e V-C sono due strong web-communities
  $x, y, z \in C$ oppure $x, y, z \in V$-C
  - Dimostrazione. Sia $C \subset V$ tale che C e V-C sono due strong web-communities
  - senza perdita di generalità, assumiamo che $x \in C$
  - se fosse $y \in V - C$ e $z \in V - C$ allora $|N(x) \cap C| = 0 < |N(x) \cap (V - C)| = 2$
  - se fosse $y \in C$ e $z \in V - C$ allora $|N(x) \cap C| = 1 = |N(x) \cap (V - C)|$
    e analogamente se fosse $y \in V - C$ e $z \in C$
  - in tutti e due i casi verrebbe contraddetta l'ipotesi che C e V-C sono due strong web-communities QED

Partizionare un grafo in due web-communities

- Problema **Strong Web-Communities Partitioning (SWCP)**: dato un grafo G=(V,E), decidere se esiste un sottoinsieme C di V tale che C e V-C sono due strong web-communities
- **Teorema**: SWCP è NP-completo
- Dimostrazione. Il problema è in NP: un certificato è un sottoinsieme $C \subset V$, e verificare che C e V-C sono strong web-communities richiede tempo polinomiale in |G|
- Per dimostrare la completezza di SWCP riduciamo ad esso 3SAT
  - siano $X = \{x_1, \dots, x_n\}$ e $f = c_1 \wedge c_2 \wedge \dots \wedge c_m$ con $c_j = \ell_{j1} \vee \ell_{j2} \vee \ell_{j3}$ e $\ell_{jh} \in X$ oppure $\neg \ell_{jh} \in X$, per $j= 1, \dots, m$ e $h = 1, 2, 3$
- costruiamo un grafo costituito da
  - due nodi "specializzati" T e F che potranno appartenere alla stessa comunità C se e soltanto se C = V (e quindi C non è una comunità, poiché non è contenuta propriamente in V)
  - un gadget per ogni variabile
  - un gadget per ogni clausola

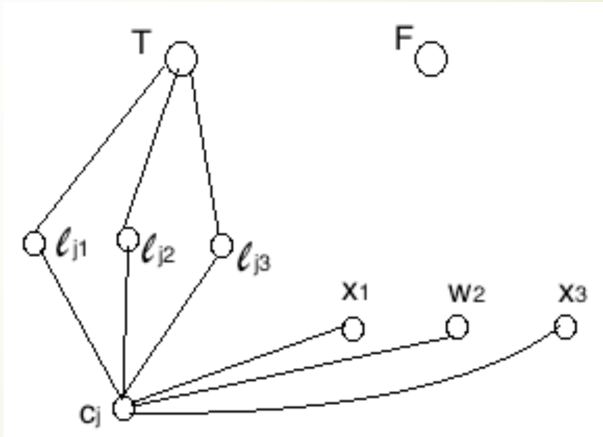
Partizionare un grafo in due web-communities

- In figura i due nodi "specializzati" T e F e il gadget per la variabile $\mathbf{x_i}$:
  - gadget che contiene i nodi x_i, w_i, y_i, z_i, t_i, f_i
  - e tanti nodi senza nome (quelli piccoli in figura): al nodo x_i (w_i) sono collegati tanti senza nome quante sono le clausole che contengono la variabile $\mathbf{x_i}$ ($\neg$ $\mathbf{x_i}$) più uno
    - nell'esempio in figura, $\mathbf{x_i}$ è contenuta in due clausole e $\neg$ $\mathbf{x_i}$ in una clausola
- **se T e F sono in due comunità distinte**: diciamo se $T \in C$ e $F \in V - C$
  - poiché t_i e f_i hanno grado 2 allora T, t_i, y_i devono essere contenuti in C e F, f_i, z_i devono essere contenuti in $V - C$
  - perché questo sia possibile **è necessario che esattamente uno dei nodi x_i e w_i sia contenuto in C e esattamente uno dei nodi x_i e w_i sia contenuto in $V - C$**
  - e, naturalmente, ciascun nodo senza nome deve essere contenuto nella stessa comunità che contiene il padre



Partizionare un grafo in due web-communities

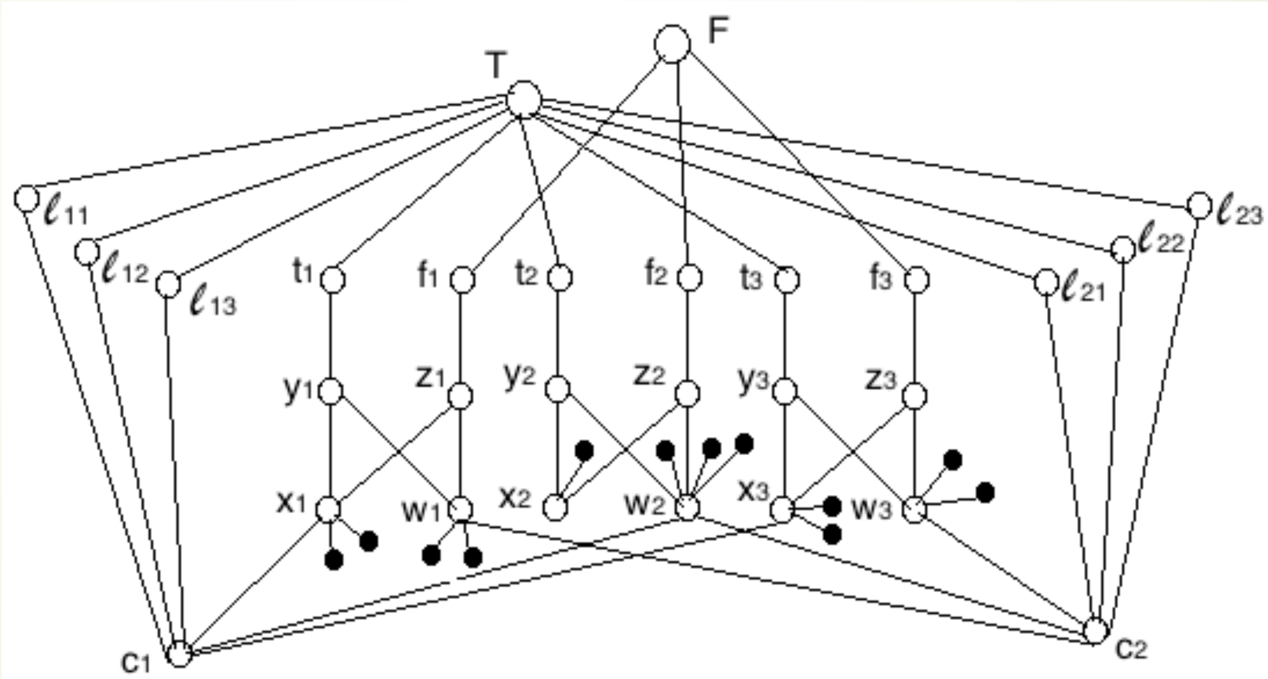
- In figura: i due nodi "specializzati" T e F, il gadget per la variabile c_j e i suoi collegamenti con i gadget variabile:
  - il gadget per la variabile c_j contiene i nodi c_j, ℓ_{j1}, ℓ_{j2}, ℓ_{j3}
  - e il nodo c_j è collegato ai letterali contenuti nella clausola c_j : al nodo x_i se c_j contiene il letterale x_i , al nodo w_i se c_j contiene il letterale $\neg x_i$
    - nell'esempio in figura, $c_j = x_1 \vee \neg x_2 \vee x_3$
- **se T e F sono in due comunità distinte**: diciamo se $T \in C$ e $F \in V - C$
  - poiché ℓ_{j1}, ℓ_{j2} e ℓ_{j3} hanno grado 2 allora T e c_j devono essere contenuti in C
  - perché questo sia possibile **è necessario che almeno uno dei nodi nei gadget variabile collegato a c_j sia contenuto in C**
    - nell'esempio in figura almeno uno fra x_1, w_2, x_3 deve essere contenuto in C
  - altrimenti c_j avrebbe tanti vicini in C quanti in V - C



