 RIASSUNTO

Il tirocinio è incentrato totalmente nel calcolo delle misure di centralità dei nodi all’interno della rete Steemit e sull’indagine statistica, applicando metodi non parametrici ai risultati ottenuti.

La centralità di un nodo quantifica l’”importanza” (centralità) di un nodo o un gruppo di nodi in una rete ; l’ipotesi è che la centralità strutturale sia un elemento in grado di motivare l’importanza di un attore in un processo di comunicazione.

Il lavoro è stato svolto a distanza utilizzando da remoto la macchina messa a disposizione dai relatori; strumento fondamentale per il calcolo delle diverse metriche interessate, in particolar modo per la pesantezza in byte dei diversi grafi composti dalle numerose interazioni tra gli utenti del Social media (miliardi di interazioni relative a milioni di utenti).

Per mezzo della ricerca svolta in precedenza da altri tirocinanti è stato possibile calcolare sette centralità tramite algoritmi ottimi o approssimati.

Gli importanti studi antecedenti a questo comprendono: il grafo Steemit (rappresentata da una lista di archi salvati in un file) contenente tutte le interazioni tra gli utenti (ad es. commenti, “Mi piace”, post, transazioni monetarie, ecc.) e la derivazione da esso di altri tre “sottografi”, logicamente distinti in base al tipo di utente o al tipo di interazione utilizzando un database.

Passati un mese tra la scrittura del codice e l’esecuzione dei vari algoritmi, abbiamo tutte le misure di tutti i nodi per i quattro grafi, con un totale di 28 file. I tre mesi successivi sono stati impiegati per analizzare le diverse misure locali e globali attraverso strumenti di statistica descrittiva utile a far vedere le distribuzioni delle centralità, e dei test d’inferenza per riuscire a trarre delle conclusioni sui legami creati all’interno della piattaforma basata su blockchain, Steemit.

Indice

1. Introduzione . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 3
2. Preambolo del tirocinio . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . .

2.1 Blockchain . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . .

2.2 Blockchain Online Social Media . . . . . . . . . . . . . . . . .

2.3 Principi di funzionamento della Blockchain . . . . . . .

2.4 Steemit . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . .

2.5 Lavori antecedenti . . . . . . . . . . . . . . . . . . . . . . . . . . .

1. Tecnologie adoperate . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . .

3.1 Tecnologie e protocolli di comunicazione . . . . . . . .

3.2 Linguaggi di programmazione impiegati . . . . . . . . .

3.2.1 Python3 . . . . . . . . . . . . . . . . . . . . . . . . .

3.2.2 R (Software) . . . . . . . . . . . . . . . . . . . . .

3.3 Librerie implementate . . . . . . . . . . . . . . . . . . . . . . .

3.3.4 Networkit . . . . . . . . . . . . . . . . . . . . . . . . . . .

3.3.4.1 Centralità Betweeness . . . . . . . . . . .

3.3.4.2 Centralità Closeness . . . . . . . . . . . . .

3.3.4.3 Centralità Eigenvector . . . . . . . . . . .

3.3.4.4 Centralità In degree . . . . . . . . . . . . .

3.3.4.5 Centralità Out degree . . . . . . . . . . . .

3.3.4.6 Centralità Katz . . . . . . . . . . . . . . . . . .

3.3.4.7 Centralità Pagerank . . . . . . . . . . . . . .

CAPITOLO 1

Introduzione

L’analisi delle reti sociali (Social Network Analysis, SNA), a volte detta anche teoria della rete sociale, è una moderna metodologia di analisi delle relazioni sociali sviluppatasi a partire dai contributi di Jacob Levi Moreno, il fondatore della sociometria, scienza che analizza le relazioni interpersonali. Per capire quali strumenti e teorie utilizza questa branca della scienza, bisogna introdurre diverse teorie che implementino i metodi e gli strumenti utilizzati sulla ricerca nella SNA, utile pure per comprendere facilmente i concetti che stanno alla base di questa moderna disciplina. Per spiegare il concetto ci serviamo dell’architettura TCP/IP (Transmission Control Protocol/Internet Protocol) ingegneria utilizzata per il funzionamento di una rete, esempio utile a instaurare nel nostro caso una gerarchia tra le diverse teorie utilizzate.

Brevemente la Suite Protocol Internet (TCP/IP) è un raggruppamento di protocolli di rete implementati appositamente e organizzati a livelli, in grado di far interagire attraverso un canale di comunicazione due macchine remote; la particolarità interessante sta nel fatto che i protocolli situati nei livelli inferiori offrono funzionalità ai protocolli presenti nei livelli superiori. Immaginando di traslare la Suite nel nostro contesto trasformandola in una “Suite logica” delle diverse teorie possiamo organizzarla come: il livello più astratto rappresentato dalla “Graph theory”, a seguire nel secondo livello la “Network theory” ed infine alcune fondamenti di queste ultime vengono utilizzate dalla “Social Network Analysis”.

(schema grafico riassuntivo, vedi immagine 1.1.)

Immagine che contiene tavolo

Descrizione generata automaticamente

Immagine 1.1: struttura logica dei diversi livelli

Descrivendo le tre teorie sommariamente, troviamo che la Teoria dei grafi (Graph Theory) è una branca della matematica la quale si occupa, appunto, dello studio dei grafi; oggetti discreti che permettono di schematizzare una grande varietà di situazioni e processi, e spesso di consentirne 3delle analisi in termini quantitativi e algoritmici.

In termini informali, per grafo si intende una struttura costituita da:

* Oggetti semplici, detti **vertici** o **nodi**;
* **Collegamenti** tra i vertici; dove tali collegamenti possono essere:
  + **Non orientati** (cioè dotati di una direzione, ma non dotati di un verso): in questo caso detti spigoli, e il grafo è detto “non orientato”;
  + **Orientati** (cioè dotati di una direzione e di un verso): in questo caso sono detti archi o cammini, e il grafo è detto “orientato”;
  + Eventuali **dati associati a nodi e/o collegamenti**; un grafo pesato è un esempio di grafo in cui a ogni collegamento è associato un valore numerico, detto “peso”.

Al secondo livello della gerarchia notiamo già l’uso dei principi instaurati dalla teoria dei grafi; infatti, la Teoria delle reti è lo studio dei grafi rappresentati come relazioni simmetriche ed asimmetriche tra oggetti discreti. In informatica, la Teoria delle reti è una parte della Teoria dei grafi. Essa studia caratteristiche e comportamenti delle reti complesse, che anche se riguardano gli argomenti più diversi, hanno strutture e dinamiche determinati non dal genere di rete, ma dall’essere una rete. Proprietà comuni si ritrovano in reti tecnologiche come Internet o il Web, biologiche come i sistemi epidemiologici, reti sociali come gli abitanti di un gruppo o sistema.

Al terzo e ultimo livello arriviamo al fulcro del tirocinio, riprendendo la descrizione sommaria utilizzata all’inizio di questo paragrafo; la teoria della rete sociale è vista e studiata come rete di relazioni, più o meno estese e strutturate. Il presupposto fondante è che ogni individuo (o attore) si relaziona con gli altri e questa sua interazione plasma e modifica il comportamento di entrambi. Lo scopo principale dell’analisi della network è appunto quello di individuare e analizzare tali legami (ties) tra gli individui (nodes). Diverse classi di misure sono disponibili in letteratura, rivolte fra l’altro all’esame delle proprietà di rete nel loro complesso (coesione, centralità, …), alla ricerca di sottoreti specifiche (gruppi, egonet) ed alla ricerca di somiglianze fra reti. La SNA fa anche ampio uso dell’algebra lineare e della statistica, come già accennato all’inizio del sommario.

L’applicazione della metodologia, in questo contesto, viene attuata a una social network basata sulla tecnologia delle Blockchain. I più comuni sistemi sociali sono sistemi centralizzati per loro natura (Facebook, Instagram, ecc.) poiché l’architettura è una tradizionale Client-Server, nella quale una banca dati (server) accumula e controlla tutti i dati di tutti gli abitanti del sistema (client); nelle blockchain, la società è organizzata democraticamente, dove ogni utente ha la possibilità di vedere l’enorme registro che forma la rete, consultando in tempo reale gli sviluppi della piattaforma tramite i dati organizzati a blocchi (cioè ogni azione di un utente viene salvato in uno dei blocchi del sistema), correttamente crittografati per i dati sensibili e immune da attacchi di utenti malevoli, di cui forniremo una breve descrizione nel primo capitolo. Un sistema molto conosciuto costruito sulla base delle Blockchain, oggetto del nostro studio, il quale vanta un milione e più utenti registrati dal marzo del 2016 è Steemit.

Per poter prendere in esame gli utenti del sistema con le loro interazioni, concetto fondamentale, è essere in possesso dei dati della popolazione del Social media, quindi essere in possesso di una lista contenente tutte le interazioni tra i diversi utenti. Utilizzando le tecnologie ed i concetti sopra citati, con il supporto di librerie ad-hoc (steem-python) fornite per l’interazione con la blockchain di Steemit, dei tirocinanti in passato hanno creato un dataset contenente tutte le interazioni salvate nei blocchi dall’arco temporale dal 2017-03-08T17:34:2 (risultato restituito dal timestamp del primo blocco) fino al 2019-01-01T00:40:2 (timestamp dell’ultimo blocco analizzato) e a partire da esso è stato possibile, tramite un programma scritto appositamente, creare il grafo globale. Il grafo è composto da 1.074.722 vertici i quali rappresentano gli utenti registrati all’interno della piattaforma, essi vengono collegati per mezzo di archi, che identificano ogni possibile azione che un utente può effettuare all’interno del sistema, quindi diremo che un nodoA ha un collegamento con un nodoB, se l’utente A ha effettuato un’azione nei confronti dell’utente B. Successivamente per l’attuazione di analisi più approfondite, è stato necessaria la costruzione di altri tre grafi, sottoinsiemi di quello globale; i parametri scelti per la suddivisione verranno discussi nella sottosezione 2.5.

Il lavoro successivo, oggetto di questa relazione, è stato svolto per mezzo delle direttive imposte dei relatori, che hanno fornito la base di partenza e aiutato con qualche consiglio durante le fasi d’implementazione e di svolgimento delle prove ponendo in maniera chiara l’obiettivo, cioè calcolare e analizzare le misure di centralità, ognuna atta ad esprimere una particolare caratteristica di un nodo in relazione a tutti gli altri nodi della rete in cui si trova. Una volta correttamente calcolate, possono essere prese in esame nella fase di analisi con la visualizzazione della distribuzione rispetto le varie centralità, cercando di chiarire i parametri della popolazione analizzata.

