Grafi e Comunità - Struttura Disomogenea parte 2

1 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 $V - C \neq V$, $V - C \neq \emptyset$ 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)).

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.

2 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 1. *SWCP è NP-completo.*

Prima di dimostrare il teorema, abbiamo bisogno di un lemma.

Lemma 1. *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 (segue)

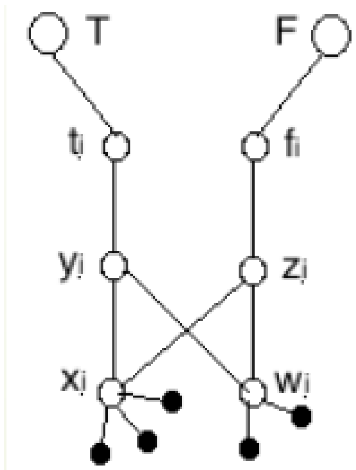
Problema Strong Web-Communities Partitioning (SWCP): dato un grafo $G = (V, E)$, decidere se esiste un sottoinsieme C di V tale che C e V-C sono due strong web-communities.

Teorema 2. *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 $X = \{x_1, ..., x_n\}$ e $f = c_1 \wedge c_2 \wedge ... \wedge c_m$ con $c_j = l_{j1} \vee l_{j2} \vee l_{j3}$ e $l_{jh} \in X$ oppure $\neg l_{jh} \in X$, per $j = 1, ..., m$, $h = 1, 2, 3$. 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.

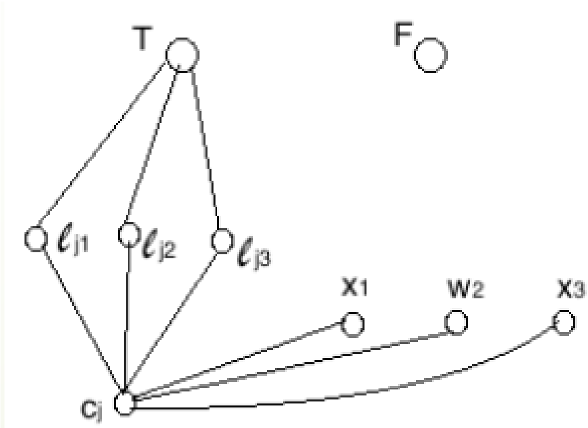
In figura



i due nodi "specializzati" T e F e il gadget per la variabile 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 x_i ($\neg x_i$) più uno. Nell'esempio in figura, x_i è contenuta in due clausole e $\neg x_i$ 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 (per il lemma) 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.

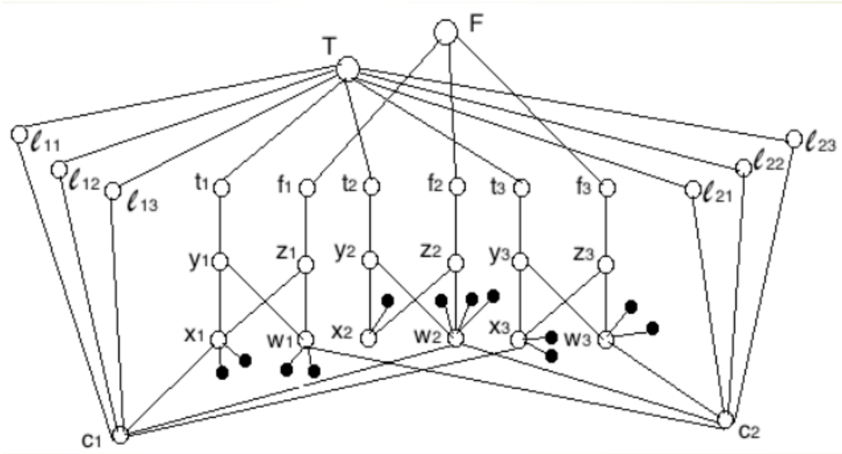
In figura:



i due nodi "specializzati" T e F, il gadget per la clausola c_j e i suoi collegamenti con i gadget variabile: il gadget per la clausola 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 (per il lemma) T, l_{j1}, l_{j2}, l_{j3} 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 $V - C$.

