Progetto: Majority dynamics in networks

Majority domination in networks

Data una network G=(V,E), per ogni vertice $v \in V$, denotiamo con

- $N(v)$ l'insieme degli adiacenti di v in G
- $d(v)$ il grado di v in G, cioè $d(v) = |N(v)|$

Un dominating set D di G è un sottinsieme dei vertici di G tale che

per ogni $v \in V - D$ vale che $|N(v) \cap D| \geq 1$

Un majority dominating set D di G è un sottinsieme dei vertici di G tale che

per ogni $v \in V - D$ vale che $|N(v) \cap D| \geq \left\lceil \frac{d(v)}{2} \right\rceil$

L'obiettivo della dominazione è determinare un dominating set D di taglia minima che domina tutto il grafo G

Majority domination in networks

Il problema di determinare un dominating set (o anche un majority dominating set) per G è NP-hard.

Molti algoritmi di approssimazione ed euristiche sono note per la soluzione del problema

Algorithm 1: Majority Domination Greedy(G, f)

$D = \emptyset$.
while *there exists* $v \in V - D$ *such that* $\Delta_v f(D) > 0$ **do**
  select $\arg\max_{v \in V-D} \Delta_v f(D)$
  $D = D \cup \{u\}$
return D

where $\Delta_v f(D) = f(D \cup \{v\}) - f(D)$ and, $f : 2^V \to R^+$ may be one of the following submodular functions:

$$f_1(D) = \sum_{v \in V} \min \left\{ |N(v) \cap D|, \ \lceil \frac{d(v)}{2} \rceil \right\}$$

$$f_2(D) = \sum_{v \in V} \sum_{i=1}^{|N(v) \cap D|} \max \left\{ \lceil \frac{d(v)}{2} \rceil - i + 1, \ 0 \right\}$$

$$f_3(D) = \sum_{v \in V} \sum_{i=1}^{|N(v) \cap D|} \max \left\{ \frac{\lceil \frac{d(v)}{2} \rceil - i + 1}{d(v) - i + 1}, \ 0 \right\}$$

Majority cascade in networks

Data una network G=(V,E),

- un seed set S⊆ V,
- un Majority dynamical process di Influence Diffusion on G è definito come una sequenza di sottinsiemi:

$$\text{Inf}[\boldsymbol{S},\mathbf{0}],\ \text{Inf}[\boldsymbol{S},\mathbf{1}],\ \ldots,\ \text{Inf}[\boldsymbol{S},\boldsymbol{r}],\ \ldots \subseteq \boldsymbol{V}$$

dove

- Inf[$\boldsymbol{S}$,$\mathbf{0}$]=$\boldsymbol{S}$
- Inf[$\boldsymbol{S}$,$\boldsymbol{r}$]=Inf[$\boldsymbol{S}$,$\boldsymbol{r}-\mathbf{1}$] ∪

$$\{\boldsymbol{v} \in V - \text{Inf}[\boldsymbol{S},\boldsymbol{r}-\mathbf{1}] : |N(\boldsymbol{v}) \cap \text{Inf}[\boldsymbol{S},\boldsymbol{r}-\mathbf{1}]| \geq \lceil d(\boldsymbol{v})/2 \rceil\}$$

- Il più piccolo t tale che Inf[$\boldsymbol{S}$,t]= Inf[$\boldsymbol{S}$,t+1] è l'istante in cui si ferma la cascade.
  - Inf[$\boldsymbol{S}$] =Inf[$\boldsymbol{S}$,t]

Majority cascade in networks

Data una network $G=(V,E)$,

- un seed set $S \subseteq V$,
- un Majority dynamical process di Influence Diffusion on G è definito come una sequenza di sottinsiemi:

$$\text{Inf}[\boldsymbol{S},\mathbf{0}],\ \text{Inf}[\boldsymbol{S},\mathbf{1}],\ \ldots,\ \text{Inf}[\boldsymbol{S},\boldsymbol{r}],\ \ldots \subseteq \boldsymbol{V}$$

  dove

  - $\text{Inf}[\boldsymbol{S},\mathbf{0}]=\boldsymbol{S}$
  - $\text{Inf}[\boldsymbol{S},\boldsymbol{r}]=\text{Inf}[\boldsymbol{S},\boldsymbol{r}-\mathbf{1}] \cup$

$$\{\boldsymbol{v} \in V - \text{Inf}[\boldsymbol{S},\boldsymbol{r}-\mathbf{1}] : |N(\boldsymbol{v}) \cap \text{Inf}[\boldsymbol{S},\boldsymbol{r}-\mathbf{1}]| \geq \lceil d(\boldsymbol{v})/2 \rceil\}$$