Partizionare un grafo in due web-communities

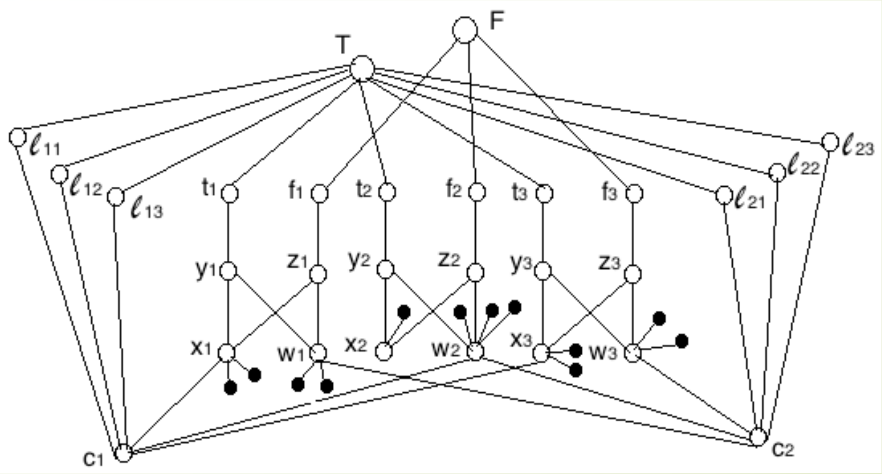
- Una visione d'insieme: in figura, la funzione $f(x_1, x_2, x_3) = c_1 \wedge c_2$

  con $\mathbf{c_1} = x_1 \vee \neg x_2 \vee x_3$ e $\mathbf{c_2} = \neg x_1 \vee \neg x_2 \vee \neg x_3$



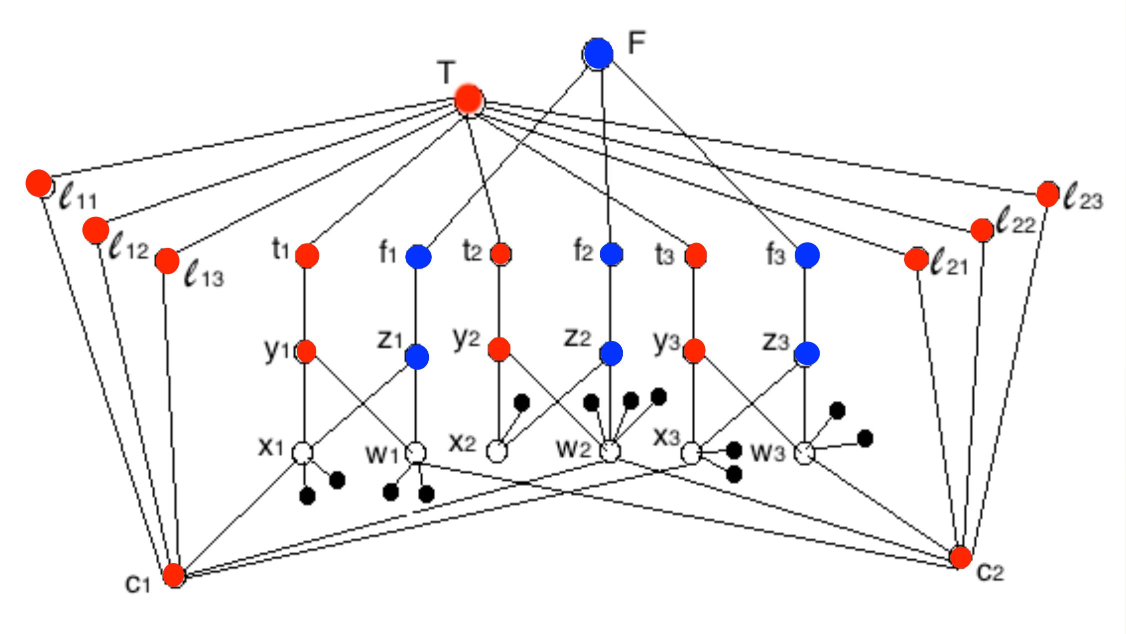
Partizionare un grafo in due web-communities

- **Se T e F sono nella stessa comunità C, allora tutti i nodi sono in C**
  - per ogni $1 \le j \le m$: poiché ℓ_{j1}, ℓ_{j2} e ℓ_{j3} hanno grado 2 allora ℓ_{j1}, ℓ_{j2} e ℓ_{j3} e c_j devono essere contenuti in C
  - per ogni $1 \le i \le n$: poiché t_i e f_i hanno grado 2 allora t_i, y_i, f_i, z_i devono essere contenuti in C
  - allora, per ogni $1 \le i \le n$: se il letterale $\mathbf{x_i}$ è contenuto in k clausole allora il nodo x_i ha k+2 vicini con nome in C
    - perciò, per poter essere inserito in V-C il nodo x_i dovrebbe avere almeno k+2 vicini in V-C
    - ma gli altri vicini di x_i sono i k+1 nodi senza nome
    - che sono k+1: non abbastanza per permettere a x_i di non far parte di C!
    - Perciò, x_i e i k+1 nodi senza nome ad esso collegati devono essere in C
  - analogamente per il letterale $\neg \mathbf{x_i}$ e il nodo w_i



