UNIVERSITÀ DEGLI STUDI DELL'AQUILA

DIPARTIMENTO DI INGEGNERIA E SCIENZE DELL'INFORMAZIONE E MATEMATICA

CORSO DI LAUREA IN INFORMATICA

Stima efficiente del numero di Network Motifs tramite Color-Coding e decomposizioni bilanciate

Relatore:

Dott. Stefano Leucci

Candidato:

Giulia Scoccia

Correlatore:

Prof. Guido Proietti

Matricola:

249503

ANNO ACCADEMICO 2019–2020

INDICE

CAPITOLO 1

INTRODUZIONE

I Motif, anche chiamati Graphlet o Pattern, sono piccoli sottografi connessi indotti di un grafo, la conta dei motif è un problema ben noto del graph mining e dell'analisi dei social network. Dato in input,un grafo G e un intero positivo k il problema richiede di contare per ogni graphlet H di k nodi, il numero di sottografi indotti di G isomorfi ad H. Comprendere la distribuzione dei motif permette di avere una conoscenza delle interazioni tra le proprietà strutturali e i nodi del grafo e inoltre fa luce sul tipo di strutture locali presenti in esso, che possono essere usate per una miriade di analisi. Poichè il conteggio dei graphlet può risultare computazionalmente impegnativo, di solito ci si accontenta di obiettivi meno ambiziosi. Uno di questi è la stima approssimata della frequenza: per ogni sottografo si richiede di stimare, nel modo più accurato possibile, la sua frequenza relativa rispetto a tutti i sottografi della stessa dimensione. Ancora meno ambiziosamente, visto che il numero di sottografi di una data dimensione cresce in modo esponenziale, si restringe l'attenzione al problema della stima della frequenza relativa solo ai sottografi che compaiono il maggior numero di volte nel grafo input. Ci sono due approcci per ottenere tali stime. Il primo è basato sull'utilizzo delle catene di Markov Monte Carlo, mentre il secondo è quello della tecnica del Color Coding introdotta da Alon, Yuster e Zwick [1]. Studi recenti mettono in luce e studiano le differenze tra i due approcci [2]. In questa tesi ci concentreremo solo sulla tecnica del Color Coding. Tale tecnica è stata introdotta da Alon, Yuster e Zwick in [1], per risolvere in ma-

niera randomizzata il problema di determinare l'esistenza di cammini ed alberi in G. Un'estensione di questa tecnica consente di ottenere garanzie statistiche forti per il problema del Motif Counting, da cui le frequenze possono essere facilmente derivate, tali tecniche sono state utilizzate per l'analisi di reti sociali e biologiche [2, 3, 4]. Tale estensione si basa su due osservazioni chiave. La prima è che il Color Coding può essere usato per costruire "un'urna" astratta che contiene un sottoinsieme statisticamente rappresentativo di tutti i sottografi di G (non necessariamente indotti) che hanno esattamente k nodi e sono alberi. La seconda osservazione è che il compito di campionare k-graphlet, ossia graphlet con k nodi, può essere ridotto, con un overhead minimale, a campionare k-alberi, alberi con k nodi, dall'urna. Si può così stimare il numero dei motif in due fasi: la "fase di costruzione", in cui si crea l'urna da G e la "fase di campionamento", dove si campionano i graphlet fino ad ottenere delle stime accurate per i graphlet di interesse.

1.1 Contributo della tesi

In questo lavoro di tesi, l'attenzione è stata concentrata sull'ottimizzazione di un algoritmo basato sulle tecnica del Color Coding per la ricerca di k-treelet all'interno di grafi più o meno grandi. Per k-treelet , si intendono alberi (non necessariamente indotti) in un grafo con k nodi. È stato visto in uno studio del 2008 [4] su una rete PPI (Protein-Protein Interaction)quanto la ricerca di k-treelet in un grafo può essere utile per la ricerca della frequenza di particolari strutture biomolecolari (unicellulari e pluricelluri). Per effettuare tale ricerca è stato necessario concentrarsi sulla fase costruttiva descritta in precedenza.

