Identifying the sources of false information in social networks

Marco Amoruso

Laurea Magistrale in Informatica
Università degli Studi di Salerno

30 settembre 2016

Relatori
Prof. Vincenzo Auletta
Dott. Diodato Ferraioli

Overview

1. Introduzione
2. Un caso di studio
3. Identificazione singola sorgente
4. Individuazione sorgenti multiple
5. Risultati
6. Conclusioni

Tag Cloud


nodes
ALGORITHMS
misinformation
Influence
attackers
SUSPECTED
ACTIVE
SET
SOURCES
SORT
PROBLEM
SOLUTION
APPROACHES
NETWORK
PROCESS
REVERSE
IMETER
RESULTS
accuracy
DETECTION
MULTIPLE
TIME
based
Suspector
probability
number
Social
SPREAD
INFECTED
TRACES
OPTIMAL
GRAPH
NEIGHBORS
FOLLOW
function
vector
originated
PROVIDES
DATA
USE
Section
content
EVALUATE
FIGURE
RATIO
experiment
CASCADE
COMPARISON
RESPECTIVELY
CLOSELY
NEW
EXTENDED
STUDY
FACEBOOK
rates
different
running
flow
makes
INCREASES
INDEPENDENT
COMPUTE
MANAGER
ESTIMATE
averaged
real
SHARING
containing
datasets
HIGHEST
SLASHDOT
WORK
IDENTIFY
report
achieves
DEFINITION
SUBMODULAR
MONTE
CARLO
returns
STARTS
LEMMA
POSSIBLE
EFFICIENT
CREATE

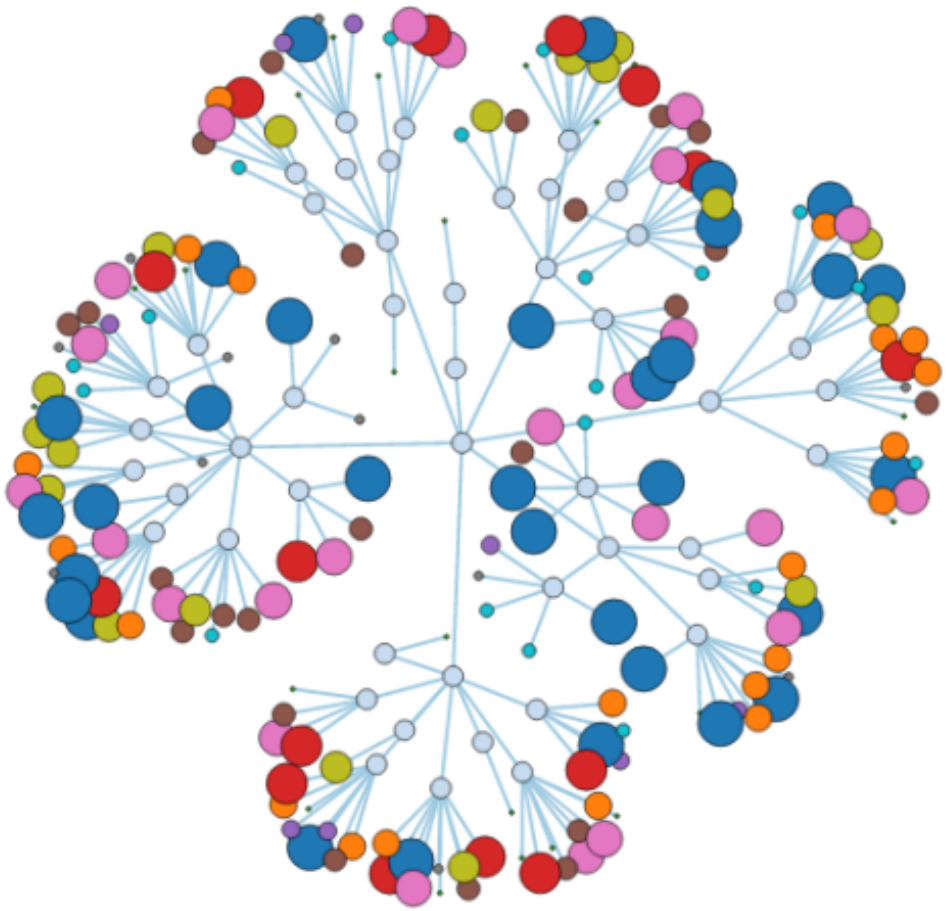
Introduzione

Una **rete sociale** consiste in un gruppo di individui connessi tra di loro da diversi legami sociali

Questi legami possono riguardare:

- Conoscenza
- Rapporti di lavoro
- Vincoli familiari
- ...

Rappresentazione grafica



Analisi

La rapida crescita delle reti sociali ha cambiato il modo in cui le persone interagiscono.

Facebook, Twitter ed altre note piattaforme online rappresentano ormai i mezzi di comunicazione più utilizzati

Problema

Diffusione di false informazioni

Analisi

La rapida crescita delle reti sociali ha cambiato il modo in cui le persone interagiscono.

