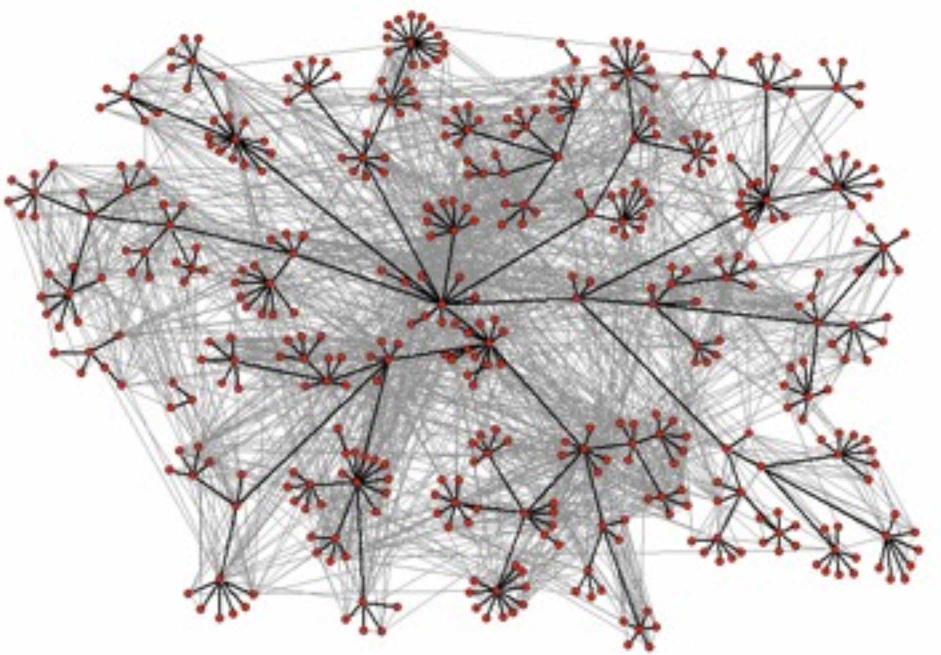
Epidemics and Information Propagation in Social Networks



Epidemic Processes

- Viruses, diseases
- Online viruses, worms
- Fashion
- Adoption of technologies
- Behavior
- Ideas

Example: Ebola virus

- First emerged in Zaire 1976 (now Democratic Republic of Kongo)
- Very lethal: it can kill somebody within a few days
- A small outbreak in 2000
- From 10/2000 – 01/2009 173 people died in African villages

Example: HIV

- Less lethal than Ebola
- Takes time to act, lots of time to infect
- First appeared in the 70s
- Initially confined in special groups: homosexual men, drug users, prostitutes
- Eventually escaped to the entire population

Example: Melissa computer worm

- Started on March 1999
- Infected MS Outlook users
- The user
  - Receives email with a word document with a virus
  - Once opened, the virus sends itself to the first 50 users in the outlook address book
- First detected on Friday, March 26
- On Monday had infected >100K computers

Example: Hotmail

- Example of Viral Marketing: Hotmail.com

- Jul 1996: Hotmail.com started service
- Aug 1996: 20K subscribers
- Dec 1996: 100K
- Jan 1997: 1 million
- Jul 1998: 12 million

Bought by Microsoft for $400 million

Marketing: At the end of each email sent there was
a message to subscribe to Hotmail.com
“Get your free email at Hotmail"

The Bass model

- Introduced in the 60s to describe product adoption
- Can be applied for viruses
- No network structure

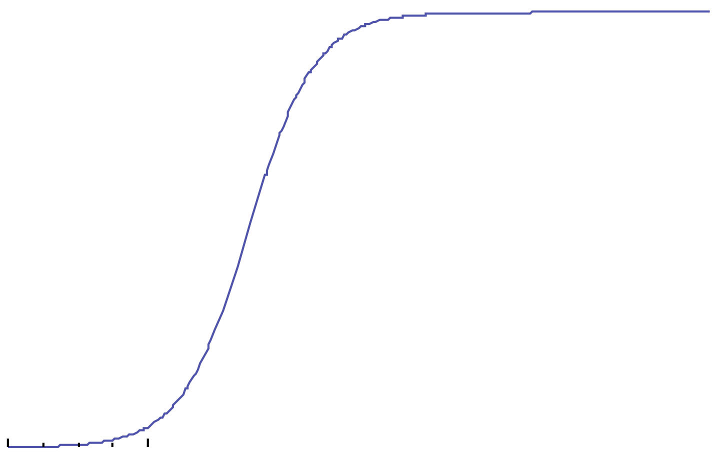
$$F(t+1) = F(t) + p(1 - F(t)) + q(1 - F(t))F(t)$$