Per rendere maggiormente chiara l’analisi sulle metriche è stato necessario isolare un determinato gruppo di nodi tra i milioni presenti nella popolazione, accuratamente selezionati tramite una classifica sui nodi più alti in termini di grado in entrata (cardinalità dei collegamenti in entrata) e grado in uscita (cardinalità dei collegamenti in uscita). Numero e modalità di scelta dei nodi è stata pianificata dal tirocinante, cercando di isolare un campione che eliminasse i nodi meno influenti della rete e che includesse quelli con valori più alti in più centralità possibili.

Una volta raccolti tutti i valori dei nodi scelti si può fare uso dei metodi più comuni della statistica descrittiva cosi da osservare le caratteristiche più importanti del campione scelto. In aggiunta, con il consiglio dei relatori è stato proposto l’uso di strumenti di statistica inferenziale tra due o più parametri indipendenti, in modo da riuscire a capire sé presente una possibile relazione tra le misure e per quale fattore intrinseco esse sono legate tra loro; per mezzo di numerosi test non parametrici è stato possibile notare delle relazioni monotone tra le diverse variabili.

La tesi percorre il cammino svolto durante il tirocinio, dividendo in capitoli i diversi argomenti studiati per comprendere le diverse materie che entrano in gioco nella ricerca. In linea generale, la struttura è organizzata in due parti: la prima utile a spiegare l’ambito del tirocinio e i concetti appresi per svolgere correttamente il lavoro; la seconda parte descrive l’attuazione nella pratica dei concetti studiati in teoria. In ordine di stesura dei capitoli, tralasciando questo capitolo dell’introduzione, descriviamo il preambolo del tirocinio come tutto ciò che riguarda le fondamenta della ricerca: da una descrizione sommaria sul sistema su cui attuiamo le analisi compreso di un excursus delle ricerche passate effettuate da colleghi. Una volta presa coscienza dell’ambito su cui si svolge la ricerca, possiamo descrivere teoricamente quali tecnologie e quali concetti della SNA vengono utilizzati durante la prima fase del lavoro, comprese di una descrizione delle librerie e delle centralità. Terminati i concetti esposti nella prima fase, possiamo descrivere le nozioni statistiche che entrano in gioco nella seconda e terza fase; descrivendo i metodi utilizzati durante la fase di statistica descrittiva (seconda fase) e i test statistici nella statistica inferenziale (terza fase), con un giusto livello di astrazione per una facile comprensione senza dilungarci troppo nella complessità della materia. Finito il quinto capitolo si è in grado di comprendere all’atto pratico il lavoro e le motivazioni per cui si sono scelti certi strumenti; infatti, il sesto capitolo può essere considerato il cuore del tirocinio, descrivendo minuziosamente tutte le fasi del lavoro comprese di tempistiche e con l’attuazione pratica di tutti i concetti descritti nei capitoli precedenti. L’ultimo capitolo offre una conclusione, con gli obiettivi raggiunti e i possibili lavori futuri utilizzando i dati di questa ricerca.

Gli appendici conclusivi, illustrano tutti i metodi grafici e i metodi analitici calcolati sui dati.

Capitolo 2

Preambolo del tirocinio

2.1 Blockchain

La blockchain (letteralmente “catena di blocchi”) è una struttura dati condivisa e “immutabile”. Essa è pensata e implementata come un vero e proprio registro digitale in grado di contenere informazioni condivise con tutta la rete. Le interazioni all’interno del sistema vengono raggruppate in “blocchi”, che vengono concatenati in ordine cronologico, e la cui integrità è garantita dall’uso della crittografia. La struttura dati essendo immutabile vuol dire che, di norma, il suo contenuto una volta scritto non è più né modificabile né eliminabile, a meno di non invalidare l’intera struttura; in modo ovvio essendo un registro ha la caratteristica intrinseca di crescere nel tempo. Ogni blocco è suddiviso in due componenti: una prima parte utile ad identificare il blocco viene chiamato header e contiene il timestamp del blocco (quindi la data di creazione) e un insieme di informazioni utili ad identificarlo; la seconda parte contiene l’informazione vera e propria, quindi un record di transazioni registrate in quel momento nel sistema. Soffermandoci sul fattore di immutabilità e di condivisione, le blockchain sono particolarmente utili in situazioni in cui due soggetti vorrebbero effettuare una transizione, ma l’uno è diffidente nei confronti dell’altro; nella società civile, sono state pensate e sviluppate istituzioni come gli studi legali e le banche, per assicurare lo scambio di denaro o di proprietà. Immaginandoci il registro digitale come un’istituzione, essa consente lo sviluppo di un rapporto sociale dove, grazie alla partecipazione di tutti, si è in grado di verificare le transazioni, in totale trasparenza, grazie al fatto che le transazioni vengono registrate su archivi inalterabili. Infatti, l’idea generale delle blockchain consisteva nel creare una forma di contante elettronico che potesse essere scambiata in modalità peer-to-peer, senza passare attraverso una qualsiasi istituzione certificata. Questo è possibile grazie all’uso di un sistema decentralizzato e distribuito, senza la presenza di un’entità centrale che decide e garantisce il corretto funzionamento della piattaforma, e soprattutto grazie alla distribuzione delle copie del registro visibile a tutti gli utenti del sistema, cosicché sia facile risalire alla veridicità delle transazioni eseguite.

I vantaggi dell’utilizzo di questa architettura peer-to-peer, porta all’inattaccabilità da parte di utenti malevoli all’enorme registro che forma la rete. Un utente malintenzionato per poter attaccare un sistema del genere, dovrebbe poter entrare in tutti gli utenti del sistema che sono in possesso di una copia del registro ed effettuare delle modifiche che siano coerenti in tutto il sistema, totalmente differente da un’architettura client-server dove tutti i dati dei client sono contenuti in un'unica macchina che offre con semplicità un bersaglio da compromettere. Le entità che creano la rete vengono chiamati nodi, tali nodi partecipano e contribuiscono alla crescita della catena di blocchi in totale democrazia, attraverso l’uso di un algoritmo di consenso. Concetto fondamentale in grado di garantire che le regole del protocollo vengano seguite e che tutte le transazioni avvengano correttamente ed è uno dei motivi principali che rende difficile modificare una blockchain. Nel pratico questo algoritmo viene utilizzato per decidere qual è il prossimo blocco da aggiungere alla catena di blocchi. Esistono diversi algoritmi di consenso, alcuni esempi sono Proof of Work (PoW), Proof of Stake (PoS) e Delegated Proof of Stake (DPos).

Una volta arrivato a un determinato passo dell’algoritmo di consenso e si aggiunge il blocco alla struttura dati, esso non può essere più alterato in maniera retroattiva, senza alterare tutti i blocchi successivi. Questo è possibile tramite l’adozione di una funzione hash, una funzione che mappa dati di dimensione arbitraria in dati di dimensione fissata. Quando un nodo esegue una nuova transazione, deve firmarla attraverso un protocollo di firma digitale e segnalare a tutti gli altri nodi della piattaforma che una nuova transazione deve essere aggiunta a un blocco. Ogni blocco, quindi, è processato in modo che la funzione generi un hash value, una stringa alfanumerica di lunghezza fissata. La funzione hash solitamente utilizzata all’interno delle applicazioni di alta sicurezza è la SHA256, che genera un’immagine di 256 bit. Quando viene creato un nuovo blocco i dati appartenenti a quel blocco vengono elaborati attraverso SHA256, incorporando l’hash dei dati del blocco precedente. Cosi facendo, ogni blocco è collegato al blocco precedente e, se dei dati contenuti nella blockchain venissero manipolati o cancellati, i blocchi successivi al blocco corrotto rifiuterebbero i tentativi di modifica.

Esistono vari tipi di blockchain, dove la particolarità che le differenziano è dal tipo di permessi assegnati agli utenti per accedere o meno al registro digitale, per questo le blockchain vengono divise in Blockchain pubbliche o permissioned.

Tornando al protocollo che rende effettive le modifiche e l’immutabilità della struttura, queste sono relativamente vitali per la sicurezza e integrità della network di una criptovaluta. Gli algoritmi di consenso sono vitali per riuscire a trovare un consenso generale in situazioni in cui è possibile la presenza di errori, siano essi in buona o cattiva fede. Il problema consiste nel trovare un accordo tra entità diverse, nel caso in cui siano presenti informazioni discordanti, in modo da mantenere intatta l’integrità, l’operatività e la consistenza della rete. Tra i tanti algoritmi, descriviamo quello presente nel nostro sistema, il DPoS. L’algoritmo è formato da due fasi: l’elezione di un gruppo di produttori (i miner) e lo scheduling della produzione. All’inizio un numero limitato di nodi viene scelto per aggiungere un nuovo blocco alla blockchain in base alla quantità di stake posseduto e al comportamento tenuto nel corso del tempo. Successivamente, nella fase di elezione, gli utenti comuni partecipano a una votazione per eleggere gli utenti che acquisiranno il diritto di validare le transazioni.

2.2 Blockchain Online Social Media

Oltre all’impiego della tecnologia delle Blockchain per il funzionamento dei social network distribuiti “classici”, si sono sviluppate novità interessanti in questo campo aggiungendo la possibilità di far circolare della moneta elettronica all’interno del sistema. Gli utenti quindi possono guadagnare della criptomoneta implementata nella piattaforma attraverso l’utilizzo classico del sistema, in maniera totalmente trasparente alla comunità, grazie a una blockchain che permetta a tutti la verifica dell’avvenuta transazione. Un sistema economico cosi costruito è basato sull’impiego dei principi della token economy e dell’attention economy. L’attention economy è un tipo di economia che fonda il suo sviluppo sull’attenzione degli utenti, in cambio di servizi offerti ai consumatori. Il profitto è generato dalle informazioni che circolano all’interno della piattaforma o dal suo valore, acquisito man mano che viene utilizzata dagli utenti. La token economy invece è una tecnica psicologica secondo cui, sfruttando determinati incentivi come i token, il soggetto è portato a comportarsi come desiderato.

