
Propagation Model 1: Independent Cascade (G, S, R, P)

```
1 totalSpread  $\leftarrow$  0
2 for iter = 1 to R do
3   tried  $\leftarrow$   $\emptyset$ 
4   newNodes  $\leftarrow$  S
5   while newNodes  $\neq$   $\emptyset$  do
6     currentNodes  $\leftarrow$  newNodes
7     newNodes  $\leftarrow$   $\emptyset$ 
8     for each node x :  $x \in$  newNodes do
9       for each node y :  $y \in N_x$  and  $y \notin S$  do
10         if Random(0, 1) < P then
11           newNodes += node y
12         tried += (node x, node y)
13           // (line 12 alternative: tried += node y)
13       S = S  $\cup$  currentNodes  $\cup$  newNodes
14   totalSpread += Len(S)
15 return (totalSpread / R)
```

Propagation Model 2: Weighted Cascade 1 (G, S, R)

```
1 totalSpread  $\leftarrow$  0
2 for  $iter = 1$  to  $R$  do
3   tried  $\leftarrow \emptyset$ 
4   newNodes  $\leftarrow S$ 
5   while newNodes  $\neq \emptyset$  do
6     currentNodes  $\leftarrow$  newNodes
7     newNodes  $\leftarrow \emptyset$ 
8     for each node  $x : x \in newNodes$  do
9       for each node  $y : y \in N_x$  and  $y \notin S$  do
10         if  $Random(0, 1) < (1 / in-degree(y))$  then
11           newNodes += node  $y$ 
12           tried += (node  $x$ , node  $y$ )
13           // (line 12 alternative: tried += node  $y$ )
14   S =  $S \cup currentNodes \cup newNodes$ 
15 totalSpread += Len( $S$ )
16 return (totalSpread /  $R$ )
```

Propagation Model 3: Weighted Cascade 2 (G, S, R)

```
1 totalSpread  $\leftarrow$  0
2 for iter = 1 to R do
3   tried  $\leftarrow$   $\emptyset$ 
4   newNodes  $\leftarrow$  S
5   while newNodes  $\neq$   $\emptyset$  do
6     currentNodes  $\leftarrow$  newNodes
7     newNodes  $\leftarrow$   $\emptyset$ 
8     for each node x :  $x \in$  newNodes do
9       for each node y :  $y \in N_x$  and  $y \notin S$  do
10         SND  $\leftarrow$  0
11         for each node z :  $z \in N_y$  do
12           SND += out-degree(z)
13         if Random(0, 1) < (out-degree(x) / SND) then
14           newNodes += node y
15         tried += (node x, node y)
16         // (line 12 alternative: tried += node y)
17     S = S  $\cup$  currentNodes  $\cup$  newNodes
18   totalSpread += Len(S)
19 return (totalSpread / R)
```