Facebook, Twitter ed altre note piattaforme online rappresentano ormai i mezzi di comunicazione più utilizzati

Problema

Diffusione di false informazioni

Diffusione di false informazioni

Impatti sulla società

- Un falso comunicato stampa su Twitter nel 2013 affermava che il presidente Obama fosse stato ferito, generando l'instabilità dei mercati finanziari
- Genitori influenzati da informazioni false sui vaccini
- Cyberbullismo: Diffusione di foto spiacevoli o invio di mail contenenti materiale offensivo può costituire un danno psicologico

Diffusione di false informazioni

Risulta importante:

- Limitare tale diffusione
- Divulgare le informazioni veritiere
- Individuare i responsabili

Obiettivo

Identificare le sorgenti di false informazioni

Diffusione di false informazioni

Risulta importante:

- Limitare tale diffusione
- Divulgare le informazioni veritiere
- Individuare i responsabili

Obiettivo

Identificare le sorgenti di false informazioni

Identificare le sorgenti di false informazioni

k-Suspector problem

Reference

Nguyen DT, Nguyen NP, Thai MT. "Sources of misinformation in online social networks: Who to suspect?'.
Military Communications Conference, MILCOM 2012, IEEE.

k-Suspector problem

Idea

Analizzare a ritroso il processo di diffusione andando ad individuarne le sorgenti:

- Per ogni utente stabilire quale persona probabilmente lo abbia influenzato, capire a sua volta da chi è stato contagiato e così via (*Reverse diffusion process*)
- Classificare ogni utente in base al livello di sospetto raggiunto (*Ranking*)

Obiettivo

Individuare i k utenti più sospetti tra l'insieme dei nodi che sono state influenzati da false informazione

Algoritmo proposto: Imeter-Sort

Input

- Grafo che modella la rete di interesse
  - Il peso dell'arco indica la probabilità di trasmissione
- Insieme di nodi influenzati da false informazioni

Fasi dell'algoritmo

- *Reverse diffusion process*
- *Ranking*

Reverse diffusion process

A partire da ogni utente u viene calcolato il flusso (*reverse flow*), che ha portato la disinformazione ad u

- Un reverse flow può fermarsi al nodo u oppure avanzare ad uno dei nodi vicini v proporzionalmente all'influenza su u
- Quando un reverse flow si ferma ad un nodo diventa inattivo
- Se più reverse flow attivi giungono allo stesso nodo si fondono in un unico

Il processo termina quando non ci sono reverse flow attivi

Ranking

Osservazione

Più volte un nodo viene attraversato da un reverse flow, più è alta la probabilità che abbia diffuso false informazioni

Ogni nodo viene classificato in base al numero di volte che compare nei reverse flow

Output

Vengono selezionati i k nodi più sospetti risultanti dal processo di classificazione

Ranking

Osservazione

Più volte un nodo viene attraversato da un reverse flow, più è alta la probabilità che abbia diffuso false informazioni

Ogni nodo viene classificato in base al numero di volte che compare nei reverse flow

Output

Vengono selezionati i k nodi più sospetti risultanti dal processo di classificazione

Independent Cascade Model

La diffusione delle false informazioni viene descritta mediante l'Independent Cascade Model

- Le sorgenti da cui far partire la falsa informazione vengono scelte in maniera casuale
- Un nodo u viene influenzato da un nodo vicino v proporzionalmente alla probabilità di trasmissione $p_{v,u}$
- Se un nodo diventa "attivo" allo step t, allora nello step $t+1$ proverà ad infettare ogni vicino

Il processo, partendo dalle sorgenti malevole, continua fino a quando nessun nuovo nodo è influenzato da false informazioni

Idea

- Analizzare i legami fra gli utenti che sono stati condizionati dalle false informazioni
- Stabilire come queste si siano diffuse attraverso la rete
  - Quale utente ha influenzato i propri vicini con probabilità maggiore
- Trovare la struttura che rappresenti al meglio il processo di diffusione avvenuto

Soluzione proposta

Trattare il problema dell'identificazione di sorgenti malevole risolvendo una variante del *Maximum Spanning Tree problem*

Idea

- Analizzare i legami fra gli utenti che sono stati condizionati dalle false informazioni
- Stabilire come queste si siano diffuse attraverso la rete
  - Quale utente ha influenzato i propri vicini con probabilità maggiore
- Trovare la struttura che rappresenti al meglio il processo di diffusione avvenuto

Soluzione proposta

Trattare il problema dell'identificazione di sorgenti malevole risolvendo una variante del *Maximum Spanning Tree problem*

Concetti base

Un Tree è una struttura in cui per ogni coppia di nodi esiste un unico percorso che li collega

Uno Spanning Tree è un tree in cui sono presenti tutti i nodi della rete considerata

Maximum Spanning Tree

Spanning tree per cui la somma delle probabilità degli archi è massima

Maximum Spanning Arborescence problem

Le reti sociali considerate hanno archi direzionati, per cui si parla di *Maximum Spanning Arborescence problem*

Una Spanning arborescence di peso massimo è uno spanning tree di peso massimo, in cui è presente una radice che dà origine alla struttura

Obiettivo

Calcolare la spanning arborescence di peso massimo ed indicare la sua radice come la sorgente di false informazioni

Identificazione di una singola sorgente

L'algoritmo di **Chu-Liu/Edmonds** calcola la spanning arborescence di peso massimo

L'algoritmo si divide nelle seguenti fasi:

- *Contrazione*
- *Espansione*

Fase di contrazione

Per ogni nodo v non visitato, seleziona l'arco (u, v) di peso massimo $p_{best[v]}$

- Aggiungilo alla soluzione corrente se non genera un *ciclo*
- Altrimenti memorizza gli archi del ciclo C
- **Contrai la rete**

Crea un nuovo nodo w e contrai i nodi di C in w, $\forall$ arco (u, v):

- Se $u \in C$ aggiungi l'arco (w, v) di peso $p_{(u,v)}$
- Se $v \in C$ aggiungi l'arco (u, w) di peso $p_{(u,v)} + p_{best[v]} + p_{min}$
- Rimuovi l'arco (u, v) se u e/o v sono coinvolti nel ciclo C