Un'altra importante caratteristica di tali sistemi, data dal loro modo di essere implementati, è la possibilità di visualizzare ed eliminare la circolazione di notizie false, tramite appunto la possibilità di premiare o meno un utente che crea del contenuto. Anche se vi sono degli indiscutibili vantaggi dall’uso delle blockchain, vi sono pure degli svantaggi sull’utilizzo di questi sistemi poiché ancora nella fase iniziale del loro studio e sviluppo. Uno svantaggio importante è la scalabilità del sistema, infatti a causa della loro struttura, le blockchain permettono di effettuare poche decine di operazioni al secondo; i server hanno un simile problema, che viene risolto con la distribuzione di più server. Un secondo svantaggio è la non banalità nella verifica dell’identità degli utenti, dal momento che a ogni utente può essere associato più di un account ed è possibile, anche se non regolamentare, usare bot programmati per eseguire certe attività meccaniche all’interno della piattaforma.

2.3 Steemit

Steemit è un social media basato sulla blockchain Steem. Dal punto di vista social ha alcune caratteristiche di Facebook, altre di Medium (una piattaforma di blogging), altre ancora di Reddit (una community online molto frequentata negli Stati Uniti). Esso vanta l’implementazione di una criptovaluta di nome Steem che mantiene viva la piattaforma, cercando di mantenere alti livelli di informazioni all’interno del sistema. Infatti, gli utenti possono essere ricompensati in STEEM pubblicando notizie o commentando post di terzi; il guadagno della persona dipende dal grado di coinvolgimento ottenuto dal proprio post: più commenti e voti si ottengono, maggiore sarà la ricompensa. Steem è una criptovaluta che utilizza la tecnologia Exchange e quindi può essere scambiata con altre monete virtuali o con valuta reale.

Il ruolo di un utente, eliminando la presenza dei bot può ricoprire un doppio ruolo: Autore e Curatore. I primi sono coloro che producono post e li pubblicano sul proprio profilo. I curatori sono invece coloro che lasciano un commento sotto i post, che li condividono ed esprimono il loro parere. Le reazioni possibili sono quattro: Upvote, Commento, Resteem e Flag. L’Upvote è come il “mi piace”, il commento è ovvio, il Resteem è la condivisione, mentre il Flag è il “pollice verso”. Una delle caratteristiche di Steemit è la possibilità di mettere un “non mi piace” ai post. Tramite tutte le relazioni, la reputazione di una persona può aumentare o diminuire. Più la reputazione è alta e maggiore sarà il proprio peso all’interno della piattaforma. Ogni post, commento e Upvote viene ricompensato: il 75% del guadagno andrà all’autore del post, il 25% sarà diviso tra i curatori.

La differenza tra Steemit e le altre Blockchain Online Social Media sta nell’uso di tre tipi di valute:

* **STEEM** – rappresentano la vera e propria cripto-moneta;
* **STEEM DOLLARS** (**SBD**) – In termini finanziari, rappresentano la liquidità all’interno di Steemit e sono costruiti in modo tale che 1 SBD corrisponderà a 1 USD;
* **STEEM POWER** (**SP**) – Rappresentano la propria reputazione e influiscono in maniera direttamente proporzionale al voto: di conseguenza, più SP si posseggono, più il voto sarà influente. In termini finanziari, rappresentano investimenti a lungo termine sul capitale, infatti convertire gli STEEM in SP è immediato ed è un procedimento chiamato power up; convertire SP in STEEM invece, è un processo chiamato power down che richiedete molto tempo

2.3 Steem e i principi di funzionamento della Blockchain

Steem è una blockchain che supporta la costruzione di una comunità e l’interazione sociale attraverso ricompense in criptomoneta. Secondo quanto dichiarato nel paper, si tratta della prima criptomoneta che tenta, in maniera accurata e chiara, di ricompensare il contributo soggettivo dato alla community da un certo gruppo di individui. Gli incentivi economici permessi da tale token possono contemporaneamente facilitare la crescita di una nuova piattaforma social media e nel frattempo avviare una nuova economia di successo. Gli sviluppatori della blockchain sono alla ricerca di un algoritmo in grado di dare un punteggio ai contributi degli utenti, che la maggioranza dei membri della community consideri equa. Steem è una blockchain basata su Graphene, la stessa tecnologia che alimenta BitShares. Essa produce nuovi token ogni volta che un blocco è stato prodotto. Al contrario delle blockchain, in cui non vi è alcun controllo su chi conduce il lavoro del miner, nella blockchain Steem questo compito viene delegato a un numero preciso di utenti fidati e conoscitori della piattaforma, chiamati witness.

Tali soggetti sono pagati dal sistema per creare in totale sicurezza i blocchi che compongono la blockchain. Essenzialmente, un Witness rappresenta un nodo della rete Steem affidabile, in quanto eletto attraverso i voti della Steemit Community, poiché ritenuto in grado di eseguire diversi compiti come:

* Price feed – Steem si basa sul concetto di libero mercato, di conseguenza il valore degli STEEM viene deciso dai Witness, che hanno anche il compito di produrre i price feed. I price feed rappresentano quanto vale uno STEEM in USD. Ognuno dei Witness pubblica il proprio price feed e il price feed effettivo è calcolato come la mediana dei top 20 Witness che sono stati scelti in un periodo di 3,5 giorni.
* Consenso – Essendo un gruppo di persone elette e fidate, i Witness attivi hanno il compito di esprime il proprio consenso sull’applicazione degli effetti delle hardfork. Un hardfork è un cambio nella logica dell’attività che alimenta la rete Steem. Il rilascio di una versione del codice da parte della Steemit Inc. non vuol dire che gli utenti siano forzati ad aggiornare il proprio codice.
* Produzione di blocchi ogni 3 secondi – un altro importante task che rende le blockchain stabile e aggiornata agli occhi dei mercati esterni. Dato che il sistema conosce in anticipo i Witness attivi in quel round. Steem è in grado di schedulare i Witness per produrre blocchi ogni tre secondi. Al tempo stesso, questi ultimi sincronizzano la loro produzioni di blocchi grazie al protocollo NTP. Ovviamente, nel fare questo lavoro i Witness vengono ricompensati con uno STEEM per ogni blocco prodotto

Gli utenti più prestigiosi, cioè quelli con alta reputazione (maggior possesso di SP) eleggono i Witness attraverso una votazione. In questo modo, i casi di Witness maliziosi sono poco probabili, per via dell’algoritmo di consenso DPoS illustrato precedentemente, che, a lungo andare, fa sì che gli utenti siano a conoscenza di quali Witness siano affidabili e onesti. Inoltre, ciascun Witness è pagato molto bene per svolgere il suo compito e comportarsi in maniera maliziosa farebbe perdere l’incarico successivamente.

2.5 Lavori precedenti

Le ricerche effettuate in passato e che vengono prese in assist da questa ricerca comprendono il download dei dati dalla blockchain e la costruzione dei diversi grafi.

Dopo l’ottenimento delle informazioni della blockchain, si avrà un dataset composto da 18 file ed occupa in memoria circa 430GB. L’intervallo di blocchi preso in considerazione va dal blocco numero 38526 (con data di creazione 2017), fino ad arrivare al blocco numero 29059999 (con data di creazione 2019). I primi blocchi della blockchain sono stati esclusi per motivi di praticità: poiché il protocollo di funzionamento di Steem prevede l’aggiunta di un nuovo blocco ogni tre secondi e considerando che a Marzo 2017 la piattaforma non era sufficientemente popolare, spesso i blocchi non contenevano transazioni o ne avevano un numero esiguo. Inoltre, l’uso delle funzioni che permettono il download dei blocchi della blockchain implica, per ogni blocco, un tempo di attesa dato dalla richiesta di scaricamento e ad ogni modo, in caso di blocco vuoto, alla memorizzazione di una stringa non utile ai fini della ricerca.

In principio, sono stati estratti le informazioni contenute nei vari blocchi della blockchain di Steem.

Attraverso queste informazioni, doveva essere costruito un grafo che rappresentasse l’intera rete Steemit; per rendere possibile la costruzione, i blocchi sono stati memorizzati in una serie di file con estensione json in modo da poterli analizzare per estrapolare le transazioni.

Diventa naturale astrarre le azioni che gli utenti eseguono su loro stessi o su altri utenti in una rete. Le informazioni sugli utenti e le loro azioni sono state estratte dai blocchi e memorizzati in un dataset, a partire da quest’ultimo è stato costruito il grafo di tutte le transazioni e alcuni suoi sottografi, mediante l’utilizzo di un database.

Il grafo di Steem rappresenta le interazioni tra gli utenti della piattaforma. Si tratta di un grafo orientato, in cui gli utenti di Steemit sono rappresentati da i nodi e le azioni che compiono sono rappresentate da archi.

Un arco che ha coda nel nodo1 e testa nel nodo2 assume il significato di azione iniziata dal nodo1 e compiuta sul nodo2. Esempi di queste azioni sono: votare, commentare, trasferire una somma di denaro, seguire, la creazione di un utente e così via.

Uno dei capisaldi della sua costruzione è il significato di interazione, in questo contesto: si potrà dire che due utenti hanno avuto un’interazione quando l’uno eseguirà almeno un’azione che influenzerà l’altro. Quindi all’interno del grafo verranno eliminate le frecce che da un nodo vanno sullo stesso (chiamati self-loop, cioè azioni che non ha un secondo soggetto che partecipa ad essa).

Il grafo delle transazioni rappresenta le interazioni che ci sono state tra gli utenti durante il periodo del 2017-2018. Sfruttando i concetti della teoria dei grafi e delle possibili misure di centralità, si può valutare il grafo a livello macroscopico, studiandone la struttura e la connettività, e a livello microscopico, analizzando il comportamento che tendono ad avere i nodi. Inizialmente sono stati condotti degli studi sul grafo ricavato considerando tutte le 38 transazioni presenti nelle blockchain. A seguito dei risultati ottenuti, è stato ritenuto necessario costruire dei sottografi a partire dal grafo iniziale.

Dall’analisi delle proprietà base del grafo risulta un grafo diretto e non pesato che rappresenta 1.24 milioni di utenti, con 191.2 milioni di interazioni.