- F(t): Ratio of infected at time t
- p: Rate of infection by outside
- q: Rate of contagion

The Bass model

- F(t): Ratio of infected at time t
- p: Rate of infection by outside
- q: Rate of contagion

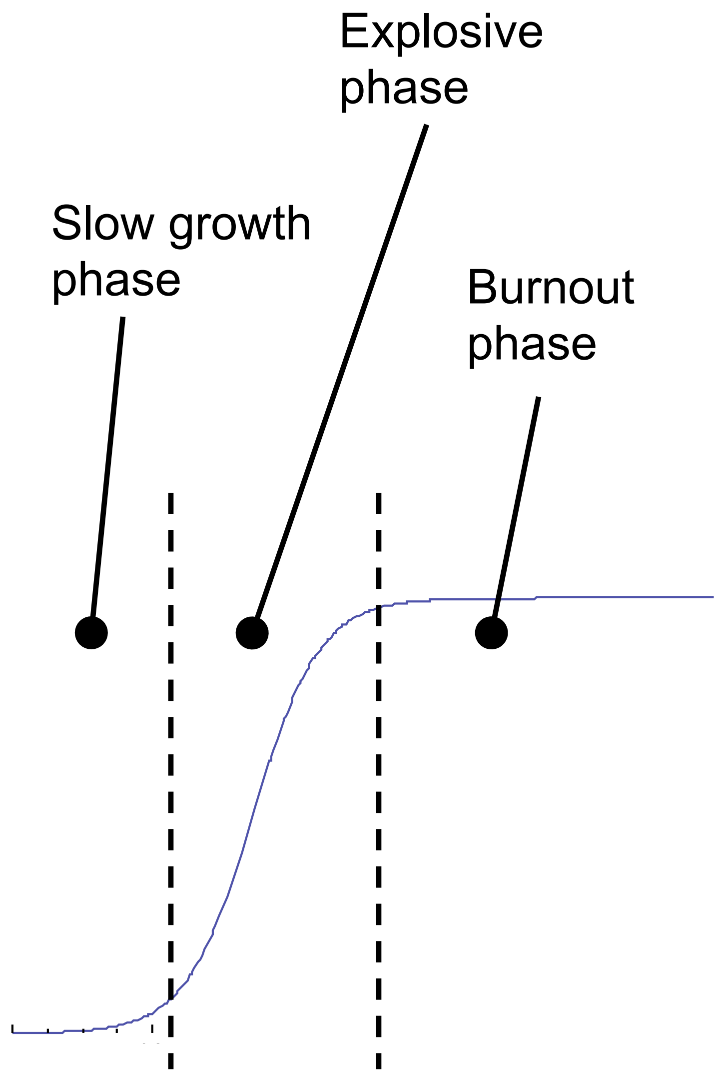
$$F(t) = \frac{1 - e^{-(p+q)t}}{1 + \frac{q}{p} e^{-(p+q)t}}$$



The Bass model

- F(t): Ratio of infected at time t
- p: Rate of infection by outside
- q: Rate of contagion

