Università degli Studi di Padova
Dipartimento di Scienze Statistiche
Corso di Laurea in
Statistica per le Tecnologie e le Scienze

Metodi Bootstrap per l'Inferenza su Grafi Casuali

Relatore Prof. Matteo Grigoletto
Dipartimento di Scienze Statistiche

Laureando Alex John Caldarone
Matricola 2033255

Anno Accademico 2023/2024

Alla mia famiglia

Indice

Introduzione

Negli ultimi anni, grazie anche allo sviluppo delle tecnologie informatiche, si è assistito ad un aumento della mole di dati disponibile da analizzare nei contesti più disparati. In alcuni di questi, come ad esempio le scienze sociali (si pensi all'impatto dei social network), è di interesse studiare le relazioni tra i vari soggetti osservati. In questo caso le relazioni tra le unità vengono rappresentate matematicamente tramite un grafo. Questo cambiamento pone alcuni dei concetti della statistica concepita in contesti più "classici" in difficoltà in quanto non si può assumere l'indipendeza delle osservazioni di cui si dispone. Anzi, è proprio la dipendenza tra le unità osservate ad essere d'interesse.

Questo nuovo contesto in cui ci si è trovati ad operare ha fatto sì che fosse necessario rivedere alcuni degli strumenti statistici concepiti in precedenza, o di concepirne di nuovi, in modo da poterli adattare a questo particolare tipo di dato.

In questo lavoro si vuole dare una rassegna di alcuni di questi metodi, concentrandosi in particolare sul campionamento e sul bootstrap applicato ai dati di rete. Nello specifico, si andrà a studiare come ottenere degli intervalli di confidenza in modo efficiente per il grado medio di una rete (il termine rete è spesso usato come sinonimo di grafo).

La tesi è organizzata nel seguente modo. Nel primo capitolo si fornirà un'introduzione alla teoria dei grafi, insieme ad alcune statistiche descrittive utilizzabili per studiarne le caratteristiche. Successivamente, si introdurranno i grafi casuali, modelli probabilistici impiegati per descrivere la struttura dei grafi. Nel Capitolo 2 si farà una rassegna del bootstrap e della sua estensione ai dati dipendenti. Nel terzo capitolo ci si concentrerà sull'inferenza su grafi casuali. In particolare, si introdurrà brevemente l'*Exponential Random Graph Model*. Si proseguirà poi con alcuni metodi di campionamento per i grafi. Infine ci si concentrerà sul *Fast Patchwork Bootstrap* (FPB), un metodo bootstrap introdotto per ottenere degli intervalli di confidenza per il grado medio di una rete in modo efficiente.
Si concluderà poi con il Capitolo 4, in cui si applicherà il FPB ad un dataset reale.

Capitolo 1

Grafi e Grafi casuali

1.1 Grafi

1.1.1 Nozioni generali

Un grafo è definito come la coppia di insiemi $G = (V, E)$, dove V è l'insieme (finito) dei vertici (chiamati anche nodi), mentre E è l'insieme degli archi (o lati), con $E \subseteq V \times V$. Dato un grafo G, gli insiemi dei vertici e degli archi sono indicati rispettivamente con $V(G)$ e $E(G)$. Il numero di vertici presenti in un grafo G viene definito come ordine del grafo, indicato con $|V(G)|$, mentre il numero di archi è definito come la grandezza del grafo, $|E(G)|$.

Nei grafi risultano particolarmente interessanti da analizzare le proprietà di archi e vertici. Sia $G = (V, E)$ un grafo. Se $u, v \in V$ ed $e = \{u, v\} \in E$, si dice che l'arco e unisce i vertici u e v. In questo caso di dice che i vertici u (o v) e l'arco e sono incidenti. Due vertici distinti sono adiacenti se esiste un arco che li congiunge. Allo stesso modo, due archi distinti sono adiacenti se hanno un vertice in comune. Un esempio di vertici e archi adiacenti è riportato nella Figura 1.1.

Dato un vertice $v \in V$, si indica con $E(v)$ l'insieme di tutti i archi incidenti a v e con $N(v)$ l'insieme di tutti i vertici adiacenti a v.

Un grafo viene detto indiretto quando gli archi $e \in E$ sono una coppia non ordinata, cioè $e = \{v_1, v_2\}$, diretto quando $e = (v_1, v_2)$.

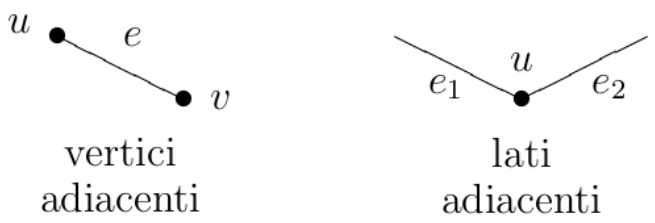


FIGURA 1.1: Esempio di vertici e archi adiacenti (*Bottacin 2008*)

I grafi, oltre ad essere rappresentati come una coppia di insiemi, possono essere rappresentati anche come delle matrici. Vi sono tre principali matrici che si possono usare per descrivere i grafi: la matrice di adiacenza, la matrice di incidenza e la matrice laplaciana.

Si consideri un grafo $G = (V, E)$. Sia $V = \{v_1, v_2, \ldots, v_n\}$ l'insieme dei vertici e $E = \{e_1, e_2, \ldots, e_m\}$ l'insieme dei archi di G. La matrice di adiacenza del grafo G, $A(G)$, è una matrice quadrata e simmetrica di dimensioni $n \times n$ con generico elemento di indice (i, j) pari a

$$a_{ij} = \begin{cases} 1 & \text{se} \quad \{v_i, v_j\} \in E \\ 0 & \text{se} \quad \{v_i, v_j\} \notin E \end{cases} \tag{1.1}$$

La matrice di adiacenza $B(G)$, invece, è una matrice $n \times m$ con generico coefficiente in posizione (i, j) pari a

$$b_{ij} = \begin{cases} 1 & \text{se vertice } v_i \text{ e arco } e_j \text{ sono incidenti} \\ 0 & \text{altrimenti} \end{cases} \tag{1.2}$$

Infine, la matrice Laplaciana di un grafo è definita come

$$L = D - A \tag{1.3}$$

con $D = diag(d(v_i))$, matrice diagonale contenente i gradi dei singoli vertici presenti all'interno del grafo. In particolare, i singoli elementi contenuti all'interno di questa matrice saranno del tipo

$$l_{ij} = \begin{cases} d(v_i) & \text{se } i = j \\ -1 & \text{se } i \neq j \text{ e } v_i \text{ adiacente a } v_j \\ 0 & \text{altrimenti} \end{cases} \tag{1.4}$$

Quest'ultima matrice ricopre un ruolo importante nell'analisi dei grafi e, in particolare, nel clustering dei vertici. Inoltre, i valori assunti dagli autovalori di questa matrice sono legati al numero di componenti connesse del grafo. Per maggiori dettagli sul clustering basato sulla matrice laplaciana si veda Kolaczyk 2009.

1.1.2 Cammini, distanze e connessione

Un cammino $P = (V', E')$ è un grafo con

$$V' = \{v_0, v_1, \ldots, v_k\},\ E' = \{v_0v_1, v_1v_2, \ldots, v_{k-1}v_k\} \tag{1.5}$$

dove i vertici v_i sono distinti. La lunghezza del cammino P è il suo numero di archi, $|E'|$. Se $v_0 = v_k$, il cammino viene chiamato ciclo. Un grafo G viene detto connesso se esiste un $u - v$ cammino che unisce qualsiasi coppia di vertici $u, v \in V(G)$. Più in generale, si parla di componente connessa di un grafo G quando esiste un sottografo H massimamente connesso e i cui vertici non sono connessi a nessun altro vertice contenuto in $G \setminus H$ (dove $G \setminus H$ è il grafo che si ottiene eliminando i vertici contenuti in H da G e tutti i archi ad essi incidenti). A volte nei dati di rete, si ha una componente connessa che domina le altre per cardinalità. In questo caso si parla di *giant component*.

La connessione in un grafo risulta essere un aspetto molto importante da analizzare, sia a livello globale che locale, in quanto può dare utili indicazioni sulla struttura del grafo ai fini dell'analisi che si sta conducendo. Di particolare rilievo sono le *clique* e i *k-core*. Una *clique* è un sottografo completo di G, mentre un *k-core* è un sottografo in cui i vertici hanno tutti almeno grado k. Queste due misure risultano però computazionalmente onerose da calcolare per grandi quantità di dati. Si preferiscono quindi altre misure per determinare quanta "connessione" c'è all'interno di un grafo.
Spesso si considera un sottografo H del grafo G e si valuta quanto questo è "denso", ovvero quante connessioni vi sono tra i vertici in quel sottografo. In un grafo G senza cappi (arco che connette un vertice a se stesso) e archi multipli, la densità del sottografo $H = (V_H, E_H)$ è definita come

$$den(H) = \frac{|E_H|}{|V_H|(|V_H| - 1)/2} \tag{1.6}$$

Considerando $H = H_v$ come l'insieme dei vicini di un vertice v, $den(H_v)$ dà un'indicazione di quanto clustering ci sia intorno al vertice v. Una misura di clustering per tutto

il grafo può essere ottenuto calcolando la media di $den(H_v)$ rispetto a tutti i vertici presenti nel grafo.

Una misura alternativa di clustering è espressa in termini di triangoli e triple connesse. Per triangolo si intende un sottografo completo di ordine 3. Per tripla connessa si intende un insieme di 3 vertici connessi da due archi. Una misura di quanto clustering ci sia in un grafo è dato dal rapporto tra il numero di triangoli e il numero di triple connesse. Sia $\tau_\Delta(v)$ il numero di triangoli in G dove v è uno dei vertici e $\tau_3(v)$ il numero di triple connesse. Si ha che $\tau_3(v) = \binom{d(v)}{2}$, con $d(v)$ il grado (numero di archi incidenti) di v. Si definisce allora il coefficiente di clustering per un vertice v come $cl(v) = \frac{\tau_\Delta(v)}{\tau_3(v)}$. A livello globale si definisce

$$cl_T(G) = \frac{\sum_{v \in V} \tau_\Delta(v)}{\sum_{v \in V} \tau_3(v)} \tag{1.7}$$

come il rapporto tra il numero di triangoli e triple connesse del grafo G.

Tralasciando ora la connessione e il clustering dei vertici, la definizione di cammino permette di definire la nozione di distanza tra due vertici in un grafo. La distanza tra due vertici u e v in un grafo G, $d_G(u, v)$, è la lunghezza del più corto cammino che unisce u e v. Se non esiste un cammino che congiunge i due vertici u e v si pone $d_G(u, v) = \infty$. La distanza tra i vertici risulta di particolare interesse nei dati di rete. Si definisca la distanza media tra vertici distinti appartenenti ad un grafo G come

$$\bar{\ell} = \frac{1}{|V|(|V|-1)/2} \sum_{u \neq v \in V} d_G(u, v) \tag{1.8}$$

Nell'analisi di alcuni dati di rete, a volte si nota che la distanza tra i vertici è molto piccola e che questi sono molto connessi tra di loro. In questo caso si parla di rete *small world*. Più formalmente, una rete viene detta *small-world* se all'aumentare del numero di vertici n, la distanza media $\bar{\ell}$ cresce al massimo in modo logaritmico rispetto ad n, ovvero $\bar{\ell} \sim O(log(n))$.

Inoltre, la definizione della distanza su un grafo permette di definire una misura di centralità dei vertici. Per centralità si intende l'importanza di un vertice all'interno del grafo di interesse. Una misura di centralità di un vertice potrebbe essere misurata, per esempio, come la vicinanza di un vertice rispetto agli altri. Vorremmo quindi una statistica che indichi quanto un vertice è lontano o vicino rispetto agli altri. L'indice di *closeness centrality* $c_{Cl}(v)$ misura la centralità di un vertice v in maniera inversa rispetto alla sua distanza da tutti gli altri,

$$c_{Cl}(v) = \frac{1}{\sum_{u \in V} d_G(v,u)} \tag{1.9}$$

Si possono definire altre misure di centralità che colgano altri aspetti del grafo. Ad esempio, la *betweenness centrality* misura quanto un vertice è collocato tra altre coppie di vertici, mentre la *eigenvector centrality* si basa sull'idea che più i vicini (dove per vicini si intendono i vertici adiacenti) di un vertice sono centrali più lo è il vertice in considerazione. Per una trattazione più approfondita di misure di centralità di vertici si veda Kolaczyk 2009.

1.1.3 Il grado dei vertici

Una caratteristica particolarmente importante dei vertici all'interno di un grafo è il loro grado.

Sia $G = (V, E)$ un grafo. Il grado di un vertice $v \in V$ è il numero di archi incidenti a v. Viene indicato con $d(v)$. Un vertice viene detto pari se il grado è pari, dispari altrimenti. Si possono definire quindi il minimo e il massimo grado osservato in un grafo rispettivamente come

$$\delta(G) = \min\{d(v) | v \in V\} \tag{1.10}$$

$$\Delta(G) = \max\{d(v) | v \in V\} \tag{1.11}$$

Disponendo delle informazioni relative al grado di ogni vertice presente in un grafo, è naturale definire il grado medio $d(G)$ del grafo G

$$d(G) = \frac{1}{|V|} \sum_{v \in V} d(v) \tag{1.12}$$

Ovviamente, essendo questa una media, si ha che

$$\delta(G) \leq d(G) \leq \Delta(G) \tag{1.13}$$

Una proprietà del grado dei vertici all'interno di un grafo che ha particolare importanza quando si decide di studiare questi oggetti con dei metodi statistici, è l'*handshaking lemma*.

Lemma 1. Sia $G = (V, E)$ un grafo. Allora la somma dei gradi dei vertici è

$$\sum_{v \in V} d(v) = 2|E|$$

Spesso è di interesse valutare se è più probabile che i vertici di grado alto siano collegati con altri vertici di grado alto piuttosto che con vertici con grado basso. Sia $G = (V, E)$ un grafo. Sia E' l'insieme degli archi diretti, tale che $|E'| = 2|E|$. Sia $(d(u), d(v))_{u,v \in E'}$ un vettore bidimensionale contenente i gradi dei vertici u e v. Si indichi con ij l'arco tra i vertici i e j. Un grafo viene detto associativo se un valore elevato di $d(u)$ corrisponde ad un valore elevato di $d(v)$, disassociativo altrimenti. Si definisce indice di associatività

$$\rho_G = \frac{\sum_{i,j \in V} \left(\mathbb{1}\{ij \in E'\} - \frac{d(i)d(j)}{|E'|} \right) d(i)d(j)}{\sum_{i,j \in V} \left(d(i)\mathbb{1}\{i = j\} - \frac{d(i)d(j)}{|E'|} \right) d(i)d(j)} \tag{1.14}$$

$$= \frac{\sum_{i,j \in V} d(i)d(j) - \left(\sum_{i \in V} d(i)^2 \right)^2 / |E'|}{\sum_{i \in V} d(i)^3 - \left(\sum_{i \in V} d(i)^2 \right)^2 / |E'|} \tag{1.15}$$

L'indice di associatività ha un'interpretazione in termini di correlazione. Si estragga casualmente un arco da E', siano X e Y rispettivamente i gradi dei vertici di partenza e arrivo dell'arco. Allora, ρ_G è il coefficiente di correlazione tra X e Y (Hofstad 2016).

1.2 I Grafi Casuali

I grafi casuali furono per la prima volta introdotti per dimostrare caratteristiche di altri grafi. Con la comparsa di grandi quantità di dati di rete negli anni '90 del secolo scorso, quest'area della matematica ha visto un notevole incremento di attività con la nascita di nuovi modelli in grado di descrivere i dati osservati. In questa sezione ci si concentrerà sulle tipologie fondamentali di grafi casuali, con particolare attenzione al grafo grafo di Erdös-Rényi, il modello di Watts-Strogratz e il *preferential attachment model*. Mentre il primo modello è quello più semplice, il secondo e il terzo sono nati in risposta all'osservazione dei fenomeni *small-world* e *scale-free* nei dati di rete.

1.2.1 Grafo casuale di Erdös-Rényi

Si consideri un insieme di n vertici distinti $V = [n] = \{1, 2, \ldots, n\}$. Sia K_n il grafo completo su $[n]$ (un grafo viene detto completo quando ogni vertice è connesso ad ogni altro vertice).