* Grafo delle interazioni sociali

Il grafo delle interazioni sociali è stato pensato e costruito considerando solamente le transazioni col quale si interagisce nell’ambito del social network. Le transazioni considerate sono: vote, comment, custom\_json, delete\_comment, comment\_options, claim\_reward\_balance. Il grafo considerato ha proprietà con valori piuttosto simili al grafo globale. Il numero die nodi è 1.23 milioni. Di conseguenza, il numero di utenti che non ha mai effettuato alcuna operazione di tipo sociale è di circa 14 mila, cioè l’1% degli utenti totali del grafo di tutte le interazioni.

* Grafo sociale senza bot

Dalla classifica degli utenti centrali per out-degree e in-degree nel grafo social, emerge la forte presenza di account di proprietà di Steem e di account bot. I corrispondenti nodi giocano sicuramente un ruolo fondamentale nel mantenere la macrostruttura del grafo coesa. Grafo aggiornato nel corso del tempo, in seguito alla scoperta di nuovi utenti bot all’interno del sistema.

* Grafo delle interazioni monetarie

Il sottografo rappresenta solamente interazioni di tipo finanziario. Le transazioni considerate per costruirlo sono elencate di seguito: conver, transfer\_to\_vesting, transfer, limit\_order\_create, limit\_order\_create2, limit\_order\_cancel, set\_withdraw\_vesting\_route, delegate\_vesting\_Shares, withdraw\_vesting, escrow\_approve, escrow\_Release, escrow\_dispute, escrow\_Transfer. Il grafo delle interazioni monetarie risulta piuttosto piccolo rispetto ai grafi precedenti. Questo va ad evidenziare quella che è la natura di Steemit: la parte monetaria è ridotta e gli utenti ricevono valuta grazie al rewarding system. Il grafo ha circa 1 milione di nodi e 4 milioni di archi.

Capitolo 3

Tecnologie adoperate

In questo capitolo viene trattato l'ambiente in cui si svolge il tirocinio. Le sezioni seguenti includono: una descrizione dei protocolli e comandi utilizzati durante tutte le fasi del tirocinio, i linguaggi di programmazione studiati e utilizzati ed infine le librerie e le diverse metriche adoperate che possono definire il posizionamento di un attore nel proprio ambiente rilevante in termini puramente relazionali, ovvero relativamente a tutti gli altri attori sociali con cui allaccia rapporti di scambio (fondamenti della centralità). L'idea di centralità è intuitiva e si basa sul fatto che, a partire da un insieme di attori sociali (organizzazione, individui) e dato un insieme di relazioni fra di loro (finanziarie, informative, ecc.) è quasi sempre possibile dare agli attori un ordinamento gerarchico in base alla loro posizione nella struttura relazionale concreta che è possibile osservare. Proprio per caratterizzare in maniera differente i diversi nodi, in base a più punti di vista vengono adottate più metriche di centralità che mettono in evidenza un diverso aspetto di ogni nodo. Viene fornita una descrizione dettagliata in questo capitolo.

3.1 Tecnologie e protocolli di comunicazione

Per poter essere in grado di svolgere correttamente il tirocinio è stato adottato l'utilizzo di alcuni strumenti atti a comunicare in remoto con la macchina del dipartimento, contenente una directory personale in cui processare tutti i nuovi file creati durante l'esecuzione di programmi. Il requisito fondamentale della macchina dipartimentale era la quantità di memoria RAM che contenesse all'interno, poiché parliamo dell'esecuzione di algoritmi che prendono in input dimensioni di file abbastanza grandi, si è impossibilitati nell'utilizzare la propria macchina locale per mettere in esecuzione i diversi codici. Tale fatto sta a evidenziare quanto importante sia avere un calcolatore abbastanza potente da poter contenere in memoria e manipolare tutti i collegamenti del grafo globale o dei grafi derivati.

Parlando delle modalità di collegamento in remoto, la soluzione più ovvia e semplice è stata l’utilizzo del servizio per la comunicazione tra macchine remote, chiamato SSH (Secure SHell). In informatica e telecomunicazioni il protocollo ssh permette di stabilire una sessione remota cifrata (quindi protetta da malintenzionati durante il flusso di dati) tramite interfaccia a riga di comando con un altro host di una rete informatica. Per poter essere in grado di utilizzare tale funzionalità, in ambiente GNU/Linux, è sufficiente installare il pacchetto openssh-server, tramite il comando:

# apt-get install openssh-server

Una volta installato il pacchetto, bisogna tenere in mente che il protocollo SSH utilizza la porta TCP 22. Quindi, per poter connettersi al server SSH, è necessario assicurarsi che tale porta sia inutilizzata da qualsiasi altro programma. Ovviamente essendo un protocollo client-server, abbiamo bisogno di una struttura client che si interfacci con la struttura server; utilizzando un sistema operativo Unix/Linux il client per la shell remota è incluso nel pacchetto OpenSSH. La connessione al server, nei sistemi Unix-like, avviene tramite il comando con sintassi generale:

ssh [opzioni] nomeutente@host [comando]

Terminata la spiegazione della modalità di connessione all’host remoto, insorge un ulteriore problema; potremmo semplicemente mettere in esecuzione i vari algoritmi scritti in precedenza utilizzando solamente la shell remota ma questo porta a vari inconvenienti. Dato che siamo connessi tramite il protocollo SSH che necessita di una connessione ADSL e che questa rimanga attiva, si potrebbe incorrere alla perdita di dati ed all’interruzione di processi messi in esecuzione, per via dell’interruzione della linea di comunicazione instaurata tra il client e il server SSH. A risolvere questo sgradevole problema, incorre in nostro aiuto un comando presente nei sistemi Unix e Unix-like, e più in generale dei sistemi POSIX, che esegue un altro comando facendogli ignorare il segnale SIGHUP, in modo da permettergli di proseguire l’esecuzione anche in caso di una disconnessione dal terminale, o della chiusura dell’emulatore di terminale.

SIGHUP è un segnale inviato a un processo quando il suo terminale di controllo è chiuso, normalmente quando si esegue un comando su SSH se la connessione si interrompe, o si disconnette, la sessione SSH viene interrotta e tutti i comandi eseguiti dal terminale si interrompono. Tale comando serve proprio per evitare l’interruzione dei comandi e quindi il programma eseguito tramite tale strumento ignorerà tutti i segnali di hangup e i comandi continueranno a funzionare. La sintassi per il comando nohup è la seguente:

nohup *comando* [*arg1* [*arg2* …] ]

Nohup eseguirà il comando “comando” in foreground e reindirizza l’output del comando al file nohup.out; file che verrà creato nella directory di lavoro corrente.

Poiché si ha la necessità di eseguire più programmi quindi più algoritmi ai diversi grafi derivati, il comando nohup è stato eseguito in background, aggiungendo il simbolo ‘&’ al termine del comando.

Ultima utilità per trasferire file in modo sicuro da e per sistemi Linux è il comando SCP, usato per avere in locale i diversi file sorgente e manipolarli in modo più semplice. SCP (secure copy) è un comando che consente di copiare file e directory in modo sicuro tra due posizioni. Con scp è possibile copiare un file o una directory dal sistema locale al sistema remoto e viceversa. Quando si trasferiscono i dati con scp, sia i file che le password vengono crittografate.

La sintassi del comando è semplice:

scp [OPTION] [user@]SRC\_HOST:]file1 [user@]DEST\_HOST:]file2

OPTION- opzioni come cifratura, configurazione ssh, porta ssh, limite, copia ricorsiva, ecc

[User@]SRC\_HOST:]path\_file – nome utente(DOT)nome host:] File di origine

[User@]DEST\_HOST;]path\_destinazione – nome utente(DOT)nome host:] directory dove inserire il file di origine nella macchina remota.

Ovviamente il comando, in base alla semantica inserita nel comando specifica la semantica dell’operazione. SCP fornisce in più una serie di opzioni che controllano ogni aspetto del suo comportamento, ad esempio specificare la porta ssh dell’host remoto, oppure preservare la modifica dei file e i tempi di accesso, ecc.

Il protocollo e i due comandi sono stati fondamentali nello svolgimento dell’analisi; i primi due artifici sono stati consigliati dai relatori, anche se il protocollo ssh era già stato utilizzato durante il corso della triennale; invece, i costrutti nohup e il comando scp sono stati studiati durante la ricerca per comprenderne le funzionalità da applicare nel contesto.

3.2 Linguaggi di programmazione impiegati

Nel corso della ricerca si è presa familiarità con due diversi linguaggi di programmazione mai utilizzati nel corso della carriera. Il primo linguaggio di programmazione è stato scelto dai relatori, questo per un fattore di semplicità poiché i costrutti utilizzati erano già conosciuti e implementati da terzi. Il secondo è stato scelto dal tirocinante, dopo aver proseguito una ricerca sui diversi ambienti utili per l’analisi statistica di popolazioni o campioni.

3.2.1 Python 3

Nel corso del tirocinio, soprattutto per l’utilizzo di librerie in grado di manipolare strutture complesse come i grafi con i relativi calcoli delle centralità di tutti i nodi; è stato necessario l’utilizzo del linguaggio di programmazione Python 3, in grado di interfacciarsi con le librerie già citate nell’introduzione, dove 3 indica la versione attuale del sistema. Python è un linguaggio di programmazione di più “alto livello” rispetto alla maggior parte degli altri linguaggi, orientato a oggetti, adatto, tra gli usi, a sviluppare applicazioni distribuite, scripting, computazione numerica e system testing. Esso è un linguaggio multi-paradigma che ha tra i principali obiettivi: dinamicità, semplicità e flessibilità. Supporta, come già detto, il paradigma “object-oriented”, la programmazione strutturata e molte caratteristiche di programmazione funzionale e riflessione.

Tale strumento è stato fortemente consigliato dai relatori per il calcolo delle diverse centralità poiché gli algoritmi utili erano già implementati e disponibili nelle librerie Networkit (implementato nel linguaggio di programmazione Python, versione 3). In aggiunta un altro consiglio è stato l’utilizzo delle librerie “stats” per l’analisi statistica e quindi per effettuare i confronti tra le varie metriche (anch’essa utilizza Python); consiglio accettato per il calcolo del solo Test di Friedman, di cui forniremo una descrizione più avanti in questo capitolo.