$$F(t) = \frac{1 - e^{-(p+q)t}}{1 + \frac{q}{p} e^{-(p+q)t}}$$

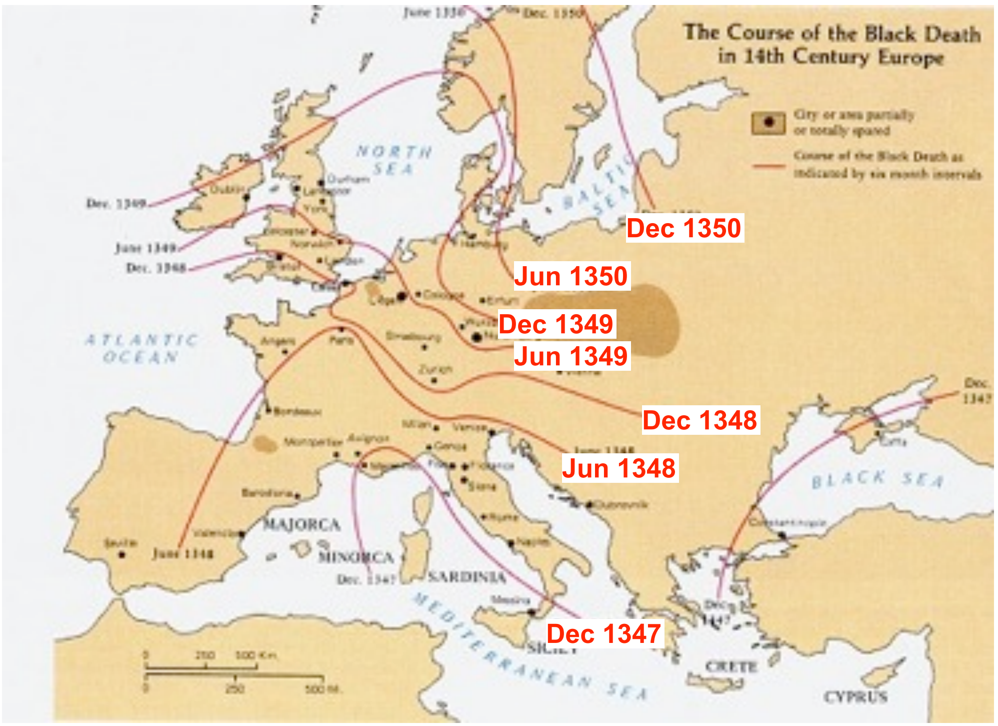


Network Structure

- The Bass model does not take into account network structure
- Let’s see some examples

Example: Black Death (Plague)

- Started in 1347 in a village in South Italy from a ship that arrived from China
- Propagated through rats, etc.



Example: Mad-cow disease

- Jan. 2001: First cases observed in UK
- Feb. 2001: 43 farms infected
- Sep. 2001: 9000 farms infected

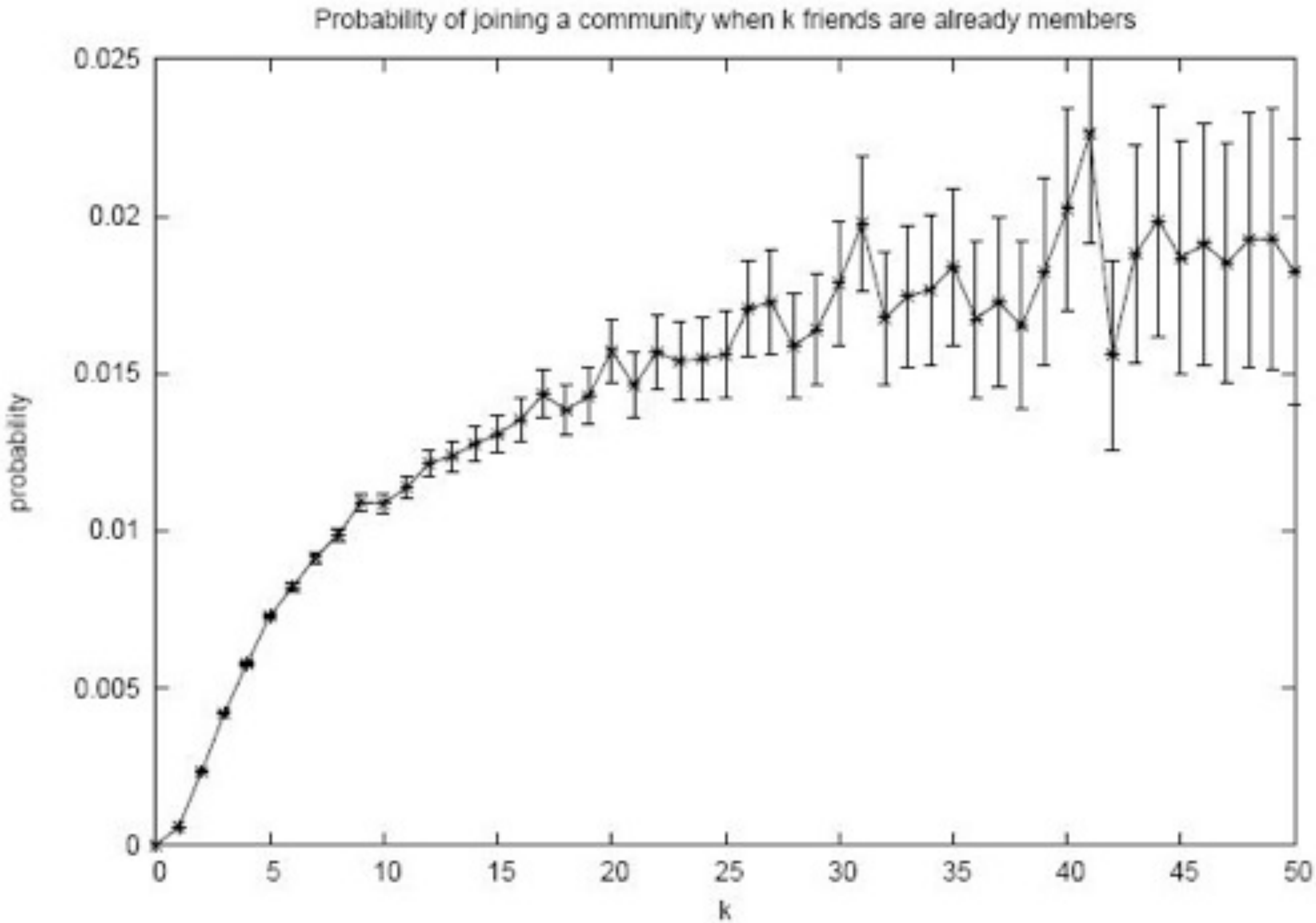
- Measures to stop: Banned movement, killed millions of animals