La fase costruttiva, è descritta mediante una programmazione dinamica , è un processo che però richiede un grande impiego di tempo e spazio. Il lavoro svolto ha portato, per prima cosa ad un'implementazione, in Java dell'algoritmo noto [2]. Il programma permette la ricerca delle occorrenze dei diversi k-treelet, all'interno del grafo. L'approccio dell'algoritmo utilizzato in questa tesi è bottom-up, per cui, supposto di dover conteggiare i treelet di dimensione k di un grafo, l'algoritmo lavora in esattamente k fasi. Nell'i-esima fase saranno conteggiati i treelet di dimensione i, ottenuti dalla composizione di tutti quelli con dimensione minore di i,perciò per

poter calcolare i treelet di dimensione k, sarà necessario aver già calcolato quelli di dimensione fino a k-1. Questo meccanismo rispetta ciò che viene dalle formule ricorsive della programmazione dinamica. Poichè il numero degli alberi cresce in maniera esponenziale rispetto a k l'algoritmo richiede, al crescere di k, sempre più tempo per essere eseguito. A tal proposito nella tesi viene proposta un'ottimizzazione, basata su opportune decomposizioni "bilanciate" degli alberi, che consente di rendere indipendenti i conteggi dei treelet di dimensione k da $\frac{1}{3}$ dei conteggi precedenti. Questo consente di eseguire le prime, circa $\frac{2}{3}$ k fasi prima della fase k, comportando un risparmio notevole di tempo. Ad esempio su un grafo con 63731 nodi e 817090 archi, l'algoritmo non ottimizzato richiede DA VEDERE tempo per la ricerca dei treelet con DA VEDERE nodi, mentre quello ottimizzato richiede un tempo DA VEDERE.

1.2 Organizzazione del testo

La descrizione del lavoro è strutturata nel seguente modo. Nel capitolo 2 viene descritta la tecnica del color coding e il suo utilizzo per il conteggio degli alberi. Si vedrà l'algoritmo di [2] e la sua formulazione "top-down". Si discuterà la scelta di adottare un approccio "bottom-up" per l'implementazione e i suoi vantaggi. Nel capitolo 3 si discuterà in dettaglio la tecnica delle decomposizioni bilanciate ed il relativo impatto sull'algoritmo. Anche in questo caso si discuterà sulle scelte effettuate in fase implementativa. Nel capitolo 4 si discuteranno i risultati di un'analisi sperimentale delle performance dell'algoritmo ottimizzato rispetto alla versione di [2]. Infine, nel capitolo 5, veranno discusse le possibili estensioni del presente lavoro di tesi.

CAPITOLO 2

COLOR CODING

Nel capitolo verrà descritta la tecnica del Color Coding utilizzata in questo studio.

La tecnica fu presentata per la prima volta nel 1995, da Alon, Yuster e Zwick [1]. In generale, dato un grafo $G = (V, E)$, il problema dell'isomorfismo dei sottografi di G è un problema $NP - completo$. Il metodo del Color Coding permette di risolvere sottocasi di questo problema in tempo polinomiale.

Dati un grafo $G = (V, E)$ ed uno $H = (V_H, E_H)$, i vertici V di G, in cui verrà cercato un sottografo isomorfo ad H, sono colorati casualmente di $k = |V_H|$ colori. Se $|V_H| = O(\log(V))$, allora, tutti i vertici del sottografo di G isomorfo ad H, se esiste, saranno colorati da colori distinti.

Il primo algoritmo descritto, però, si limitava alla ricerca di sottografi indotti in un grafo, senza farne un conteggio.

È per questo motivo che in questo capitolo presenteremo un'estensione dell'algoritmo dato da Alon [1], per effettuare un conteggio dei Motif all'interno del grafo. Dato in input un grafo $G = (V, E)$ e un numero k, per prima cosa il color coding assegna uniformemente e indipendentemente per ogni nodo di G un'etichetta in $[k] := \{1, ..., k\}$, indicato come un colore. L'obiettivo ÃÍ quello di conteggiare il numero di alberi colorati non indotti di $k - nodi$ in G - chiamati *treelet* - i cui colori non sono ripetuti. Questo viene fatto in maniera efficiente mediante una programmazione dinamica, tecnica bottom-up che identifica dei sottoproblemi del

problema originario, procedendo logicamente dai problemi piÃź piccoli verso quelli piÃź grandi. Grazie al fatto che alberi con insiemi disgiunti di colori devono giacere su insiemi digiunti di nodi.