Fase di contrazione

Per ogni nodo v non visitato, seleziona l'arco (u, v) di peso massimo $p_{best[v]}$

- Aggiungilo alla soluzione corrente se non genera un *ciclo*
- Altrimenti memorizza gli archi del ciclo C
- **Contrai la rete**

Crea un nuovo nodo w e contrai i nodi di C in w, $\forall$ arco (u, v):

- Se $u \in C$ aggiungi l'arco (w, v) di peso $p_{(u,v)}$
- Se $v \in C$ aggiungi l'arco (u, w) di peso $p_{(u,v)} + p_{best[v]} + p_{min}$
- Rimuovi l'arco (u, v) se u e/o v sono coinvolti nel ciclo C

Fase di espansione

Per ogni ciclo riscontrato:

- Seleziona il nodo w che corrisponde alla contrazione di C
- Aggiungi gli archi del ciclo alla soluzione
- Elimina dalla soluzione l'arco di peso minimo in grado di rompere il ciclo C

Output

La soluzione finale contiene gli archi che formano la spanning arborescence di peso massimo relativa alla rete iniziale

Identificazione di sorgenti multiple

Dopo aver sperimentato l'approccio discusso si è esteso il problema dell'individuazione di un'unica sorgente a quello di più fonti di false informazioni

Invece di calcolare una singola spanning arborescence, l'idea è quella di determinare una o più **Branching** di peso massimo

Definizione

Una *Branching* è una foresta di arborescence disgiunte

Identificazione di sorgenti multiple

Obiettivo

Indicare le *radici* delle branching calcolate come le sorgenti che hanno diffuso false informazioni

Soluzione adottata

L'algoritmo proposto da *Camerini*, *Fratta* e *Mattioli* permette di calcolare le k branching di peso massimo

K best branching

Idea

Le k branching di peso massimo differiscono tra loro di un arco

Algoritmo

- Calcolo della branching di peso massimo
- Si ricerca l'arco e che non deve far parte della seconda branching di peso massimo in favore dell'arco f
- La terza branching viene determinata tra la branching migliore che abbia l'arco e e quella che non lo contiene

Problema

Le radici delle branching calcolate generalmente non differiscono tra di loro

K best branching

Idea

Le k branching di peso massimo differiscono tra loro di un arco

Algoritmo

- Calcolo della branching di peso massimo
- Si ricerca l'arco e che non deve far parte della seconda branching di peso massimo in favore dell'arco f
- La terza branching viene determinata tra la branching migliore che abbia l'arco e e quella che non lo contiene

Problema

Le radici delle branching calcolate generalmente non differiscono tra di loro

K best branching

Idea

Le k branching di peso massimo differiscono tra loro di un arco

Algoritmo

- Calcolo della branching di peso massimo
- Si ricerca l'arco e che non deve far parte della seconda branching di peso massimo in favore dell'arco f
- La terza branching viene determinata tra la branching migliore che abbia l'arco e e quella che non lo contiene

Problema

Le radici delle branching calcolate generalmente non differiscono tra di loro

Euristica applicata: ISFI

- Si applica l'algoritmo per il calcolo della branching di peso massimo
- Le j radici vengono aggiunte alla soluzione finale
- Se $j < k$:
  - Gli archi contenenti le j radici vengono eliminati dalla rete iniziale
  - L'algoritmo viene eseguito sulla rete
- Tale processo viene ripetuto fino ad ottenere k radici

Output

La soluzione finale contiene i k nodi che vengono indicati come le sorgenti che hanno diffuso false informazioni

Caratteristiche macchina

- Sistema operativo Ubuntu 14.04
- 8 processori AMD Opteron(tm) Processor 6376, 16 core, 2.3GHz, 16MB L3 cache
- 16GB di Memoria RAM
- 180GB di Disco.

Scelte effettuate

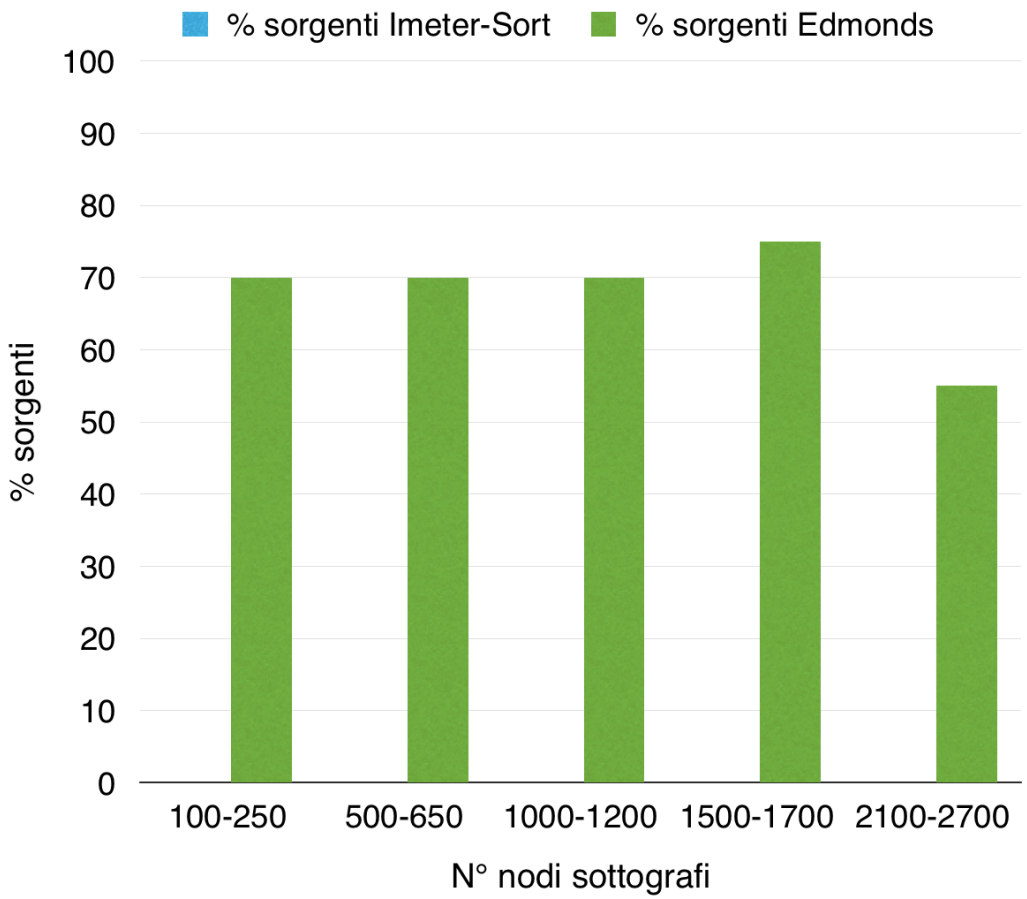
Per gli esperimenti effettuati sono stati considerati i seguenti parametri:

- Il numero di sorgenti da individuare k
- Le reti sociali di partenza:
  - *Wiki-Vote*: Votazioni per la scelta dell'admin di Wikipedia
  - *Epinions*: Recensioni di prodotti fatte dagli utenti basate su interazioni di fiducia
- I grafi derivanti dalla simulazione del processo di diffusione di false informazioni

Caratteristiche reti reali

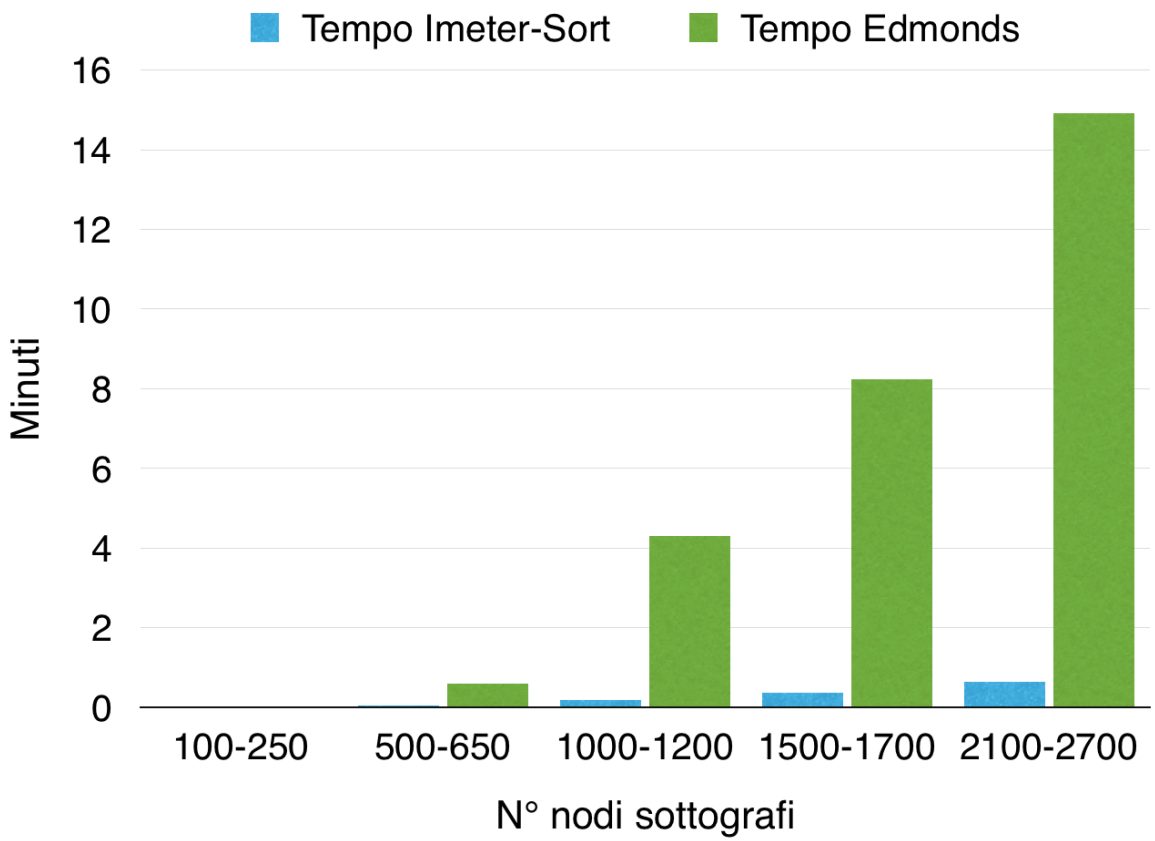
Dataset information	Wiki-Vote	Epinions
Nodi	7115	75879
Archi	103689	508837
Average Clustering	0,1409	0,1378
# Triangoli	608389	1624481
Frazione Triangoli Chiusi	0,04564	0,0229
Diametro	7	14