Una visione d'insieme: in figura,

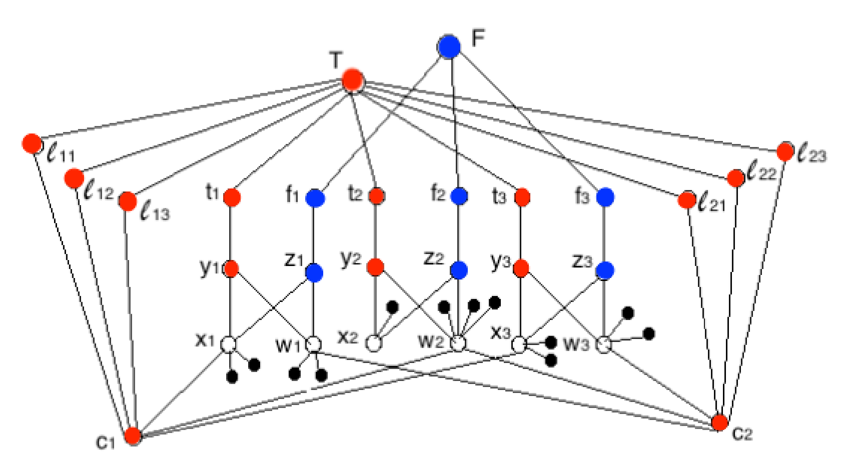


la funzione $f(x_1, x_2, x_3) = c_1 \wedge c_2$ con $c_1 = x_1 \vee \neg x_2 \vee x_3$ e $c_2 = \neg x_1 \vee \neg x_2 \vee \neg x_3$.

Se T e F sono nella stessa comunità C, allora tutti i nodi sono in C.

- Per ogni $1 \leq j \leq m$, poiché l_{j1}, l_{j2} e l_{j3} hanno grado 2, allora (per il lemma) l_{j1}, l_{j2} e l_{j3} e c_j devono essere contenuti in C.
- Per ogni $1 \leq i \leq n$, poiché t_i e f_i hanno grado 2, allora (per il lemma) t_i, y_i, f_i, z_i devono essere contenuti in C.
- Allora, per ogni $1 \leq i \leq n$: se il letterale x_i è contenuto in k clausole, allora il nodo x_i ha $k + 2$ vicini "con nome" in C (y_i, z_i e k nodi c_j).
- Perciò, per poter essere inserito in $V - C$ il nodo x_i dovrebbe avere almeno $k + 3$ 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 $\neg x_i$ e il nodo w_i.

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.

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 $V - C$.
2. Allora, ogni partizione di G in due strong web communities corrisponde ad una assegnazione di verità α 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 $\alpha(x_i) =$ vero, mentre
   - se $x_i \in V - C$ e $w_i \in C$ allora $\alpha(x_i) =$ falso
   - o, viceversa, che se $x_i \in C$ (insieme con T) e $w_i \in V - C$ allora $\alpha(x_i) =$ falso, mentre se $x_i \in V - C$ e $w_i \in C$ allora $\alpha(x_i) =$ vero.
3. Ma in base a quale criterio scegliere fra le due opzioni?!
4. In base all'insieme nel quale collochiamo i nodi c_j (lo vedremo più avanti)..

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 comunità, tutti i nodi in G sono in quella comunità!). 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 $V - C$.
2. Allora, ogni partizione di G in due strong web communities corrisponde ad una assegnazione di verità α per X.
3. Per ogni clausola 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. 27).
4. 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 (Conclusione)

Concludiamo la prova: mostriamo che f è soddisfacibile se e soltanto se G è partizionabile in due strong web-communities.

($\Leftarrow$): se G è partizionabile in due strong web-communities C e V-C. Allora T e F non sono nella stessa comunità: sia $T \in C$ e $F \in V - C$. Allora, 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. Allora poniamo:

- $\alpha(x_i)$ = vero per tutti gli $x_i \in X$ tali che $x_i \in C$
- $\alpha(x_i)$ = falso per tutti gli $x_i \in X$ tali che $x_i \in V - C$

Inoltre, per ogni clausola 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 $l_{jh} = x_i$ allora $x_i \in C \Rightarrow \alpha(x_i)$ = vero; se $l_{jh} = \neg x_i$ allora $w_i \in C \Rightarrow x_i \in V - C \Rightarrow \alpha(x_i)$ = falso. Quindi, α è una assegnazione di verità che soddisfa ogni clausola in f, ossia, f è soddisfacibile.

($\Rightarrow$): se f è soddisfacibile. Sia α 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, 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 $\alpha(x_i)$ = vero
- per $i \in [n]$, i nodi w_i, y_i, t_i e i senza nome adiacenti a w_i tali che $\alpha(x_i)$ = 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 (quindi $\alpha(x_i)$ = falso) 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. $\Rightarrow$ ossia, V-C non sarebbe una comunità. (e lo stesso dicasi per $w_i \in V - C$). $\square$

QED.