Network Analysis
Network robustness

Samuele Crea

June 2024

1 The Study

In questo ultimo assignment lo scopo è quello di simulare failures di nodi all'interno di un grafo per misurarne la robustezza.

Per far ciò verranno eseguiti diversi tipi di attacchi: alcuni meno efficaci come dei random failure ed altri più mirati, come l'eliminazione di nodi basandosi sul loro valore di betweenness, degree e di pagerank.

Una volta fatto ciò su grafi più piccoli tenterò di applicare lo script anche sul grafo utilizzato per i primi due assignment.

A questo punto, la seconda parte dell'assignment sta nel migliorare la robustezza del grafo, cercando di capire il modo più efficace per farlo.

2 Early tests on small graph

Come già detto, i 4 tipi di failure sono stati testati su grafi piccoli, creati con l'ausilio della libreria networkX.
Il primo grafo utilizzato è stato quello dal quale sono partito per molti test degli assignment precedenti: il karate club graph offerto da networkX.

Questo grafo ha davvero pochi nodi (34) ma può essere interessante vedere come questi tipi di attacchi si comportano su un grafo di dimensioni ridotte.

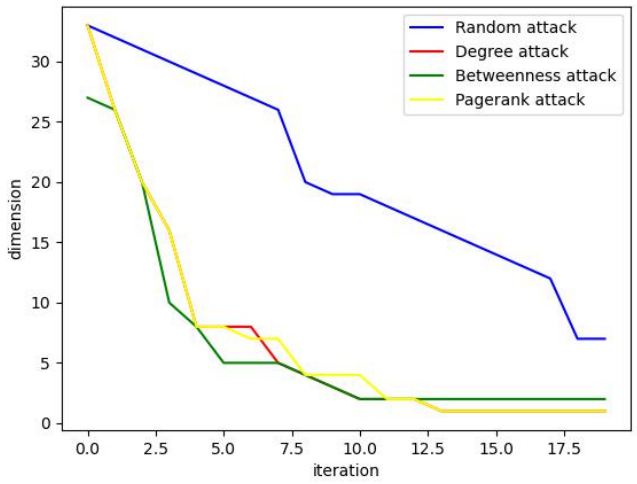

Figure 1: dimension of giant component in a small graph

Come possiamo vedere, il grafico in figura 1 rappresenta la dimensione del giant component ad ogni iterazione del processo di simulazione di failure.

Come possiamo vedere dopo 20 failure il giant component tende quasi sempre a 0.
Questo è chiaramente dovuto anche al fatto che più del 60 percento del grafo è stato rimosso.

Come era immaginabile i failure randomici sono il metodo peggiore per ridurre velocemente e significativamente il giant component anche se in questo caso anche questo tipo di approccio sembra essere piuttosto efficace.

Tutti gli altri tipi di failure mirati per un grafo così piccolo hanno bene o male lo stesso comportamento.

Il secondo grafo che ho deciso di prendere in analisi è un grafo randomico di tipo barabasi albert, anche questo generato con 2000 nodi con l'ausilio della libreria networkX.

Questo tipo di grafo è scale-free quindi quello che ci aspettiamo è di vedere una resistenza abbastanza marcata ai random failure e invece una risposta piuttosto debole ad attacchi mirati.

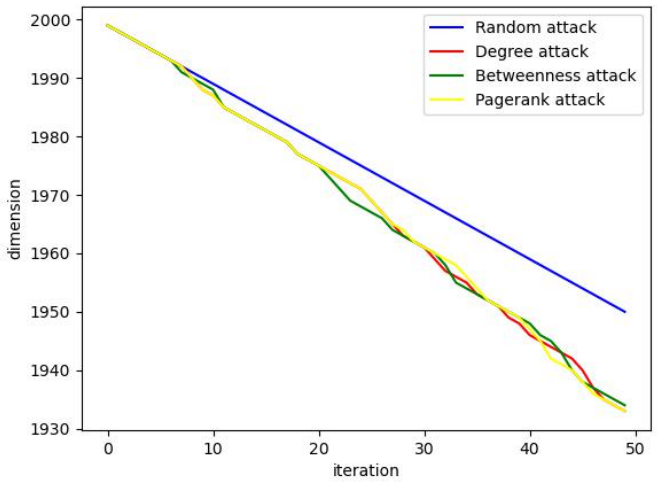

Figure 2: dimension of giant component in a barabasi albert graph

Come immaginabile il grafo reagisce meglio a random failure rispetto che agli attacchi mirati dopo 50 round nei quali viene tolto un nodo alla volta.
La presenza di soli 2000 nodi forse non rende estremamente significativo l'esperimento ma comunque è già visibile una sostanziale differenza tra i due tipi di failure.

3 Test on a real graph

Dopo aver provato lo script su grafi piccoli passiamo allo studio su un grafo reale, quello usato nei primi due assignment.

Anche per questo esperimento, come in quello precedente,ho utilizzato una versione ridotta del grafo del social network GPlus.
Questa decisione è stata presa ancora una volta a causa degli elevati tempi computazionali dello script, che mi rendevano impossibile testare il mio grafo.

Il grafo di GPlus, come è stato già detto nel primo assignment dopo l'attenta analisi delle sue proprietà, è un grafo che presenta alcuni nodi estremamente connessi e dal ruolo centrale (hub) e altri con pochi collegamenti.

E' chiaro come questo sia un punto importante nell'analisi della robustezza di un grafo, e che la rimozione di nodi come ad esempio il 2300 porta a una riduzione drastica del giant component.