Edmonds vs Imeter-Sort



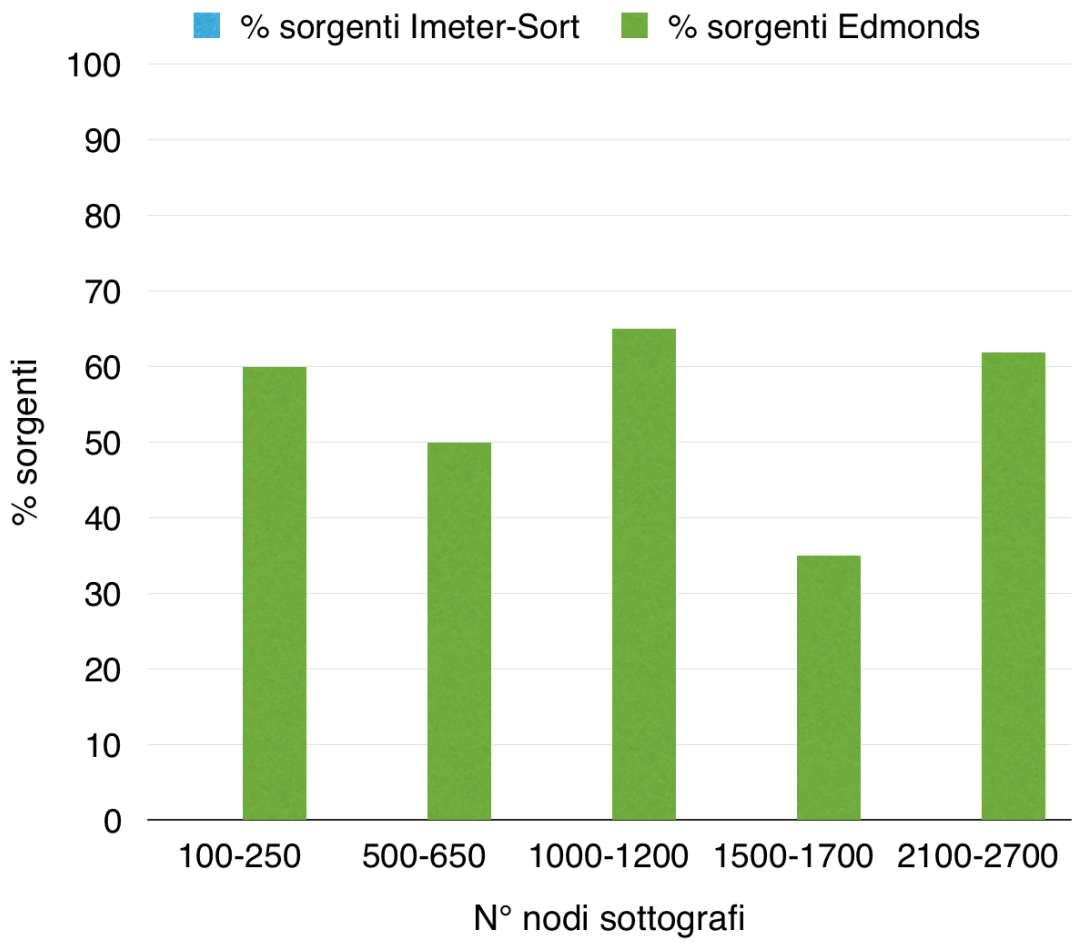
Accuratezza con $k = 1$ su **Wiki-Vote**

Edmonds vs Imeter-Sort



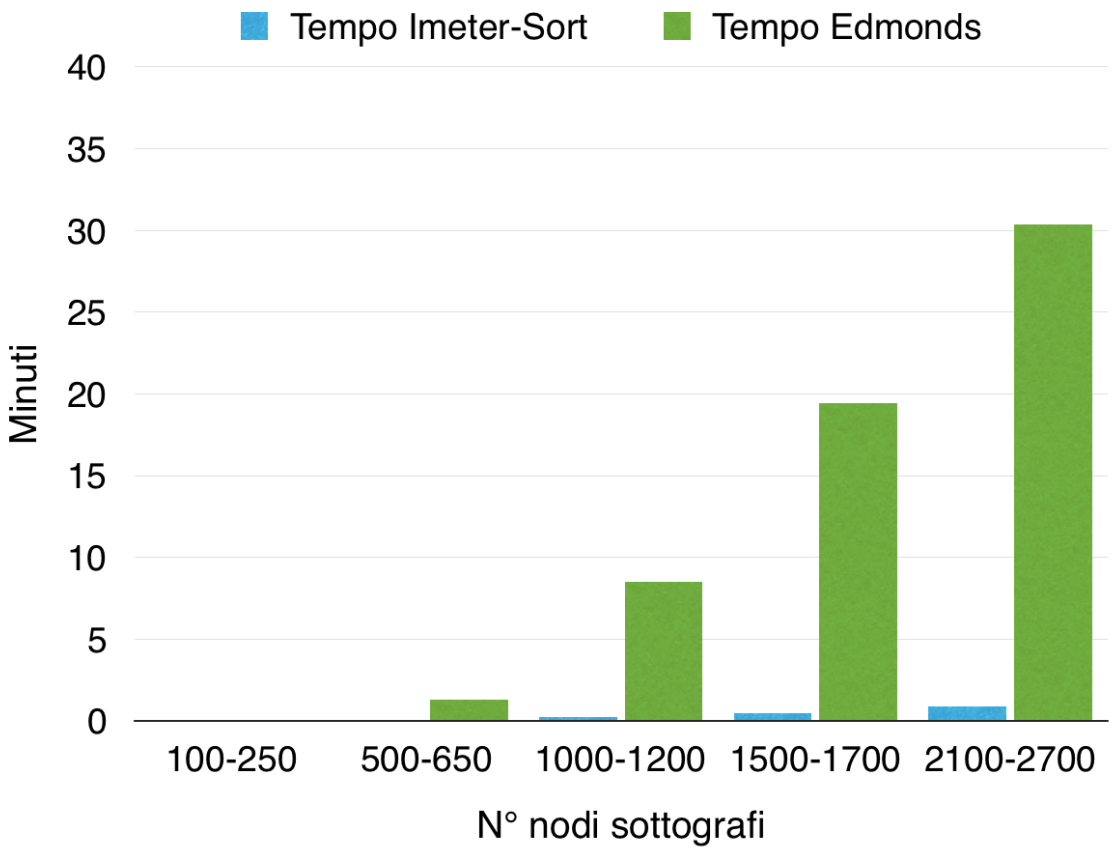
Tempo d'esecuzione con $k = 1$ su **Wiki-Vote**

