Possibili soluzioni alla applicazione di RI per grafi temporali

Locicero Giorgio

July 16, 2020

Contents

1 Definizioni tecniche

La definizione di grafo temporale (temporalmente discreto e non basato su intervalli di tempo continuo) è molto simile a quella di grafo normale $G = \{V, E\}$ dove gli archi tra i nodi $E = \{u, v, t\}$ sono visti come singoli contatti avvenuti in un certo quanto temporale ben definito (molto simile ad una macchina a stati), non considero il caso di intervalli temporali continui perché gli studi a riguardo sono inesistenti e mi sembra che lo state of the art sia utilizzare grafi temporali con contatti ($t \in \mathbb{N}$ e non $t \in [a, b] | a, b \in \mathbb{R}$).

La maggior parte delle definizioni ed intuizioni per la risoluzione del problema è stata trovata in tutta la bibliografia , in particolare in [5].

Vengono considerati grafi temporali direzionati dove gli archi hanno una sola direzione, ma i risultati possono essere espansi a grafi indirezionati molto semplicemente

Di seguito presento delle possibili definizioni utili per l'attuazione del subgraph matching(eventuali modifiche possono essere supposte sulla base dei modelli e degli obiettivi):

Definition 1. Un grafo G1 è isomorfico ad un sottografo di un grafo G2 se e solo se esiste una corrispondenza uno a uno tra i nodi del sottografo di G2 e G1 che preserva l'adiacenza, più formalmente $\exists f : V_{G1} \to V_{G2} | (s, d) \in E_{G1} \implies (f(s), f(d)) \in E_{G2}$

Per la definizione di isomorfismi su sottografi temporali, definizioni concrete non sono state trovate, dato che ognuno degli autori che hanno esposto delle

soluzioni lo hanno fatto sulla base dei loro bisogni, senza generalizzare il concetto.

Anche [5] non ha dato una definizione concreta, concentrandosi sulla definizione di sottografo temporale che rispetti la proprietà di time-respecting (che tenterò di modificare in seguito), ed ha tralasciato la generalizzazione del problema andando direttamente alla soluzione adatta al problema che voleva risolvere (quindi considerando i sottografi matchati nel grafo target che rispettassero la loro definizione di time-respecting subgraph).

I problemi sui possibili match restano comunque anche perchè gli autori dei paper citati nelle referenze (su isomorfismi di grafi temporali) sembravano avere le idee confuse sul matching stesso, talvolta non considerando totalmente il parametro temporale del grafo query, ma svolgendo un matching (edge-driven nella maggior parte dei casi) che rispettasse le proprietà già citate di archi time-respecting (a coppie con destinazione e sorgente uguale o viceversa), che tenterò di descrivere prossimamente

Definition 2. Data una coppia di archi $(e_r, e_j), e = (s, d, t)$ dove $d_r = s_j$ o viceversa, dato $\delta \in \mathbb{N}, \delta > 0$, si deve avere $0 \leq |t_j - t_r| \leq \delta$

Definition 3. Data una sequenza di archi $\{e_1, \ldots, e_n\}$, questa sequenza forma un time-respecting path se e solo se $r \in [1, n[\subseteq \mathbb{N}, e_r = (s_r, d_r, t_r) \in E_G, d_r = s_{r+1}, t_r < t_{r+1}$

Cioè un cammino time-respecting deve avere delle interazioni tra i vari nodi che siano non decrescenti .

Definition 4. Un grafo temporale GT1 è isomorfico ad un sottografo temporale di un grafo Temporale GT2 se e solo se esiste una corrispondenza uno a uno tra i nodi del sottografo temporale di GT2 e GT1 che preserva l'adiacenza ed il sottografo

Stranamente, nei paper più specifici per la risoluzione di subgraph-matching temporale [5] sembra che vengano considerati come grafi query semplici grafi statici, o più nello specifico non vengono stabiliti particolari tempi di contatto, ma semplici linee guida (cosa abbastanza comprensibile dato che i grafi query sono user-defined), ed i ragionamenti temporali sono principalmente sul grafo target, considerato come grafo temporale, e questo mi ha fatto sorgere dei dubbi non indifferenti, specialmente nel caso in cui il grafo query sia esso stesso un grafo temporale complesso.

Le strategie che verranno presentate nel prossimo capitolo rimangono comunque le stesse, solo che bisogna vedere effettivamente come fare il matching nel caso in qui anche il grafo query sia un grafo temporale.

Andando a prendere ispirazione da altri paper e di mia intuizione, penso che si possano attuare due tecniche nel caso in cui il grafo query sia esso stesso un grafo temporale:

- Utilizzare una trasformazione del grafo temporale query in un grafo statico, sempre secondo i pattern di trasformazione che tenterò di definire nel prossimo capitolo, e poi utilizzare questo grafo statico per il submatching sul grafo temporale target.