Per $0 \leq M \leq n$, lo spazio $\mathcal{G}(N, M)$ è composto da tutti i $\binom{N}{M}$ sottografi di K_n con M archi, i cui elementi sono tutti equiprobabili. Sia $G_M \in \mathcal{G}(n, M)$ un grafo appartenente allo spazio $\mathcal{G}(N, M)$, allora la probabilità che questo sia esattamente pari ad un grafo fissato H su $[n]$ con M archi è

$$\mathbb{P}_M(G_M = H) = \binom{N}{M}^{-1} \tag{1.16}$$

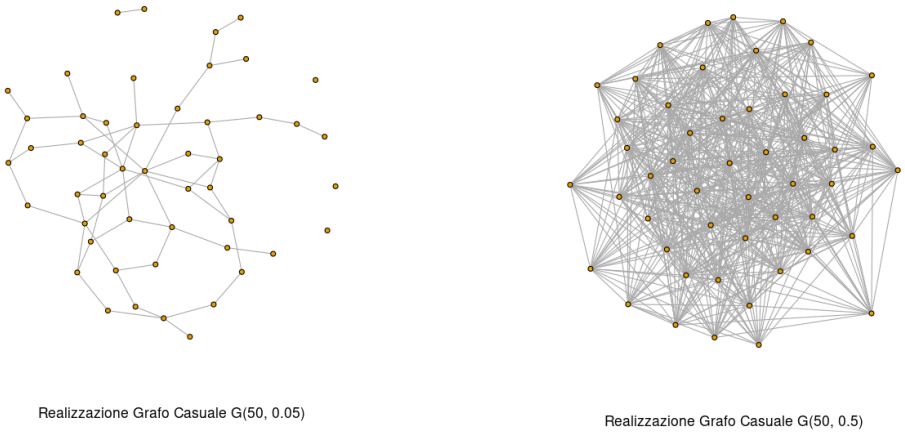
con $N = \binom{n}{2}$. Questo tipologia di grafi viene chiamata grafo casuale di Erdős-Renyi e fu introdotta in Erdös e Rényi 1959.

Un altro spazio che può essere definito è $\mathcal{G}(n, p)$, dove $0 \leq p \leq 1$ è la probabilità che vi sia un arco tra due vertici. Secondo questa definizione di grafo casuale, introdotta in Gilbert 1959, viene fissata solamente la cardinalità dell'insieme dei vertici e si ha che ogni coppia di vertici può essere connessa, indipendentemente dalle altre, da un arco con probabilità p. I grafi appartenenti a questo spazio, quindi, sono composti da n vertici e da archi scelti indipendentemente con probabilità p. In Figura 1.2 sono riportati due realizzazioni di questa tipologia di grafo casuale per due diversi valori di p.
Sia $G_p \in \mathcal{G}(n, p)$ un elemento di $\mathcal{G}(n, p)$, allora la probabilità che questo sia pari ad un grafo H definito su $[n]$ con m archi è pari a

$$\mathbb{P}_p(G_p = H) = p^m(1-p)^{n-m} \tag{1.17}$$

In questa seconda famiglia di grafi casuali, ogni vertice ha un numero atteso di archi incidenti pari a $c = (n-1)p$. Infatti, ogni vertice può essere connesso con qualsiasi degli altri $n-1$ vertici nel grafo con probabilità pari a p. Più nello specifico, il grado di un vertice segue una distribuzione binomiale. Sia $X \sim Bin(n-1, p)$ la variabile casuale che indica il grado $d(v)$ del generico vertice v che appartiene al grafo considerato. Allora

$$\mathbb{P}(X = k) = \binom{n-1}{k} p^k (1-p)^{n-1-k} \tag{1.18}$$



(a) (b)

FIGURA 1.2: A sinistra una realizzazione di un grafo casuale $\mathcal{G}(n,p)$ con $n = 50$ e $p = 0.05$. A destra una realizzazione di un grafo casuale $\mathcal{G}(n,p)$ con $n = 50$ e $p = 0.50$.

Sfruttando l'espressione ottenuta per il numero medio di archi incidenti per vertice, si può riscrivere $p = \frac{c}{n-1}$. Si considera ora la distribuzione della variabile X quando $n \to \infty$,

$$\mathbb{P}(X = k) = \binom{n-1}{k} p^k (1-p)^{n-1-k} \tag{1.19}$$

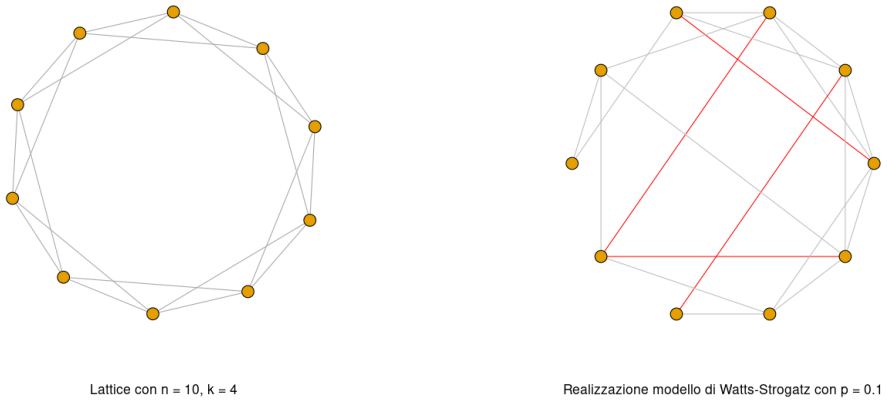
$$= \binom{n-1}{k} \left(\frac{c}{n-1}\right)^k \left(1 - \frac{c}{n-1}\right)^{n-1-k} \tag{1.20}$$

$$\approx \frac{c^k}{k!} e^{-c} \tag{1.21}$$

Quindi, quando il numero di vertici presenti nel grafo casuale tende all'infinito, la distribuzione del grado dei vertici segue una distribuzione $Poisson(c)$, con $c = (n-1)p$.

1.2.2 Modello di Watts-Strogatz

Il modello di Watts-Strogatz, introdotto in Watts e Strogatz 1998, viene introdotto per meglio descrivere il comportamento delle reti osservate rispetto al modello di Erös-Rényi. In particolare, questo modello nasce per spiegare il fenomeno osservato in molti dati di rete dove si hanno contemporaneamente un alto coefficiente di clustering e una bassa distanza media $\bar{\ell}$. In altre parole questo modello nasce per descrivere le reti cosiddette *small-world*.



(a) (b)

FIGURA 1.3: A sinistra un lattice con 10 vertici, dove ogni vertice ha altri 4 vertici adiacenti. A destra realizzazione del modello di Watts-Strogatz con p = 0.1. Sono evidenziati in rosso gli archi "deviati".

Per creare un grafo con queste caratteristiche si inizia con un lattice ad anello con n vertici, ovvero una struttura ad anello dove ogni vertice è connesso ai k vertici a lui più vicini. Un esempio di lattice è rappresentato in Figura 1.3(a). Una volta ottenuto il lattice, si modificano in modo indipendente e con probabilità p gli archi. In particolare, questi vengono deviati in modo da collegare un vertice lontano rispetto al vertice da cui parte l'arco. Una realizzazione di questo modello è rappresentata in Figura 1.3(b).

Quando $p \to 0$, si ha $\bar{\ell} \sim \frac{n}{2k}$ e $C \sim \frac{3}{4}$ (qui con C si intende la densità definita nell'equazione 1.6), mentre quando $p \to 1$, si ha $\bar{\ell} \sim \frac{log(n)}{log(k)}$ e $C \sim \frac{k}{n}$. Quindi quando $p = 0$, il lattice è una *large world*, in quando la distanza media cresce linearmente con n, con un alto coefficiente di clustering, invece quando $p = 1$ si ottiene un grafo *small-world* con un basso coefficiente di clustering.

Quando p è piccolo si hanno dei grafi che presentano sia la proprietà di *small-world* sia un alto coefficiente di clustering. A livello intuitivo questo potrebbe essere spiegato dal fatto che bastano pochi archi che connettano vertici tra di loro lontani per abbassare in modo non lineare la distanza media tra i vertici contenuti nel grafo. Questi archi non hanno un grande effetto sul coefficiente di clustering.

In Figura 1.4, tratto dal paper originale Watts e Strogatz 1998, si nota esattamente questo comportamento. In questo grafo viene considerato l'andamento della distanza media (indicata con L) e del coefficiente di clustering (indicato con C) per vari valori di p e li si confronta con i valori assunti per $p = 0$. Gli esperimenti condotti dagli autori mostrano come basti incrementare di poco p affinché questo abbia un effetto più

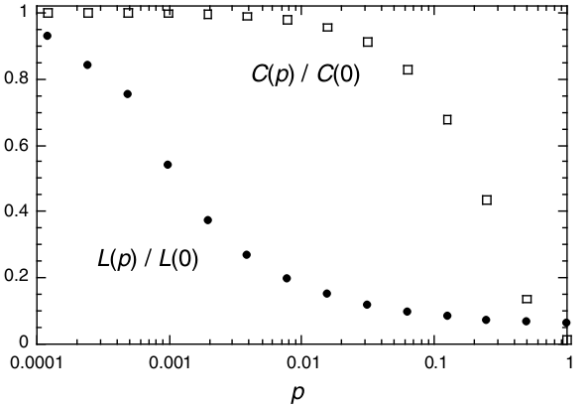


FIGURA 1.4: Andamento del coefficiente di clustering e distanza media al variare di p (*Watts e Strogatz 1998)*

che lineare nel ridurre la distanza media, mantenendo essenzialmente un coefficiente di clustering invariato.

1.2.3 Modello di Barabási-Albert

Molti dei dati di rete osservati riportano la proprietà di essere *scale-free*. Ciò vuol dire che la distribuzione del grado dei vertici anziché seguire una distribuzione di Poisson, come nel caso del grafo di Erdös-Rényi, segue una *power law*. Sia N_k il numero di vertici con grado k, allora la proprietà *scale-free* si traduce in una frequenza del tipo

$$N_k \approx c_n k^{-\tau} \tag{1.22}$$

Il modello di Barabási-Albert è un modello per descrivere la crescita di grafi non diretti. Si inizia con un grafo $G^{(0)}$ of $N_v^{(0)}$ vertici e $N_e^{(0)}$ archi. Dopodiché ad ogni iterazione t dell'algoritmo di crescita, il grafo corrente $G^{(t-1)}$ viene modificato aggiungendo un vertice di grado $m \geq 1$, con gli m archi che vengono collegati ad m vertici diversi in $G^{(t-1)}$. La probabilità che il nuovo vertice sia collegato ad un vertice v è pari a

$$\frac{d(v)}{\sum_{v' \in V} d(v')} \tag{1.23}$$

Quindi, ad ogni passo dell'algoritmo, il nuovo vertice è collegato agli m vertici presenti nel grafo in modo da preferire i vertici con un grado maggiore. Dopo t iterazioni si avrà $N_v^{(t)} = N_v^{(0)} + t$ e $N_e^{(t)} = N_e^{(0)} + tm$.

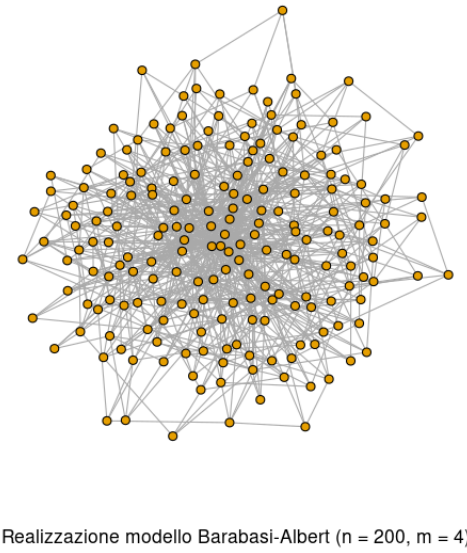
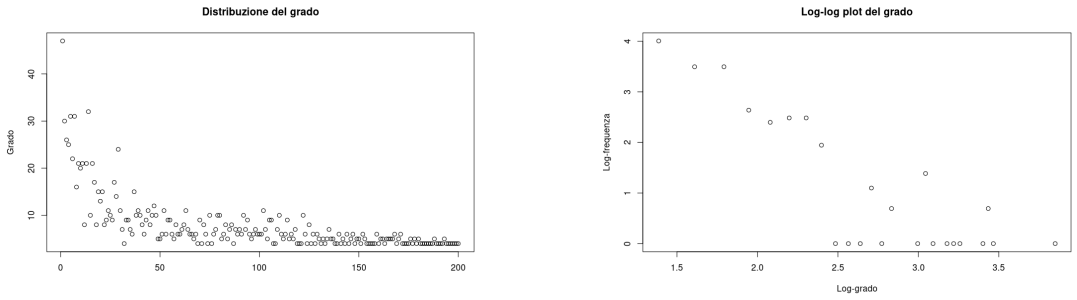


FIGURA 1.5: Realizzazione del modello di Barabási-Albert con 200 vertici e $m = 4$



(a) (b)

FIGURA 1.6: A sinistra la distribuzione del grado del grafo simulato tramite modello di Barabási-Albert. A destra il log-log plot corrispondente.

La rete generata da questo modello sarà di tipo *scale-free* in quanto i vertici con grado elevato saranno quelli che con probabilità maggiore vedranno aggiungersi un arco incidente nelle iterazioni dell'algoritmo. Empiricamente si è verificato che per questo particolare modello si ha, $N_k \approx k^{-3}$. Una realizzazione di una rete generata dal modello di Barabási-Albert è riportata in Figura 1.5. La distribuzione del grado e il log-log plot dei gradi del grafo simulato sono riportati in Figura 1.6. Risulta evidente la proprietà di *scale-free*.

Capitolo 2

Il Bootstrap

2.1 Bootstrap

Spesso, durante l'analisi di un dataset, si vuole stimare un parametro di interesse θ partendo dai dati a disposizione. Si ottiene così uno stimatore $\hat{\theta}$, funzione dei dati. Una semplice stima puntuale però può risultare insoddisfacente. In pratica, si desidera associare alla stima del parametro una misura di incertezza. Questa è spesso rappresentata da un intervallo di confidenza. Nel caso in cui si conosce la distribuzione probabilistica dalla quale provengono i dati, questa operazione risulta facile grazie alle proprietà degli stimatori di massima verosimiglianza. Tuttavia, spesso non si sa, oppure non si può ipotizzare, quale sia la legge probabilistica che ha generato i dati. Il bootstrap è un metodo che permette di approssimare la distribuzione di una statistica dei dati utilizzando solamente l'informazione contenuta nel campione, senza necessità quindi di alcuna ipotesi distributiva del fenomeno generatore dei dati.

Si supponga di avere a disposizione un campione di numerosità n, $\mathbf{y} = (y_1, \ldots, y_n)$ di osservazioni indipendenti ed identicamente distribuite secondo una distribuzione ignota F. Sia $\hat{F}$ la distribuzione empirica dei dati.
Il bootstrap si basa sull'utilizzare la distribuzione empirica dei dati come sostituto di quella teorica. Se θ è il parametro di interesse, calcolabile applicando la funzione f ai dati, una sua stima può essere ottenuta come

$$\hat{\theta} = f(\mathbf{y}) = f(y_1, \ldots, y_n) \tag{2.1}$$

Un campione bootstrap può essere ottenuto a partire dalla distribuzione empirica dei dati $\hat{F}$, dove ad ogni osservazione viene data lo stesso peso $\frac{1}{n}$. In particolare, si vanno a "ricampionare" (con reinserimento) i dati osservati, formando così un nuovo campione $\mathbf{y}^* = (y_1^*, \ldots, y_n^*)$ (il simbolo $*$ viene utilizzato per denotare il fatto che si tratta di un campione bootstrap). Questa procedura di "ricampionamento" viene effettuata B volte, ottenendo quindi i campioni bootstrap

$$\mathbf{y}_1^*, \mathbf{y}_2^*, \ldots, \mathbf{y}_B^* \tag{2.2}$$

L'i-esima replicazione dello stimatore è quindi calcolabile applicando la funzione f ai campioni bootstrap ottenuti

$$\hat{\theta}_i^* = f(\mathbf{y}_i^*) = f(y_{i1}^*, \ldots, y_{in}^*),\ i = 1, \ldots, B \tag{2.3}$$

dove y_{ij}^* è la j-esima osservazione dell'i-esimo campione bootstrap.

Una volta ottenute B replicazioni del parametro di interesse è possibile calcolare lo standard error dello stimatore bootstrap come

$$se(\hat{\theta}^*) = \sqrt{\frac{\sum_{i=1}^{B}(\hat{\theta}_i^* - \bar{\theta^*})^2}{B-1}} \tag{2.4}$$

con $\bar{\theta^*} = \frac{1}{B}\sum_{i=1}^{B}\hat{\theta}^*$.

Questa procedura illustrata finora prende il nome di *bootstrap non parametrico*, in quanto non viene fatta alcuna assunzione sulla distribuzione dalla quale provengono i dati.

A quanto visto finora si contrappone il *boostrap parametrico*. In questo contesto, si conosce la distribuzione dalla quale provengono i dati, a meno di un parametro. Questo parametro può essere stimato dal campione osservato. Una volta stimato, si ottengono i campioni bootstrap generando i dati dalla distribuzione dalla quale è stato generato il campione originale, utilizzando come parametro quello stimato sul campione osservato.