Edmonds vs Imeter-Sort



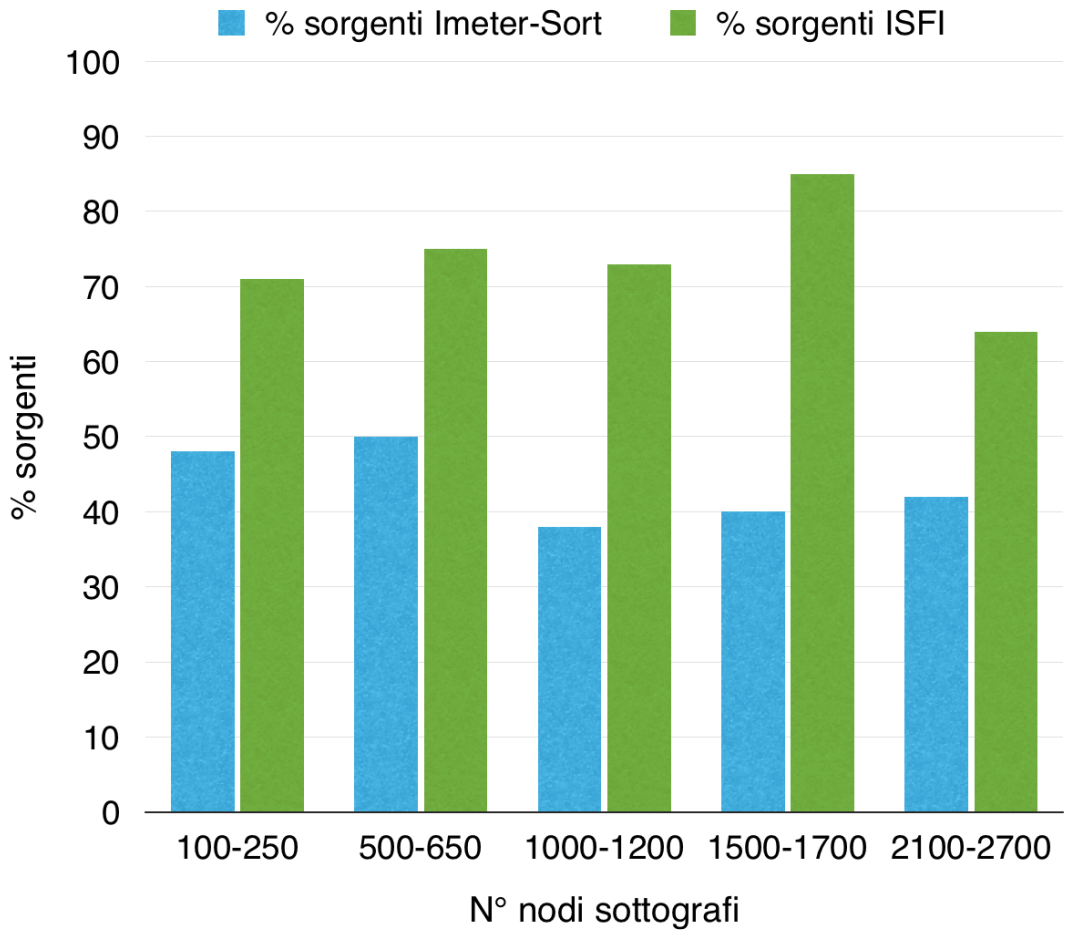
Accuratezza con $k = 1$ su **Epinions**

Edmonds vs Imeter-Sort



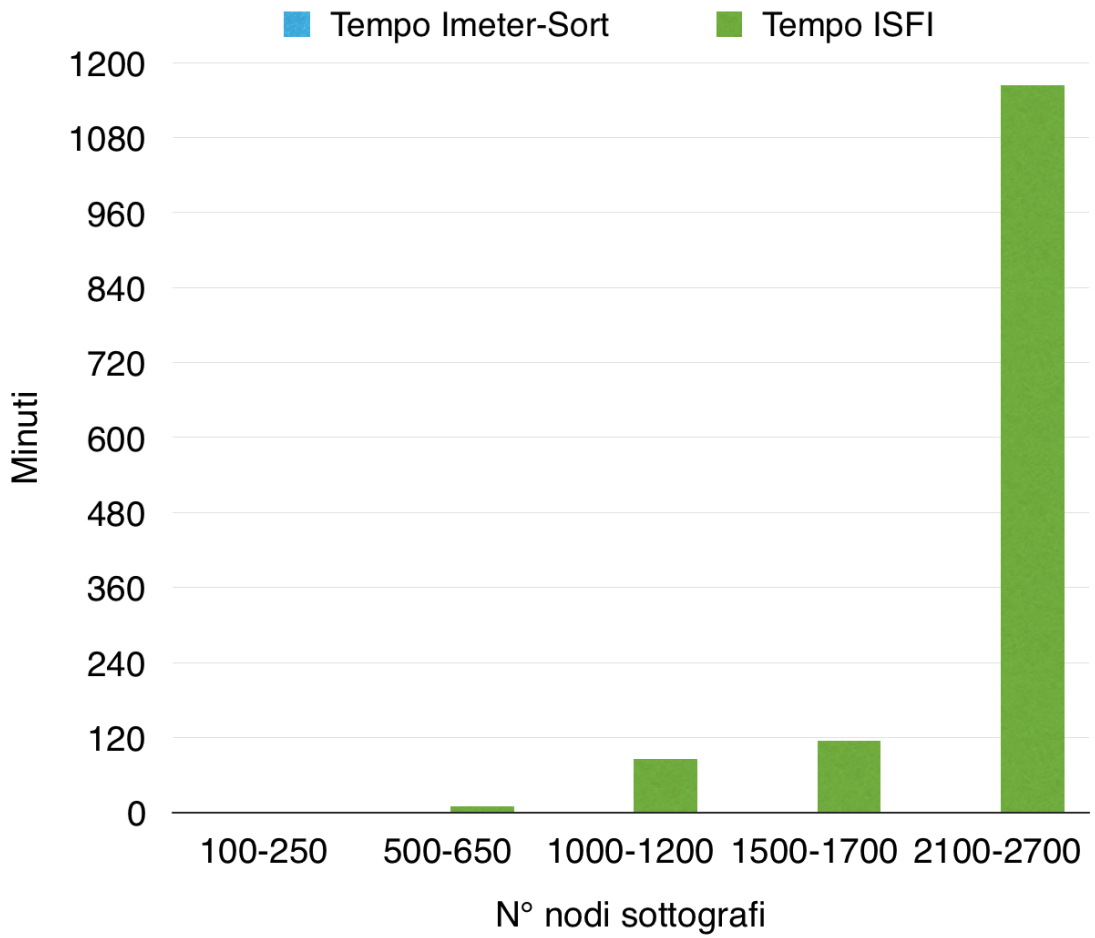
