

## Porting Polya Filter

### Descrizione del Problema

La funzione Polya Filter implementa un nuovo metodo per filtrare le informazioni all'interno di reti complesse. Le reti complesse sono reti costituite da un gran numero di nodi interconnessi tra loro, come ad esempio le reti sociali o le reti di comunicazione. Gli autori della funzione propongono una metodologia di filtraggio ispirata all'urna di Pólya, il quale rappresenta un modello combinatorio guidato da un meccanismo di auto-rinforzo, che si basa su una famiglia di ipotesi nulle che possono essere calibrate per valutare quali archi sono statisticamente significativi rispetto all'eterogeneità propria di una rete.

L'urna di Pólya è un modello probabilistico in cui una serie di palline (o "urne") di colori diversi vengono estratte e rimesse in altre urne che contengono palline dello stesso colore. In questo modo, il contenuto dell'urna cambia nel tempo, riflettendo l'evoluzione delle frequenze dei colori.

Nel contesto della propagazione delle informazioni, l'urna di Pólya può essere utilizzata per modellizzare il modo in cui un'idea o un'informazione si diffonde all'interno di una rete. Ad esempio, si può immaginare che ogni nodo della rete abbia un colore associato (in base al contenuto dell'informazione) e che i nodi si scambino informazioni tra di loro, modificando così le frequenze dei colori. Gli autori della funzione utilizzano l'urna di Pólya per modellizzare la propagazione di informazioni all'interno di una rete sociale, in cui ogni nodo rappresenta un individuo e ogni arco rappresenta una connessione sociale. In particolare, il modello viene utilizzato per studiare il processo di "filtraggio" delle informazioni all'interno della rete, ovvero il modo in cui le informazioni vengono selezionate, modificate e trasmesse dai nodi della rete.

Lo studio dimostra l'efficacia del metodo attraverso una serie di esperimenti su reti complesse reali e artificiali. In particolare, si evidenzia come il metodo proposto riesca a migliorare significativamente la precisione della selezione dei nodi da cui diffondere l'informazione rispetto ad altri metodi tradizionali. Gli autori dimostrano che questo approccio può essere utilizzato per prevedere l'evoluzione della rete e identificare i nodi influenti nella diffusione delle informazioni. Inoltre, il modello può essere esteso per considerare l'influenza di nodi esterni alla rete e per prevedere l'effetto di diverse strategie di diffusione delle informazioni.

In pratica, il metodo prevede di assegnare un punteggio a ogni nodo della rete in base alla sua importanza relativa e di utilizzare questi punteggi per selezionare i nodi da cui diffondere l'informazione. Inoltre, il metodo tiene conto del fatto che l'importanza dei nodi può variare nel tempo, ad esempio a seguito di nuove informazioni che emergono nella rete.

### Descrizione codice

La funzione prende in input una matrice di adiacenza "W" che rappresenta la rete, un parametro "a" libero del filtro, un livello di approssimazione "apr\_lvl" e un flag "parallel" per abilitare o disabilitare l'elaborazione parallela.

Il parametro "a" si riferisce alla probabilità che un nodo della rete decida di seguire un'altra fonte di informazione invece di continuare a seguire la sua attuale fonte. In particolare, il modello considera un'urna di Pólya contenente biglie di due colori, dove il colore rosso rappresenta la fonte di informazione corrente del nodo e il colore nero rappresenta una nuova fonte di informazione potenziale. Il parametro "a" specifica la probabilità che un nodo decida di prendere una biglia nera invece di una rossa dalla urna, cioè la probabilità che un nodo cambi la sua attuale fonte di informazione. Questo parametro è importante perché influenza la dinamica del modello e la velocità di diffusione delle informazioni nella rete. Ad esempio, se il valore di "a" è molto basso, i nodi avranno una maggiore probabilità di rimanere fedeli alla loro attuale fonte di informazione, rallentando la diffusione delle nuove informazioni nella rete. Al contrario, se il valore di "a" è molto alto, i nodi saranno più propensi a cambiare la loro fonte di informazione, accelerando la diffusione delle informazioni nella rete.

Il codice esegue diverse operazioni per calcolare i p-value. In primo luogo, controlla se la rete è simmetrica, estrae gli archi  $(i, j, w)$  dalla matrice di adiacenza W in base alla simmetria e ottiene l'elenco degli archi. Inoltre, calcola i gradi  $(k_{in}, k_{out})$  e le forze  $(s_{in}, s_{out})$  di ogni nodo utilizzando la forma asintotica se sono presenti pesi non interi.

Infine, la funzione calcola i p-value per ogni arco  $(i, j, w)$  utilizzando la funzione `polya_cdf` e restituisce il valore minimo di p, essa implementa la distribuzione cumulativa della Polya. Questa funzione calcola i p-value che possono essere approssimati e quelli che non possono essere approssimati utilizzando due forme diverse a seconda del valore di a: se  $a = 0$  usa la distribuzione binomiale altrimenti usa un'approssimazione asintotica.

La funzione restituisce un vettore di valori "P" che rappresentano i p-value prescritti dal filtro di Polya per ciascun link della rete

### Come utilizzarlo

Aprire la cartella "idmdefinitivo" e avviare il progetto in R.

Librerie utilizzate :

igraph

mvtnorm

Matrix

## Esempio di prova con grafo creato a mano

Creazione grafo vuoto

```
g <- make_empty_graph()
```

aggiunta dei nodi e degli archi

```
g <- add_vertices(g, 5)
```

```
g <- add_edges(g, c(1,2, 1,3, 2,4, 3,4, 4,5),attr=list(weight=c(0.5, 0.8, 0.2, 1.0, 0.6)))
```

chiamata delle funzione

```
PF(adj_matrixg,4.5,2,FALSE)
```

## Esempio di prova con grafo acquisito da Network Repository

grafo contenuto nel file bio-CE-CX.edges

lettura da file e creazione struttura grafo

```
edge_list <- read.csv("bio-CE-CX.edges", header = FALSE, sep = " ")
```

```
graph <- graph_from_data_frame(d = edge_list, directed = FALSE, vertices = NULL)
```

creazione matrice di adiacenza

```
adj_matrix <- as_adjacency_matrix(graph, type = "both",sparse = FALSE)
```

chiamata della funzione PF

```
PF(adj_matrix,4.5,2,FALSE)
```