2.1.1 Intervalli di Confidenza Bootstrap

Come anticipato, spesso si vuole ottenere un intervallo di confidenza per il parametro di interesse θ. La costruzione di un intervallo di confidenza si basa sull'utilizzo di una

quantità pivotale, ovvero una funzione dei dati e del parametro θ la cui distribuzione non dipende da θ.
Una naturale quantità pivotale per costruire un intervallo di confidenza è l'errore di stima $\hat{\theta} - \theta$. Questa quantità però non risulta direttamente utilizzabile in quanto non se ne conosce la distribuzione. Per ovviare a questo problema la si sostituisce con il suo equivalente bootstrap $\hat{\theta}^* - \hat{\theta}$. All'aumentare del numero di campioni bootstrap generati la distribuzione di $\hat{\theta}^* - \hat{\theta}$ approssimerà con una precisione sempre maggiore quella della quantità $\hat{\theta} - \theta$, pertanto il suo utilizzo è ragionevole.

L'intervallo di confidenza più semplice che si può costruire utilizzando questa quantità pivotale è l'intervallo di confidenza pivotale non studentizzato.

Intervallo di confidenza pivotale non studentizzato

Sia q^*_α il generico quantile di ordine α della distribuzione bootstrap di $\hat{\theta}^* - \hat{\theta}$. Questo verrà utilizzato come surrogato del quantile di ordine α della distribuzione di $\hat{\theta} - \theta$. Un intervallo di confidenza con copertura $1 - \alpha$ si può ottenere come

$$\mathbb{P}(q^*_{\alpha/2} \leq \hat{\theta} - \theta \leq q^*_{1-\alpha/2}) \approx 1 - \alpha \tag{2.5}$$

$$\mathbb{P}(\hat{\theta} - q^*_{1-\alpha/2} \leq \theta \leq \hat{\theta} - q^*_{\alpha/2}) \approx 1 - \alpha \tag{2.6}$$

da cui si ottiene il seguente intervallo di confidenza:

$$[\hat{\theta} - q^*_{1-\alpha/2}, \hat{\theta} - q^*_{\alpha/2}] \tag{2.7}$$

Essendo che il quantile di ordine α, q^*_α, della distribuzione dell'errore di stima bootstrap è pari a $\hat{\theta}^*_{(\alpha B)} - \hat{\theta}$, dove $\hat{\theta}^*_{(\alpha B)}$ è la statistica d'ordine α, l'intervallo può essere riscritto anche come

$$[\hat{\theta} - \hat{\theta}^*_{((1-\alpha/2)B)} + \hat{\theta}, \hat{\theta} - \hat{\theta}^*_{(\alpha/2B)} + \hat{\theta}] = [2\hat{\theta} - \hat{\theta}^*_{((1-\alpha/2)B)}, 2\hat{\theta} - \hat{\theta}^*_{(\alpha/2B)}] \tag{2.8}$$

Intervallo di confidenza pivotale studentizzato

La precisione degli intervalli di confidenza pivotali non studentizzati dipende fortemente dalla somiglianza delle distribuzioni di $\hat{\theta}^* - \hat{\theta}$ e $\hat{\theta} - \theta$. Nel caso in cui queste due distribuzioni abbiano diversa varianza si può usare un intervallo di confidenza studentizzato, basato sulla quantità pivotale

$$T = \frac{\hat{\theta} - \theta}{\hat{\sigma}} \tag{2.9}$$

dove $\hat{\sigma} = sd(\hat{\theta})$. Il corrispondente bootstrap di questa quantità è

$$T^* = \frac{\hat{\theta}^* - \hat{\theta}}{\hat{\sigma}^*} \tag{2.10}$$

qui, $\hat{\sigma}^*$ è calcolata sulle realizzazione del campione bootstrap. Sia t^*_α il quantile di ordine α della distribuzione di T^*, formata dai B valori ottenuti tramite bootstrap. Utilizzando una procedura analoga a quanto fatto sopra si può ottenere un intervallo di confidenza di livello $1 - \alpha$ come

$$\mathbb{P}(t^*_{\alpha/2} \leq T \leq t^*_{1-\alpha/2}) \approx 1 - \alpha \tag{2.11}$$

$$\mathbb{P}(t^*_{\alpha/2} \leq \frac{\hat{\theta} - \theta}{\hat{\sigma}} \leq t^*_{1-\alpha/2}) \approx 1 - \alpha \tag{2.12}$$

$$\mathbb{P}(\hat{\theta} - \hat{\sigma}t^*_{1-\alpha/2} \leq \theta \leq \hat{\theta} - \hat{\sigma}t^*_{\alpha/2}) \approx 1 - \alpha \tag{2.13}$$

da cui si ottiene l'intervallo di confidenza

$$[\hat{\theta} - \hat{\sigma}t^*_{1-\alpha/2}, \hat{\theta} - \hat{\sigma}t^*_{\alpha/2}] \tag{2.14}$$

Questo intervallo di confidenza dipende da $\hat{\sigma}$, che non sempre è possibile calcolare esplicitamente. Quando questo non è possibile si ricorre ad un "bootstrap di II° livello". Con questo si intende una procedura bootstrap dove per ognuno dei B campioni bootstrap $\mathbf{y}^*$, si ricampionano ulteriori B_2 volte le osservazioni contenute in $\mathbf{y}^*$. Su ognuno di questi campioni di "II° livello" si calcola una realizzazione della statistica di interesse e ne viene calcolata la deviazione standard facendo uso delle sue B_2 replicazioni. In questo

modo, alla statistica calcolata sul campione bootstrap $f(\mathbf{y}^*)$ si riesce ad affiancare una misura di incertezza $\hat{\sigma}^*$.

Intervallo di confidenza percentile

Mentre gli intervalli di confidenza finora trattati si basavano su delle quantità pivotali, si introdurrà brevemente ora un intervallo di confidenza che non faccia uso di quantità pivotali.

Si supponga di avere a disposizione B campioni bootstrap $\mathbf{y_1}^*, \ldots, \mathbf{y_B}^*$ e le corrispondenti realizzazioni del parametro di interesse θ, $\hat{\theta}_1^*, \ldots, \hat{\theta}_B^*$. Sia $\hat{G}$ la funzione di ripartizione empirica di $\hat{\theta}^*$. L'intervallo di confidenza di livello $1-\alpha$ è definito come

$$\left[G^{-1}\left(\frac{\alpha}{2}\right),\ G^{-1}\left(1-\frac{\alpha}{2}\right)\right] \tag{2.15}$$

Essendo che $G^{-1}(\alpha)$ corrisponde al percentile di livello α della distribuzione bootstrap di $\hat{\theta}^*$, l'intervallo può essere riscritto come

$$\left[\hat{\theta}^*_{(\frac{\alpha}{2}B)}, \hat{\theta}^*_{\left((1-\frac{\alpha}{2})B\right)}\right] \tag{2.16}$$

Per ulteriori dettagli sull'intervallo di confidenza percentile e altri tipi di intervalli di confidenza bootstrap si rimanda a Efron e Tibshirani 1993.

2.2 Bootstrap per Dati Dipendenti

Finora si è ipotizzato che i valori osservati nel campione siano indipendenti. Questa ipotesi di indipendenza permette di ricampionare i dati e calcolare le quantità di interesse in modo semplice. Non sempre però si hanno osservazioni indipendenti. È questo il caso delle serie storiche (si veda Di Fonzo e Lisi 2005 per un'introduzione), che in questa sezione verranno prese come esempio di riferimento per introdurre metodi bootstrap per ricampionare dati dipendenti. Le metodologie presentate si dividono in due categorie: parametrici (o *model-based*), dove viene prima stimato un modello e vengono ricampionati residui dal modello stimato, e metodi non-parametrici, che nel caso delle serie storiche prendono il nome di metodi a "blocchi".

2.2.1 Metodi parametrici

Si consideri un processo AR(1) per semplicità, definito come

$$Y_t = \phi_1 Y_{t-1} + \varepsilon_t \tag{2.17}$$

con $\varepsilon_t \sim WN(0, \sigma^2)$. Si supponga di osservare una serie storica $y_1, \ldots, y_n$, realizzazione del modello AR(1).

Negli approcci parametrici, viene prima stimato il modello generatore dei dati, ottenendo così una stima di massima verosimiglianza $\hat{\phi}$. Successivamente, si generano i residui del modello come

$$e_t = y_t - \hat{\phi} y_{t-1}, \ t = 2, \ldots, m \tag{2.18}$$

Si noti che non è possibile calcolare il valore del residuo e_1 in quanto non si dispone dell'osservazione y_0. Una volta ottenuta la serie storica dei residui del modello $\{e_t\}$, questa viene ricampionata in modo da ottenere il campione bootstrap dei residui $e_2^*, \ldots, e_n^*$. Una volta ottenuti i residui bootstrap questi, si applica la definizione del modello AR(1) e si ottiene una nuova serie storica tramite ricorsione, ponendo il valore di $y_2^* = e_2^*$,

$$y_2^* = e_2^*, y_3^* = \hat{\phi} y_2^* + e_3^*, \ldots, y_n^* = \hat{\phi} y_{n-1}^* + e_n^* \tag{2.19}$$

Si noti che anche in questo caso non è possibile ottenere una realizzazione bootstrap per la prima osservazione della serie storica. Una volta ottenuta la realizzazione bootstrap della serie storica, su di questa viene stimata il parametro di interesse ϕ, ottenendo quindi lo stimatore bootstrap $\hat{\phi}^*$. Questa procedura di ricampionamento e stima del parametro viene eseguita B volte, ottenendo quindi le realizzazioni bootstrap del parametro di interesse $\hat{\phi}_1^*, \hat{\phi}_2^*, \ldots, \hat{\phi}_B^*$.

Questo metodo può essere facilmente esteso al processo AR(p),

$$Y_t = \phi_1 Y_{t-1} + \phi_2 Y_{t-2} + \cdots + \phi_p Y_{t-p} + \varepsilon_t \tag{2.20}$$

con $\varepsilon \sim WN(0, \sigma^2)$. Si noti che in questo caso i parametri da stimare sono p. Questo comporterà che nelle serie storiche ottenute tramite ricampionamento dei residui non sarà possibile ottenere delle realizzazioni bootstrap per le prime p osservazioni della serie.

Sotto alcune condizioni (riportate in Chernick e La Budde 2011) si ha che la stima del vettore di parametri $\hat{\phi}$ è consistente per il vero valore dei parametri ϕ. Questo garantisce

che lo stimatore boostrap è consistente per $\hat{\phi}$ e che la distribuzione dell'errore di stima $\hat{\phi}^* - \hat{\phi}$ convergerà alla distribuzione dell'errore di stima $\hat{\phi} - \phi$.

2.2.2 *Block Bootstrap*

Affinché i metodi parametrici diano dei buoni risultati, è necessaria la corretta specificazione del modello sottostante. Se questa è errata allora tutte le analisi successive risulteranno anch'esse errate. Per rimediare a questo svantaggio si sono sviluppati metodi di ricampionamento dei dati dipendenti che non necessitassero di assunzioni parametriche e che quindi possono essere applicati ad un'ampia gamma di processi stazionari. Tra questi hanno avuto particolare successi i metodi cosiddetti a "blocchi".

Il metodo di ricampionamento a blocchi più comune è il *moving block bootstrap*. L'idea fondamentale alla base di questo metodo, utilizzato per serie storiche stazionarie, è che quando delle osservazioni sono sufficientemente distanza temporalmente la loro correlazione è pressoché nulla. Questo permette di trattare blocchi che sono "lontani" in modo interscambiabile. Si consideri una serie storica di n osservazioni, dove n può essere espresso come $n = bl$, dove b e l sono interi che indicano rispettivamente il numero di blocchi e la lunghezza di ogni blocco. Va posta particolare attenzione alla scelta di l, che deve essere sufficientemente grande da catturare la struttura di dipendenza presente all'interno della serie storica. La dimensione dei blocchi (e quanta dipendenza questa cattura) risulta inoltre di fondamentale importanza per valutare la distribuzione bootstrap della statistica di interesse. I blocchi costruiti possono essere sovrapposti o meno.
Nel caso in cui si costruiscano blocchi sovrapposti di lunghezza l per una serie storica di lunghezza n, si otterrebbero i seguenti blocchi

$$(y_1, y_2, \ldots, y_{l-1}, y_l), (y_2, y_3, \ldots, y_l, y_{l+1}), \ldots, (y_{n-l}, y_{n-l+1}, \ldots, y_{n-1}, y_n) \tag{2.21}$$

Si ottengono così $n - l + 1$ blocchi che possono essere poi ricampionati per ottenere una realizzazione bootstrap della serie storica. Si nota che costruendo i blocchi in questo modo le prime e le ultime $l-1$ unità appariranno in meno blocchi rispetto alle altre. Per rimediare a questa problematica è stato introdotto il *Circular Block Bootstrap*, dove le osservazioni vengono disposte "lungo un cerchio" in modo che l'osservazione successiva ad y_n sia y_1.

Il difetto principale di cui soffrono i metodi a "blocchi" è spesso un'impossibilità nel ricostruire la struttura di dipendenza presente nella serie storica originale, che può essere

rafforzata o indebolita in base alla posizione in cui vengono posti blocchi che originariamente erano vicini. Sono stati proposte varie estensioni per cercare di porre rimedio ad eventuali svantaggi del *block bootstrap*. Una prima estensione consiste nel considerare la lunghezza del blocco l non come una quantità fissata ma come una realizzazione di una variabile causale $L \sim Geom(p)$. Un altro metodo che cerca di cogliere in modo migliore la struttura dei dati è il *post-blackening*. Il *post-blackening* consiste nello stimare un modello per la serie storica osservata. Si considerano poi i residui del modello stimato e si applica a questi il *block bootstrap*, ottenendo così una nuova serie composta da blocchi di residui. A questi viene poi applicato il modello stimato per ottenere una nuova serie. Per questo motivo, il *post-blackening* viene spesso visto come un punto di incontro tra metodi parametrici e non parametrici.

Per un'introduzione al bootstrap per serie storiche si veda Chernick e La Budde 2011, mentre una trattazione più approfondita sui diversi aspetti del bootstrap per dati dipendenti (sia per serie storiche che per dati spaziali) può essere trovata in Davison e Hinkley 1997.

Capitolo 3

Inferenza su Grafi Casuali

Ci sono essenzialmente due "paradigmi" di inferenza sulle reti. Nel primo è di interesse modellare la struttura stessa della rete, ovvero utilizzare un modello statistico che sia in grado di spiegare la configurazione della rete. Nel secondo approccio, invece, si è interessati a condurre inferenza su un parametro della rete G di interesse.
In questo capitolo si introdurrà prima l'ERGM (*Exponential Random Graph Model*), una classe di modelli usata per modellare la struttura delle reti. Nelle sezioni successive ci si concentrerà invece sul campionamento di vertici e archi in reti di grandi dimensioni e, infine, si introdurrà il Fast Patchwork Bootstrap, un metodo bootstrap non parametrico utilizzato per fare inferenza sul grado medio di una rete.

3.1 *Exponential Random Graph Model*

Si consideri un grafo y di ordine n, dove si indica con $V = \{1, 2, \ldots, n\}$ l'insieme dei vertici. Questi possono essere connessi tra di loro mediante degli archi y_{ij} (dove con y_{ij} si intende l'arco che connette i vertici i e j). Inoltre, ogni vertice può avere associato un vettore di covariate $\mathbf{x}$, che ne descrive le caratteristiche. Quando si costruisce un modello per studiare la struttura di un grafo, si considerano gli archi realizzazioni di variabili casuali Y_{ij}. Si indica con $\mathcal{Y}$ lo spazio campionario, ovvero l'insieme all'interno del quale le variabili Y_{ij} sono definite e possono assumere valori. Nel caso di un grafo non diretto di ordine n, si ha $\mathcal{Y} = \{0, 1\}^{\binom{n}{2}}$.

Un *Exponential Random Graph Model* (o ERGM) per y è definito come

$$\mathbb{P}(Y = y; \theta) = \frac{\exp\{\theta^T g(y)\}}{k(\theta)}, \quad y \in \mathcal{Y} \tag{3.1}$$

dove, $k(\theta) = \sum_{y \in \mathcal{Y}} \exp\{\theta^T g(y)\}$ è una costante di normalizzazione, $\theta \in \Theta \subseteq \mathbb{R}^p$ è un vettore p-dimensionale di parametri, $g(y)$ è un vettore p-dimensionale di statistiche di rete. In generale, un ERGM permette di estendere il concetto di famiglia esponenziale ai grafi.

Questi modelli hanno alcune "peculiarità" che li distinguono rispetto ai modelli che si costruirebbero in situazioni in cui non si ha una forte dipendenza tra le osservazioni. Negli ERGM infatti, oltre ad utilizzare le covariate dei vertici $\mathbf{x}$, si utilizzano delle statistiche della rete stessa per modellarne la struttura. Un'altra peculiarità della modellazione dei dati di rete è che si dispone di una sola osservazione, anziché di un campione di osservazioni.