3.2.2 R (Software)

R è un linguaggio di programmazione e un ambiente di sviluppo specifico per l’analisi statistica dei dati. Esso è un software libero in quanto viene distribuito con la licenza GNU GPL (*General Public License*), ed è disponibile per diversi sistemi operativi. Il suo linguaggio orientato agli oggetti deriva direttamente dal pacchetto S distribuito con una licenza non open source. La sua popolarità e semplicità è dovuta anche all’ampia disponibilità di moduli distribuiti con la licenza GPL e organizzati in un apposito sito chiamato “CRAN” (“Comprehensive R Archive Network"). Tramite questi moduli è possibile estendere di molto le capacità del programma, aggiungendo anche di propria iniziativa moduli utili agli utilizzatori del sistema. Anche se il linguaggio è fornito con un’interfaccia a riga di comando, sono disponibili diverse interfacce grafiche che consentono di integrare R con diversi pacchetti come Emacs (Editor di testo libero). R è stato scelto per idea del tirocinante, entusiasta dalla sua semplicità nell’organizzazione dei pacchetti e del calcolo dei risultati. Esso è stato utilizzato per le misure statistiche generali, per grafici delle distribuzioni e per le correlazioni tra le centralità.

3.3 Libreria networkit

Questa sezione comprende una descrizione sulla libreria implementata nel corso della ricerca.

Networkit è uno strumento “open source” per l’analisi di grafi su larga scala. Fornisce tutti gli strumenti utili per l’analisi ed è in grado di gestire grafi che possiedono dalle centinaia ai milioni di archi in maniera semplice. Una delle motivazioni che ha portato a preferire l’uso di Networkit è sicuramente l’ottimo compromesso tra buone prestazioni e perdita delle informazioni: poiché l’intento era quello di costruire un grafo che rappresentasse l’intera community di un social media. Si sospettava che il grafo avesse grandi dimensioni: perciò era necessario uno strumento che permettesse processi di calcolo eseguibili in tempi ragionevoli. Per realizzare tutto questo, gli algoritmi e le strutture dati implementate da Networkit sono i risultati di un duro lavoro di software engineering studiato per ottenere alte prestazioni e parallelismo, con un codice ibrido per combinare le prestazioni del kernel utilizzando il linguaggio C++ unito all’uso di un’interfaccia Python. Molte delle implementazioni sono le più veloci nel campo della ricerca, grazie ad algoritmi paralleli che sfruttano architetture multicore; così facendo in tempi più brevi possono essere calcolate diverse proprietà importanti sul grafo. Tra le tipologie di misure/analisi che Networkit può eseguire abbiamo quella a noi interessata, cioè il calcolo delle **misure di centralità**.

Il modulo che interessa a noi sono il networkit.centrality e il modulo networkit.base. Nell’implementazione dei moduli è presente il meccanismo dell’ereditarietà, concetto della programmazione orientata agli oggetti che consente di scomporre il modello dati in una gerarchia di classi che definiscono strutture dati e comportamenti differenziati; tale meccanismo permette di definire classi astratte con funzionalità comuni a tutti gli oggetti appartenenti a quella categoria fino ad arrivare a classi sempre più specifiche. Tra i numerosi vantaggi nell’usare questo meccanismo vi è la riutilizzazione di porzioni di codice, la possibilità di adattare o estendere anche in un secondo momento il comportamento a nuovi casi d’uso specifici estendendo in questo modo anche il ciclo di vita del codice; per funzionare però la gerarchia delle classi deve seguire alcune regole di buona progettazione. Descrivendo le diverse classi dei moduli, la gerarchia è composta dalla classe Oggetto, che generalizza la classe astratta networkit.base.Algorithm, che generalizza la classe networkit.centrality.Centrality la cui specializzazione è data dall’implementazione delle classi networkit.centrality.ApproxBetweenness, networkit.centrality.Closeness, networkit.centrality.Eigenvector, networkit.centrality.DegreeCentrality, networkit.centrality.KatzCentrality e networkit.centrality.PageRank (vedi immagine 3.1).

Di seguito una descrizione sommaria delle diverse classi:

-networkit.base.Algorithm: Classe astratta per la definizione di algoritmi, tra i metodi implementati vi è il metodo run() che esegue l’algoritmo rappresentato dall’oggetto;

- networkit.centrality.Centrality: Classe astratta per la definizione delle misure di centralità, i metodi implementati tra gli altri sono ranking() che restituisce una lista i cui elementi rappresentano il numero del nodo e il suo valore di centralità in ordine decrescente e il metodo scores() che restituisce i valori di centralità ordinati per numero di nodo;

-networkit.centrality.Valore: Classe specializzata per il calcolo di una ben precisa centralità tra quelle elencate, la descrizione è fornita nel prossimo capitolo.

3.4 Indicatori di centralità

Nella teoria dei grafi e nell’analisi delle reti, gli indicatori di centralità identificano i vertici più importanti all’interno di un grafo. L’applicazione include l’identificazione degli attori più influenti all’interno della rete sociale. Questo capitolo con le sue sottosezioni è utile a spiegare le caratteristiche fondamentali che contraddistinguono le diverse metriche da calcolare su un nodo all’interno di una rete. Fino ad oggi, possiamo raggruppare queste misure di centralità in due macro-gruppi; nella quale le caratteristiche analizzate dalle quantità appartenenti al primo gruppo sono totalmente differenti dalle caratteristiche analizzate nel secondo gruppo.

Possiamo categorizzare le prime quattro misure (Betweeness, Closeness, In-degree, Out-degree), inserendole nel primo gruppo, come i valori che esprimono l’importanza all’interno della rete in merito ai propri collegamenti ed ai collegamenti degli altri verso il nodo stesso; invece il resto delle misure possono rientrare nel secondo gruppo (Eigenvector, Katz, Pagerank) poiché sono misure perlopiù equivalenti che utilizzano come metro di giudizio la quantità di collegamenti o il prestigio di tutti i nodi vicini a lui. Quindi diremo che i nodi con alta centralità Eigenvector/Katz/Pagerank avranno molti contatti con nodi importanti della rete.

3.4.1 Centralità Betweeness

**Nella teoria dei grafi, la quantità betweeness è una misura di centralità nei grafi basata sul concetto di “cammino minimo”; come cammino minimo viene inteso rispetto la lunghezza di un cammino, cioè il numero di collegamenti attraversati per andare da un nodo all'altro.**

**Per ogni due vertici di un grafo connesso, esiste almeno un cammino minimo tra i vertici tale che il minor numero di archi che passano attraverso al cammino sia minimizzato. La centralità Betweeness per ogni vertice è il numero di cammini minimi che passano attraverso il medesimo. Il calcolo della misura per il nodo “v” avviene tramite il calcolo dell’espressione:**

nella quale sigma\\_st è il numero totale di cammini minimi dal nodo ‘s’ al nodo ‘t’ e sigma\\_st(v)

è la cardinalità del sottoinsieme formato dai cammini minimi che passano per il nodo ‘v’.

\\Algoritmo delle centralità più costoso da effettuare in un grafo in termini di tempo, dato che esso è un problema di minimo e quindi incorre il problema dell’enumerazione di tutti i possibili cammini (path), scegliendo tra essi quelli di minor lunghezza. Per accentuare questo problema, incorrendo però nella precisione dei risultati ottenuti; facciamo uso di un algoritmo approssimato rappresentato dalla classe networkit.centrality.ApproxBetweenness descritto in Matteo Riondato e Evgenios M. Kornaropoulos: “Fast Approximation of Betweeness Centrality through Sampling”.

Il costruttore della classe networkit.centrality.ApproxBetweenness prende in ingresso quattro parametri: il grafo, epsilon, delta e universalConstant. La dicitura del costruttore è la seguente:

networkit.centrality.ApproxBetweenness(G, epsilon=0.01, delta=0.1, universalConstant=1.0);

Dalla documentazione, l’algoritmo approssima la quantità Betweeness di tutti i vertici in modo che i punteggi rientrino in un errore additivo epsilon con probabilità almeno 1 – delta, dove epsilon e delta sono parametri dati in input all'oggetto come visto nella dicitura. I valori vengono normalizzati di default. Il metodo run() ha un costo in tempo di O(m) per campione, dove ‘m’ è il numero di archi.

* + - 1. Centralità Closeness

In un grafo connesso, la centralità closeness di un nodo è calcolata come il reciproco della somma delle lunghezze dei cammini minimi tra il nodo da esaminare e tutti i nodi all'interno del grafo. Quindi, più la quantità di closeness di un nodo è alta più il nodo è "vicino" a tutti gli altri nodi della rete. La misura è stata definita da Bavelas (1950) come il reciproco della farness (misura che calcola la somma dei cammini minimi):

dove d(y, x) è la distanza tra i vertici di ‘x’ e ‘y’.

Il costruttore della classe prende in ingresso

La classe crea un oggetto algoritmo che calcola l’esatta quantità di closeness di tutti i nodi del grafo. Il costruttore presenta la seguente dicitura:

Closeness(G, normalized=FALSE, variant=networkit.centrality.ClosenessVariant.Standard)

L’algoritmo si aspetta due parametri oltre al grafo da analizzare, di cui una variabile booleana per la normalizzazione dei risultati e una variabile “variant” per decidere quale definizione di closeness scegliere per il calcolo, nell’implementazione le possibilità per questa variabile sono 2: la versione standard (utilizzata nel nostro caso) per il calcolo su grafi connessi, e una versione generalizzata per il calcolo su grafi non connessi.

L’algoritmo calcola i valori non normalizzati, dopo il settaggio del parametro “Normalized” a “False”. Il metodo run() ha un costo in tempo di O(n\*m), dove ‘n’ è il numero di nodi e ‘m’ il numero di archi del grafo.

* + - 1. Centralità In degree

Storicamente la prima misura di centralità considerata, per la sua intuitività è la misura del grado, la quale definizione comprende il numero di archi che entrano nel nodo. Il grado di collegamento è la caratterizzazione del rischio immediato che quel nodo riceva ciò che sta fluendo all’interno della rete. Nel nostro particolare caso, gli archi associati sono aspetti positivi (cioè scambio di informazioni) e quindi l’in-degree può essere considerato come una forma di popolarità nella rete. Il grado di centralità di un vertice ‘v’, per un grafo G:=(V,E) con |V| vertici e |E| archi, è definita come:

La dicitura del costruttore per il calcolo di questo algoritmo è:

DegreeCentrality(G, normalized=False, outDeg=False,ignoreSelfLoops=True).