Network Impact

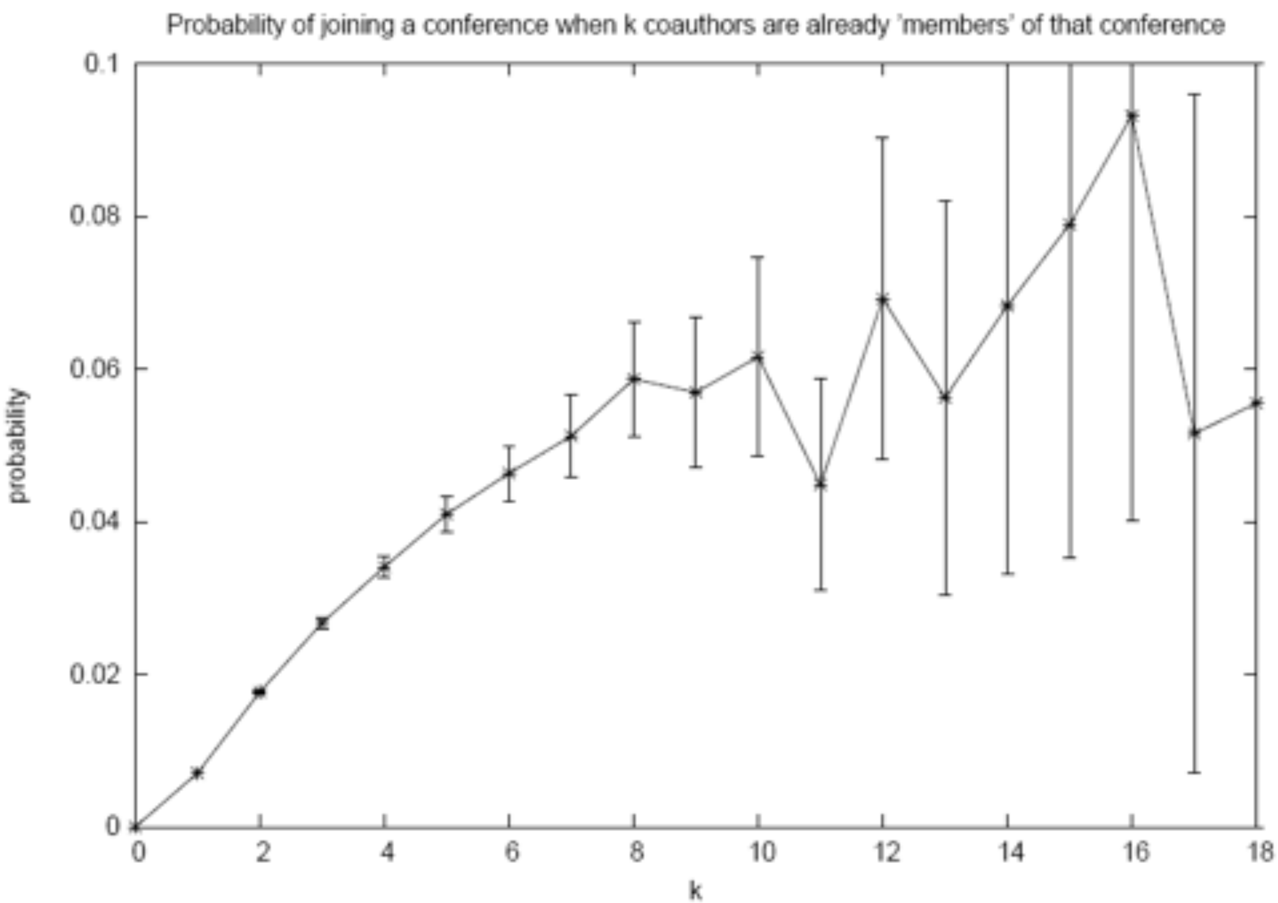
- In the case of the plague it is like moving in a lattice
- In the mad cow we have weak ties, so we have a small world
  - Animals being bought and sold
  - Soil from tourists, etc.