La natura dei dati che si stanno considerando, ovvero dati in cui si hanno relazioni di dipendenza tra i vari vertici, fa sì che vi siano diverse sfide nella stima e specificazione dei modelli. Ad esempio, una prima difficoltà nella stima dei modelli ERGM è quella di valutare la costante di normalizzazione $k(\theta)$, che spesso non ha un'espressione esplicita. Questo rende necessario l'utilizzo di simulazioni Monte Carlo. Più in generale, nel contesto dei modelli per dati di rete, i metodi Monte Carlo trovano una grande applicazione sia nella stima dei parametri stessi del modello che nella valutazione della bontà di adattamento di questo.

Nonostante le difficoltà che si trovano nella stima dei modelli ERGM, vi sono casi però in cui questa risulta semplice. Si consideri ad esempio il caso in cui gli archi Y_{ij} sono tutti indipendenti tra di loro e sono realizzazione di una variabile bernoulliana, $Y_{ij} \sim Ber(p)$, dove p rappresenta la probabilità che un arco connetta i vertici i e j. Si può facilmente mostrare che in questo caso un ERGM è equivalente ad un modello di regressione logistica con verosimiglianza

$$\mathcal{L}(\theta) = exp\left\{\theta \sum_{i<j} y_{ij} - c \cdot log(1 + e^{\theta})\right\} \tag{3.2}$$

con $\theta = logit(p)$ e $c = \frac{|V|(|V|-1)}{2}$.

Mentre l'assunzione di indipendenza degli archi permette di facilitare di molto la stima dei parametri di interesse, come si può ben immaginare, questa è irrealistica e fa si che non si colga la struttura di dipendenza tra i vari vertici che spesso risulta la caratteristica di maggiore interesse quando si trattano i dati di rete. Proprio per questo, spesso si preferiscono modelli in cui è possibile considerare la dipendenza tra i vari vertici. Una classe di modelli che permette dipendenza tra i vertici sono i modelli Markoviani, dove due archi vengono considerati indipendenti solo se non sono incidenti allo stesso

vertice. È inoltre possibile costruire modelli dove si tiene conto della transitività della rete o di altre statistiche di rete che è di interesse considerare. La specificazione della struttura di dipendenza all'interno della rete diventa quindi una componente cruciale nella modellazione dei dati. Infatti, se questa viene specificata in modo errato spesso, durante le simulazioni necessarie alle stime dei parametri, si avranno dei modelli degeneri (dove per degeneri si intendono dei modelli che generano delle reti che non hanno archi o che hanno il massimo numero di archi possibili).

Nel corso degli anni sono state proposte molte estensioni ai modelli ERGM. Ad esempio, si possono considerare modelli ERGM per reti con struttura a blocchi, per reti con struttura spaziale e/o temporale. Finora abbiamo considerato archi non diretti. Esistono però modelli ERGM anche per archi diretti dove, anziché considerare gli archi $Y_{ij} = Y_{ji}$ indipendenti, si considerano delle diadi (Y_{ij}, Y_{ji}) tra loro indipendenti. Il più famoso di questi modelli è il $p1$-model (Holland e Leinhardt 1981).

Per un'introduzione agli ERGM si rimanda il lettore a Kolaczyk 2009, mentre per una trattazione più approfondita si veda Schweinberger et al. 2020.

3.2 Metodi di Campionamento sui Grafi Casuali

Spesso si è interessati a conoscere una caratteristica $\eta(G)$ del grafo $G = (V, E)$, utilizzato per rappresentare il sistema che si sta studiando, ma non si dispone di un'osservazione di G nella sua interezza. Si deve quindi ricorrere ad un campione $G^* = (V^*, E^*)$, con $V^* \subseteq V$ ed $E^* \subseteq E$. Una stima del parametro di interesse, disponendo di G^*, potrebbe essere data da $\hat{\eta}(G) = \eta(G^*)$, utilizzando il metodo di *plug-in*.

Si hanno quindi due problemi da affrontare: come ottenere G^* partendo da G e come stimare il parametro di interesse $\eta(G)$.

Per quanto riguarda la stima del parametro $\eta(G)$, nell'ambito dell'analisi dei dati di rete risultano molto utili gli stimatori di Horvitz-Thompson. Questo perché nei metodi di campionamento che si introdurranno qui si conoscono le probabilità di inclusione dei vertici e degli archi della rete. Questo fa sì che si possano stimare agevolmente i totali. Si consideri l'esempio dei totali sui vertici. Sia y_i una proprietà del vertice i che è di interesse studiare. Il totale a livello di popolazione è

$$\tau = \sum_{i \in V} y_i \tag{3.3}$$

Questo totale (in questo caso sui vertici), una volta che si dispone di un grafo campionato $G^* = (V^*, E^*)$, può essere stimato usando uno stimatore di Horvitz-Thompson

$$\hat{\tau}_H = \sum_{i \in V^*} \frac{y_i}{\pi_i} \tag{3.4}$$

dove π_i è la probabilità che il vertice i sia incluso nell'insieme dei vertici campionati V^*. Nel caso in cui la variabile y_i che si è interessati a studiare è il grado $d(i)$ allora uno stimatore del grado medio della rete G è

$$\hat{\bar{d}} = \frac{1}{|V|} \sum_{i \in V^*} \frac{d(i)}{\pi_i}. \tag{3.5}$$

Un ragionamento analogo può essere seguito per costruire stimatori sui totali degli archi.

Prima di considerare stimatori del parametro di interesse $\eta(G)$, si deve disporre di un campione G^* che, nel caso dei dati di rete, deve essere costruito in modo consono con il fatto che si ha a che fare con dati rappresentati tramite un grafo, dove sono quindi di fondamentale interesse le relazioni che vi sono tra i vertici. Le metodologie utilizzate spesso sono composte di due fasi. Nella prima fase, chiamata di *selezione*, si campionano le unità dall'insieme dei vertici V oppure dall'insieme degli archi E. Nella seconda fase, di *osservazione*, si campionano le unità dell'insieme non campionato nella fase di selezione (a volte nella fase di osservazione si campionano le unità sia dall'insieme dei vertici che dall'insieme degli archi).

3.2.1 *Induced Subgraph Sampling*

Nella prima fase dell'*induced subgraph sampling* si estraggono dall'insieme V n vertici senza reinserimento, ottenendo così un campione di vertici $V^* = \{v_1, \ldots, v_n\}$. Dopodiché si osservano gli archi per ogni coppia di vertici $i, j \in V^*$ per cui esiste un arco corrispondente $\{i, j\} \in E$. Si ottiene così un insieme di archi campionati E^* composto da tutti gli archi contenuti in E che sono incidenti ai vertici contenuti in V^* (quindi campionati nella prima fase).

La probabilità di inclusione nel campione dei vertici è

$$\pi_i = \frac{n}{|V|} \tag{3.6}$$

mentre la probabilità in inclusione in un determinato arco $\{i, j\}$ è

$$\pi_{ij} = \frac{n(n-1)}{|V|(|V|-1)} \tag{3.7}$$

Questo metodo richiede la conoscenza di $|V|$ per poter calcolare le probabilità di inclusione π_i e π_{ij}. Un esempio di *Induced Subgraph Sampling* dove si sono campionati 5 vertici su un grafo simulato di tipo $\mathcal{G}(n, p)$ è riportato in Figura 3.1(a).

3.2.2 *Incident Subgraph Sampling*

Questo metodo procede in modo speculare rispetto al precedente. Infatti, nella prima fase viene estratto un campione casuale semplice di dimensione n senza reinserimento, denotato con E^*, dall'insieme degli archi E, mentre nella seconda fase si osservano tutti i vertici incidenti agli archi estratti, che vanno a formare l'insieme V^*.

In questo caso, la probabilità di inclusione degli archi è

$$\pi_{\{i,j\}} = \frac{n}{|E|} \tag{3.8}$$

mentre la probabilità che un vertice venga incluso nel campione può essere espressa come

$$\mathbb{P}(\text{vertice } i \text{ nel campione}) = 1 - \mathbb{P}(\text{nessun arco incidente ad } i \text{ viene campionato}) \tag{3.9}$$

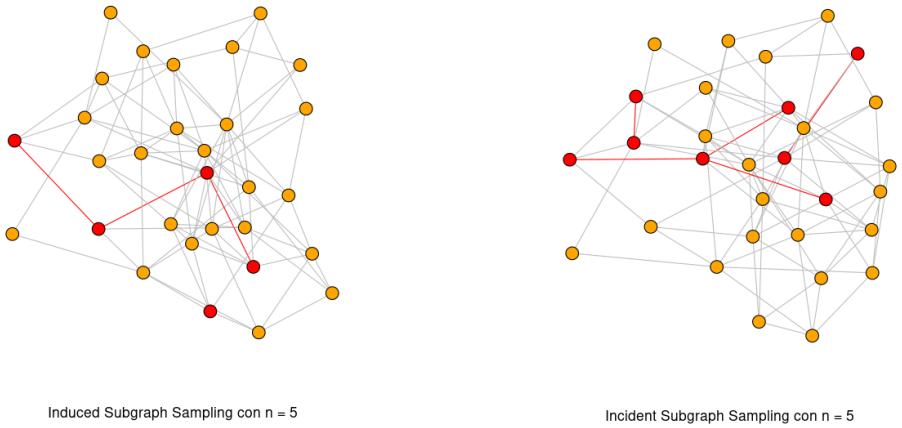
$$= \begin{cases} 1 - \frac{\binom{|E|-d(i)}{n}}{\binom{|E|}{n}}, & \text{se } n \leq |E| - d(i) \\ 1, & \text{altrimenti} \end{cases} \tag{3.10}$$

dove $d(i)$ è il grado del vertice i.

Un esempio di *Incident Subgraph Sampling* dove si sono campionati 5 vertici su un grafo simulato di tipo $\mathcal{G}(n, p)$ è riportato in Figura 3.1(b).

3.2.3 *Star Sampling*

In questo metodo di campionamento, anziché campionare dei vertici dal grafo G e poi osservare gli archi tra tutte le possibili coppie di vertici che sono presenti in E, come



(a) (b)

FIGURA 3.1: Due schemi di campionamento sul medesimo grafo $\mathcal{G}(n,p)$ con $n = 30$, $p = 0.25$. A sinistra è stato applicato l'*Induced Subgraph Sampling*, a destra l'*Incident Subgraph sampling*

viene fatto nell'*induced subgraph sampling*, viene estratto un campione iniziale di vertici V_0^* e successivamente si osservano tutti gli archi incidenti ai vertici contenuti in V_0^*.

Lo *star sampling* può avere due varianti. Nel caso in cui dopo aver estratto i vertici iniziali in V_0^* si osservano solo gli archi ad essi incidenti, si parla di *unlabelled star sampling*, mentre se oltre ad osservare gli archi incidenti ai vertici in V_0^* si osservano anche i vertici incidenti a questi archi che appartengono a $V \setminus V_0^*$ (cioè se si osservano entrambi i vertici indicenti ad ogni arco osservato), si parla di *labelled star sampling*.

La probabilità di inclusione dei vertici nell'insieme V_0^* è pari a

$$\pi_i = \frac{n}{|V|} \tag{3.11}$$

mentre la probabilità di estrazione di un arco $\{i,j\}$ è

$$\pi_{\{i,j\}} = 1 - \mathbb{P}(\text{nè il vertice } i \text{ nè } j \text{ sono nel campione iniziale}) \tag{3.12}$$

$$= 1 - \frac{\binom{|V|-2}{n}}{\binom{|V|}{n}} \tag{3.13}$$

3.2.4 *Snowball Sampling*

Lo *snowball sampling* è una generalizzazione dello *star sampling* dove, anziché limitarsi ad osservare i vicini immediati dei vertici contenuti nel campione iniziale V_0^*, si osservano tutti i vertici adiacenti a quelli nel campione iniziale fino ad una distanza pari a k.

Più formalmente, sia V_0^* l'insieme dei vertici inizialmente estratti con un campionamento casuale semplice senza reinserimento. Sia $\mathcal{N}(S)$ l'insieme dei vertici adiacenti a quelli contenuti nell'insieme S.
Allora l'insieme dei vertici adiacenti a quelli estratti nella prima fase può essere definito come

$$V_1^* = \mathcal{N}(V_0^*) \cap \overline{V_0^*} \tag{3.14}$$

qui $\overline{V_0^*}$ è l'insieme dei vertici che non sono stati estratti. Continuando l'insieme dei vertici adiacenti a quelli inclusi in V_1^* può essere scritto come

$$V_2^* = \mathcal{N}(V_1^*) \cap \overline{V_1^*} \cap \overline{V_0^*} \tag{3.15}$$

e così via fino ad ottenere V_k^*.

L'insieme dei vertici osservati alla fine sarà pari a

$$V^* = V_0^* \cap V_1^* \cap \cdots \cap V_k^* \tag{3.16}$$

mentre l'insieme degli archi osservati E^* sarà dato da tutti gli archi incidenti ai vertici contenuti in V^*.

Mentre per gli altri tipi di campionamento si possono ottenere probabilità di inclusione nel campione dei vertici e degli archi, questo risulta difficile per lo snowball sampling.

3.3 *Fast Patchwork Bootstrap* (FPB)

Sia $G = (V, E)$ un grafo non diretto, con distribuzione del grado dei vertici pari a $F = \{f(k) : k \geq 0\}$, dove $f(k)$ è la probabilità che un vertice scelto casualmente abbia grado k. Sia $\mu(G)$ il grado medio del grafo G.
Il grafo G rappresenta un ipotetico "vero" grafo casuale che non è mai pienamente osservato. Si osserva invece G_n, un grafo casuale di ordine n con distribuzione del grado dei vertici pari a $F_n = \{f_n(k), k \geq 0\}$.

Il Fast Patchwork Bootstrap (FPB), introdotto in Gel, Lyubchich e Ramirez Ramirez 2017, è un metodo bootstrap non parametrico per effettuare inferenza su F partendo dalla realizzazione G_n di G. Sia $\eta(G)$ il parametro di interesse e $\hat{\eta}(G_n)$ lo stimatore di $\eta(G)$ ottenuto dalla realizzazione G_n. Utilizzando il metodo FPB si valuterà l'incertezza legata a $\eta(G)$ utilizzando la distribuzione bootstrap di $\hat{\eta}(G_n)$.

L'algoritmo FPB consiste di due fasi: la fase di *campionamento*, dove vengono creati i *patch*, e la fase di *ricampionamento* (bootstrap), dove vengono utilizzati i *patch* campionati in precedenza per quantificare l'incertezza della stima del parametro di interesse.

La prima fase si basa sull'algoritmo Labelled Snowball with Multiple Inclusions, o LSMI, introdotto in Thompson et al. 2016. In questo algoritmo si campionano m vertici, chiamati *semi*, senza reinserimento da G_n. A partire da questi *semi*, si creano dei *patch* di ampiezza d. In particolare, partendo dal seme che si sta considerando, si crea un *patch* formando un insieme che contiene il seme stesso e i suoi vicini (vertici a lui incidenti). Ad ogni iterazione dell'algortimo si aggiungono i vicini dei vertici aggiunti nell'ultima iterazione. Tutti i vertici vengono contati con la loro molteplicità, ovvero se un vertice appare due volte in un *patch*, questo deve essere contato entrambe le volte. Un esempio di questo algoritmo è riportato in Figura 3.2.

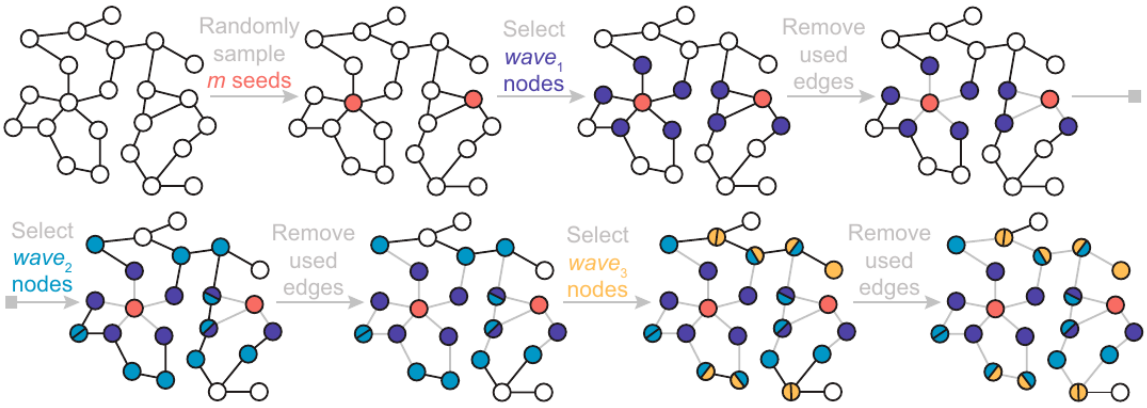


FIGURA 3.2: Passi dell'algoritmo LSMI con $m = 2$ e $d = 3$ (*Gel, Lyubchich e Ramirez Ramirez 2017*)

La seconda fase dell'algortimo si basa sul ricampionare B volte l'insieme dei *seed* e l'insieme dei vertici *non-seed*, ovvero i vertici inclusi nei *patch* ma che non erano stati usati come semi di questi.
Lo pseudo-codice dell'algoritmo, come riportato in Gel, Lyubchich e Ramirez Ramirez 2017, è nell'Algoritmo 1.