Il metodo si aspetta come parametri il grafo G, la normalizzazione dei risultati sé settato a True, l’opzione “outDeg” impostato a False in quanto vogliamo calcolare la centralità in ingresso e non in uscita e infine si definisce il parametro per ignorare i self-loop, poiché non vogliamo contare nella quantità di in degree il numero di operazioni svolte dal nodo su sé stesso in quanto questo particolare andrebbe a inquinare il reale risultato.

Nell’algoritmo utilizzato su Networkit, il metodo run() utilizza in tempo O(m), dove ‘m’ è il numero degli archi nel grafo.

* + - 1. Centralità Out degree

Equivalente alla misura della centralità in degree, la misura out degree utilizza il grado in uscita dei nodi, cioè il numero di archi che si vanno a collegare ad altri nodi della rete. Quindi, equivalentemente alla misura in degree rappresenta una forma di popolarità all’interno della rete. Il grado di centralità di un vertice ‘v’, per un grafo G:=(V,E) con |V| vertici e |E| archi, è definita come:

L’algoritmo utilizzato dalla libreria è lo stesso del grado in entrata, con la differenza che il parametro “outDeg” è impostato a True. Per il resto, dato che trattiamo la stessa classe dell’indegree centrality, i comportamenti sono identici.

* + - 1. Centralità Eigenvector

Nella teoria dei grafi, la centralità Eigenvector (anche chiamata “eigencentrality” o “punteggio prestigio”) è una misura dell’influenza di un nodo all’interno di una rete. I relativi punteggi vengono assegnati a tutti i nodi della rete in base al concetto che le connessioni ai nodi con punteggio alto contribuiscono maggiormente al punteggio del nodo in questione rispetto alle connessioni ai nodi con punteggio basso. Un punteggio di eigenvector alto significa che un nodo è connesso a molti nodi che hanno punteggi alti.

Dato un grafo G:= (V,E) con |V| (cardinalità di V) vertici; definito A = () come la matrice di adiacenza, nella quale se il vertice ‘v’ è collegato al vertice ‘t’, e altrimenti . La relativa centralità, chiamata ‘x’ nell’equazione, per il vertice ‘v’ può essere definita come:

dove M(v) è l’insieme dei nodi vicini a v e λ è una costante. Con un piccolo riarrangiamento questo può essere riscritto in notazione vettoriale come equazione per il calcolo degli autovettori relativi all’autovalore λ:

In generale, ci sono diversi autovalori λ corrispondenti ad autovettori diversi da zero. Tuttavia, la matrice di adiacenza ha tutte le sue componenti non negative e ciò implica che, per il teorema di Perron-Frobenius, l’autovalore di modulo massimo lamda di A è reale positivo e l’autovettore corrispondente ha tutte le componenti positive. La componente v-esima del relativo autovettore fornisce quindi il punteggio di centralità relativa del vertice ‘v’ nella rete.

Il costruttore della classe networkit.centrality.Eigenvector inizializza, tramite i suoi parametri, il grafo e la tolleranza per il calcolo dell’autovettore; essendo la matrice di dimensioni elevate non può essere una stima accurata quindi ci si deve accontentare di un margine di errore.

La dicitura della classe è la seguente: networkit.centrality.Eigenvector(G,tol=1e-9).

* + - 1. Centralità Katz

La centralità secondo Katz è una generalizzazione della centralità per grado, introdotto da Leo Katz nel 1953. Come abbiamo visto, la centralità per grado misura il numero di nodi direttamente collegati al nodo che si sta misurando; la centralità secondo Katz misura il numero di tutti i nodi la quale sono connessi attraverso un cammino, allo stesso tempo il contributo dei nodi distanti è penalizzato. Matematicamente, è definito come:

Dove è il fattore di attenuazione compreso tra (0,1).

La centralità secondo Katz può essere vista come variante della centralità per autovettore (eigenvector). Un’altra forma di centralità per Katz è:

Messa a confronto con l’espressione della centralità per autovettore, si nota che al posto di abbiamo solamente il termine .

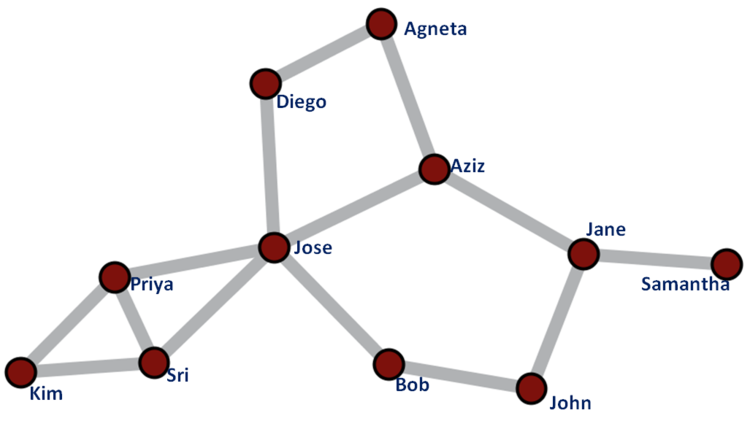
È dimostrato che l’autovettore principale (associato al più grande autovalore di A, dove A è la matrice di adiacenza) è il limite della centralità secondo Katz poiché si avvicina a da sotto.

Per quanto riguarda l’algoritmo utilizzato, esso richiede tempo O(m). Il numero di iterazioni dipende da quanto tempo ci vuole per raggiungere la convergenza (e quindi sulla tolleranza desiderata, specificata come parametro della classe)

La dicitura del costruttore è:

networkit.centrality.KatzCentrality(G,alpha=0,beta=0.1,tol=1e-8)

dove alfa e beta sono parametri utilizzati per il calcolo dell’algoritmo.

Ad esempio, nella figura sotto, assumiamo di voler calcolare la centralità di John con alfa=0.5. Il peso assegnato ad ogni collegamento con i nodi direttamente collegati a John avranno un peso nella sommatoria di (0.5)^1 = 0.5. Jose è connesso indirettamente a John, attraverso la connessione con Bob, dato che il cammino è composto da due collegamenti avremo che il peso nella sommatoria sarà (0.5)^2=0.25. Similmente i collegamenti con Agneta attraverso Aziz e Jane sarà (0.5)^3 = 0.125 e cosi via.

* + - 1. PageRank

Ultima ma non mento importante, è la centralità Pagerank. Analizzando le centralità già viste, notiamo che la centralità secondo Katz ha il problema seguente: se il nodo con alta centralità è collegato a molti altri nodi, allora anche questi ultimi automaticamente avranno un’alta centralità. In molti casi, tuttavia, significa meno se un nodo è solo uno tra i tanti da collegare, cioè la centralità acquisita da un nodo dovrebbe essere maggiore se e solo se questo nodo ha un basso grado di connessioni, in modo da eliminare il fattore per cui un nodo avendo a disposizione tanti collegamenti, si dovrebbe andare a considerare proprio il collegamento con il nodo in questione.

La centralità PageRank è una correzione della centralità secondo Katz poiché tiene in considerazione questo problema. Ci sono tre distinti fattori che determinano la centralità per un nodo:

1. Il numero di archi collegati a lui;
2. La propensione al collegamento dei suoi vicini;
3. La centralità dei suoi vicini.

Il primo fattore è ovvio, infatti più un nodo è “linkato” (riferito) dagli altri nodi più è centrale; ragionevolmente, il valore dell’appoggio si deprezza proporzionalmente al numero di collegamenti forniti dal nodo: i link provenienti da nodi parsimoniosi sono più degni di quelli emanati da nodi con un eccesso di link.

Infine, non tutti i nodi anche con ugual grado di connessione vengono considerati uguali: i collegamenti dai vertici importanti sono più preziosi di quelli dai vertici sconosciuti.

Questa classificazione è utilizzata da Google per assegnare un valore di “prestigio” alle pagine Web. Si tratta di una distribuzione probabilistica che misura la probabilità che un utente generico, navigando in maniera random attraverso i link delle pagine visitate, arrivi ad una pagina target. L’algoritmo è composto da un’inizializzazione e un aggiornamento:

(Inizializzazione)

(Aggiornamento)

* PR( è il PageRank di un nodo ;
* M( è l’insieme di tutti i nodi che puntano ;
* L() è l’out degree della pagina ;

d è un fattore di damping factor. La differenza 1-d è la probabilità che l’utente che naviga in maniera random seguendo i vari collegamenti web , possa decidere di smettere di seguire i collegamenti e di scegliere di aprire una pagina a caso

Per quanto riguarda i grafi che si stanno analizzando, semanticamente parlando, un nodo avrà un alto valore di PageRank quando gli utenti saranno interessati ad agire con quel particolare utente.

Nell’algoritmo utilizzato nel nostro caso abbiamo settato il valore del damp=0.85.

La dicitura del costruttore è la seguente:

networkit.centrality.PageRank(G, damp=0.85, tol=1e-9)

ELEMENTI DI STATISTICA

Viene dedicato un capitolo alla presentazione dei metodi ricorrenti nella ricerca statistica.

La statistica è una disciplina che ha come fine lo studio quantitativo e qualitativo di un particolare fenomeno in condizioni di incertezza.

Tale scienza è comunemente suddivisa in due categorie in base alle proprietà che vogliamo definire nello studio da effettuare sul campione o sulla popolazione:

* Statistica descrittiva: studia il fenomeno relativo o un campione (parte di una popolazione);
* Statistica inferenziale: estende i risultati ottenuti su un campione, all’intera popolazione.

Le due sottosezioni seguenti sono divise in base a queste due branche della statistica.

STATISTICA DESCRITTIVA

Nella statistica descrittiva sono definiti gli strumenti utilizzati per sintetizzare e classificare le diverse distribuzioni dei nodi in merito alle loro centralità, i dati dell’indagine possono essere rappresentate attraverso molteplici rappresentazioni grafiche e risultati analitici.

Le rappresentazioni grafiche servono per evidenziare in modo semplice, le quattro caratteristiche fondamentali di una distribuzione di frequenza (tendenza centrale, variabilità, simmetria e curtosi). Insieme con i vantaggi di fornire una visione sintetica e di essere di facile lettura, hanno però l’inconveniente fondamentale di mancare di precisione; per avere una visione più dettagliata ed analitica vengono presentati pure metodi di sintesi dei dati.

ISTOGRAMMA E STEM AND LEAF (Metodo grafico)

Per dati quantitativi, riferiti a variabili continue misurate su scale ad intervalli, di norma si ricorre a istogrammi oppure tramite un altro tipo di rappresentazione chiamato stem-and-leaf.