Tempo d'esecuzione con $k = 1$ su **Epinions**

ISFI vs Imeter-Sort



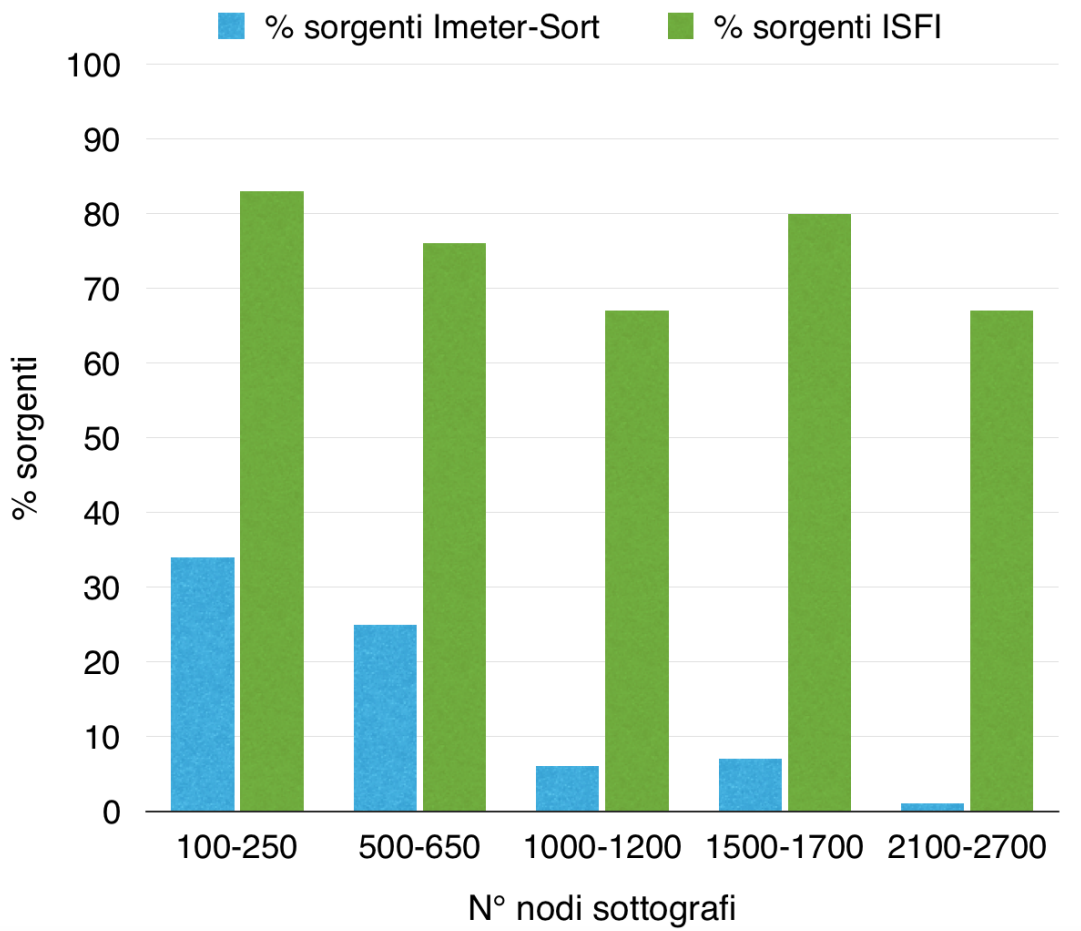
Accuratezza con $k = 4$ su **Wiki-Vote**