Una volta concluso l'algoritmo si può ottenere uno stimatore bootstrap per la distribuzione del grado dei vertici. In Gel, Lyubchich e Ramirez Ramirez 2017 viene proposto uno stimatore che faccia uso sia dei vertici *seed* che dei vertici *non-seed*. Lo stimatore è costruito considerando gli insiemi $\{v_s^*\}$ e $\{v_{ns}^*\}$, su di questi si rileva la frequenza con cui appaiono vertici con un particolare grado k. Per quanto riguarda i vertici contenuti

Algoritmo 1: Fast Patchwork Bootstrap (FPB)

Input: G_n, numero di semi m, numero di wave d, numero di campioni bootstrap B
Output: un campione di m semi con $\{v_s\}$ con d wave intorno ad ogni seme $\{v_{ns}\}$, corrispondenti campioni bootstrap $\{v_s^*\}_b$ e $\{v_{ns}^*\}_b$, $b = 1, \ldots, B$

```
{v_s} = campiona senza reinserimento m vertici da V(G_n)
for i = 1, ..., m do
    si inizia con la rete originale G_n
    included_0 = {v_s}_i
    for j = 1, ..., d do
        siano wave_j i vicini di tutti i vertici dell'insieme included_{j-1}
        included_j = included_{j-1} ∪ wave_j
        elimina gli archi che sono stati utilizzati per trovare wave_j
    end for
    {v_ns}_i = ⋃_{j=1}^{d} wave_j
end for
for b = 1, ..., B do
    {v_s^*}_b = campiona con reinserimento m elementi di {v_s}
    {v_ns^*}_b = campiona con reinserimento |{v_ns}| elementi di {v_ns} con pesi
    proporzionali all'inverso dei loro gradi
end for
```

all'interno dell'insieme $\{v_{ns}^*\}$ bisogna fare particolare attenzione. Infatti, un vertice apparirà in questo insieme solamente se il vertice *seed* corrispondente ha grado maggiore di zero (un vertice con grado uguale a zero è isolato e quindi partendo da questo non si possono osservare altri vertici).
Si indichi con p_0 la probabilità che il vertice *seed* corrispondente ad vertice *non-seed* in considerazione abbia grado 0. Sia inoltre $d(v_{i_s}^{*b})$ il grado dell'i-esimo vertice appartenente al b-esimo campione bootstrap dei *seed* e, allo stesso modo, sia $d(v_{i_{ns}}^{*b})$ il grado dell'i-esimo vertice appartenente al b-esimo campione bootstrap dei *non-seed*. Sia, infine, $|\{v_s^*(k)\}|$ il numero di vertici con grado k all'interno dell'insieme dei *seed* ricampionati tramite bootstrap e sia $|\{v_{ns}^*(k)\}|$ il numero di vertici con grado k all'interno dell'insieme dei *non-seed* ricampionati tramite bootstrap.

Si può costruire uno stimatore per il numero di vertici con un particolare grado N_k come

$$\hat{N}_k^* = \sum_{b=1}^{B}\sum_{i=1}^{n} \mathbb{1}\{d(v_{i_s}^{*b}) = k\} + \tag{3.17}$$

$$\sum_{b=1}^{B}\sum_{i=1}^{n} \mathbb{1}\{d(v_{i_{ns}}^{*b}) = k|\text{grado seme} > 0\}(1-p_0) + \mathbb{1}\{d(v_{i_{ns}}^{*b}) = k|\text{grado seme} = 0\}p_0 \tag{3.18}$$

$$= \sum_{b=1}^{B}\sum_{i=1}^{n} \mathbb{1}\{d(v_{i_s}^{*b}) = k\} + \sum_{b=1}^{B}\sum_{i=1}^{n} \mathbb{1}\{d(v_{i_{ns}}^{*b}) = k|\text{grado seme} > 0\}(1-p_0) \tag{3.19}$$

$$= |\{v_s^*(k)\}| + (1-p_0)\sum_{b=1}^{B}\sum_{i=1}^{n} \mathbb{1}\{d(v_{i_{ns}}^{*b}) = k|\text{grado seme} > 0\} \tag{3.20}$$

$$= |\{v_s^*(k)\}| + (1-p_0)|v_{ns}^*(k)| \tag{3.21}$$

stimando p_0 con $\hat{p}_0^*$, ovvero la frequenza di semi che hanno grado zero all'interno dell'insieme $\{v_s^*\}$, si ottiene

$$= |\{v_s^*(k)\}| + (1-\hat{p}_0^*)|\{v_{ns}^*(k)\}| \tag{3.22}$$

Dividendo lo stimatore ottenuto per il numero totale di vertici ricampionati (sia *seed* che *non-seed*) si ottiene il seguente stimatore bootstrap per la frequenza dei vertici con grado pari a k

$$\hat{f}^*(k) = \frac{|\{v_s^*(k)\}| + (1-\hat{p}_0^*)|\{v_{ns}^*(k)\}|}{|\{v_s^*\}| + |\{v_{ns}^*\}|} \tag{3.23}$$

Lo stimatore bootstrap per il grado medio $\mu(G)$ basato $\hat{f}^*(k)$ è

$$\hat{\mu}^*(G_n) = \sum_{k\geq 0} k\hat{f}^*(k) = \frac{\sum_{k\geq 0} k|\{v_s^*(k)\}| + (1-\hat{p}_0^*)\sum_{k\geq 1} k|\{v_{ns}^*(k)\}|}{|\{v_s^*\}| + |\{v_{ns}^*\}|} \tag{3.24}$$

Una volta definito questo stimatore rimangono da scegliere gli iperparametri. Questi sono, fissato il numero B di campioni bootstrap, il numero di *seed* e il numero d di *wave* utilizzate per costruire i *patch*. Si considerano S diversi possibili valori per il numero di *seed* e D diversi valori per l'ampiezza delle *wave* costruite attorno ai *seed*. Si hanno quindi $J = SD$ combinazioni di valori. Per ognuna di queste combinazioni viene applicato l'Algortimo 1, calcolato il parametro $\eta(G_n^*)$ di interesse (in questo caso il grado medio) e si determina un intervallo di confidenza del tipo

$$BCI_j = \left(\hat{\eta}^{j*}_{(B\alpha/2)}, \hat{\eta}^{j*}_{(B(1-\alpha/2))}\right) \tag{3.25}$$

dove $\hat{\eta}^{j*}_{(B\alpha)}$ è la statistica d'ordine $B\alpha$ della distribuzione bootstrap del parametro η, per la j-esima combinazione di *seed-wave*.

L'algoritmo di convalida incrociata presentato in Gel, Lyubchich e Ramirez Ramirez 2017, e riportato nell'Algoritmo 2, ha come obiettivo quello di determinare in modo efficace quale sia la migliore combinazione dei valori di *seed* e *wave*. Per fare ciò, si fa utilizzo di campioni chiamati *proxy*. L'algoritmo per poter essere utilizzato necessita della rete osservata, dell'insieme dei semi usati per generare i patch, degli intervalli di confidenza bootstrap (uno per ogni combinazione), il numero e la dimensione dei campioni *proxy* da generare e un livello di confidenza 1 - α.

Algoritmo 2: Algoritmo di convalida incrociata per selezione della combinazione ottimale di *seed-wave*

Input: G_n, insieme U dei semi usati per i patch, intervalli di cofidenza bootstrap per J combinazioni seme-wave $j = 1, \ldots, J$, dimensione campione proxy h, N numero di campioni proxy da ottenere, livello di significatività α

Output: combinazione seed-wave ottimale j_{opt}

for $i = 1, \ldots, N$ **do**
  campiona h vertici da U
  stima $\hat{\eta}^{proxy}_i$ sugli h vertici campionati
  for $j = 1, \ldots, J$ **do**
    $count_{i,j} = \begin{cases} 1 & \text{if } \hat{\eta}^{proxy}_i \in BCI_j \\ 0 & \text{altrimenti} \end{cases}$
  end for
end for
$j_{opt} = \arg\min_{j=1,\ldots,J} \left| \frac{1}{N} \sum_{i=1}^{N} count_{i,j} - (1-\alpha) \right|$
$BCI_{j_{opt}}$

Si generano N campioni proxy. Per ognuno di questi si calcola il valore del parametro di interesse, chiamato $\hat{\eta}^{proxy}$. Poi, si considerano tutte le J possibili combinazioni dei parametri di interesse e si definisce una variabile dicotomica che assume valore 1 se $\hat{\eta}^{proxy}$ appartiene al j-esimo intervallo di confidenza, 0 altrimenti (alla fine si avranno NJ variabili dicotomiche). Per ogni combinazione di *seed* e *wave* si fa una media delle variabili dicotomiche corrispondenti a quella combinazione, andando ad ottenere la frazione di volte che il parametro proxy è contenuto nell'intervallo di confidenza corrispondente alla j-esima combinazione. Si ottiene quindi un livello di copertura effettivo dell'intervallo di confidenza costruito usando la j-esima combinazione degli iperparametri. Per ognuno di questi si considera la differenza con il livello nominale al

quale è costruito l'intervallo di confidenza. L'algoritmo termina restituendo l'intervallo di confidenza $BCI_{j_{opt}}$ che minimizza la differenza tra il livello di copertura effettivo e nominale.

Capitolo 4

Applicazione a dati reali

4.1 I Dati

In questo capitolo si passerà ora ad un'applicazione pratica di quanto discusso finora. In particolare, si considera il dataset proveniente dall'articolo Ji e Jin 2016, relativo alla rete di collaborazione tra statistici che hanno pubblicato articoli scientifici tra gli anni 2003 e 2012 sui giornali scientifici *Annals of Statistics*, *Biometrika*, *Journal of the American Statistical Association* e *Journal of the Royal Statistical Society - Series B*.

Si procederà prima con un'analisi esplorativa dei dati e poi si applicherà il *Fast Patchwork Bootstrap* (FPB) per ottenere una stima bootstrap del grado medio della rete. Si applicherà l'algortimo di validazione incrociata presentato nella sezione precendete e si confronteranno i risultati con quelli che si sarebbero ottenuti se non si fosse applicato il FPB.

Le analisi sono state svolte tramite il linguaggio di programmazione R (R Core Team 2024).

4.2 Analisi esplorativa

La rete considerata è composta da 3607 vertici, dove ogni vertice rappresenta uno statistico che ha pubblicato almeno un articolo in uno dei 4 giornali scientifici considerati nel periodo 2003-2012. Un arco è presente tra due vertici quando i due statistici presi in considerazione hanno pubblicato almeno un articolo insieme. In totale sono presenti 5615 archi.

Numero di collaboratori	
Statistico	**# Collaboratori**
Peter Hall	65
Raymond J. Carroll	55
Joseph G. Ibrahim	41
Jianqing Fan	38
David Dunson	32
Hongtu Zhu	25

Tabella 4.1: Elenco statistici con maggiore numero di collaboratori

Una visualizzazione della rete nella sua interezza è riportata nella Figura 4.1.

In questa rappresentazione si nota come vi siano alcuni statistici che hanno collaborato molto con altri, mentre una grande percentuale ha collaborato con un numero ristretto di persone. La dimensione dei vertici è proporzionale al loro grado. In particolare, rappresentata al centro della figura si nota la presenza di una grande componente connessa dove sono presenti i vertici con un grado elevato, mentre all'esterno si hanno i vari statistici che non hanno collaborato con altri o che hanno collaborato con un numero ristretto di persone.

Questa caratteristica della rete si riflette anche nell'istogramma del grado della rete, riportato in Figura 4.2. Da questo grafico è evidente come la maggior parte dei vertici ha un grado minore o uguale di 3, mentre una piccola parte dei vertici ha un grado maggiore. Una naturale interpretazione è la presenza di alcuni statistici molto prolifici, che collaborano con un elevato numero di persone. Considerando il log-log plot del grado della rete (Figura 4.3), ovvero un grafico in cui si riporta il logaritmo del grado dei vertici e la corrispondente log-frequenza osservata, si nota un andamento rettilineo (con coefficiente angolare negativo). Questo andamento è tipico delle rete *scale-free*.

Nella Tabella 4.1 sono riportati i sei statistici con grado maggiore all'interno della rete considerata. Si nota come la persona con il maggior numero di collaboratori è Peter Hall. Il sottografo indotto dalle collaborazioni di Peter Hall (ovvero il grafo dove si considerano solo quei vertici che hanno un arco incidente al vertice a cui si fa riferimento) è riportato in Figura 4.4.

Oltre al grado vi sono altre misure per determinare quali siano i vertici più importanti all'interno di una rete. Una di queste è la closeness centrality, introdotta nel Capitolo 1, che misura l'importanza di un vertice in modo inversamente proporzionale alla sua distanza dagli altri vertici. In altre parole, vertici che sono vicini a molti vertici avranno una closeness centrality alta, mentre i vertici che sono lontani da molti vertici avranno una closeness centrality bassa. I primi vengono considerati come i vertici "importanti".