Partizionare un grafo in due web-communities

- Una visione d'insieme: se T e F sono in due diverse comunità allora
  - per ogni clausola c_j i nodi c_j, ℓ_{j1}, ℓ_{j2}, ℓ_{j3} sono con T (rossi)
  - per ogni variabile x_i i nodi t_i e y_i sono con T (rossi) e i nodi f_i e z_i sono con F (blu)
  - Allora, dobbiamo ''colorare'' i nodi x_i, w_i, per ogni variabile x_i, in modo che l'insieme dei nodi rossi e l'insieme dei nodi blu siano due strong web-communities



Partizionare un grafo in due web-communities

- G è partizionabile in due strong web-communities C e V-C solo se T e F ***non*** sono entrambi in C e non sono entrambi in V - C
  - altrimenti, se T e F sono nello stesso insieme, tutti i nodi in G sono in quell'insieme!
- Affinché $T \in C$ e $F \in V - C$
  - 1) Per ogni variabile x_i in X, **esattamente uno dei nodi x_i e w_i deve essere contenuto in C e esattamente uno dei nodi x_i e w_i deve essere contenuto in V – C** (vedi pag. 25)
    - allora, ogni partizione di G in due strong web communities corrisponde ad una assegnazione di verità **a** per X: possiamo decidere, per ogni $i \in [n]$,
    - che se $x_i \in C$ (insieme con T) e $w_i \in V - C$ allora $\mathbf{a}(\mathbf{x_i})$ = vero, mentre se $x_i \in V - C$ e $w_i \in C$ allora $\mathbf{a}(\mathbf{x_i})$ = falso
    - o, viceversa, che se $x_i \in C$ (insieme con T) e $w_i \in V - C$ allora $\mathbf{a}(\mathbf{x_i})$ = falso, mentre se $x_i \in V - C$ e $w_i \in C$ allora $\mathbf{a}(\mathbf{x_i})$ = vero
    - ma in base a quale criterio scegliere fra le due opzioni?!
      - In base all'insieme nel quale collochiamo i nodi c_j (lo vedremo più avanti)..

Partizionare un grafo in due web-communities

- G è partizionabile in due strong web-communities solo se T e F ***non*** sono nella stessa comunità,
  - altrimenti, se T e F sono nella stessa comunita, tutti i nodi in G sono in quella comunità!
- Affinché $T \in C$ e $F \in V - C$
  - 1) Per ogni variabile $\mathbf{x_i}$ in X, esattamente uno dei nodi x_i e w_i deve essere contenuto in C e esattamente uno dei nodi x_i e w_i deve essere contenuto in $V - C$
    - allora, ogni partizione di G in due strong web communities corrisponde ad una assegnazione di verità **a** per X
  - 2) Per ogni clausola $\mathbf{c_j}$, il nodo c_j **deve** appartenere a C (che contiene T),
  - e perché questo sia possibile **è necessario che almeno uno dei nodi nei gadget variabile collegato a c_j sia contenuto in C** (vedi pag. 26)
    - **ossia, uno dei nodi corrispondenti a un letterale nella clausola c_j deve essere contenuto in C**
- Non resta che concludere la prova: mostriamo che G è partizionabile in due strong web-communities se e soltanto se f è soddisfacibile

Partizionare un grafo in due web-communities

- Concludiamo la prova: mostriamo che f è soddisfacibile se e soltanto se G è partizionabile in due strong web-communities
- $\Rightarrow$: se G è partizionabile in due strong web-communities C e V - C
- allora T e F ***non*** sono nella stessa comunità: sia T ∈ C e F ∈ V – C
- allora, per ogni variabile $\mathbf{x_i}$ in X, esattamente uno dei nodi x_i e w_i deve essere contenuto in C e esattamente uno dei nodi x_i e w_i deve essere contenuto in V–C
  - allora poniamo $\mathbf{a}(\mathbf{x_i})$ = vero per tutti gli $\mathbf{x_i}$ ∈ X tali che x_i ∈ C
    e $\mathbf{a}(\mathbf{x_i})$ = falso per tutti gli $\mathbf{x_i}$ ∈ X tali che x_i ∈ V - C
- inoltre, per ogni clausola $\mathbf{c_j}$, il nodo c_j deve appartenere a C,
  - e perché questo sia possibile è necessario che almeno uno dei nodi nei gadget variabile collegato a c_j sia contenuto in C
  - ossia, uno dei nodi corrispondenti a un letterale nella clausola c_j deve essere contenuto in C: sia ℓ_{jh} tale letterale
  - ossia: se $\ell_{jh} = \mathbf{x_i}$ allora x_i ∈ C e $\mathbf{a}(\mathbf{x_i})$ = vero, se $\ell_{jh} = \neg\, \mathbf{x_i}$ allora w_i ∈ C e $\mathbf{a}(\mathbf{x_i})$ = falso
- Quindi, **a** è una assegnazione di verità che soddisfa ogni clausola in f, ossia, f è soddisfacibile

Partizionare un grafo in due web-communities

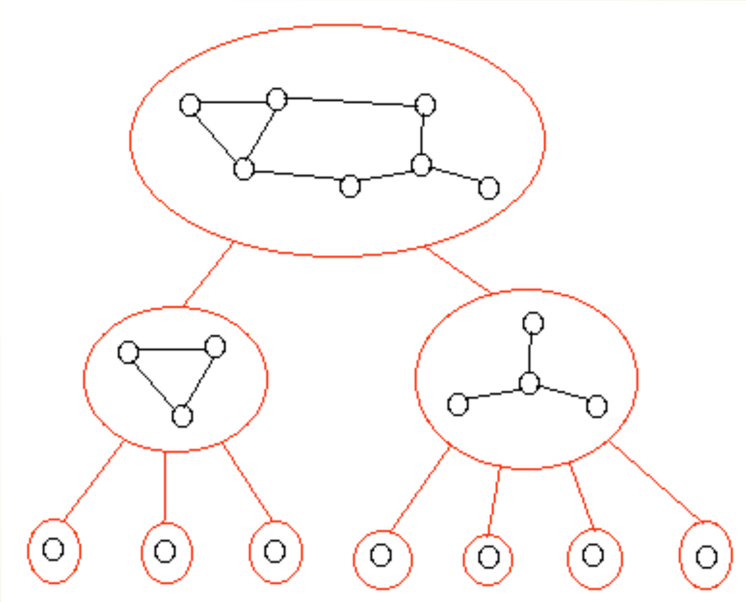
- $\Leftarrow$ se f è soddisfacibile
- sia **a** una assegnazione di verità per X che soddisfa ogni clausola **c_j** in f
- Costruiamo C: inseriamo in C
  - il nodo T e, per ogni $j \in [m]$, i nodi c_j e ℓ_{j1}, ℓ_{j2}, ℓ_{j3} e inoltre
  - per $i \in [n]$, i nodi x_i, y_i, t_i e i senza nome adiacenti a x_i tali che **a**(**x_i**) = vero
  - per $i \in [n]$, i nodi w_i, y_i, t_i e i senza nome adiacenti a w_i tali che **a**(**x_i**) = falso
- C è una comunità: lascio come facilissimo esercizio verificare che ogni nodo che abbiamo inserito in C ha un numero maggiore di vicini in C che non in V-C
- V – C è una comunità: anche questa verifica è molto semplice
  - vale soltanto la pena osservare il ruolo dei nodi senza nome:
  - se x_i è stato collocato in V-C e **x_i** compare in k clausole di f, poiché i k nodi c_j tali che **x_i** $\in$ **c_j** sono in C, allora k vicini di x_i sono in C
  - allora, se non ci fossero i nodi senza nome, x_i avrebbe k vicini in C e 1 vicino, y_i, in V-C
  - ossia, V-C non sarebbe una comunità
  - (e lo stesso dicasi per $w_i \in V – C$) QED