2.1 Algoritmo

Qui descriviamo l'estensione dell'algoritmo del color coding che può contare e campionare treelet colorati non indotti uniformemnte a caso. L'algoritmo consiste in una fase di costruzione e una fase di campionamento, in questo studio però non si vedrá quest'ultima fase, poichè non rilevante ai fini di questa tesi, ma ci concentreremo sulla prima fase. L'algoritmo inizia con una prima fase di colorazione dove per ogni nodo $v \in V$ di G è assegnato un colore $c(v)$, scelto indipendentemente e uniformemente a caso da $[k]$. L'obiettivo della fase di costruzione è quello di costruire una tabella del conteggio dei treelet. Per ogni v e per ogni albero colorato T_C con k nodi, si vuole un conteggio $c(T_C, v)$ del numero di copie di T_C in G che sono radicate in v (si noti che qui si intendono copie non indotte). A questo fine per ogni v si inizializza $c(T_C, v) = 1$, dove T è il treelet triviale di 1 nodo e $C = \{c_v\}$. Successivamente si esegue una programmazione dinamica per il conteggio di treelet di dimensione $h = 2, ..., k$. Per ogni h a turno, si considera ogni possibile albero radicato T con $h \leq k$ nodi e ogni possibile insieme $C \subseteq [k]$ con $|C| = h$. Poi, $\forall v \in V$,si calcola come segue il numero $c(T_C, v)$ di occorrenze dei treelet (non indotti) radicati in v isomorfi a T e i cui colori giacciono nell'insieme C. Si divide idealmente T in due sottoalberi radicati T' e T'' radicati rispettivamente nella radice r di T e in uno dei figli di r.
Perciò $c(T_C, v)$ è dato come segue:

$$c(T_c, v) = \frac{1}{\beta_T} \sum_{u \sim v} \sum_{\substack{C', C'' \subset C \\ C' \cap C'' = 0}} c(T'_{C'}, v) \cdot c(T''_{C''}, u) \tag{1}$$

dove β_T è una costante di normalizzazione che é uguale al numero di alberi di T isomorfi a T'' radicati in un figlio di r. Per calcolare $c(T_C, v)$ si passa attraverso tutti gli archi uv di G, combinando i contatori di u e v. La correttezza e la complessità di questa costruzione, non sono trattate qui,ma vengono dimostrate in [1].

Algorithm 1: Fase di costruzione

input : Grafo G, dimensione del treelet k ;

for v *in* G **do**

  $C = \{c_v\}$;

  $c(T_{c(v)}, v) = 1$;

end

for $h = 2$ *to* k **do**

  for v *in* G **do**

    foreach $T : |T| = h$ **do**

      $c(T_c, v) = \frac{1}{\beta_T} \sum_{u \sim v} \sum_{\substack{C', C'' \subset C \\ C' \cap C'' = 0}} c(T'_{C'}, v) \cdot c(T''_{C''}, u)$

    end

  end

end

CAPITOLO 3

DECOMPOSIZIONI BILANCIATE

BIBLIOGRAFIA

[1] Noga Alon, Raphael Yuster, and Uri Zwick. Color-coding. *Journal of the ACM (JACM)*, 42(4):844–856, 1995.

[2] Marco Bressan, Flavio Chierichetti, Ravi Kumar, Stefano Leucci, and Alessandro Panconesi. Motif counting beyond five nodes. *ACM Transactions on Knowledge Discovery from Data (TKDD)*, 12(4):1–25, 2018.

[3] Marco Bressan, Stefano Leucci, and Alessandro Panconesi. Motivo: fast motif counting via succinct color coding and adaptive sampling. *Proceedings of the VLDB Endowment*, 12(11):1651–1663, 2019.

[4] Noga Alon, Phuong Dao, Iman Hajirasouliha, Fereydoun Hormozdiari, and S Cenk Sahinalp. Biomolecular network motif counting and discovery by color coding. *Bioinformatics*, 24(13):i241–i249, 2008.