- Il più piccolo t tale che $\text{Inf}[\boldsymbol{S},t]=\text{Inf}[\boldsymbol{S},t+1]$ è l'istante in cui si ferma la cascade.
  - $\text{Inf}[\boldsymbol{S}] = \text{Inf}[\boldsymbol{S},t]$

Si noti che il fatto che i nodi attivati nello step $\boldsymbol{r}$ sono scelti in $V-\text{Inf}[\boldsymbol{S},\boldsymbol{r}-\mathbf{1}]$ significa che se un nodo si attiva, esso rimarrà attivato per sempre

Majority cascade in cost networks

Data una network $G=(V,E\ )$ e costi associati ai nodi della rete $c\colon V\ {-}{>}\ N$

- un seed set $S\subseteq V$ produce un Majority dynamical process di Influence Diffusion on G

$$\mathrm{Inf}[\boldsymbol{S},\mathbf{0}],\ \mathrm{Inf}[\boldsymbol{S},\mathbf{1}],\ \ldots,\ \mathrm{Inf}[\boldsymbol{S},\boldsymbol{r}],\ \ldots \subseteq \boldsymbol{V}$$

dove t è Il più piccolo l'istante in cui si ferma la cascade, $\mathrm{Inf}[\boldsymbol{S}] = \mathrm{Inf}[\boldsymbol{S},t]$

- Il costo del seed set S è definito $c(S) = \sum_{u\in S} c(u)$
- Dato un badget k, l'obiettivo è determinare un seed set S tale che $c(S) \le k$ che massimizzi $|\mathrm{Inf}[\boldsymbol{S}]|$

Majority cascade in cost networks

Data una network $G=(V,E\ )$ e costi associati ai nodi della rete $c\colon V\ {-}{>}\ N$

- Il problema di determinare il seed set S con $c(S) \leq k$ che massimizzi $|\mathrm{Inf}[\boldsymbol{S}]|$ è NP-hard ed è hard da approssimare, visto che esso nel caso in cui tutti costi sono $c(u) = 1$ per ciascun vertice $u \in V$ è NP-hard ed è hard da approssimare

- Quindi l'unica possibilità è determinare delle euristiche

Considereremo algoritmi che determinano un seed set S massimale cioè al più grande seed set S tale $c(S) \leq k$ e valuteremo il numero di vertici che tale seed set S è capace di influenzare, cioè $|\mathrm{Inf}[\boldsymbol{S}]|$.

Majority cascade in cost networks

Considereremo algoritmi che determinano un seed set S massimale cioè al più grande seed set S tale $c(S) \leq k$ e valuteremo il numero di vertici che tale seed set S è capace di influenzare, cioè $|\mathrm{Inf}[\boldsymbol{S}]|$.

Individuare un seed set massimale per cost majority cascade: Algorithm 1

Fissato un intero k, una funzione costi $c: V \rightarrow N$, una funzione f_i per $i = 1,2,3$, determinare un seed set massimale di $G=(V,E)$

Algorithm Cost-Seeds-Greedy (G, k, c, f_i)

$S_p = S_d = \emptyset$.
repeat
  select $u = \arg\max_{v \in V - S_d} \frac{\Delta_v f_i(S_d)}{c(u)}$
  $S_p = S_d$
  $S_d = S_p \cup \{u\}$
until $c(S_d) > k$
return S_p

$$f_1(S) = \sum_{v \in V} \min\left\{|N(v) \cap S|,\ \left\lceil \frac{d(v)}{2} \right\rceil\right\}$$

$$f_2(S) = \sum_{v \in V} \sum_{i=1}^{|N(v) \cap S|} \max\left\{\left\lceil \frac{d(v)}{2} \right\rceil - i + 1,\ 0\right\}$$

$$f_3(S) = \sum_{v \in V} \sum_{i=1}^{|N(v) \cap S|} \max\left\{\frac{\left\lceil \frac{d(v)}{2} \right\rceil - i + 1}{d(v) - i + 1},\ 0\right\}$$