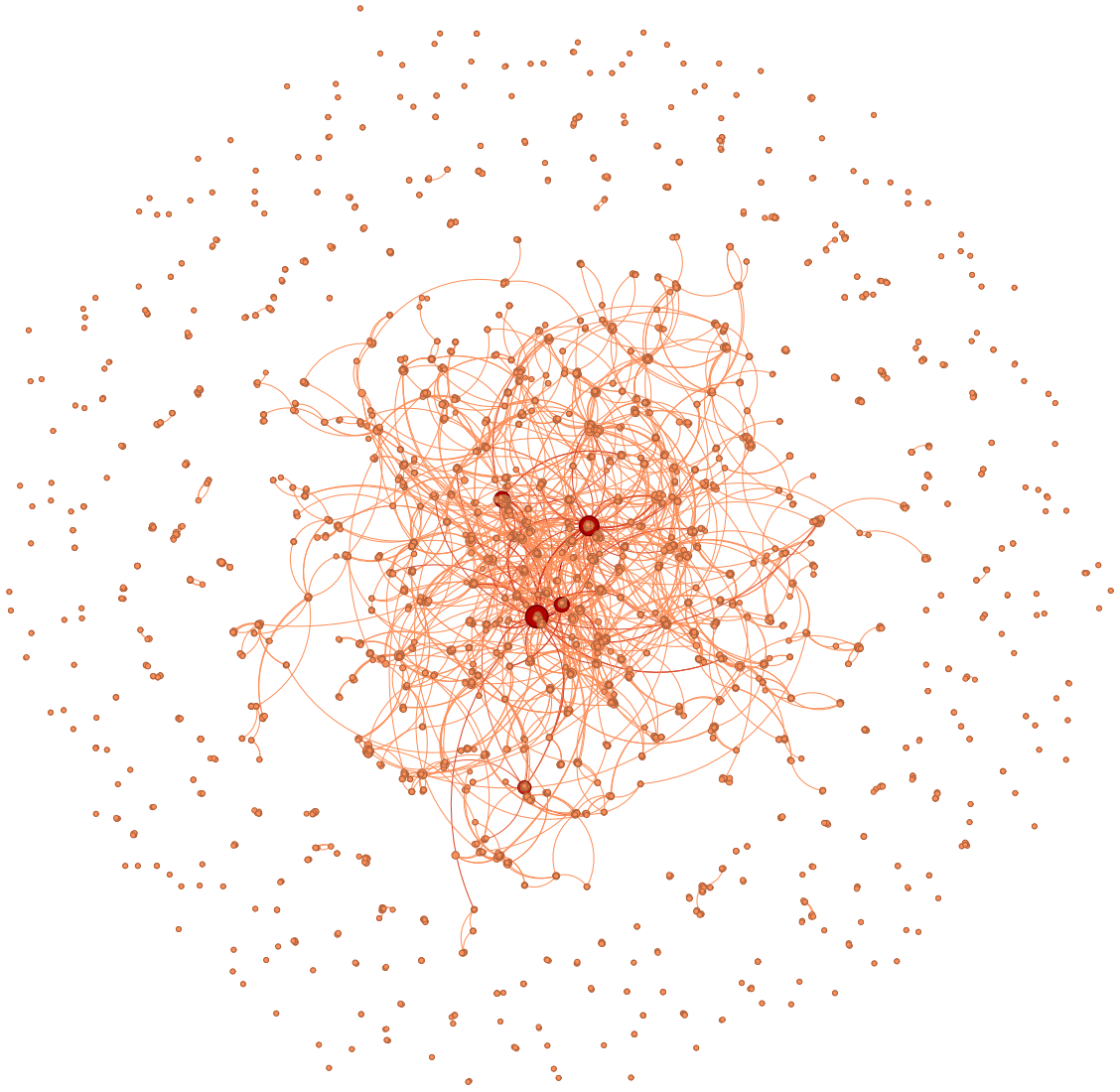


FIGURA 4.1: Visualizzazione della rete delle citazioni tra statistici

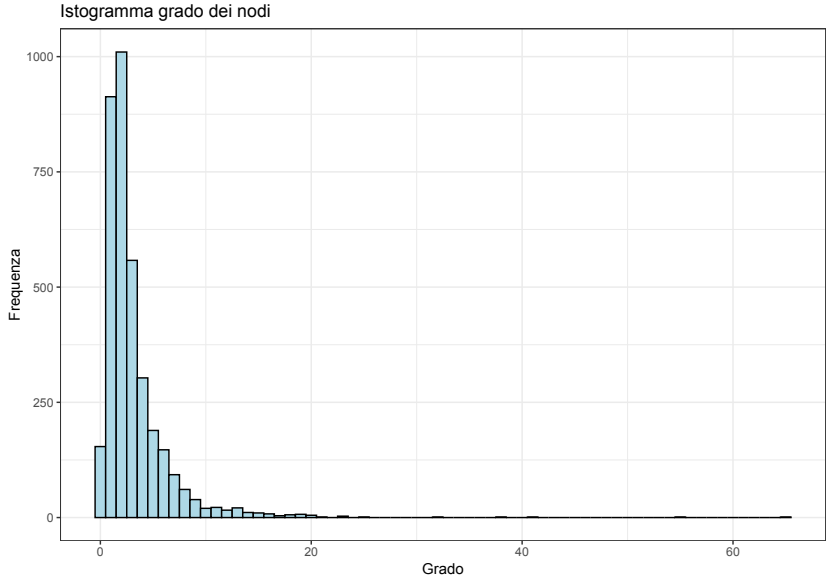


FIGURA 4.2: Istogramma del grado della rete di collaborazione degli statistici

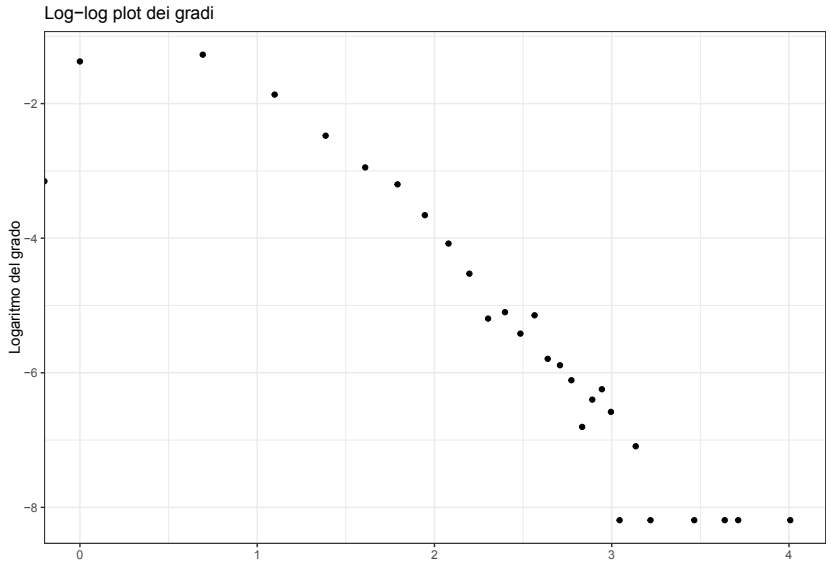


FIGURA 4.3: Log-log plot della rete di collaborazione tra statistici

La Tabella 4.2 riporta i vertici che hanno una closeness centrality più alta all'interno della rete. Si nota come tre gli statistici che mostrano una closeness centrality alta sono tra i vertici con un grado maggiore.
Un'altra misura di importanza dei vertici è la betweenness centrality, secondo la quale un vertice è importante quando collega tante altre coppie di vertici. In questo caso, secondo questa misura, un vertice (quindi uno statistico) può essere considerato importante quando mette in contatto molti altri statistici. Nella Tabella 4.3 sono riportati gli statistici con una maggiore betweenness centrality nella rete. Anche in questo caso varie persone che appaiono in questa tabella sono quelle con il maggior numero di

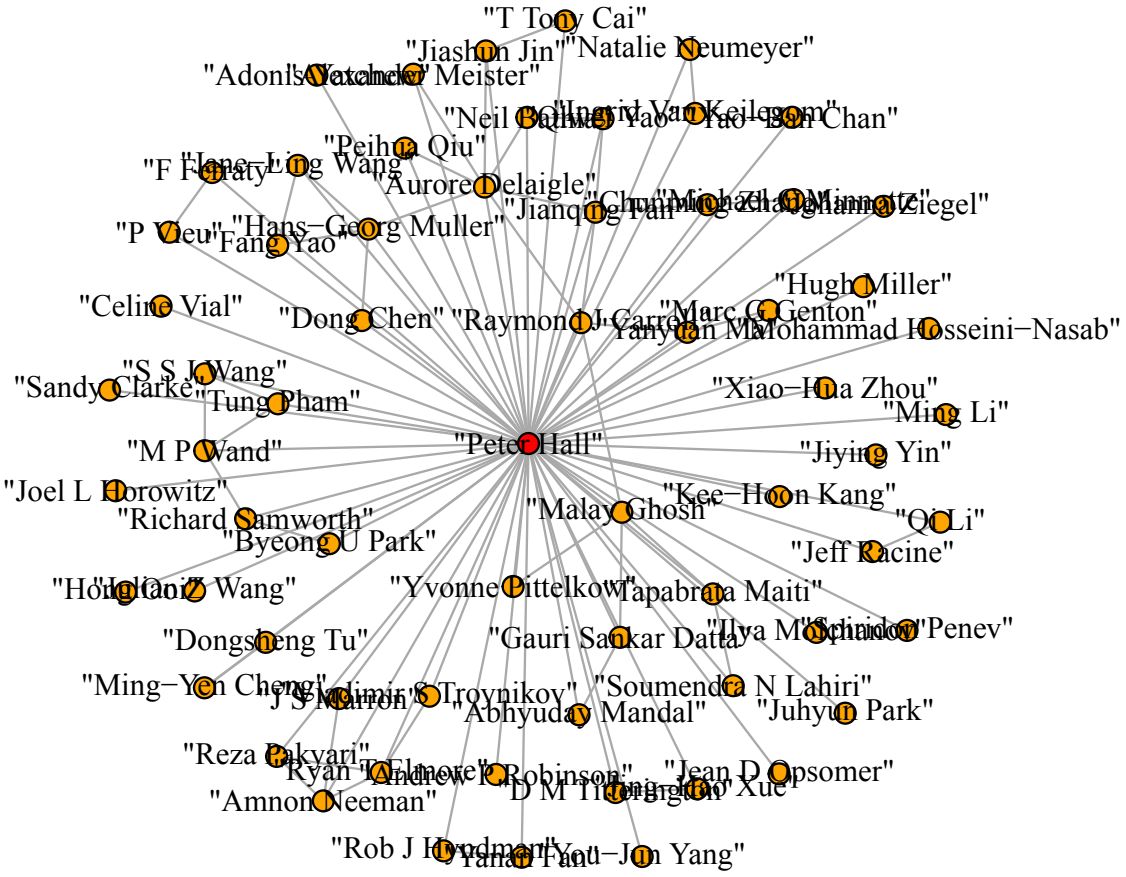


FIGURA 4.4: Sottografo indotto dei co-autori di Peter Hall (statistico con maggiori collaboratori nel dataset)

Closeness centrality	
Statistico	**Score**
Raymond J. Carroll	2.058898e-7
Peter Hall	2.058867e-7
Jianqing Fan	2.058603e-7
Yanyuan Ma	2.058545e-7
Malay Ghosh	2.058414e-7
Aurore Delaigle	2.058395e-7

TABELLA 4.2: Elenco statistici con maggiore closeness centrality

collaboratori. Questa breve analisi esplorativa ha rivelato come nel dataset considerato le unità statistiche più importanti siano Peter Hall, Raymond J. Carroll e Jianqing Fan.

Betweenness centrality	
Statistico	**Score**
Raymond J. Carroll	692374.5
Peter Hall	688296.1
Jianqing Fan	259634.8
Runze Li	2088451.9
James R. Robins	199810.9
Gerda Claeskens	177996.7

Tabella 4.3: Elenco statistici con maggiore betweenness centrality

4.3 Applicazione Fast Patchwork Bootstrap

Si vuole ora applicare il *Fast Patchwork Bootstrap* (FPB) alla rete di collaborazione tra statistici, replicando così alcuni dei risultati riportati in Gel, Lyubchich e Ramirez Ramirez 2017.

In R è disponibile un'implementazione del FPB nella libreria Snowboot (Chen et al. 2018).

A differenza del bootstrap "classico", il *Fast Patchwork Bootstrap* ha performance che dipendono da alcuni iper-parametri. I due più importanti sono il numero di vertici e il numero di wave da utilizzare per campionare i vertici all'interno di una rete. Nella Figura 4.5 sono riportate le distribuzioni bootstrap del grado medio della rete, al variare di numero di *seed* e *wave*. Si nota come, in linea generale, all'aumentare del numero di semi utilizzati e numero di wave si ha una distribuzione via via più "stretta", in grado di fornire degli intervalli di confidenza migliori. Nei vari istogrammi in Figura 4.5 è riportata una linea verticale tratteggiata che indica il vero valore del grado medio nella rete di citazione tra statistici in considerazione ($\hat{\mu} = 3.113$). Tutte le distribuzioni boostrap costruite contengono il vero valore del parametro, ciò che cambia è l'incertezza associata ad esso.

Utilizzando il *Fast Patchwork Bootstrap* si può stimare la distribuzione del grado dei vertici all'interno della rete. La stima di questa sarà influenzata anch'essa dalla scelta dei parametri *seed* e *wave*.
Nelle Figure 4.6 e 4.7 sono riportate le distribuzioni stimate del grado della rete utilizzando prima 50 *seed* e 1 *wave* e poi 100 *seed* (mantenendo uguale il numero di wave). Nella realizzazione di questi due grafici si è considerata la rete di collaborazione tra statistici dopo la rimozione di tutti i vertici isolati, ovvero dopo aver rimosso tutti i vertici corrispondenti a statistici che non hanno pubblicato articoli scientifici assieme ad altri.

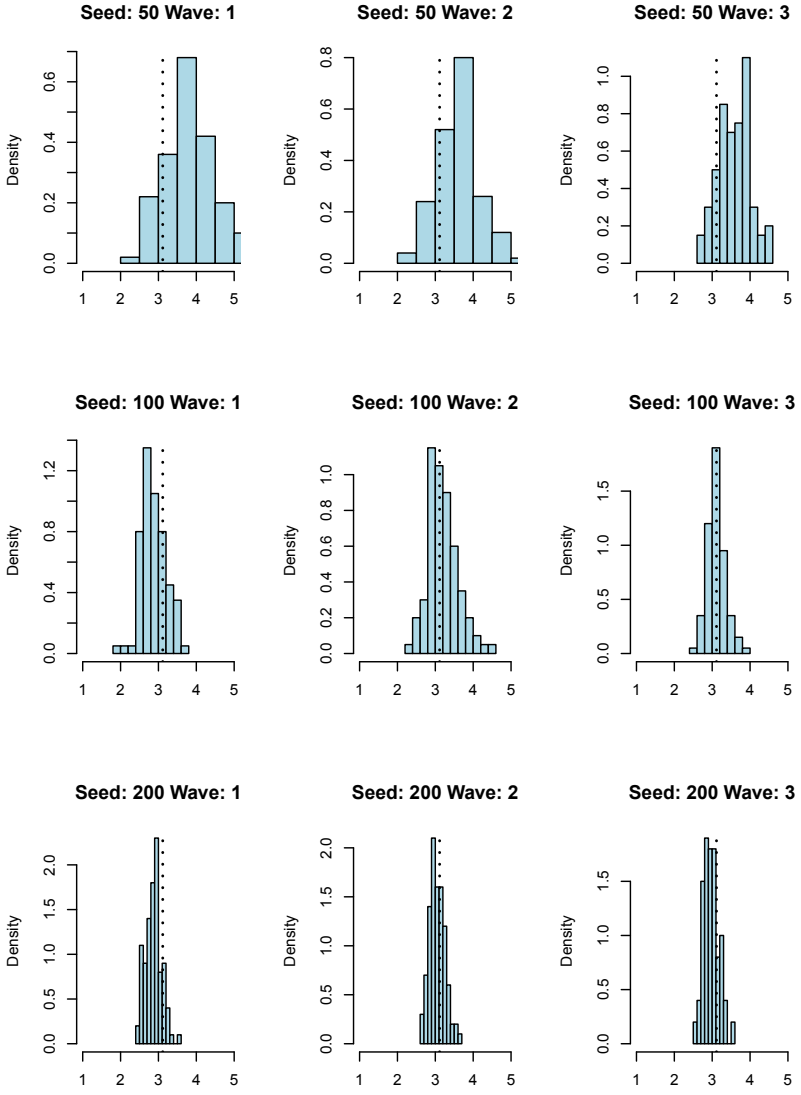


FIGURA 4.5: Distribuzione del gradio medio calcolato tramite FPB al variare dei parametri *seed* e *wave*

Come si può notare le due distribuzioni differiscono in modo non trascurabile. In particolare si nota una differenza nella coda delle distribuzioni. Nella distribuzione stimata con 50 *seed* si osservano vertici con un grado fino a 18, mentre nella distribuzione stimata con 100 *seed* il massimo grado osservato è 8. Nonostante questa differenza nelle code della distribuzione, aumentare il numero di *seed* migliora l'incertezza legata alla stima della distribuzione. Si prendano come esempio le stime delle frequenze associate ai gradi 1, 2 e 3. Si ha una riduzione nell'ampiezza degli intervalli di confidenza per le stime delle frequenze dei gradi considerati all'interno della rete. Questa riduzione dell'incertezza può essere intuitivamente spiegata dalla presenza di più vertici con con un particolare grado all'interno del campione.

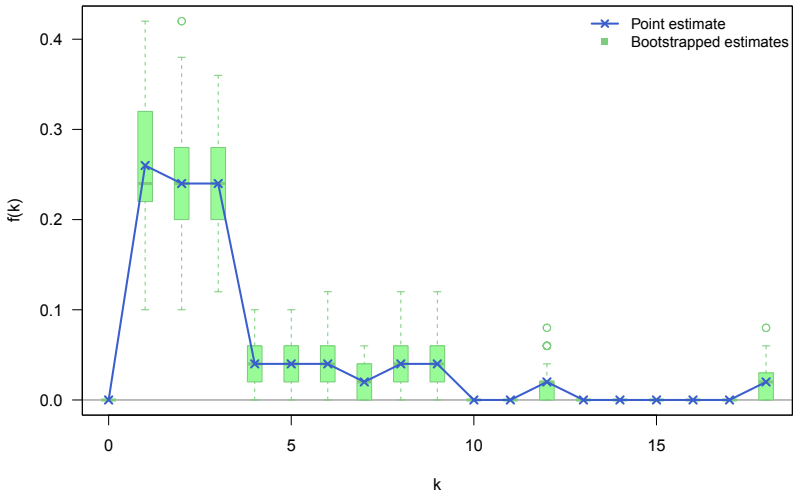


FIGURA 4.6: Distribuzione del grado della rete stimata tramite FPB con 50 *seed* e 1 *wave*

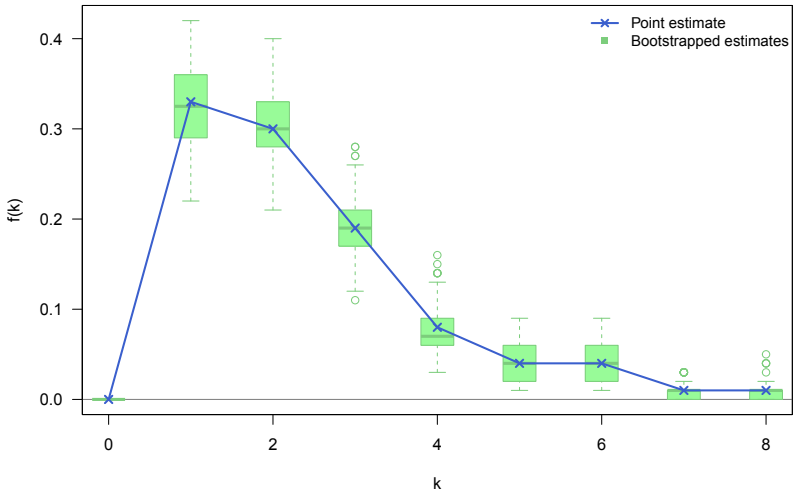


FIGURA 4.7: Distribuzione del grado della rete stimata tramite FPB con 100 *seed* e 1 *wave*

k	$f(k)$	$\widehat{f(k)}_{FPB}$	I.C. FPB	$\widehat{f(k)}_{BOOT}$	I.C. Boot
1	0.264	0.275	(0.125, 0.425)	0.165	(0.060, 0.270)
2	0.292	0.225	(0.100, 0.350)	0.380	(0.260, 0.520)
3	0.161	0.125	(0.050, 0.250)	0.219	(0.120, 0.340)
4	0.087	0.075	(0.000, 0.175)	0.083	(0.020, 0.160)
5	0.054	0.050	(0.000, 0.125)	0.038	(0.000, 0.100)

TABELLA 4.4: Stime e intervalli di confidenza per distribuzione del grado nella rete di citazioni tra statistici usando FPB e bootstrap

4.3.1 Confronto tra FPB e Bootstrap

Si vuole confrontare il *Fast Patchwork Bootstrap* con il bootstrap "classico". Per questo confronto si è voluto seguire quanto riportato in Gel, Lyubchich e Ramirez Ramirez 2017, dove si è considerata la rete di collaborazione tra statistici dopo aver rimosso tutti i vertici con un grado pari a 0. Questa rete avrà 3453 vertici.