- svolgere dei confronti e un matching basato su confronti complessi

2 Possibilità utilizzabili ed implementabili

Le possibili soluzioni di subgraph matching possono essere principalmente 3:

- Time before topology: estrarre i sottografi temporali che rispettano le ipotesi di time-respecting, poi procedere con il matching dei sottografi topologico semplice. Questo approccio è particolarmente pesante come prestazioni, anche perché possono presentarsi ridondanze non trascurabili.
- Topology before time: svolgere il matching direttamente sulla topologia statica del grafo, e poi svolgere delle operazioni sulle componenti temporali dei grafi ottenuti, quindi pruning sui sottografi temporali effettivi che rispettino le ipotesi, questa è la strategia più semplice, anche perché si possono utilizzare direttamente algoritmi di subgraph matching noti, per poi gestire come si vuole i risultati ottenuti.
- Time and topology together: Il controllo temporale ed il matching vengono fatti nello stesso momento e quindi, prendendo come esempio un algoritmo di Ullman semplice, viene fatto un controllo sul matching di ogni nodo e di ogni edge, facendo pruning di soluzioni che non rispettino le ipotesi di time-respecting match.

Di seguito presento delle soluzioni per ogni categoria descritta, prima parlando di una soluzione che prima calcola i grafi statici indotti dai grafi temporali

Una effettiva possibilità utilizzabile per ridurre lo spazio di ricerca delle soluzioni è l'utilizzo di un intervallo temporale in cui i contatti possono avvenire in sequenza, cioè , supposto un insieme ordinato di contatti $\{e_1, e_2, \ldots, e_n\} | r \in [1, n] \subseteq \mathbb{N}, e_r = (s_r, d_r, t_r) \in E_G, t_r \leq t_{r+1}$ dove G è un grafo temporale, allora si può fare un pruning delle possibilità in base alla seguente regola: data una coppia di archi $(e_r, e_{r+e}), e > 0$ dove $d_r = s_{r+e}$si ha che ,dato $\delta \in \mathbb{N}, \delta > 0$,si deve avere $t_{r+e} - t_r \leq \delta$, cioè la propagazione deve essere avvenuta entro un tempo definito (in questo caso definito nei naturali ma la definizione può essere adattata in base alle esigenze), per una definizione che non ha bisogno dell'ordinamento dei contatti iniziale, e per vedere nel dettaglio delle strategie di risoluzione al problema, vedere [5].

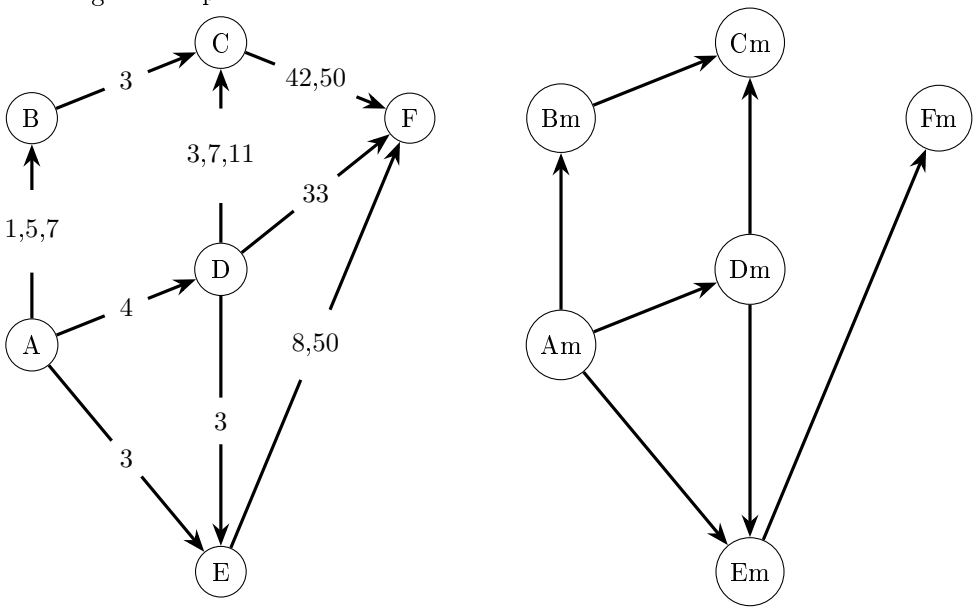
Il controllo dell'intervallo temporale è una parte chiave nella maggior parte degli algoritmi e nelle tecniche utilizzate dagli altri autori, anche perché simula bene i comportamenti reali

Una possilbilità di risoluzione per gli algoritmi di subgraph matching di grafi temporali utilizzata da altri autori [4, 2] è la conversione del grafo temporale in una o più forme statiche indotte che permettano il matching statico della rete, in particolare le strategie utilizzate introducono dei metodi molto semplici nella conversione, che contano semplicemente gli archi che rispettino una o più condizioni temporali

