**Presentazione Stefano**

**L-GRAAL 1**

Il terzo metodo che vediamo è stato Sviluppato nel 2015, L-GRAAL è un metodo basato sull’idea di mappare insieme nodi che costituiscono un *pattern* (in questo caso dei *sottograﬁ* chiamati **graphlet**) deﬁnito da una grande quantità di interazioni condivise.

L-GRAAL è in grado di individuare l’*overlap* tra le reti e fornisce risultati migliori di tutti gli altri metodi GO-based a livello di mapping delle proteine e delle interazioni tra le stesse.

**L-GRAAL 2**

Questo metodo ottimizza una funzione obiettivo (indicare IP), che fonde le informazioni derivanti dalle sequenze di proteine con le interazioni tra i vari graphlet. Questa funzione viene risolta con la Programmazione Intera in O(blabla) dove |V| indica il numero di nodi nelle reti e *d* è il valore di grado massimo. Dal momento che con la sola Programmazione Intera otteniamo una soluzione parziale, è necessario ricondursi alla *formulazione duale* del problema.

Sfortunatamente, anche questo problema è -completo e, in pratica, si risolve fermando, di fatto, l’algoritmo dopo un determinato limite temporale o dopo un numero di iterazioni ﬁssato.

In tutti i test svolti L-GRAAL ha mostrato una percentuale di successo non indiﬀerente, superiore a tutti gli altri metodi con cui è stato confrontato, come IsoRank (indica nel grafico).

**STRUC2VEC 1**

Esiste un’altra branca di approcci al problema che invece fanno uso del machine learning. Che sono DeepWalk, node2vec e struct2vec. Vi presentiamo quella che raggiunge le prestazioni più elevate e supera i limiti delle altre. Si chi chiama struct2vec.

L’algoritmo si basa sul concetto di somiglianza strutturale tra due nodi. Intuitivamente, due nodi che hanno lo stesso numero di archi sono strutturalmente simili, ma se i loro vicini hanno anch’essi lo stesso numero di archi, allora sono ancora di più strutturalmente simili.

Ad esempio come potete vedere nell’immagine, il nodo U e V hanno 5 e 4 archi, quindi sono più o meno simili. Sono circondati da entrambi d 5 nodi che hanno 3 e 5 archi. L’unica differenza è l’arco tra C e U…

* è un framework ﬂessibile, ovvero polte sue componenti possono essere cambiate a piacere.
* apprende le *latent representations* delle somiglianze strutturali di ogni nodo, ovvero per ogni nodo l’algoritmo restituisce un vettore che rappresenta la sua struttura all’interno del grafo.
* è molto resistente in caso di rumore, ovvero archi mancanti.

**STRUC2VEC 2**

L’algoritmo si di vide in 4 fasi:

* Nella prima fase calcola per ogni coppia di nodi la loro somiglianza strutturale considerando il loro vicinato a dimensioni crescenti. Ovvero prima si considerano solo i nodi a distanza di 1 arco poi a distanza 2, ecc…
* Viene costruito un Grafo multilivello pesato.

Ogni livello è un grafo completo composto da tutti i nodi del grafo originale, il peso tra ogni coppia di nodi è calcolato con la seguente formula.

Cioè al primo livello compaiono le somiglianze strutturali calcolate considerando solo i nodi ad 1 arco di distanza.

Inoltre ogni nodo è collegato col corrispondente nel layer inferiore e superiore.

* Si attraversa più volte il grafo multilivello con un algoritmo semi-randomico, generando diverse sequenze di nodi.
* Infine, utilizza *Skip-Gram*, una tecnica di *unsupervised learning* per generare la *latent representations* (un vettore) per ogni nodo.

In sostanza si trasforma il grafo originale in tante sequenze di nodi, da cui poi un algoritmo non supervisionato genera per ogni nodo un vettore che rappresenta la sua somiglianza strutturale.

2 nodi strutturalmente simili hanno latent rappresentations vicine, ovvero sono punti in uno spazio multidimensionale vicini.

**STRUC2VEC 3**

*Adesso vediamo un esempio per chiarire.*

*struc2vec* è stato testato in diversi scenari e confrontato con gli algoritmi allo stato dell’arte (*DeepWalk* e *node2vec*).

Il primo test che vediamo, utilizza il *barbell graph*.

Prendiamo in considerazione il nodo GIALLO: …

Quindi i 2 nodi gialli sono strutturalmente identici……..

*struc2vec* individua le *latent representations* posizionando i nodi strutturalmente equivalenti gli uni vicino agli altri.

Mentre algoritmi allo stato dell’arte come *DeepWalk* e *node2vec* falliscono.

**STRUC2VEC 4**

Un secondo test svolto su una rete più complessa chiamata *Zachary’s Karate Club network*.

Questa rete è composta da 34 nodi e 78 archi. La rete è stata duplicata (mostrare in alto a destra) I due graﬁ sono stati connessi tramite un arco fra i nodi 1 e 37 (indicare il collegamento).

Anche in questo caso *DeepWalk* e *node2vec* falliscono nell’individuare le *latent* *representations* di nodi strutturalmente equivalenti, mentre *struc2vec* fornisce i risultati migliori.

Ad esempio, i 2 nodi VERDI che collegano i due grafi… sono vicini. in node2vec e deepwalk no.

**CONCLUSIONI**

Concludendo abbiamo presentato il problema del Network Alignement che invece di allineare stringhe di DNA, cerca di allineare e trovare similitudini tra grafi.

Il più importante grafo che si va a studiare è quello delle Interazione Proteina Proteina degli organismi, in quanto negli ultimi anni, il corpus di dati PPI è cresciuto esponenzialmente.

Abbiamo presentato 4 metodi:

* MTGO: possiede solo 13 citazioni
* Isorank: ha 500 citazioni ma in calo dal 2008
* LGRAAL: come abbiamo visto è empiricamente migliore di Isorank

Purtroppo, non abbiamo trovato paper che mettessero a confronto questi 3 con struct2vec

* Struct2vec: anche se è molto complesso, e noi ci siamo limitati a presentarlo a grandi linee, ad ogni step molte sue parti possono essere cambiate, ad esempio la formula della somiglianza strutturale può essere modificata, l’algoritmo di unsupervised learning può essere cambiato, ecc…

è stato citato oltre 300 volte dal 2017 e per gli ambiti più diﬀerenti.

Ha una complessità migliore di Isorank.

Quindi sembra che la ricerca si muova verso algoritmi basati sul Machine Learning come struct2vec.