Gli istogrammi sono grafici a barre verticali nei quali:

-le misure della variabile casuale sono riportate lungo l’asse orizzontale;

-mentre l’asse verticale rappresenta il numero assoluto, oppure la frequenza relativa o quella percentuale, con cui compaiono i valori di ogni classe.

I lati dei rettangoli sono costruiti in corrispondenza degli estremi di ciascuna classe. Un istogramma deve essere inteso come una rappresentazione areale: nella quale le superfici dei vari rettangoli devono essere proporzionali alle frequenze corrispondenti. Quando le classi hanno la stessa ampiezza, le basi dei rettangoli sono uguali; di conseguenza, le loro altezze risultano proporzionali alle frequenze che rappresentano. Solo quando le basi sono uguali, è indifferente ragionare in termini di altezze o di aree di ogni rettangolo. Ma nel caso in cui le ampiezze delle classi sono diverse, bisogna ricordare il concetto generale che le frequenze sono rappresentate dalle superfici e quindi è necessario rendere l’altezza proporzionale.

Descrivendo il diagramma a ramo e foglia (stem-and-leaf plot), esso è un tipo di tecnica semi-grafica, che può essere descritta come un incrocio tra un istogramma e una tabella di frequenza. Il metodo è utile per una prima descrizione di una distribuzione di dati.

I principi di costruzione sono semplici:

* Ogni numero è diviso in due parti, il ramo (stem) e la foglia (leaf);
* Il ramo è il numero, collocato a sinistra, che include tutte le cifre eccetto l’ultima;
* La foglia, collocata a destra, è sempre un numero con una cifra sola (single digit), che può essere esclusivamente l’ultima di tutto il numero.

Anche questo grafico insieme all’istogramma ha lo scopo di mostrare le caratteristiche fondamentali di una distribuzione di dati:

* Valore minimo e massimo e quindi l’intervallo di variazione;
* I valori più frequenti;
* La presenza di uno o più picchi;
* La forma della distribuzione, in relazione soprattuto alla simmetria;
* La presenza di outlier o valori anomali, quelli troppo distanti dal gruppo principale di valori.

BOX-AND-WHISKER (Metodo grafico)

I diagrammi Box-and-Whisker, chiamati anche box-plot, sono un metodo grafico diffuso recentemente e reso di uso corrente dai programmi informatici, che possono costruirlo con rapidità. La quantità di informazioni che forniscono è elevata, essi riassumono cinque informazioni statistiche contenute nelle distribuzioni:

* La mediana
* Il primo e il terzo quartile
* Il valore minimo e quello massimo

Anche questo metodo grafico come i precedenti mette in rilievo quattro caratteristiche fondamentali di una distribuzione statistica di dati:

1. La misura di tendenza centrale, attraverso la mediana;
2. Il grado di dispersione o variabilità dei dati, rispetto alla mediana;
3. La forma della distribuzione dei dati, in particolare la simmetria;
4. La presenza di outliers, cioè valori anomali.

La sua realizzazione richiede una serie di passaggi logici, che può essere riassunta in uno schema composto da cinque punti, dalla quali derivano gli elementi metodologici:

1. Viene tracciata una linea orizzontale, interna alla scatola, che rappresenta la mediana;
2. La scatola è delimitata da due linee orizzontali; dove la linea inferiore indicata con Q\_1 che rappresenta il primo quartile (i quantili vengono illustrati nella sezione successiva) o quartile inferiore, e la linea superiore indicata con Q\_3 che rappresenta il terzo quartile o quartile superiore.
3. La distanza tra il terzo e il primo quartile, detta distanza interquartilica è una misura della dispersione della distribuzione, anche se poco precisa;
4. Le linee che si allungano dai bordi della scatola e che si concludono con altre due linee orizzontali, i baffi (whiskers), che delimitano gli intervalli nei quali sono collocati i valori minori di Q\_1 (nella parte inferiore) e quelli maggiori di Q\_3 (nella parte superiore) non classificati come outliers. Il valore più alto tra quelli presenti nella variabile che non identifica un valore anomalo definisce la fine del baffo superiore.
5. Nel caso in cui ci fossero dei valori anomali, questi ultimi sarebbero rappresentati nel box-plot come dei punti isolati posizionati al di sopra e/o al di sotto dei baffi della distribuzione

QUANTILI

Il quantile di ordine alfa o alfa-quantili (con alfa numero reale nell’intervallo [0,1]) è un valore che divide la popolazione in due parti, proporzionali ad alfa e (1-alfa) e caratterizzate da valori rispettivamente minori o maggiori di q\_a. Per poter calcolare un quantile di un certo ordine, il campione deve essere almeno ordinato, cioè almeno deve essere definibile un ordinamento tra gli elementi che compongono i dati. I diversi quantili sono molto utili per poter dare una descrizione anche se sommaria, in base alla suddivisione e la scelta di alfa, della distribuzione dei dati. Nel caso del box-plot, vale che Q\_1 è indicato con q\_alfa dove alfa=0.25 (Q\_1 è detto quindi 25esimo percentile), Q\_2 è indicato con q\_alfa dove alfa=0.5, che rappresenta la mediana della distribuzione e infine Q\_3 è indicato con q\_alfa dove alfa=0.75

FAMIGLIE DI INDICI

Viene fornito una lista di indici utilizzati, per completezza nell’analisi statistica, durante questa ricerca:

* Deviazione standard (sigma) – Lo scarto quadratico medio, o appunto deviazione standard, è un indice di dispersione statistico, vale a dire una stima della variabilità di una popolazione di dati. È uno dei modi per esprimere la dispersione dei dati intorno a un indice di posizione, come la media aritmetica o una sua stima. Dato che è una misura effettuata con l’utilizzo dei dati espressi in una certa misura, tale indice avrà la stessa misura;
* Indice di asimmetria – In questa famiglia di indici, sono presenti diverse formule matematiche per il calcolo dell’asimmetria di una distribuzione. Un indice di asimmetria di una distribuzione è un valore che cerca di fornire una misura della sua mancanza di simmetria. Un’asimmetria sta a indicare che la distribuzione ha una densità maggiore nei valori più bassi o nei valori più alti rispetto a una misura statistica nota, come potrebbe essere la mediana o la media. L’aspetto comune a tutti gli indici con questo scopo è che il valore 0 ritornato fornisce una condizione necessaria, ma non sufficiente, affinché una distribuzione sia simmetrica. Gli indici di asimmetria comunemente utilizzati si basano su alcune proprietà delle distribuzioni simmetriche, o in particolare, della distribuzione normale. Tra gli indici utilizzati abbiamo l’indice skewness e l’indice di curtosi, dove l’ultimo ha come obiettivo la visualizzazione della forma di una distribuzione, che costituisce una misura dello “spessore” delle cose di una funzione di densità, ovvero il grado di “appiattimento” di una distribuzione, sempre in confronto a una distribuzione normale;
* Coefficiente di variazione – Il coefficiente di variazione è un indicatore statistico di dispersione relativa, calcolato come rapporto tra la deviazione standard e la media della distribuzione. La dispersione relativa serve a indicare la variabilità di un fenomeno in termini percentuali. È usata per confrontare la variabilità dei fenomeni, senza prendere in considerazione l’unità di misura.

STATISTICA INFERENZIALE

Nella statistica inferenziale vengono descritti i test utilizzati per affermare una determinata ipotesi. In statistica il test è un processo logico-matematico che porta alla conclusione di non poter respingere oppure di poter respingere l’ipotesi di casualità, mediante il calcolo di probabilità specifiche di commettere un errore con queste affermazioni. L’ipotesi che il risultato ottenuto con i dati sperimentali raccolti sia dovuto solo al caso è chiamato ipotesi nulla ed è indicata con H\_0. Di norma, con essa si afferma che le differenze tra due o più gruppi, quelle tra un gruppo e il valore atteso oppure le tendenze riscontrate siano imputabili essenzialmente al caso; per giungere a queste conclusioni si deve ricorrere all’inferenza. La domanda inferenziale di verifica dell’ipotesi nulla a cui si deve rispondere tramite questi strumenti è:

\begin{quotation}

“Nell’ipotesi che le differenze fra gruppi di osservazioni empiriche siano dovute a fattori esclusivamente casuali, qual è la probabilità che fra tutte le alternative possibili si presenti proprio la situazione descritta dei dati raccolti o una ancora più estrema?”

\end{quotation}

Se tale probabilità risulta alta, convenzionalmente uguale o superiore al 5%, si imputeranno le differenze a fattori puramente casuali. Al contrario, se la probabilità risulta bassa, inferiore al valore prefissato, si accetta come verosimile che le differenze siano dovute a fattori non casuali, rientranti tra i criteri che distinguono i gruppi di dati.

I test utilizzati in questa ricerca vengono detti test non parametrici; un test statistico è parametrico quando nel controllo delle ipotesi vi è la presenza di un parametro come la media, la varianza o la deviazione standard. Al contrario un test non parametrico non presuppone alcun tipo di distribuzione, quindi non utilizzano i valori effettivi dei dati; mentre un test parametrico ha il vincolo di poter essere applicato solamente a una distribuzione normale o comunque che la distribuzione si avvicini a una distribuzione simmetrica, con il vantaggio che i risultati riscontrati con questi test sono più attendibili rispetto a quelli non parametrici.

Descrivendo sommariamente i test non parametrici, molti hanno la caratteristica identica di trasformare i valori della distribuzione in ranghi. Il punto cardine sta nel fatto che adesso la distribuzione perde una quantità di informazione, passando dalla scala di rapporti che trova il vantaggio di avere un’origine reale e la proprietà di poter rapportare una coppia di valori; mentre con la trasformazione delle distribuzioni in ranghi si perde dell’informazione passando da una scala di rapporti a una scala ordinale, con le proprietà di equivalenza e di relazioni di maggiore e minore tra i dati. Un risultato importante di questi test è rappresentato dal valore p-value; nei test di verifica d’ipotesi, il valore p è la probabilità di ottenere risultati uguali o meno probabili di quelli osservati durante il test, supposta vera l’ipotesi nulla. Talvolta tale valore viene chiamato livello di significatività osservato.