ISFI vs Imeter-Sort



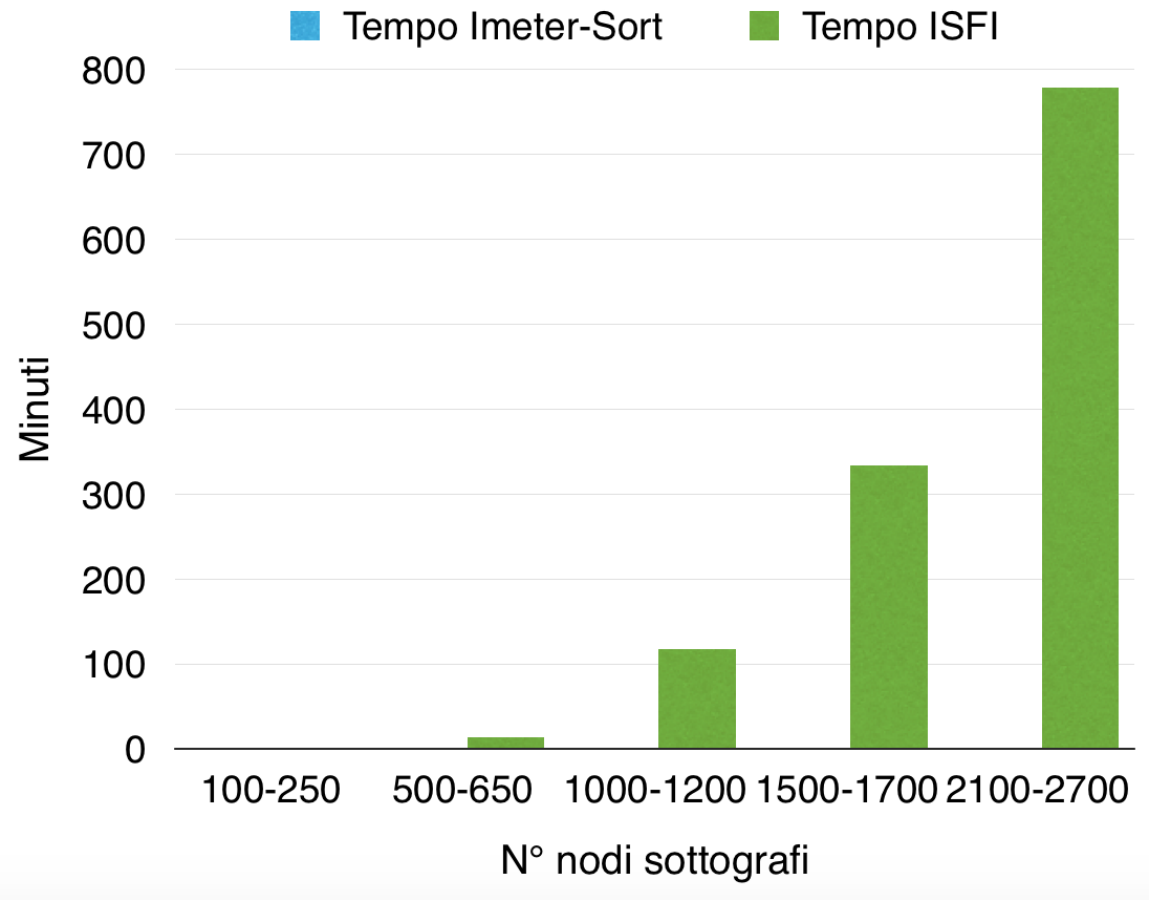
Tempo d'esecuzione con $k = 4$ su **Wiki-Vote**

ISFI vs Imeter-Sort



Accuratezza con $k = 4$ su **Epinions**

ISFI vs Imeter-Sort



Tempo d'esecuzione con $k = 4$ su **Epinions**

Conclusioni e sviluppi futuri

L'algoritmo di Edmonds e l'euristica ISFI impiegano una quantità di tempo superiore rispetto ad Imeter-Sort

Pro

In tutte le reti analizzate gli algoritmi Edmonds ed ISFI presentano un'accuratezza elevata

Sviluppi futuri

- Parallelizzare la contrazione della rete per ISFI, superando i limiti posti dal *GIL* di *Python*
- Sperimentare l'utilizzo di altre euristiche per l'individuazione di sorgenti multiple

Conclusioni e sviluppi futuri

L'algoritmo di Edmonds e l'euristica ISFI impiegano una quantità di tempo superiore rispetto ad Imeter-Sort

Pro

In tutte le reti analizzate gli algoritmi Edmonds ed ISFI presentano un'accuratezza elevata

Sviluppi futuri

- Parallelizzare la contrazione della rete per ISFI, superando i limiti posti dal *GIL* di *Python*
- Sperimentare l'utilizzo di altre euristiche per l'individuazione di sorgenti multiple

Conclusioni e sviluppi futuri

L'algoritmo di Edmonds e l'euristica ISFI impiegano una quantità di tempo superiore rispetto ad Imeter-Sort

Pro

In tutte le reti analizzate gli algoritmi Edmonds ed ISFI presentano un'accuratezza elevata

Sviluppi futuri

- Parallelizzare la contrazione della rete per ISFI, superando i limiti posti dal *GIL* di *Python*
- Sperimentare l'utilizzo di altre euristiche per l'individuazione di sorgenti multiple

Grazie per l'attenzione!