- To protect:
  - Make contagion harder
  - Remove weak ties (e.g., mad cows, HIV)

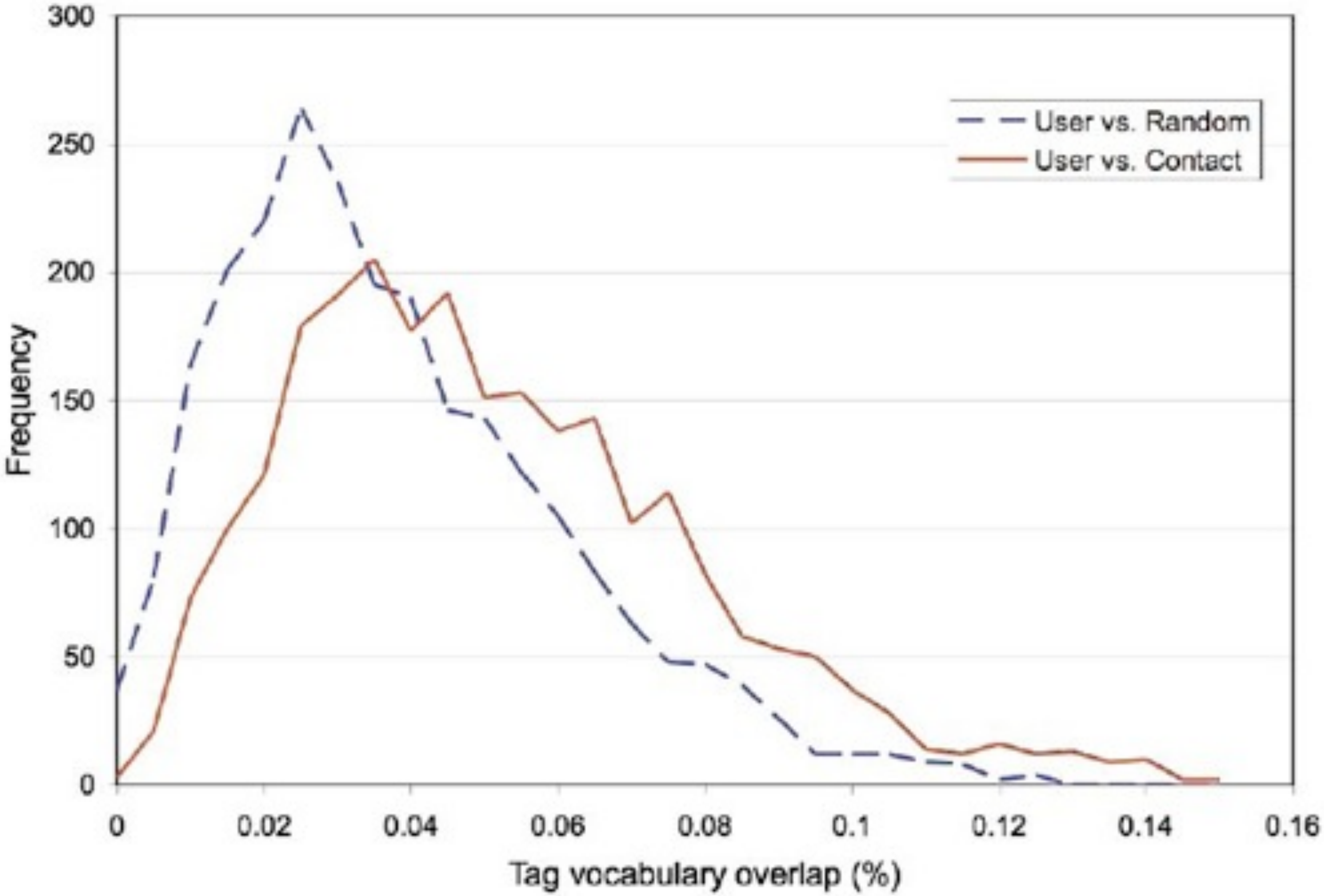
Example: Join an online group



Example: Publish in a conference



Example: Use the same tag



Models of Influence

- We saw that often decision is correlated with the number/fraction of friends
- This suggests that there might be influence: the more the number of friends, the higher the influence
- Models to capture that behavior:
  – Linear threshold model