Partizionare un grafo in comunità: approccio euristico

- Sono stati proposti numerosi metodi “euristici” per partizionare un grafo in comunità
  - dove con “comunità” si intende ora, genericamente, un insieme coeso di nodi
  - perché, in genere, le definizioni combinatoriche che sono state proposte portano a problemi intrattabili - come abbiamo visto con le web-communities
- Per grandi linee, possiamo classificare le tecniche per il partizionamento di grafi in metodi partitivi (o divisivi) e metodi agglomerativi
- in un metodo partitivo si inizia considerando l’intero grafo come un’unica grande comunità e poi, man mano, si rimuovono gli archi fino a quando il grafo risulta partizionato in componenti connesse;
  - il procedimento viene poi iterato su ciascuna componente, fino a quando si ottiene un livello di granularità ritenuto adeguato o un insieme di comunità di dimensioni ritenute adeguate
- in un metodo agglomerativo si inizia considerando ciascun nodo come una piccola comunità e poi, man mano, si aggiungono gli archi del grafo fino a quando
  - si ottengono un numero di comunità ritenuto adeguato o comunità di dimensioni ritenute adeguate

Partizionare un grafo in comunità: approccio euristico

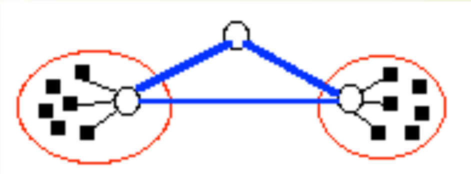
- Sia i metodi partitivi che quelli agglomerativi permettono di ottenere partizionamenti nidificati
  - in un metodo partitivo: ad ogni passo otteniamo comunità contenute in quelle ottenute al passo precedente
  - in un metodo agglomerativo: ad ogni passo otteniamo comunità che contengono quelle ottenute al passo precedente
- così che otteniamo uno schema di partizionamento ad albero



- I diversi metodi partitivi / agglomerativi proposti si distiguono per il criterio utilizzato per scegliere ad ogni passo
  - quale arco del grafo rimuovere in un metodo partitivo
  - quale arco del grafo aggiungere in un metodo agglomerativo

Betweenness di un arco

- Vediamo ora un particolare criterio per rimuovere gli archi in un metodo partitivo
- il criterio è basato sul concetto di ***betweennss*** di un arco
- a sua volta basato sui concetti di bridge e local bridge:
  - un bridge (per definizione) connette due regioni del grafo altrimenti non connesse
  - un local bridge connette due regioni che, senza di esso, sarebbero connesse in modo meno efficiente
  - perciò, possiamo dire che bridges e local bridges connettono regioni che, senza di loro, avrebbero difficoltà ad interagire
- Inoltre, abbiamo visto che bridges e local bridges sono weak ties
  - e gli archi che rimangono dopo la loro rimozione sono gli strong ties – quelli delle relazioni forti
- E da queste considerazioni nasce l'idea: rimuovendo bridges e local bridges il grafo viene partizionato in componenti che bene possono essere considerate comunità
- ma se la rete appare come in figura?
  - Nessun local bridge ma due regioni dense!



Betweenness di un arco

- Proviamo a utilizzare una proprietà diversa da quella di (local) bridge
  - anche se, come il (local) bridge, vuole ancora descrivere una crescente difficoltà di collegamento indotta dalla rimozione di un arco che la soddisfa
- essa è basata sulla nozione di traffico: gli archi che possiamo considerare i “nuovi ponti” sono quelli attraverso i quali passa più traffico
  - ragionevole!
- E il traffico lo misuriamo con una sorta di flusso di un qualche fluido:
  - per ogni coppia di nodi s e t assumiamo che s voglia inviare a t una unità di fluido
  - che, viaggiando nella rete, per raggiungere t, si suddivide equamente fra tutti gli shortest paths che collegano s a t
- La ***betweenness*** di un arco è la quantità totale di fluido che lo attraversa
  - ottenuta sommando le frazioni di fluido per tutte le coppie 〈 s ,t 〉
- Un arco è tanto più un “nuovo ponte” quanto maggiore è la sua betweenness

Betweenness di un arco

- Formalmente: dato un grafo G=(V,E) (non orientato)

- per ogni coppia di nodi s,t ∈ V e per ogni arco (u,v) ∈ E definiamo
  $\sigma_{st}(u,v)$ = numero di shortest paths fra s e t che attraversano (u,v)

- la *betweenness relativa di (u,v) rispetto alla coppia ⟨ s ,t ⟩* $b_{st}(u,v)$ è la frazione degli shortest paths fra s e t che attraversano (u,v):

$$b_{st}(u,v) = \frac{\sigma_{st}(u,v)}{\sigma_{st}}$$

  - dove σ_{st} è il numero totale di shortest paths fra s e t

- Infine, la ***betweenness* b(u,v)** di un arco (u,v) ∈ E è la semi-somma delle betweenness relative ad ogni coppia di nodi:

$$b(u,v) = \frac{1}{2} \sum_{s,t \in V} b_{st}(u,v)$$

- NB: quella che abbiamo definito è la <u>edge</u>-*betweenness*; analogamente si può definire la node-betweenness

