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

In questo capitolo verrà descritta la tecnica del Color Coding introdotta da Alon, Yuster e Zwick in [1] , la sua estensione algoritmica e l'utilizzo che ne è stato fatto in questa tesi , per il calcolo della frequenza di k-treelet in un grafo.

2.1 Tecnica del Color Coding

La tecnica del Color Coding 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.

La tecnica del *color-coding* è una tecnica randomizzata. 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.

Probabilmente i sottocasi più interessanti e semplici di problemi di sottografi isomorfi sono i seguenti. Dato un grafo $G = (V, E)$ (diretto o non) e un numero k, G può contenere un percorso(diretto o non) di lunghezza k? G può contenere un ciclo (diretto o non) di lunghezza *esattamente* k?

Sia $G = (V, E)$ un grafo diretto o indiretto. Si consideri il problema di trovare un percorso semplice (diretto o no) di lunghezza $k-1$ in G. Data una colorazione dei vertici di G con k colori. Un percorso in G è detto *colorato* se ogni suo vertice è colorato da un colore distinto. Un percorso colorato è chiaramente semplice. Ogni percorso semplice di lunghezza $k-1$, d'altro canto, ha una possibilità di diventare colorato di $k!/k^k > e^{-k}$. Si noti che questa quantità è solo esponenzialmente piccola in k. Quello che si andrà però a vedere, è il tempo necessario per trovare un percorso colorato di lunghezza $k-1$ in G, se esiste, o tutte le coppie di vertici connesse da percorsi colorati di lunghezza $k-1$ in G.

Lemma 2.1.1. *Sia $G = (V, E)$ un grafo diretto o indiretto e sia $c : V \to \{1, ..., k\}$ una colorazione dei suoi vertici con k colori. Un percorso colorato in G, se esiste, può essere trovato, nel caso peggiore, in $2^{O(k)} \cdot E$*

Dimostrazione. Per prima cosa viene descritto un algoritmo di complessità temporale $2^{O(k)} \cdot E$, che preso in input il grafo $G = (V, E)$, la colorazione $c : V \to \{1, ..., k\}$ ed un vertice $s \in V$ restituisce, se esiste, un percorso colorato di lunghezza $k-1$ che parte da s. Per trovare un percorso colorato di lunghezza $k-1$ in G che parte in un qualche punto si aggiunge semplicemente a G un nuovo vertice s', colorato con un nuovo colore 0 e connesso con archi a tutti i vertici di V. A questo punto si cerca un percorso colorato di lunghezza k che parte da s'. Un percorso colorato di lunghezza $k-1$ che inizia ad un specifico vertice s è trovato usando un approccio di programmazione dinamica. Si suppone di aver già trovato per ogni vertice $v \in V$ i possibili insiemi di colori dei percorsi di lunghezza i che connettono s a v. Si noti che non vengono registrati tutti i percorsi colorati da s a v, ma solo l'insieme dei colori di ogni percorso. Per ogni vertice v si ha perció una collezione di al più $\binom{k}{i}$ insiemi di colori. Si ispeziona ogni sotto insieme C che appartiene alla collezione di v, e ogni arco $(u, v) \in E$. Se $c(u) \notin C$, si aggiunge l'insieme $C \cup \{c(u)\}$ alla collezione di u degli insiemi di colori dei percorsi colorati di lunghezza $i+1$. Il grafo G contiene un percorso colorato di lunghezza $k-1$ rispetto alla colorazione c se e solo se la collezione finale, che corrisponde ai percorsi di dimensione $k-1$ di almeno un vertice è non vuota. Il numero di operazioni svolte dall'algoritmo appena descritto sono al più $O(\sum_{i=o}^{k} i\binom{k}{i} \cdot |E|)$ che è chiaramente $O(k2^k \cdot E)$. □

Lemma 2.1.2. *Sia $G = (V, E)$ un grafo diretto o indiretto e sia $c : V \to \{1, ..., k\}$ una colorazione dei suoi vertici con k colori. Tutte le coppie di vertici connessi da percorsi colorati di lunghezza $k-1$ in G può essere trovato in tempo $2^{O(k)} \cdot VE$ o $2^{O(k)} \cdot V^{\omega}$ nei casi peggiore.*

Dimostrazione. L'algoritmo con complessità $2^{O(k)} \cdot VE$ è ottenuto dalla semplice esecuzione dell'algoritmo descritto nella prova del Lemma 2.1.1, un numero $|V|$ di volte, una volta per ogni vertice iniziale.
Per ottenere l'algoritmo con complessità $2^{O(k)} \cdot V^{\omega}$, invece, si usa il seguente approccio. Si enumerano tutte le partizioni dell'insieme dei colori $\{1, 2, ..., k\}$ in due sottoinsiemi C_1 e C_2 di taglia $k/2$ ciascuno. Ci sono solo $\binom{k}{k/2} < 2^k$ di queste partizioni. Per ognuna delle partizioni C_1 e C_2, siano V_1 l'insieme dei vertici di G colorati da colori di C_1 e V_2 l'insieme dei vertici colorati da colori di C_2. Siano G_1 e G_2 i sottografi di G indotti da V_1 e V_2 rispettivamente. Ricorsivamente si trovano tutte le coppie di vertici di G_1 e G_2 connesse da percorsi colorati di lunghezza $k/2 - 1$. Tutte le informazioni vengono poi raccolte in due matrici booleane A_1 e A_2. Sia B una matrice booleana che le relazioni di adiacenza tra i vertici di V_1 e quelli di V_2. Il prodotto booleano A_1BA_2 restituisce tutte le coppie di V che sono connesse da percorsi colorati di lunghezza esattamente $k-1$, dove i primi $k/2$ vertici del percorso sono colorati da colori di C_1 e i rimanenti da colori di C_2. Dall'unione (OR) delle matrici ottenute per tutte le possibili partizioni si ottiene il risultato desiderato. È facile da verificare che la complessità di questo approccio è infatti $2^{O(k)} \cdot V^{\omega}$, come il numero di moltiplicazione di matrice utilizzato, $t(k)$, soddisfa la ricorrenza $t(k) \leq 2^k t(k/2)$. □

Usando i precedenti Lemma si ottengono i seguenti risultati.

Teorema 2.1.3. *Un percorso semplice diretto o indiretto di lunghezza $k-1$ in un grafo (diretto o indiretto) $G = (V, E)$ che contiene tale percorso può essere trovato in un tempo atteso di $2^{O(k)} \cdot V$ nel caso indiretto e in $2^{O(k)} \cdot E$ nel caso diretto.*

Teorema 2.1.4. *Un ciclo semplice diretto o indiretto di taglia k in un grafo $G = (V, E)$ (diretto o indiretto) può essere trovato in un tempo atteso di $2^{O(k)} \cdot VE$ o $2^{O(k)} \cdot V^{\omega}$.*

Il metodo di color-coding può essere usato in maniera efficiente non solo per trovare cicli o percorsi colorati, ma ogni sottografo con una larghezza dell'albero delimitata.

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.