  – Independent cascade model

Linear Threshold Model

$$\theta_v \sim U[0,1]$$

$$\sum_{w \in N(v)} b_{vw} \leq 1$$

$$\sum_{w \in N(v) \text{ and } w \text{ is active}} b_{vw} \geq \theta_v$$

Linear Threshold Model

- A node *v* has threshold $\theta_v \sim U[0,1]$
- A node *v* is influenced by each neighbor *w* according to a *weight* b_{vw} such that

$$\sum_{w \in N(v)} b_{vw} \leq 1$$

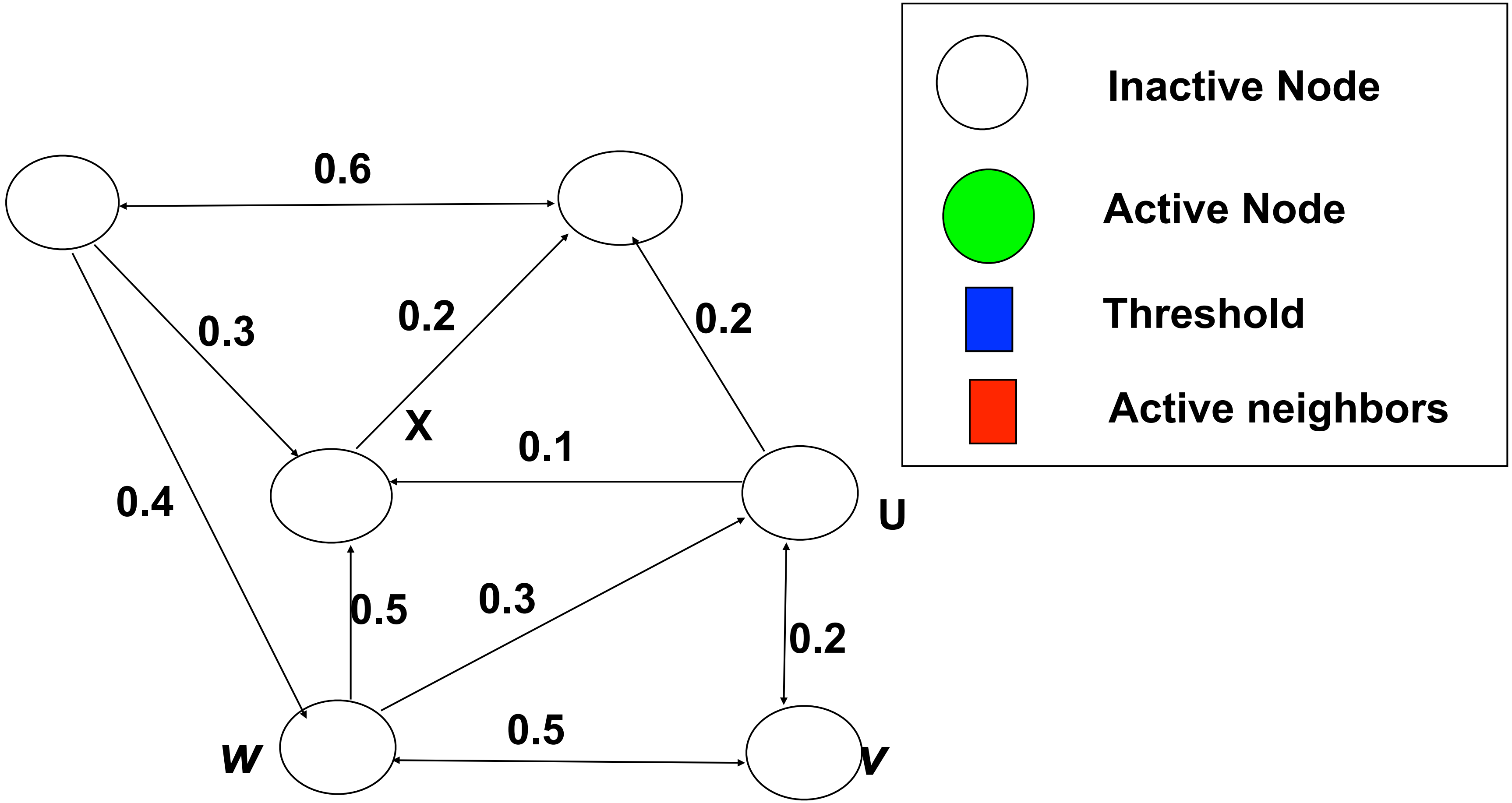
- A node *v* becomes **active** when at least