Più nel dettaglio, si utilizzerà l'algoritmo di convalida incrociata presentato nell'Algoritmo 2 per scegliere la migliore combinazione di seed e *wave*. Una determinata la migliore combinazione si andrà a campionare la rete utilizzando i parametri ottenuti nella fase di convalida incrociata.
La stima della distribuzione del grado ottenuta verrà confrontata con quella ottenuta applicando un normale bootstrap. Per semplicità espositiva ci si concentrerà sul confronto delle frequenze corrispondenti a gradi che vanno da 1 a 5. Per quanto riguarda la procedura bootstrap, si campionano senza reinserimento M vertici dalla rete e si osserva il grado di questi vertici. Una volta ottenuto un campione di gradi, si applica il bootstrap e si ottengono così B campioni bootstrap. In questo caso si sono fissati $M = 50$ e $B = 500$.

Per la convalida incrociata del FPB si sono utilizzati i seguenti parametri: la dimensione del campione proxy h è stata fissata a 13, il numero di campioni utilizzati è stato posto uguale a 100, il livello di significatività α è stato fissato a 0.05. Per quanto riguarda il numero di semi, si sono considerati i valori 40, 50, 100 e 200, mentre il numero di *wave* da considerare sono state fissate a 1, 2 e 3.

La miglior combinazione di *seed* e *wave* ottenuta nella fase di convalida incrociata è 40 *seed* e 2 *wave*. Una volta determinata la migliore combinazione di *seed* e *wave* si applica la procedura riportata nell'Algoritmo 1 per ottenere dei campioni bootstrap a partire dal campione ottenuto con la miglior combinazione dei parametri. Il numero di campioni bootstrap è stato fissato anche in questo caso a 500.

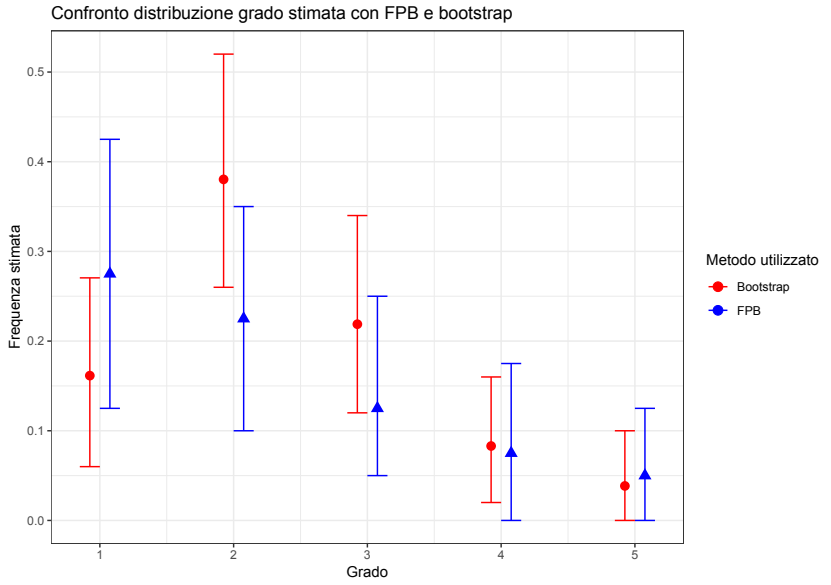


FIGURA 4.8: Confronto distribuzioni del grado stimate tramite FPB e Bootstrap

Le stime e gli intervalli di confidenza bootstrap ad esse associate, per i due metodi, sono riportate nella Tabella 4.4. Come si può notare dai valori riportati, semplicemente ricampionare i vertici porta ad una sottostima per la frequenza di vertici con grado 1 e una sovrastima dei vertici con frequenze 2 e 3. Applicando il FPB, invece, si hanno delle stime più vicine alle vere frequenze osservate nella rete. Le ampiezze degli intervalli di confidenza risultano pressoché uguali nei due metodi. Una rappresentazione grafica dei risultati si trova nella Figura 4.8.

Si considera ora solo la distribuzione del grado medio. Si nota che nella rete di collaborazione, dopo aver rimosso i vertici con grado nullo, il grado medio assume valore $\hat{\mu} = 3.252$. In Tabella 4.5 sono riportate le stime bootstrap del grado medio nella rete considerata, assieme agli intervalli di confidenza quantile di livello 95 %. Si nota che con i parametri fissati in questo esperimento il FPB è in grado di restituire degli intervalli di confidenza migliori rispetto al normale bootstrap. Nella Figura 4.9 è riportato un boxplot delle distribuzioni bootstrap del grado medio ottenute con il *Fast Patchwork Bootstrap* e con il bootstrap normale. La linea orizzontale riportata in rosso indica il vero valore del grado medio nella rete.
Come si può notare il FPB fornisce dei risultati di gran lunga migliori rispetto al normale bootstrap. Nella Figura 4.10 si è riportato lo stesso grado, dove però si è confrontato il FPB con il bootstrap effettuato a partire da un campione con 50 vertici e un campione composto da 100. Per ottenere dei risultati comparabili a quelli ottenuti con il FPB è necessario raddoppiare il numero di vertici contenuti nel campione originale dal quale si andrà poi a ricampionare. La stima del grado medio ottenuto con il bootstrap con

Metodo utilizzato	$\hat{\mu}^*$	**I.C. 95%**
FPB	3.525	(2.737, 4.414)
Bootstrap	4.308	(2.700, 6.770)

TABELLA 4.5: Stime e intervalli di confidenza per grado medio nella rete di citazioni tra statistici usando FPB e bootstrap

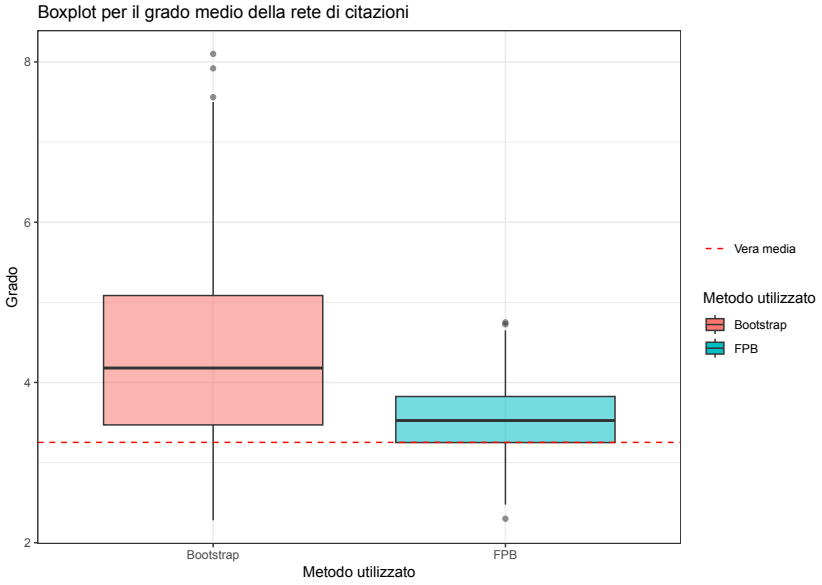


FIGURA 4.9: Boxplot grado medio stimato tramite FPB e Bootstrap

un campione originale di 100 vertici è $\hat{\mu}^*_{BOOT} = 3.496$, con un intervallo di confidenza quantile di livello 95% pari a (2.715, 4.490).

Questi risultati evidenziano la maggiore efficienza del FPB dal punto di vista delle osservazioni necessarie per ottenere dei "buoni" intervalli di confidenza per il grado medio della rete considerata in questo capitolo.

In linea generale, tramite il FPB si può ottenere una stima del grado medio di una rete e valutare in modo efficiente l'incertezza legata a questa stima. A vantaggio di questo metodo è lo schema di campionamento che, per come è costruito, è in grado di ottenere una "panoramica" migliore dei vertici nella rete. L'idea di campionare una rete utilizzando dei *patch* (o chiazze) si è rivelata vincente, nel senso che con un numero ristretto di queste si riescono ad ottenere delle buone stime. Il metodo risulta pertanto particolarmente utile in quei casi in cui non si riesce ad osservare la rete nella sua interezza oppure quando risulta particolarmente costoso campionare un numero elevato di vertici.

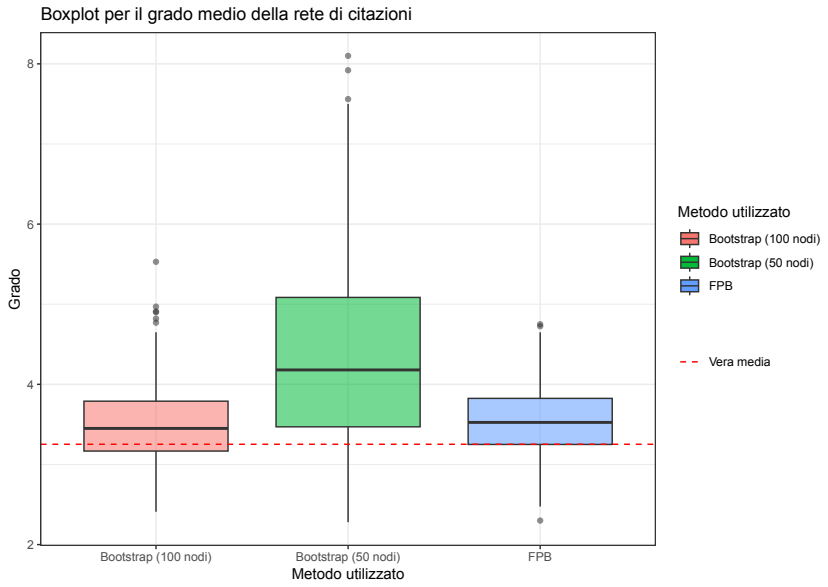


FIGURA 4.10: Boxplot grado medio stimato tramite FPB e Bootstrap con 50 e 100 vertici

Conclusione

In questa relazione finale si è data una rassegna di alcuni metodi statistici per dati di rete, con una particolare attenzione a metodi di ricampionamento bootstrap. Dopo un'introduzione su grafi, grafi casuali (di cui si sono descritti i principali modelli), bootstrap e campionamento su grafi, ci si è soffermati sul *Fast Patchwork Bootstrap* (Gel, Lyubchich e Ramirez Ramirez 2017).

Questo metodo si basa sul campionare la rete osservata (che non necessariamente coincide con quella intera) a "chiazze" (*patch*), basandosi su dei vertici *seed*. Partendo da questi, se ne osservano i vicini fino ad una determinata distanza d. I vertici *seed* e *non-seed* osservati che vanno a formare i *patch*, vengono poi utilizzati per creare dei campioni bootstrap.

Nell'ultimo capitolo, questo metodo è stato applicato ad un dataset relativo alle collaborazioni tra statistici. Lo si è confrontato con il bootstrap applicato secondo la sua definizione classica (si è partiti da un campione casuale semplice di gradi di vertici e si sono ricampionati per formare il campione bootstrap). In questo contesto il FPB si è dimostrato essere un metodo valido ed efficiente nella stima del grado medio di una rete e nel valutare l'incertezza legata a questa stima. Se n'è evidenziata la maggiore efficienza dal punto di vista di osservazioni necessarie per ottenere dei buoni risultati rispetto al metodo con il quale lo si è confrontato.

Appendice A

Codice R

Capitolo 1

```
1   library(igraph)
2
3   g1 <- erdos.renyi.game(n = 50,
4   p.or.m = 0.5,
5   type = "gnp")
6
7   plot(g1,
8   vertex.label= NA,
9   edge.arrow.size=0.02,
10  vertex.size = 3,
11  xlab = 'Realizzazione Grafo Casuale G(50, 0.5)')
12
13  g2 <- erdos.renyi.game(50,
14  p.or.m = 0.05,
15  type = "gnp")
16
17  plot(g2,
18  vertex.label= NA,
19  edge.arrow.size=0.02,
20  vertex.size = 3,
21  xlab = 'Realizzazione Grafo Casuale G(50, 0.05)')
```

Listing A.1: Gerazione dei grafi casuali di Erdös-Rényi riportati in Figura 1.2

```
1   library(igraph)
2
3   ring2 <- graph.lattice(c(10), nei = 2, circular = TRUE)
4   plot(ring2,
5   vertex.label = NA,
6   vertex.size = 8,
7   xlab = "Lattice con n = 10, k = 4")
8
9   ws <- sample_smallworld(dim = 1, size = 10, nei = 2, p = 0.05)
10  E(ws)[10]$color = "red"
```

```
11
12      plot(ws,
13      layout = layout.circle,
14      vertex.label = NA,
15      vertex.size = 8,
16      xlab = "Realizzazione modello di Watts-Strogatz con p = 0.05")
17
18      ws2 <- sample_smallworld(dim = 1, size = 10, nei = 2, p = 0.1)
19      E(ws2)
20      modified <- c(4, 6, 7, 18)
21      E(ws2)[modified]$color = "red"
22      E(ws2)[-modified]$color = "grey"
23
24      plot(ws2,
25      layout = layout.circle,
26      vertex.label = NA,
27      vertex.size = 8,
28      xlab = "Realizzazione modello di Watts-Strogatz con p = 0.1")
```

LISTING A.2: Generazione del lattice e grafo casuale secondo il modello di Watts-Strogatz in Figura 1.3

```
1       library(igraph)
2
3       # Simulating graph with Barabasi-Albert model
4       g <- barabasi.game(200,
5       power = 1,
6       m = 4,
7       directed = FALSE)
8
9       plot(g, vertex.size = 4, vertex.label = NA,
10      layout = layout.fruchterman.reingold(g),
11      xlab = "Realizzazione modello Barabasi-Albert (n = 200, m = 4)")
12
13      plot(degree(g),
14      main = "Distribuzione del grado",
15      ylab = "Grado",
16      xlab = "")
17
18      plot(log(sort(unique(degree(g)))),
19      as.vector(log(table(degree(g)))),
20      main = "Log-log plot del grado",
21      xlab = "Log-grado",
22      ylab = "Log-frequenza")
```

LISTING A.3: Generazione del grafo casuale secondo il modello di Barabási-Albert in Figura 1.5

Capitolo 3

```
1       library(igraph)
2       library(rlang)
3
```

```
4       # generating an erdos-renyi random graph with n = 30 nodes
5       g <- erdos.renyi.game(30, 0.2, type = "gnp")
6
7       plot(g, vertex.size = 8,
8       vertex.label = NA)
9
10      # incident subgraph sampling
11      g1 <- rlang::duplicate(g)
12      campione.vertici <- sample(V(g1), size = 5, replace = FALSE)
13      V(g1)[campione.vertici]$color = "red"
14      V(g1)[-campione.vertici]$color = "orange"
15
16      archi.incidenti <- c()
17      for (e in campione.vertici) {
18              for (k in campione.vertici) {
19                      if (length(E(g1)[e %--% k]) == 1) {
20                              archi.incidenti <- c(archi.incidenti, E(g1)[e %--% k])
21                      }
22              }
23      }
24      E(g1)[archi.incidenti]$color = "red"
25      E(g1)[-archi.incidenti]$color = "grey"
26
27      plot(g1, vertex.size = 8, vertex.label = NA,
28      xlab = "Induced Subgraph Sampling con n = 5")
29
30      # induced subgraph sampling
31      g2 <- rlang::duplicate(g)
32
33      campione.archi <- sample(E(g2), size = 5, replace = FALSE)
34      E(g2)[campione.archi]$color = "red"
35      E(g2)[-campione.archi]$color = "grey"
36
37      vertici.incidenti <- as.vector(ends(g2, campione.archi))
38      V(g2)[vertici.incidenti]$color = "red"
39      V(g2)[-vertici.incidenti]$color = "orange"
40
41      plot(g2, vertex.size = 8, vertex.label = NA,
42      xlab = "Incident Subgraph Sampling con n = 5")
```