Il metodo di Girvan-Newman

- Il metodo di Girvan-Newman è un metodo partitivo basato sulla betweenness:
  - si inizia considerando l'intero grafo come un'unica grande comunità
  - poi si calcola l'arco di betweenness massima e si rimuove: se il grafo residuo è non connesso allora è stata ottenuta una prima partizione in comunità
  - il procedimento viene poi iterato calcolando gli archi di betweenness massima nel grafo rimanente e rimuovendoli
  - il procedimento ha termine quando si è ottenuto un livello di granularità ritenuto adeguato

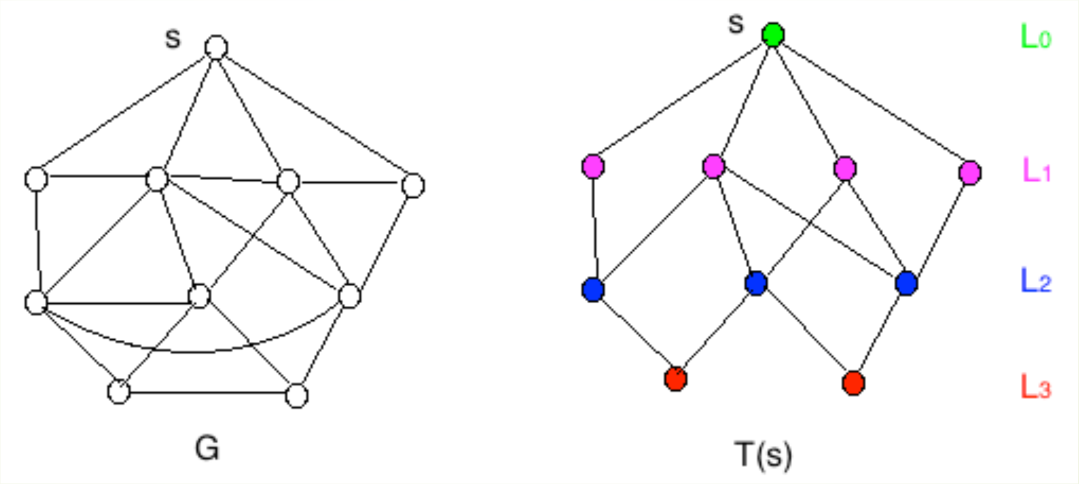
- Sì, ma come si fa a calcolare le betweenness degli archi?
  - Non possiamo mica enumerare tutti gli shortest paths di un grafo...
  - (in generale, il loro numero è esponenziale nelle dimensioni del grafo!)

Calcolo delle betweenness degli archi

- Lo schema dell'algoritmo per il calcolo delle betweeness degli archi è il seguente:
- per ogni s ∈ V esegui i seguenti tre passi
  - 1) calcola il sottografo T(s) degli shortest paths uscenti da s mediante una Breadth First Search
  - 2) mediante una visita top-down di T(s), per ogni v ∈ V calcola $\boldsymbol{\sigma_{sv}}$
  - 3) mediante una visita bottom-up di T(s), e usando quanto calcolato al punto 2), per ogni (u,v) ∈ T(s) calcola $\mathbf{b_s(u,v)} = \sum_{t \in V-\{s\}} \mathbf{b_{st}(u,v)}$
- per ogni (u,v) ∈ E calcola $\mathbf{b(u,v)} = \frac{1}{2} \sum_{s \in V} \mathbf{b_s(u,v)}$
- *Osservazione: al punto 3) calcoliamo* $\mathbf{b_s(u,v)}$ *per i soli archi (u,v) in T(s). Infatti, gli archi che non sono in T(s) non fanno parte di alcuno shortest path uscente da s*
- e ora vediamo in dettaglio i singoli passi 1), 2) e 3)

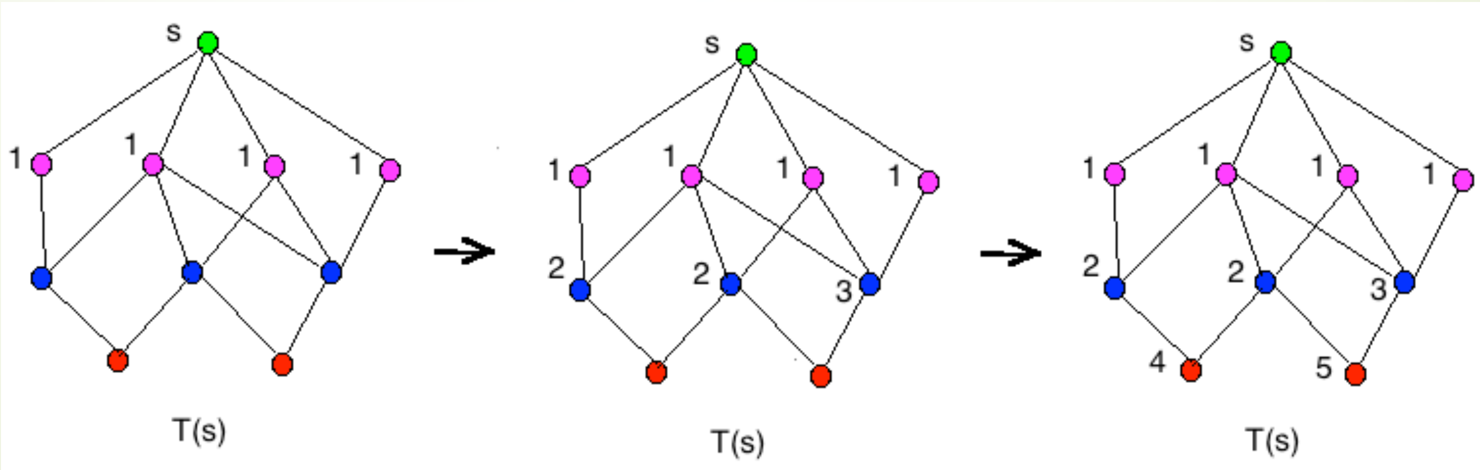
1) Breadth First Search da $s \in V$

- Calcoliamo T(s) come *insieme di archi* e, contemporaneamente, una partizione in livelli di V:
  - $L_0 \leftarrow \{s\}$ e $T(s) \leftarrow \emptyset$
  - per $h \geq 0$ e finché $L_h \neq \emptyset$ calcola: $L_{h+1} \leftarrow \{u \in V - \bigcup_{0 \leq i \leq h} L_i : \exists\, v \in L_h \text{ tale che } (v,u) \in E\}$
    e $T(s) \leftarrow T(s) \cup \{(v,u) \in E: v \in L_h \text{ e } u \in L_{h+1}\}$