TEST DI CORRELAZIONE PER RANGHI (SPEARMAN e KENDALL)

La correlazione è uno dei metodi statistici più antichi, diffuso già all’inizio del 900. La metodologia non parametrica proposta da C. Spearman è una correlazione basata sui ranghi, che ricorre agli stessi concetti della correlazione parametrica r di Pearson.

In riferimento a test di correlazione non parametrica, in letteratura sono ricorrenti i termini di correlazione tra ranghi e cograduazione utilizzati in modo appropriato solo con scale almeno di tipo ordinale, mentre associazione e concordanza con scale qualitative o categoriali.

Ci soffermiamo sul concetto di correlazione tra ranghi, e descriviamo due metodi concettualmente differenti ma che rispondono allo stesso quesito.

Il primo metodo calcola il coefficiente di correlazione di Spearman, indicato con il simbolo greco rho. Questa quantità può variare tra +1 e -1 quando la correlazione è massima, con valore positivo oppure negativo, ed è vicino a zero quando non esiste correlazione. Il metodo richiede che entrambe le variabili siamo misurate su una scala almeno ordinale che nel corso di questo paragrafo indicheremo con X e Y per semplicità. Il coefficiente di correlazione per ranghi di Spearman serve per verificare l’ipotesi nulla (H\_0) dell’indipendenza tra due variabili, nel senso che gli N valori della variabile Y hanno le stesse probabilità di associarsi con ognuno degli N valori di X. L’ipotesi alternativa di esistenza di una associazione può prevedere un risultato positivo oppure negativo, nel primo caso è detta associazione diretta: le coppie di valori sono contemporaneamente alti o bassi sia per X che per Y; nel secondo caso è chiamata associazione indiretta e a valori alti di X corrispondono valori bassi di Y o viceversa.

La metodologia di calcolo è la seguente:

1. Indiciamo l’ipotesi nulla H\_0, in base a ciò che vogliamo scoprire; l’ipotesi può essere bilaterale o unilaterale:
   1. Bilaterale, quando vogliamo che un valore sia diverso o uguale a una certa quantità; un esempio H\_0: RHO = 0;
   2. Unilaterale in una direzione, quando vogliamo osservare se Rho sia maggiore o minore di un determinato valore; un esempio H\_0: RHO>=0 .
2. Successivamente, occorre ordinare i ranghi della variabile X, assegnando 1 al valore più piccolo e progressivamente valori interi maggiori, fino ad N per il valore più alto, se i dati della variabile hanno due o più valori uguali, è necessario assegnare a ognuno di essi come rango la media delle loro posizioni.
3. Sostituire anche gli N valori di Y con i ranghi rispettivi;
4. Se le due distribuzioni
   1. Sono correlate in modo positivo (rho=+1), i valori della variabile X e della Y relativi allo stesso soggetto saranno uguali;
   2. Sono correlate in modo negativo (rho=-1), a valori alti di X saranno associati valori bassi di Y e viceversa;
   3. Se tra le due variabili non esiste correlazione (rho=0), i valori di X e di Y relativi agli stessi soggetti saranno associati in modo casuale

Per quantificare questo grado di correlazione o concordanza, Spearman ha proposto la distanza tra le coppie dei ranghi (d\_Ri), quindi l’indicatore di correlazione è la somma dei quadrati

\sum{d\_R\_i^2}

1. Quando rho=+1, le coppie di osservazioni di X e Y hanno lo stesso rango e pertanto questa sommatoria è uguale a 0, quando rho=-1, se X è ordinato in modo crescente, Y è ordinato in modo decrescente: di conseguenza, le differenze sono massime e la sommatoria raggiunge un valore massimo determinato dal numero di coppie di osservazioni (N), quando rho=0, mentre i ranghi di X sono ordinati in modo crescente quelli di Y hanno una distribuzione casuale: la sommatoria delle d^2\_R\_i tende a un valore medio, determinato dal numero di coppie di osservazioni (N).

Un altro tipo di test nato 30 anni dopo la rho di Spearman, è stato scoperto da Kendall proponendo il suo test tau. Questo metodo ha le stesse assunzioni, può essere utilizzato nelle medesime condizioni e sui medesimi dati del test rho di Spearman. I risultati tra i due test sono molto simili, anche se matematicamente non equivalenti.

La metodologia utilizzata per calcolare la tau di Kendall segue 6 passaggi:

1. Occorre ordinare per ranghi la variabile X, assegnando il rango 1 al valore più piccolo e progressivamente un rango maggiore, fino ad N, al valore più grande. Anche in questo caso se sono presenti due o più valori uguali nella variabile X, assegnare ad ognuno come rango la media delle loro posizioni.
2. Sostituire gli N valori di Y anch’essi con i ranghi rispettivi.
3. Se le distribuzioni sono correlate in modo positivo (tau=+1), anche i ranghi della variabile Y sono ordinati in modo crescente, concordanti con l’ordine naturale, se correlate in modo negativo (tau=-1) vale che i valori di Y risulteranno ordinati in modo decrescente e saranno discordanti dall’ordine naturale, se tra le due variabili non esiste correlazione (tau=0), l’ordine della variabile Y risulterà casuale e il numero di ranghi concordanti e di quelli discordanti dall’ordine naturale tenderà ad essere uguale, con somma 0.
4. Per quantificare il grado di correlazione o concordanza, Kendall ha proposto di contare la sola variabile Y: quante sono le coppie di ranghi che sono concordanti e quante quelle discordanti dall’ordine naturale; quando si ha che il valore precedente è maggiore del valore successivo, quindi non è nell’ordine naturale e pertanto contribuirà con -1 alla correlazione, viceversa nel caso in cui nell’ordine della variabile Y due valori contigui sono nell’ordine naturale questo concorrerà con un valore +1. La misura della concordanza complessiva con la variabile X è data dalla somma algebrica di tutte le concordanze e discordanze;
5. Per ricondurre il valore calcolato a un campo di variazione compreso tra +1 e -1, il numero totale di concordanze e discordanze di una serie di valori deve essere rapportato al massimo totale possibile, quindi secondo il metodo proposto da Kendall, il grado di relazione o concordanza (tau) tra la variabile X e Y può essere quantifica dal rapporto

Tau=frac(totale(concordanze-discordanze))/massimo totale possibile

TEST DI FRIEDMAN

Il test proposto da Milton Friedman nel 1937 è utile quando si vuole osservare se le diverse mediane di più campioni dipendenti sono uguali o meno; questo tipo di test è utilizzabile quando si dispone di dati categorizzati su una scala quantitativa continua. Il test verifica l’ipotesi nulla sulla tendenza centrale:

H\_0 : me1=me2=…=meK

Con l’ipotesi alternativa

H\_1: non tutte le k mediane sono uguali

Esso è l’alternativa non parametrica all’ANOVA a due criteri di classificazione o a blocchi randomizzati, quando non sono rispettate le condizioni di validità richieste dai test parametrici. E’ uno dei test non parametrici più potenti e di uso più generale. Il procedimento è semplice e richiede pochi passaggi:

1. Data una tabella a doppia entrata dove nelle righe vengono inseriti gli individui da analizzare e nelle colonne le diverse metriche da confrontare;
2. Trasformare i punteggi o le misure in ranghi entro la stessa riga, assegnando 1 al punteggio minore e progressivamente valori maggiori fino a k;
3. Successivamente bisogna sommare per colonna i valori dei ranghi;
4. Se l’ipotesi nulla H\_0 è vera, nelle colonne a confronto i ranghi minori e quelli maggiori dovrebbero essere distribuiti casualmente, pertanto le somme dei ranghi nelle k colonne dovrebbero essere tra loro tutte equivalente ed essere uguali ad un valore atteso (T\_i) che dipende dal numero di osservazioni; se l’ipotesi è falsa in almeno una colonna si concentrano i ranghi minori o maggiori, di conseguenza tale somma tende ad essere significativamente differente dal valore T\_i atteso.

T\_i(attesi)=N(k+1)/2

1. Per decidere se queste somme sono significativamente differenti dall’atteso, si calcola la statistica Fr che rappresenta la sommatoria dei quadrati degli scarti tra i k totali osservati e i corrispondenti attesi. Il valore di Fr tenderà a 0 nel caso di accordo tra totali osservati e totali attesi (H\_0 vera), mentre tenderà a un valore alto al crescere dello scarto tra essi (H\_0 falsa)

Fr= sommatoria(T\_i – N(k+1)/2)^2

TEST DI WILCOXON

Il test dei segni per ranghi di Wilcoxon è un test non parametrico che si applica nel caso in un singolo campione con due misure accoppiate. Esso normalmente viene utilizzato per verificare se esistono differenze significative nei livelli mediani di due quantità dello stesso campione; nella nostra metodologia, viene inserito dopo aver attuato il test di Friedman per individuare i valori con differente mediana. Quindi per semplicità potremmo dire che piuttosto di verificare l’ipotesi su k valori dello stesso campione, come nel caso del test di Friedman esso verifica l’ipotesi nulla su una coppia di valori.

La metodologia risponde al confronto delle due mediane come ipotesi nulla H\_0 che può essere unilaterale o bilaterale. I passaggi per calcolare il valore W del test di Wilcoxon è la seguente:

1. Inserire come righe di una tabella il campione a cui sono associate 4 colonne; nella prima colonna viene inserito il valore della prima misura, nella seconda il valore della seconda misura, nella terza la differenza tra le precedenti misure e nell’ultima il rango corrispondente alla differenza delle misure;
2. Dopodiché vengono eliminate dall’analisi le differenze nulle; la numerosità del campione sarà proporzionalmente ridotta;
3. Trasformare le differenze considerate in valore assoluto nel loro rango. Nel caso di due o più dati uguali, assegnare lo stesso valore, calcolato come media dei ranghi;
4. Attribuire ad ogni rango il segno della differenza corrispondente
5. Sommare i ranghi con lo stesso segno;
6. Scegliere il totale minore, esso è il valore di T
7. Secondo l’ipotesi nulla H\_0, la differenza tra le due serie di osservazioni appaiate dovrebbe essere uguale a zero. Di conseguenza, nella colonna delle differenze la somma dei ranghi con segno positivo e la somma dei ranghi con segno negativo dovrebbero essere uguali. Perciò il totale minore dovrebbe tendere a un valore medio atteso mu\_t determinato da N, secondo la relazione

Mu\_t=N(N+1)/4

1. La significatività della differenza tra le due serie di dati appaiati è tradotta nella significatività della differenza tra T e mu\_t