LISTING A.4: Simulazione di *Induced Subgraph Sampling* e *Indicent Subgraph Sampling* riportato in Figura 3.1

Capitolo 4

```
1       library(igraph)
2       library(tidyverse)
3
4       # ------------------------
5
6       # reading data
7       # grafo delle citazioni tra paper
8       g <- read_graph("paperCitAdj.gml", format = "gml")
```

```
9
10      # paper properties
11      prop  <- read.csv("paperList.csv")
12
13      # adding properities to nodes in the graph
14      V(g)$doi <- prop$DOI
15      V(g)$year <- prop$year
16      V(g)$title <- prop$title
17      V(g)$citcount <- prop$citCounts
18
19      # grafo degli statistici
20      papAutbiadj <- as.matrix(read.table("authorPaperBiadj.txt", sep="\t", header=F))
21
22      auth.cocit <- papAutbiadj %*% t(papAutbiadj)
23      auth.cocit <- (auth.cocit >= 1) - 0
24      diag(auth.cocit) = 0
25
26      g.auth <- graph_from_adjacency_matrix(auth.cocit,
27      mode = "undirected")
28      V(g.auth)$name <- readLines("authorList.txt")
29      V(g.auth)$grado <- degree(g.auth)
30
31
32      # Analisi esplorativa ------------------------------------------------------
33      # numero di nodi nel grafo
34      gorder(g.auth)
35
36      # numero di archi
37      gsize(g.auth)
38
39      # istogramma grado nodi
40      deg <- g.auth %>% degree()
41
42      head(sort(deg, decreasing = TRUE))
43
44      ggplot() +
45      aes(deg) +
46      geom_histogram(col = "black", fill = "light blue", binwidth = 1) +
47      ggtitle("Istogramma grado dei nodi") +
48      xlab("Grado") +
49      ylab("Frequenza") +
50      theme_bw()
51
52      # log-log degree distribution
53      dd.g <- g.auth %>% degree_distribution()
54      d <- 1:max(deg)-1
55      ind <- which(dd.g != 0)
56      #plot(d[ind], dd.g[ind], log = "xy")
57
58      ggplot(
59      data.frame(d[ind], dd.g[ind]),
60      aes(x = log(d[ind]), y = log(dd.g[ind]))
61      ) +
62      geom_point() +
63      ggtitle("Log-log plot dei gradi") +
64      xlab("") +
65      ylab("Logaritmo del grado") +
66      theme_bw()
```

```
67
68      head(sort(betweenness(g.auth), decreasing = TRUE))
69
70      head(sort(closeness(g.auth), decreasing = TRUE))
71
72      edge_density(g)
73
74
75      # nodo con grado massimo
76      max.deg.node <- which.max(deg)
77      adjacent.nodes <- neighbors(g.auth, max.deg.node)
78      neigh.graph <- induced.subgraph(g.auth, c(max.deg.node, adjacent.nodes))
79      V(neigh.graph)[45]$color = "red"
80      V(neigh.graph)[-45]$color = "orange"
81      #V(neigh.graph)$size = degree(neigh.graph)
82
83      plot(neigh.graph,
84      vertex.size = 5,
85      vertex.label.cex = 0.8,
86      vertex.label.color = "black",
87      layout = layout_with_fr(neigh.graph))
```

LISTING A.5: Codice utilizzato per l'analisi esplorativa dei dati

```
1       # library imports
2       library(igraph)
3       library(tidyverse)
4       library(snowboot)
5       library(boot)
6
7       set.seed(12345)
8
9       # ------------------------
10
11      # reading data
12      papAutbiadj <- as.matrix(read.table("authorPaperBiadj.txt", sep="\t", header=F))
13
14      auth.cocit <- papAutbiadj %*% t(papAutbiadj)
15      auth.cocit <- (auth.cocit >= 1) - 0
16      diag(auth.cocit) = 0
17
18      g.auth <- graph_from_adjacency_matrix(auth.cocit,mode = "undirected")
19      V(g.auth)$name <- readLines("authorList.txt")
20      V(g.auth)$grado <- degree(g.auth)
21
22      # number of authors
23      gorder(g.auth)
24
25      # number of collaborations
26      gsize(g.auth)
27
28      true.grado <- mean(degree(g.auth))
29
30      hist.grado.boot <- function(rete, n.seed, n.wave, true.deg, bin.width = 0.05) {
31              samp.union <- lsmi_union(rete, n.seeds = n.seed, n.wave = n.wave)
32              dist.lsmi <- lsmi_dd(samp.union$lsmi_big, rete)
33              dist.boot <- boot_dd(dist.lsmi)
34
```

```
35              hist(dist.boot$mub,
36              freq = FALSE,
37              xlim = c(1, 5),
38              col = "light blue",
39              xlab = "",
40              main = paste0("Seed: ", n.seed, " Wave: ", n.wave))
41              abline(v = true.deg, lty = "dotted", lwd = 2)
42      }
43
44      seeds <- c(50, 100, 200)
45      wave <- c(1, 2, 3)
46
47      # Create a list of ggplot objects
48      par(mfrow = c(3, 3))
49      invisible(lapply(c(50, 100, 200), function(i) {
50              invisible(lapply(c(1, 2, 3), function(j) {
51                      hist.grado.boot(g.net, i, j, true.grado)
52              }))
53      }))
54
55      par(mfrow=c(1,1))
```

LISTING A.6: Codice utilizzato per la realizzazione di Figura 4.5

```
1       isolated.authors <- which(degree(g.auth) == 0)
2       g.auth.nozeros <- delete_vertices(g.auth, isolated.authors)
3       g.net.nozeros <- snowboot::igraph_to_network(g.auth.nozeros)
4
5       # grafico della distribuzione stimata del grado con 50 seed e 1 wave
6       samp.union.50 <- lsmi_union(g.net.nozeros, n.seeds = 50, n.wave = 1)
7       dist.lsmi.50 <- lsmi_dd(samp.union.50$lsmi_big, g.net.nozeros)
8       dist.boot.50 <- boot_dd(dist.lsmi.50)
9       plot(dist.boot.50, plotmu = FALSE)
10
11      dist.boot.50
12      # grafico della distribuzione stimata del grado con 500 seed e 1 wave
13      samp.union.500 <- lsmi_union(g.net.nozeros, n.seeds = 100, n.wave = 1)
14      dist.lsmi.500 <- lsmi_dd(samp.union.500$lsmi_big, g.net.nozeros)
15      dist.boot.500 <- boot_dd(dist.lsmi.500)
16      plot(dist.boot.500, plotmu = FALSE)
```

LISTING A.7: Codice utilizzato per la realizzazione di Figure 4.6 e 4.7

```
1       isolated.authors <- which(degree(g.auth) == 0)
2       g.auth.nozeros <- delete_vertices(g.auth, isolated.authors)
3       g.net.nozeros <- snowboot::igraph_to_network(g.auth.nozeros)
4
5       set.seed(1234)
6
7       param <- lsmi_cv(g.net.nozeros,
8       n.seeds = c(40, 50, 100, 200),
9       n.wave = 3,
10      B = 100,
11      proxyRep = 100,
12      proxySize = 13)
13      param
14      samp.union <- lsmi_union(g.net.nozeros, n.seeds = param$best_combination[1],
```

```
15        n.wave = param$best_combination[2])
16        dist.lsmi <- lsmi_dd(samp.union$lsmi_big, g.net.nozeros)
17        dist.boot <- boot_dd(dist.lsmi, B = 500)
18        dist.boot
19        plot(dist.boot)
20
21        # intervallo confidenza media con FPB
22        boot_ci(dist.boot)
23
24        plot(boot_ci(dist.boot))
25
26        dist.boot
27
28        dist.stim <- dist.boot$fk[2:6]
29        dist.stim
30
31        # generazione di un ccs di gradi
32        M <- 50
33        n <- gorder(g.auth.nozeros)
34        idx <- sample(1:n, size = M)
35        degrees <- V(g.auth.nozeros)[idx]$grado
36        degrees
37        table(degrees)
38
39        true.deg <- (table(degree(g.auth.nozeros)) / 3453)[1:5]
40
41        B <- 500
42        boot.matrix <- matrix(NA, B, M)
43        # bormal bootstrap
44        boot.matrix <- sapply(1:B, function(i) sample(degrees, size = M, replace = TRUE))
45
46        # stima grado medio e i.c.
47        medie.boot <- apply(boot.matrix, 2, mean)
48        mean(medie.boot)
49        quantile(medie.boot, c(0.025, 0.975))
50
51        # bootstrap distribution of degree
52        prop.num <- function(col, num) sum(col == num)/50
53        res1 <- apply(boot.matrix, 2, function(col) prop.num(col, 1))
54        res2 <- apply(boot.matrix, 2, function(col) prop.num(col, 2))
55        res3 <- apply(boot.matrix, 2, function(col) prop.num(col, 3))
56        res4 <- apply(boot.matrix, 2, function(col) prop.num(col, 4))
57        res5 <- apply(boot.matrix, 2, function(col) prop.num(col, 5))
58
59        means.freq.boot <- c(mean(res1), mean(res2), mean(res3),
60        mean(res4), mean(res5))
61
62        quantile.mat <- matrix(NA, 5, 2)
63        quantile.mat[1,] <- quantile(res1, c(0.025, 0.975))
64        quantile.mat[2,] <- quantile(res2, c(0.025, 0.975))
65        quantile.mat[3,] <- quantile(res3, c(0.025, 0.975))
66        quantile.mat[4,] <- quantile(res4, c(0.025, 0.975))
67        quantile.mat[5,] <- quantile(res5, c(0.025, 0.975))
68
69        quantile.mat
70
71        df.estimates <- data.frame(
72        x = rep(1:5, 2),
```

```
73        estimate = c(dist.stim, means.freq.boot),
74        fpb = as.factor(c(rep(1, 5), rep(0, 5))),
75        lower.ci = c(ci.sim$fk_ci[1,2:6], quantile.mat[,1]),
76        upper.ci = c(ci.sim$fk_ci[2,2:6], quantile.mat[,2])
77        )
78
79        df.true <- data.frame(
80        x = 1:5,
81        true = true.deg
82        )
83
84        ggplot(df.estimates, aes(x = x, y = estimate, col = fpb)) +
85        geom_point(aes(shape = fpb), position = position_dodge(width = .3),
86        size = 3) +
87        geom_errorbar(aes(ymin = lower.ci, ymax = upper.ci),
88        position = "dodge", width = .3) +
89        labs(title = "Confronto distribuzione grado stimata con FPB e bootstrap",
90        x = "Grado", y = "Frequenza stimata") +
91        scale_color_manual("Metodo utilizzato",
92        values = c("red", "blue"),
93        labels = c("Bootstrap", "FPB")) +
94        guides(shape = "none") +
95        theme_bw()
96
97        # boxplot per intervallo di confidenza del grado medio
98        df.medie.stim <- data.frame(
99        medie = c(dist.boot$mub, medie.boot),
100       method = as.factor(c(rep("FPB", 500), rep("Bootstrap", 500)))
101       )
102
103       ggplot(df.medie.stim) +
104       geom_boxplot(aes(x = method, y = medie, fill = method, alpha = 0.5)) +
105       geom_hline(aes(yintercept = mean(V(g.auth.nozeros)$grado),
106       linetype = "Vera media"),
107       col = "red") +
108       labs(title = "Boxplot per il grado medio della rete di citazioni",
109       x = "Metodo utilizzato", y = "Grado",
110       fill = "Metodo utilizzato") +
111       scale_linetype_manual(name = "", values = c("Vera media" = "dashed")) +
112       guides(alpha = "none") +
113       theme_bw()
114
115       # proviamo ad ottenere un nuovo grafico utilizzando 100 nodi per il bootstrap
116       # normale
117       M.2 <- 100
118       idx.100 <- sample(1:n, size = M.2)
119       degrees.100 <- V(g.auth.nozeros)[idx.100]$grado
120
121       B <- 500
122       boot.matrix.100 <- matrix(NA, B, M.2)
123       # bormal bootstrap
124       boot.matrix.100 <- sapply(1:B, function(i) sample(degrees.100, size = M.2,
125       replace = TRUE))
126
127       # stima grado medio e i.c.
128       medie.boot.100 <- apply(boot.matrix.100, 2, mean)
129       mean(medie.boot.100)
130       quantile(medie.boot.100, c(0.025, 0.975))
```

```
131
132         df.medie.stim.2 <- data.frame(
133         medie = c(dist.boot$mub, medie.boot, medie.boot.100),
134         method = as.factor(c(rep("FPB", 500), rep("Bootstrap (50 nodi)", 500),
135         rep("Bootstrap (100 nodi)", 500)))
136         )
137
138         ggplot(df.medie.stim.2) +
139         geom_boxplot(aes(x = method, y = medie, fill = method, alpha = 0.5)) +
140         geom_hline(aes(yintercept = mean(V(g.auth.nozeros)$grado),
141         linetype = "Vera media"),
142         col = "red") +
143         labs(title = "Boxplot per il grado medio della rete di citazioni",
144         x = "Metodo utilizzato", y = "Grado",
145         fill = "Metodo utilizzato") +
146         scale_linetype_manual(name = "", values = c("Vera media" = "dashed")) +
147         guides(alpha = "none") +
148         theme_bw()
```

Listing A.8: Codice utilizzato per il confronto tra FPB e Bootstrap

Bibliografia

Bottacin, Francesco (2008). *Note del Corso di TEORIA DEI GRAFI*. https://www.math.unipd.it/~bottacin/books/grafi_note.pdf.

Chen, Yuzhou et al. (2018). "Snowboot: Bootstrap Methods for Network Inference". In: *The R Journal* 10.2, pp. 95–113. DOI: 10.32614/RJ-2018-056. URL: https://doi.org/10.32614/RJ-2018-056.

Chernick, Michael R. e Robert A. La Budde (2011). *An introduction to bootstrap methods with application to R*. Hoboken: Wiley.

Davison, A. C. e D. V. Hinkley (1997). *Bootstrap Methods and their Application*. Cambridge Series in Statistical and Probabilistic Mathematics. Cambridge University Press.

Di Fonzo, Tommaso e Francesco Lisi (2005). *Serie storiche economiche: analisi statistiche e applicazioni*. Roma: Carrocci.

Efron, Bradley e Robert J. Tibshirani (1993). *An Introduction to the Bootstrap*. Monographs on Statistics and Applied Probability 57. Boca Raton, Florida, USA: Chapman & Hall/CRC.

Erdös, P e A Rényi (1959). "On Random Graphs I". In: *Publicationes Mathematicae Debrecen* 6, pp. 290–297.

Gel, Yulia R., Vyacheslav Lyubchich e L. Leticia Ramirez Ramirez (2017). "Bootstrap quantification of estimation uncertainties in network degree distributions". In: *Scientific Reports* 7.1, p. 5807. ISSN: 2045-2322. DOI: 10.1038/s41598-017-05885-x. URL: https://doi.org/10.1038/s41598-017-05885-x.

Gilbert, E. N. (1959). "Random Graphs". In: *The Annals of Mathematical Statistics* 30.4, pp. 1141–1144. ISSN: 00034851. URL: http://www.jstor.org/stable/2237458.

Hofstad, Remco van der (2016). *Random Graphs and Complex Networks*. Cambridge Series in Statistical and Probabilistic Mathematics. Cambridge University Press.

Holland, Paul W. e Samuel Leinhardt (1981). "An Exponential Family of Probability Distributions for Directed Graphs". In: *Journal of the American Statistical Association* 76.373, pp. 33–50. DOI: 10.1080/01621459.1981.10477598.

Ji, Pengsheng e Jiashun Jin (2016). "Coauthorship and citation networks for statisticians". In: *The Annals of Applied Statistics* 10.4, pp. 1779 –1812. DOI: 10.1214/15-AOAS896. URL: https://doi.org/10.1214/15-AOAS896.

Kolaczyk, Eric D. (2009). *Statistical Analysis of Network Data*. Springer Series in Statistics. Springer.

R Core Team (2024). *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing. Vienna, Austria. URL: https://www.R-project.org/.

Schweinberger, Michael et al. (2020). "Exponential-Family Models of Random Graphs: Inference in Finite, Super and Infinite Population Scenarios". In: *Statistical Science* 35.4, pp. 627–662. ISSN: 08834237, 21688745. URL: https://www.jstor.org/stable/26997939 (visitato il 25/04/2024).

Thompson, Mary et al. (2016). "Using the bootstrap for statistical inference on random graphs". In: *Canadian Journal of Statistics* 44, pp. 3–24. DOI: 10.1002/cjs.11271.

Watts, Duncan J. e Steven H. Strogatz (1998). "Collective dynamics of 'small-world' networks". In: *Nature* 393.6684, pp. 440–442. DOI: 10.1038/30918.