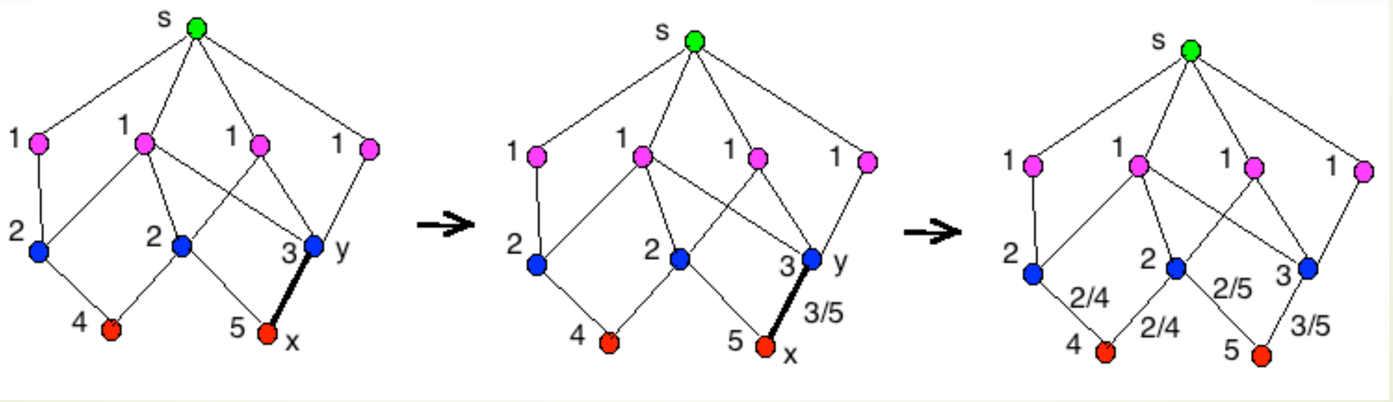
2) visita top-down di T(s)

- 2) mediante una visita top-down di T(s), per ogni v ∈ V calcola $\boldsymbol{\sigma_{sv}}$
  - il numero di shortest paths dalla radice s a un nodo v ∈ L_h è pari alla somma dei numeri dei percorsi da s a qualunque "padre" di v
  - ossia, dopo aver inizializzato $\boldsymbol{\sigma_{su}}$ **= 1 per ogni u** ∈ $\mathbf{L_1}$,
    una volta calcolato $\boldsymbol{\sigma_{su}}$ per ogni u ∈ L_{h-1} ,
  - possiamo calcolare $\boldsymbol{\sigma_{sv}} = \sum_{(u,v)\in E:\ u\in L_{h-1}} \boldsymbol{\sigma_{su}}$, per ogni v ∈ L_h



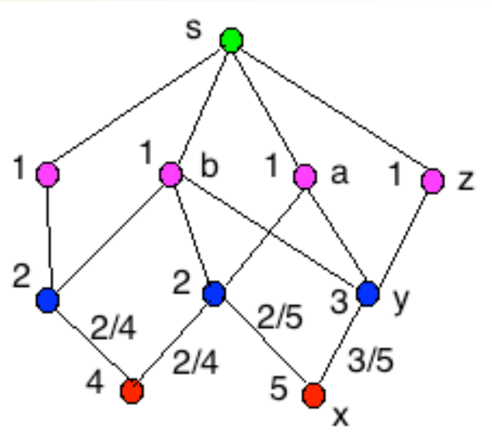
3) visita bottom-up di T(s)

- 3) per ogni (u,v) ∈ T(s), calcola $\mathbf{b_s(u,v)} = \sum_{t \in V-\{s\}} \mathbf{b_{st}(u,v)}$
- sia d il numero di livelli di T(s)
- sia (y,x) ∈ T(s) con x ∈ L_d: gli unici shortest paths uscenti da s che passano attraverso (y,x) sono gli shortest paths da s a y
  - e ciò implica anche che $\mathbf{b_{sz}(y,x) = 0}$ **per ogni** $\mathbf{z \neq x}$
- e dunque, se x ∈ L_d, $\mathbf{b_s(y,x)} = \sum_{z \in V} \mathbf{b_{sz}(y,x)} = \mathbf{b_{sx}(y,x)} = \frac{\sigma_{sx}(y,x)}{\sigma_{sx}} = \frac{\sigma_{sy}}{\sigma_{sx}}$
  - $\sigma_{sx}(y,x) = \sigma_{sy}$ perché il numero di shortest paths da s a x che attraversano (x,y) è uguale al numero di shortest paths da s a y!



3) visita bottom-up di T(s)

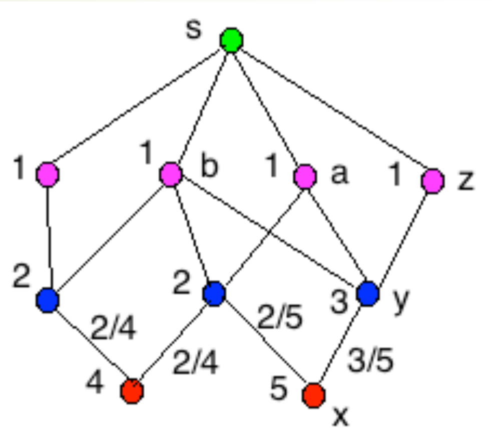
- 3) per ogni (u,v) ∈ T(s), calcola $\mathbf{b_s(u,v)} = \sum_{t \in V} \mathbf{b_{st}(u,v)}$
- sia (z,y) ∈ T(s) con y ∉ L_d: gli shortest path che passano attraverso (z,y) sono
  - *alcuni* shortest path da s a y – più precisamente, quelli che passano attraverso z
    - per lo stesso ragionamento fatto nel caso precedente
  - e, per ogni discendente x di y, *alcuni* shortest path da s a x – più precisamente, quelli che passano attraverso z e attraverso y



- Perciò, $\mathbf{b_s(z,y)}$, ossia la frazione di tutti gli shortest path uscenti da s che passa attraverso l'arco (z,y), è la somma dei seguenti termini:
  - $\frac{\sigma_{sz}}{\sigma_{sy}}$, ossia, la frazione degli shortest paths da s a y che passa attraverso (z,y)
  - per ogni discendente x di y, una frazione $\frac{\sigma_{sz}}{\sigma_{sy}}$ della frazione di shortest paths da s a x che passano attraverso (y,x), ossia, $\frac{\sigma_{sz}}{\sigma_{sy}} \times \frac{\sigma_{sy}}{\sigma_{sx}}$
  - e adesso chiariamo con l'esempio...