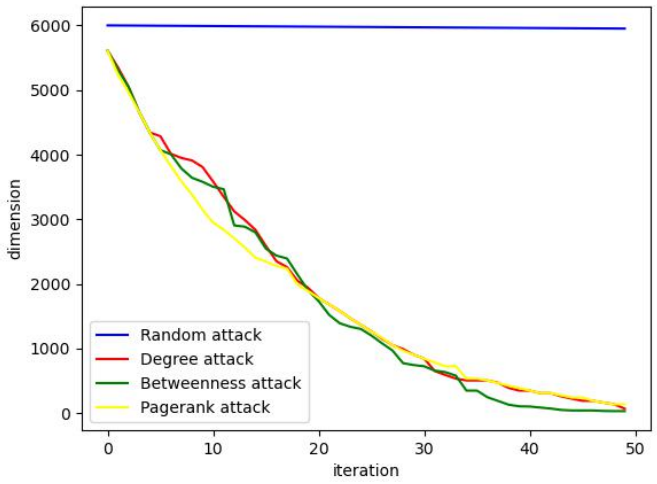

Figure 3: dimension of giant component of Gplus graph

Come possiamo notare dalla figura 3, il grafo reagisce davvero male ad attacchi mirati mentre quelli randomici non hanno quasi alcun effetto su di lui. La debolezza di questo grafo sta nel fatto che presenta davvero pochi nodi importanti che una volta rimossi riducono drasticamente la dimensione del componente gigante fino a farlo addirittura scomparire.

Il grafo quindi non risulta assolutamento robusto visto che, con l'eliminazione di solo 50 nodi su 6000 disponibili, il componente gigante è pressochè scomparso.

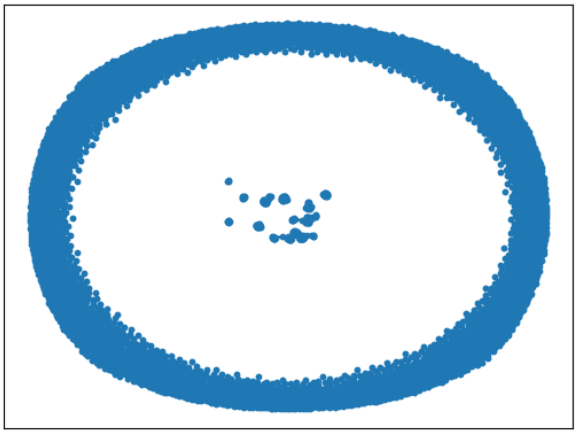

Figure 4: the graph after a degree failure of 50 nodes

Nella figura 4 è rappresentato il grafo dopo che una simulazione di attacco è stata completata.
In questo caso sono stati rimossi 50 nodi in base al loro degree, e alla fine possiamo vedere come il giant component è rimasto delle dimensioni di 72 nodi con 243 edges totali nel grafo.
L'anello esterno di nodi non presenta alcun tipo di collegamento, mentre i nodi centrali solo quello che rimangono di un giant component ormai estremamente ridotto.

4 Building robustness

E' possibile aumentare la robustezza di un grafo aggiungendo archi tra nodi esistenti.
La metrica che utilizzerò per calcolare la robustezza del grafo è il critical threshold f_c.

Questo rappresenta la frazione di nodi che devono essere rimossi da una rete affinché la componente gigante si frammenti in molte componenti più piccole.

$$f_c = 1 - \frac{1}{\frac{\langle k^2 \rangle}{\langle k \rangle} - 1}$$

Il critical threshold è stato calcolato attraverso la formula soprastante.

Per l'analisi ho deciso di aggiungere nodi attraverso la funzione add edges di networkX con tre metodi diversi:

- aggiunta di archi tra nodi con degree più alto
- aggiunta di archi tra nodi con degree più basso
- aggiunta di archi tra nodi con betweenness più alta

Ho poi eseguito lo script con un grafo di prova: il karate club graph.
Come possiamo vedere dalla tabella il metodo migliore consiste nel aggiungere

Threshold	before	after-degree	after-peripheral	after-betweenness
f_c	0.5414	0.8651	0.5414	0.8650

Table 1: critical threshold in a small graph

archi tra nodi "importanti" nel grafo, che sia per degree o betweenness.
Potenziare nodi periferici non sembra avere effetti significativi.

Nella tabella 1 sono stati aggiunti 50 archi in tutto, e nell'immagine sottostante possiamo vedere la struttura del grafo prima e dopo l'aggiunta.

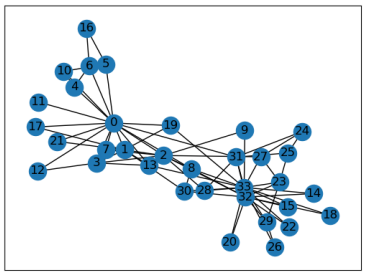

Figure 5: Before

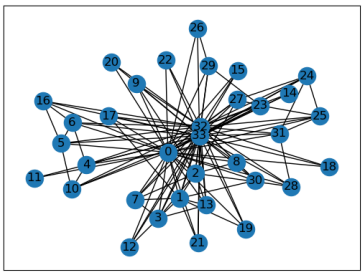

Figure 6: After

Concludendo ho applicato lo stesso algoritmo sul grafo reale di GPlus per vedere come l'aggiunta di nodi impatta su un grafo di dimensioni notevolmente maggiori.
In questo caso gli archi aggiunti sono stati 1000 archi.

Threshold	before	after-degree
f_c	0.994654	0.995776

Table 2: critical threshold in a small graph

In questo caso il critical threshold era già molto alto e attraverso l'aggiunta di ulteriori archi è ulteriormente migliorato anche se di poco.