Componenti connesse

Niccolò Parlanti

Ottobre 2021

1 Introduzione

Lo scopo dell'esercizio è di trovare e valutare il numero di componenti connesse utilizzando l'algoritmo di kruskal. Per fare ciò si parte generando un grafo casuale, con un numero di nodi a scelta e una probabilità di presenza di archi tra vertici che varia tra 0 e 1. Fatto ciò si assegna casualmente un peso all'arco, e si ricercano infine le componenti connesse tramite l'algoritmo di kruskal (utilizzando la struttura dati union-find). Si implementano infine alcuni test, variando il numero di nodi del grafo (in particolare ho scelto di eseguire i test con 5, 50 e 500 nodi).

2 Teoria

2.1 Grafo

Si definisce grafo una coppia ordinata G = (V , E) di insiemi, con V insieme dei nodi ed E insieme degli archi, tali che gli elementi di E siano coppie di elementi di V. Due vertici u , v connessi da un arco prendono il nome di estremi dell'arco; l'arco "e" viene anche identificato con la coppia formata dai suoi estremi (u , v). Se E è una relazione simmetrica allora si dice che il grafo è non orientato (o indiretto), altrimenti si dice che è orientato (o diretto).

2.2 Connettività di un grafo

Dato un grafo G = (V, E) due vertici v, u appartenenti a V si dicono "connessi" se esiste un cammino con estremi v e u. Se tale cammino non esiste, v e u sono detti "sconnessi". La relazione di connessione tra vertici è una relazione di equivalenza.

Per i = 1, ..., k (k classi di equivalenza) sono definibili i sottografi Gi = (Vi, Ei) come i sottografi massimali che contengono tutti gli elementi connessi tra loro, che prendono il nome di *componenti connesse di G.*
Se la cardinalità di G è 1, allora G si dice "connesso".

Per la rappresentazione delle componenti connesse si possono utilizzare le strutture dati per insiemi disgiunti.

2.3 Union-Find

Union-find è una struttura dati che rappresenta una partizione di un insieme finito. Questo si compone di tre operazioni fondamentali:

MAKE-SET(x): Costruisce una classe di equivalenza contenente un singolo elemento, in particolare consiste nel selezionare un elemento da ciascuna classe, chiamato rappresentante, per identificare l'intera classe.

UNION(x , y): unisce i due insiemi contenenti x ed y in un'unico insieme.

FIND(x): Ritorna il rappresentante dell'insieme che contiene x.

2.4 Algoritmo di Kruskal

L'algoritmo di Kruskal è un algoritmo ottimo utilizzato per calcolare gli MST di un grafo non orientato e con gli archi con costi non negativi.

MST-Minimum spanning tree: L'albero ricoprente minimo è un albero ricoprente nel quale sommando i pesi degli archi si ottiene un valore minimo.

L'algoritmo di Kruskal si basa sull'ordinare gli archi in ordine crescente di costo e successivamente analizzarli singolarmente per poi inserire l'arco nella soluzione se non forma cicli con gli archi precedentemente selezionati. Per farlo si serve dell'Union-Find.

```
MST-KRUSKAL(G,w)
    A <- []
    for ogni vertice v in G.V
        MAKE-SET(v)
    for ogni arco (u,v) in G.E, preso in ordine di peso non decrescente
        if FIND-SET(u)!=FIND-SET(v)
            A <- A union {(u,v)}
            UNION(u,v)
    return A
```

3 Aspettative sui test

L'aspettativa è che il numero di componenti connesse nel grafo tenderà, partendo da un picco in corrispondenza del valore riferito al numero di nodi presenti del grafo (quindi dove nessun nodo è collegato ad un altro), a diminuire (con andamento non costante) all'aumentare della probabilità di presenza di archi tra nodi, fino ad arrivare a stabilizzarsi al valore 1, dove tutti i nodi sono connessi tra loro.

4 Test

4.1 Grafo con 5 nodi

Di seguito è riportato il grafico ottenuto utilizzando l'algoritmo di kruskal su un grafo con 5 nodi.

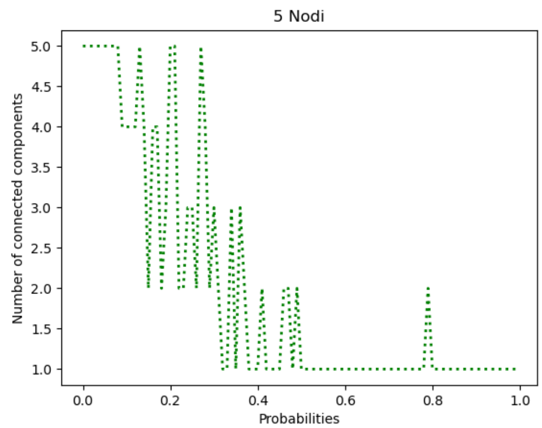

Figure 1: Andamento del numero di componenti connesse al variare della probabilità

4.2 Grafo con 50 nodi

Di seguito è riportato il grafico ottenuto utilizzando l'algoritmo di kruskal su un grafo con 50 nodi.