3) visita bottom-up di T(s)

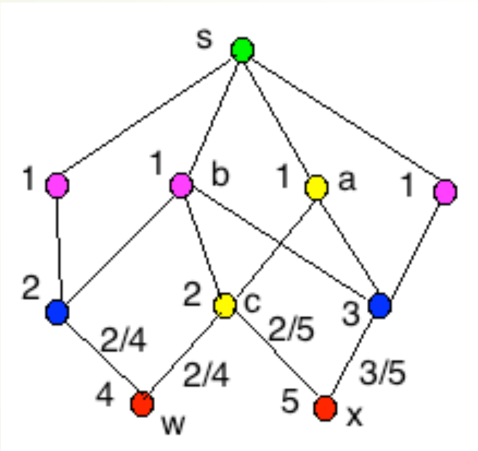
- 3) per ogni (u,v) ∈ T(s), calcola $\mathbf{b_s(u,v) = \sum_{t \in V} b_{st}(u,v)}$
- sia (z,y) ∈ T(s) con $y \notin L_d$: attraverso (z,y) passano
  - una frazione pari a $\frac{\sigma_{sz}}{\sigma_{sy}}$ degli shortest paths da s a y
  - e, per ogni discendente x di y, una frazione $\frac{\sigma_{sz}}{\sigma_{sy}}$ della frazione di shortest paths da s a x che passano attraverso (y,x)



- Esempio: i 3/5 degli shortest paths da s a x passano per (y,x)
  - di questi 3/5, 1/3 passa per (z,y), 1/3 per (a,y) e 1/3 per (b,y)
- perciò, per (z,y) passano: 1/3 degli shortest paths da s a y e 1/3 (3/5) degli shortest paths da s a x
- dunque, $\mathbf{b_s(z,y) = \frac{1}{3} + \frac{1}{3} \times \frac{3}{5} = \frac{8}{15}}$
  - perché $b_{st}(z,y) = 0$ per ogni t che non è successore di y

3) visita bottom-up di T(s)

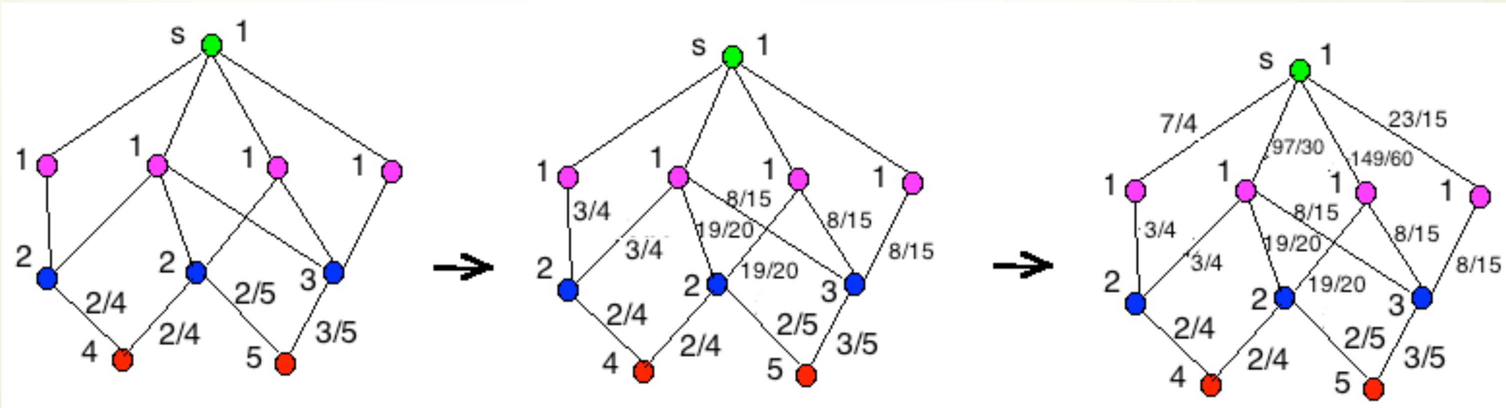
- 3) per ogni (u,v) ∈ T(s), calcola $\mathbf{b_s(u,v)} = \sum_{t \in V} \mathbf{b_{st}(u,v)}$
  - Esempio: consideriamo ora l'arco (a,c):
  - una frazione pari a $\frac{\sigma_{sa}}{\sigma_{sc}} = \frac{1}{2}$ degli shortest paths da s a c passano attraverso (a,c)
  - e, per ogni discendente di c, una frazione $\frac{\sigma_{sa}}{\sigma_{sc}} = \frac{1}{2}$ della frazione di shortest paths da s a quel discendente che passano attraverso (a,c)



- nell'esempio, i discendenti di c sono x e w
- dei 2/5 di shortest paths da s a x che passano per (c,x), 1/2 passano per (a,c) e 1/2 passano per (b,c)
- dei 2/4 di shortest paths da s a w che passano per (c,x), 1/2 passano per (a,c) e 1/2 passano per (b,c)
- perciò, per (a,c) passano:
  - 1/2 degli shortest paths da s a c,
  - 1/2 (2/5) degli shortest paths da s a x
  - e 1/2 (2/4) degli shortest paths da s a w
- dunque, $\mathbf{b_s(a,c)} = \frac{1}{2} + \frac{1}{2} \times \frac{2}{5} + \frac{1}{2} \times \frac{2}{4} = \frac{19}{20}$

3) visita bottom-up di T(s)

- 3) mediante una visita bottom-up di T, per ogni (u,v) ∈ T(s) calcola
  $\mathbf{b_s(u,v)} = \sum_{t \in V} \mathbf{b_{st}(u,v)}$
- Più in dettaglio:
  - for(h ← d; h > 0; h ← h -1) do
    calcola $\mathbf{b_s(u,v)}$ per ogni (u,v) ∈ T(s) tale che v ∈ L_h : $\mathbf{b_s(u,v)} = \frac{\sigma_{su}}{\sigma_{sv}} + \frac{\sigma_{su}}{\sigma_{sv}} \sum_{(v,x) \in T(s)} \mathbf{b_s(v,x)}$
- è una visita bottom-up: dai livelli più in basso a salire!
- *Osservazione 1*. T(s) è un insieme di archi orientati: se v ∈ L_h e (v,x) ∈ T(s) allora x ∈ L_{h+1}
- *Osservazione 2*. Abbiamo assunto $\boldsymbol{\sigma_{ss} = 1}$ (vedi figura)



Rilassare il modello

- Strong/weak ties, bridge/non bridge: queste distinzioni sembrano un po' troppo nette…
  - tipicamente, fra le nostre relazioni, abbiamo amici più stretti di altri
  - le nostre relazioni, cioè, si differenziano per "grado di confidenza"
  - ma è poco probabile che, in generale, riusciamo a discernere in modo netto fra amici e conoscenti...
- Allora, proviamo a rilassare il modello:
- Invece di considerare un grafo $G = (V, S \cup W)$
  - nel quale gli archi modellano relazioni forti e relazioni deboli
- consideriamo un grafo con archi pesati $G = (V, E, w)$, con $w: E \rightarrow \mathbb{N}$
  - nel quale il peso di un arco è tanto maggiore quanto più forte è la relazione che esso modella