Individuare un seed set massimale per cost majority cascade: Algorithm 2

Fissato k e prendendo come threshold $t(u) = \left\lceil \frac{d(u)}{2} \right\rceil$ e una funzione costi $c: V \text{—>} N$, fermare l'algoritmo WTSS presente nella slide successiva al seed set S massimale cioè al più grande seed set S tale $c(S) \leq k$

Algorithm WTSS(G)

Input: A graph $G = (V, E)$ with thresholds $t(v)$ and costs $c(v)$, for $v \in V$.

Output: A target set S for G.

$S = \emptyset$; $\quad U = V$

for each $v \in V$ **do** $\{ \delta(v) = d_G(v); \quad k(v) = t(v); \quad N(v) = \Gamma_G(v) \}$

while $U \neq \emptyset$ **do**

    [Select one vertex and eliminate it from the graph as specified in the following cases]

    if there exists $v \in U$ s.t. $k(v) = 0$ **then**

        [Case 1: The selected vertex v is activated by the influence of its neighbors in $V - U$ only; it can then influence its neighbors in U]

        for each $u \in N(v)$ **do** $k(u) = \max\{0, k(u) - 1\}$

    else

        if there exists $v \in U$ s.t. $\delta(v) < k(v)$ **then**

            [Case 2: The vertex v is added to S, since no sufficient neighbors remain in U to activate it; v can then influence its neighbors in U]

            $S = S \cup \{v\}$

            for each $u \in N(v)$ **do** $k(u) = k(u) - 1$

        else *[Case 3: The selected vertex v will be activated by its neighbors in U]*

            $v = \mathtt{argmax}_{u \in U} \left\{ \frac{c(u)\, k(u)}{\delta(u)(\delta(u)+1)} \right\}$

    [Remove the selected vertex v from the graph]

    for each $u \in N(v)$ **do** $\{ \delta(u) = \delta(u) - 1; \quad N(u) = N(u) - \{v\} \}$

    $U = U - \{v\}$

Individuare un seed set massimale per cost majority cascade: Algorithm 3

Fissato un intero k, una funzione costi $c: V \rightarrow N$, individuare un algoritmo che si ritiene interessante per determinare un seed set massimale S di $G=(V,E)$

Algorithm 5: My-Seeds (G, k)

Modello di attivazione con majority e Progetto

Scelto un grafo G=(V,E), un intero k, una funzione costi $c: V \rightarrow N$

1. Si applica Algorithm ?? per determinare il seed set massimale S con $c(S) \leq k$
2. Si fa girare il processo di attivazione e si determina l'insieme dei nodi attivati Inf[S]

- Rappresentare con grafici i risultati ottenuti
  - Per ogni fissata funzione costi $c: V \rightarrow N$
  - al variare del costo badget k del seed set S devono essere rappresentati i valori di |Inf[S]| per diversi algoritmi di selezione del seed set S

Modello di attivazione con majority e Progetto

- Gli algoritmi da utilizzare devono essere almeno due di quelli proposti + un altro da voi ideato e che pensate possa essere interessante per il problema trattato

- Le funzione costi $c: V \rightarrow N$ da utilizzare devono essere tre per ogni $u \in V$:
  - $c(u) = valore\ random$ scelto in un fissato range
  - $c(u) = \left\lceil \frac{d(u)}{2} \right\rceil$
  - Scegliete voi una funzione costi che ritenete adatta

Relazione finale

Scrivere una relazione in cui si descrive

- il problema
- la rete scelta (motivando la scelta e descrivendone le caratteristiche)
- I risultati rappresentati attraverso grafici che mostrano,
  - Per ogni fissata funzione costi,
    - per ciascuno dei tre algoritmi scelti, al variare del budget k del seed set S, i valori |Inf[S]| ottenuti
- conclusioni