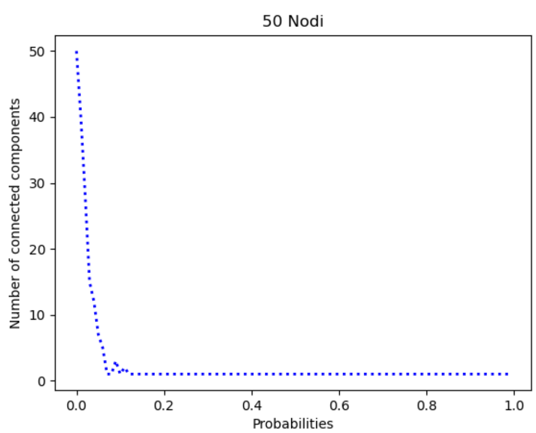

Figure 2: Andamento del numero di componenti connesse al variare della probabilità

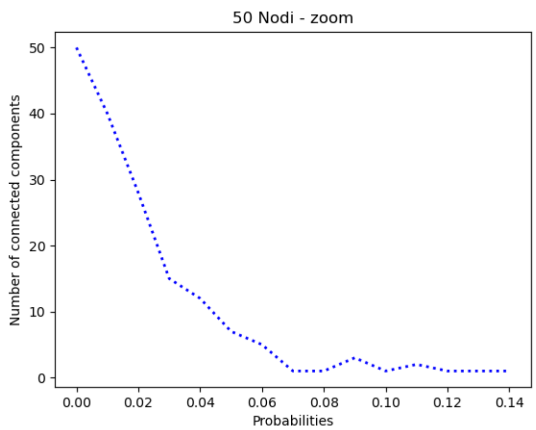

Figure 3: Andamento del numero di componenti connesse al variare della probabilità

4.3 Grafo con 500 nodi

Di seguito è riportato il grafico ottenuto utilizzando l'algoritmo di kruskal su un grafo con 500 nodi.

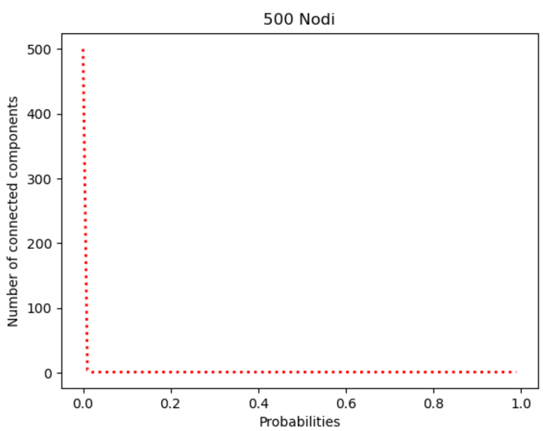

Figure 4: Andamento del numero di componenti connesse al variare della probabilità

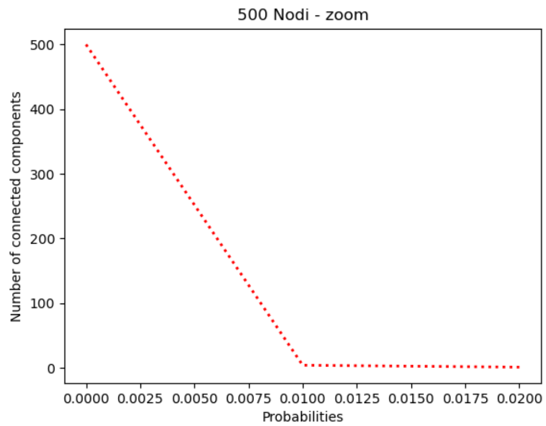

Figure 5: Andamento del numero di componenti connesse al variare della probabilità

5 Conclusioni

Dai grafici possiamo vedere come l'andamento del numero di componenti connesse al variare della probabilità tende, dopo un iniziale picco (quando la probabilità della presenza di un arco tra nodi è vicino allo zero), a calare velocemente e stabilizzarsi al valore 1. Questo perchè quando è presente un numero di archi abbastanza elevato, tutti i nodi sono connessi tra loro attraverso un cammino, ed il grafo presenta quindi una sola componente connessa. Aumentando il numero di nodi, inoltre, si può facilmente notare come la funzioni sia più lineare, ed arrivi a stabilizzarsi al valore 1 più velocemente (figura 5 e figura 3). La spiegazione si può facilemte trovare nel fatto che maggiore è il numero di nodi, maggiore è il numero di archi generati. Infatti, anche se la probabilità della presenza di un singolo arco è inizialmente piuttosto bassa, lavorando su un numero elevato di nodi, è molto più probabile che si generi un arco tra due nodi, facendo si che all'aumentare della probabilità, quasi subito tutti i nodi siano collegati tra loro tramite un cammino.