Algorithm 1 InducedStaticGraphFromTemporal(G)

```
1: function InducedStaticGraphFromTemporal(G,δ)
2:     edges sorted in sequence, select a node that have the less time of contact
3:     BFS modified with control over temporal attribute and δ, add edge to
   Gtemp if conditions are satisfied
4:     for tedge ∈ E_Gtemp do
5:         SG.addedge(tedge.s,tedge.d)
6:     end for
7:     return SG
8: end function
```

I seguenti grafi rappresentano una applicazione dell'algoritmo di trasformazione da grafo temporale:



Si è scelto un intervallo temporale di attivazione $\delta = 10$. Come si può facilmente notare, si è ottenuto un solo grafo statico, questa non è sempre la situazione, specialmente quando si ha il bisogno di tenere conto di tutti i pattern di attivazione che avvengono nel grafo temporale, quindi la soluzione presentata è abbastanza temporanea e non porterebbe risultati concreti (il problema resta comunque il caso di archi con più contatti).

Per implementare la strategia di $topologybeforetime$ bisogna semplicemente fare il matching sulla struttura statica del grafo target e poi fare il controllo sulla struttura dinamica, per esempio svolgendo un controllo sui nodi mappati e sui tempi di contatto di ogni nodo, in modo che rispettino le regole di time-respecting path definite.

Per implementare la strategia di $Timeandtopologytogether$, il ragionamento

dovrebbe essere abbastanza semplice e si concentrerebbe principalmente sulla fase di matching vera e propria (prendendo ad esempio Ullman).

Dopo la definizione del problema e dei punti chiave sul matching di grafi temporali, non dovrebbe essere difficile utilizzare le ipotesi di time-respecting path e contatti per ampliare e implementare regole di look-ahead e rollback.

È possibile comunque espandere altri concetti, come per esempio le fasi di preprocessing di RI di calcolo dei compatibility domains e dell'ordine dei nodi del grafo query per la fase di matching [3, 1], sempre utilizzando le definizioni definite precedentemente.

Per esempio, per il calcolo dei compatibility domains, si può fare comunque un controllo sul grado del nodo, e si può alternare ad un controllo sui tempi di contatto per ogni arco uscente dal nodo.

Per il calcolo dell'ordinamento dei nodi del grafo query, si potrebbe fare un ordinamento locale (rispetto al grado dei nodi) ed in seguito un ordinamento sul numero di contatti per arco (potrebbero sorgere dei problemi che verranno visti nel prossimo capitolo).

3 Problemi con l'utilizzo di grafi temporali

Gli autori delle referenze hanno principalmente utilizzato grafi temporali con archi aventi singoli contatti, considerazione abbastanza forte e che semplifica abbastanza il modello, specialmente nella fase di matching, che diventa semplicemente il controllo di una condizione aggiuntiva.

Nel caso di soluzione di matching basata su *Timeandtopologytogether* , la modifica dell'algoritmo di matching potrebbe portare all'inutilità della fase di preprocessing, in particolare l'ordinamento dei nodi della query (perchè in alcuni casi ci sarebbe l'obbligo di iniziare dal nodo che ha l'edge con il tempo minore), anche se questo rischio è discutibile e bisognerebbe provare una implementazione vera e propria.

La definizione del grafo temporale di query è abbastanza complessa, e influenza pesantemente le possibili implementazioni e soluzioni che possono essere utilizzate, nello specifico, quando si hanno più contatti in tempi diversi sullo stesso arco, cosa si dovrebbe cercare nel grafo target? Gli autori dei paper nelle referenze usano la scusa dei grafi query user-defined, che vogliono semplicemente rappresentare un pattern, quindi il problema non si pone.

Il problema resta comunque nel caso in cui si vogliano utilizzare dei grafi query con più contatti per arco, e questa possibilità non è di semplice soluzione e definizione.

References

[1] Vincenzo Bonnici, Rosalba Giugno, Alfredo Pulvirenti, Dennis Shasha, and Alfredo Ferro. A subgraph isomorphism algorithm and its application to biochemical data. *BMC bioinformatics*, 14(S7):S13, 2013.

[2] Patrick Mackey, Katherine Porterfield, Erin Fitzhenry, Sutanay Choudhury, and George Chin. A chronological edge-driven approach to temporal subgraph isomorphism. *mackey2018chronological*, pages 3972–3979, 2018.

[3] Giovanni Micale, Vincenzo Bonnici, Alfredo Ferro, Dennis Shasha, Rosalba Giugno, and Alfredo Pulvirenti. Multiri: Fast subgraph matching in labeled multigraphs. *arXiv preprint arXiv:2003.11546*, 2020.

[4] Ashwin Paranjape, Austin R Benson, and Jure Leskovec. Motifs in temporal networks. In *Proceedings of the Tenth ACM International Conference on Web Search and Data Mining*, pages 601–610, 2017.

[5] Ursula Redmond and Pádraig Cunningham. Subgraph isomorphism in temporal networks. *arXiv preprint arXiv:1605.02174*, 2016.