Rilassare il modello

- Analogamente, invece di utilizzare i concetti di bridge e local bridge
  - che, in una rete reale, sono davvero pochi
- consideriamo il concetto di **neighborhood overlap**: dato un arco (u,v),

$$\mathbf{NO(u,v)} = \frac{|\mathbf{N(u)} \cap \mathbf{N(v)}|}{|\mathbf{N(u)} \cup \mathbf{N(v)} - \{\mathbf{u,v}\}|}$$

- un local bridge è un arco (u,v) tale che NO(u,v) = 0

- Certo, è ragionevole rilassare il modello
- ma nel modello dal quale siamo partiti si individuava una sorta di coincidenza fra archi deboli (proprietà locale) e local bridges (proprietà globali)
  - ogni qualvolta il grafo soddisfa la STCP

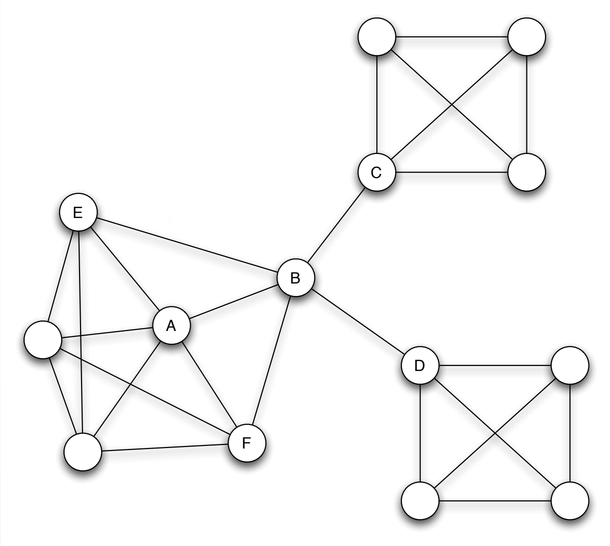
- Possiamo affermare che qualcosa di analogo valga nel modello rilassato?
- Ossia, che il neighborhood overlap di un arco è tanto più piccolo quanto minore è il suo peso?

Rilassare il modello

- [Onnela et al., 2007] ci hanno fornito evidenza sperimentale in favore di questa "coincidenza": nel loro esperimento
  - hanno considerato una rete, con archi pesati coerentemente con la "forza" delle corrispondenti relazioni
    - ad esempio, in una rete di connessioni telefoniche, il peso di un arco potrebbe essere proporzionale alla durata complessiva delle comunicazioni di due utenti
  - poi hanno iniziato a rimuovere gli archi a partire da quelli più pesanti e a seguire, via via, quelli sempre più leggeri
    - e si sono accorti che la rete si disconnetteva molto lentamente
  - poi hanno ripetuto l'esperimento "al contrario": hanno iniziato a rimuovere gli archi a partire da quelli più leggeri e a seguire, via via, quelli sempre più pesanti
    - e si sono accorti che la rete si disconnetteva molto più velocemente!
- L'esperimento di Onnela mostra che la corrispondenza vale anche nel modello rilassato:
  - poiché è ragionevole assumere che gli archi con neighborhood overlap più basso sono quelli la cui rimozione sconnette la rete
- allora il neighborhood overlap di un arco è tanto più piccolo quanto minore è il suo peso

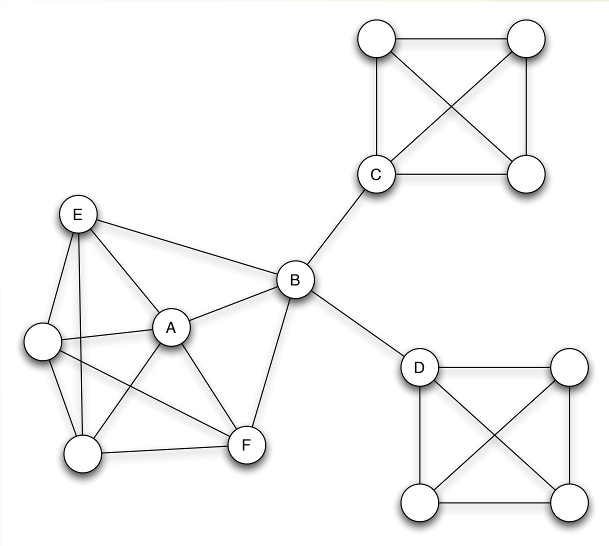
Disomogeneità

- La discussione condotta finora ci restituisce l'immagine di una rete costituita da agglomerati di nuclei fortemente connessi collegati da weak ties
  - in questa immagine gli archi non sono distribuiti uniformemente fra i nodi
- alcuni nodi sono centrali all'interno dei gruppi coesi cui appartengono
  - come il nodo A in figura
- altri nodi hanno posizioni periferiche
  - ossia, hanno un ruolo di interfaccia con altri gruppi
  - come il nodo B in figura
- Le posizioni di A e di B sono strutturalmente diverse all'interno della rete



Disomogeneità

- Le posizioni di A e di B sono strutturalmente diverse all'interno della rete
- Definiamo l'***embeddedness*** di un arco come il numero di vicini che gli estremi di quell'arco hanno in comune: **Emb(u,v) = |N(u) ∩ N(v)|**
  - se (u,v) è un local bridge allora Emb(u,v) = 0
- Tutti gli archi incidenti sul nodo A hanno embeddedness elevata
  - possiamo interpretare questa caratteristica come una situazione in cui tutti i vicini di A si fidano di A e viceversa
- Le cose per il nodo B sono parecchio diverse
  - tuttavia, anche la posizione di B ha i suoi vantaggi
- su B incidono numerosi local bridges
- ossia, B si trova al centro di un ***buco strutturale***
- e questo, come sappiamo, gli permette di avere accesso a fonti di informazione inaccessibili ai nodi centrati nelle rispettive comunità



Disomogeneità

- In conclusione:
- gli individui derivano benefici dalla struttura della rete nella quale sono immersi
  - benefici che dipendono dalla loro posizione all'interno della rete
- benefici in termini di affidabilità del tessuto sociale
- benefici in termini di contenuti informativi

- Alejandro Portes definisce ***capitale sociale*** la "capacità degli attori/agenti di assicurarsi benefici in virtù del loro essere membri di una rete o struttura sociale"