(weighted) θ_v fraction of its neighbors are **active**

$$\sum_{w \in N(v) \text{ and } w \text{ is active}} b_{vw} \geq \theta_v$$

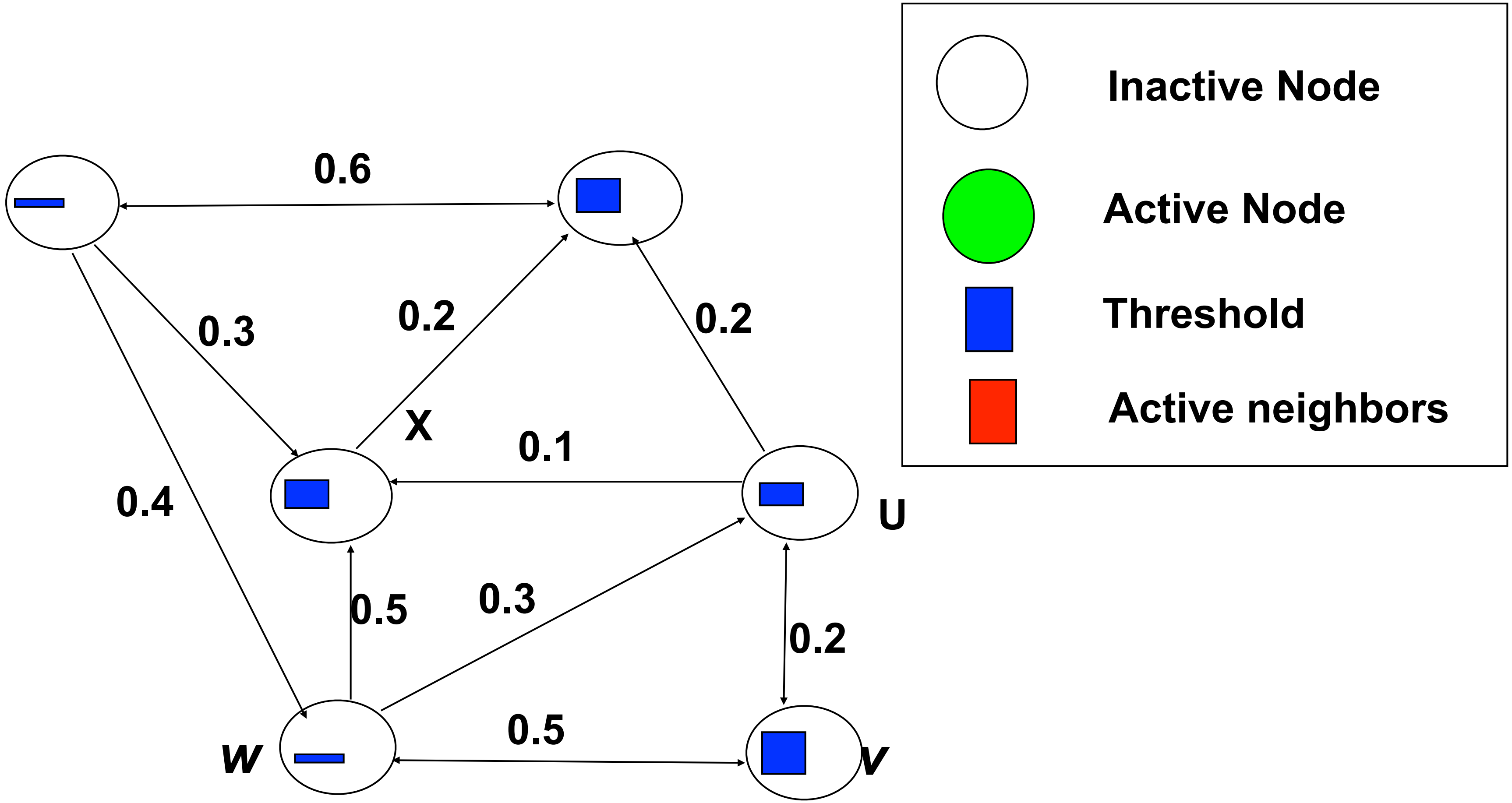
Examples: riots, mobile phone networks

Example

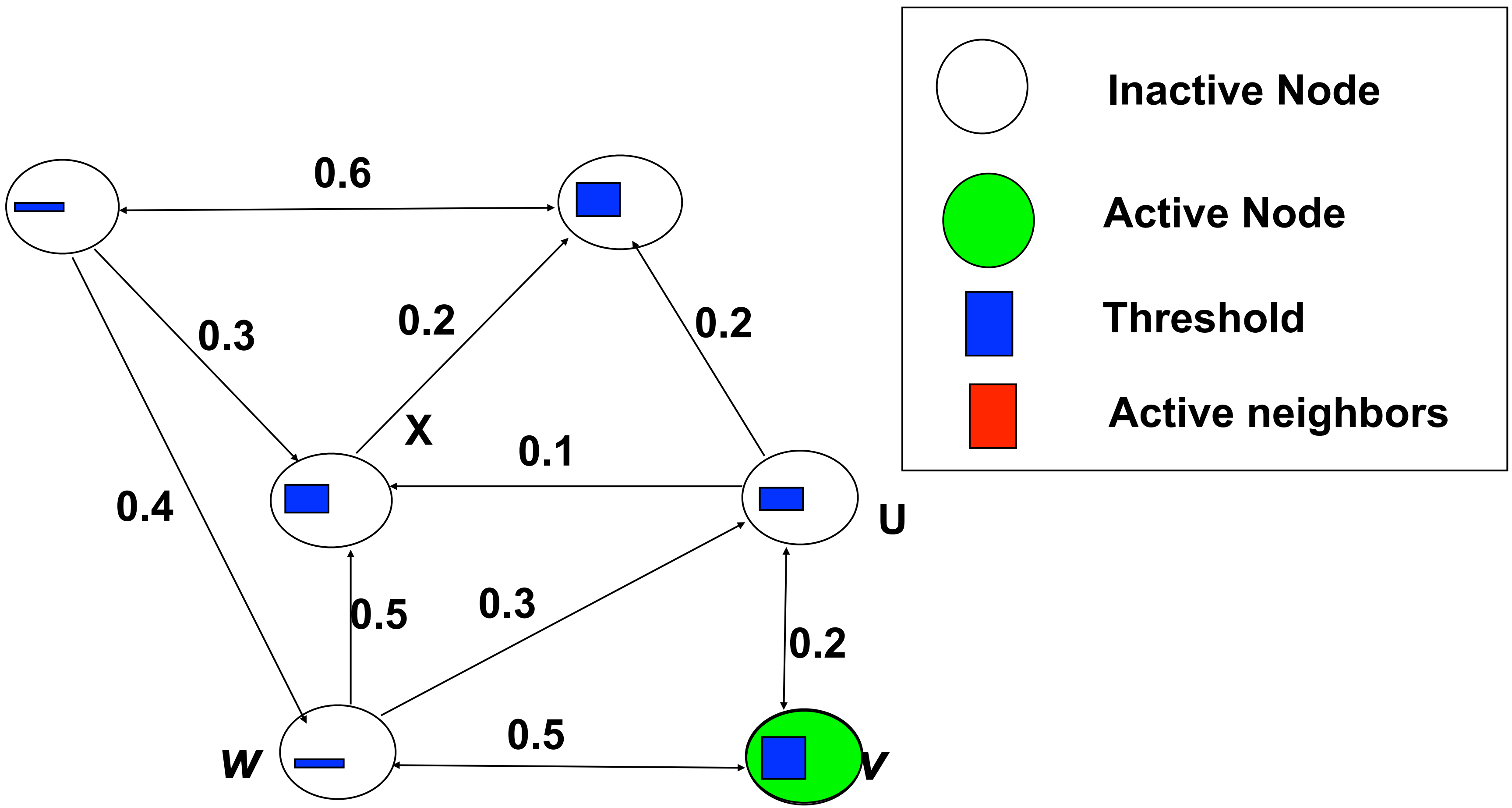


Inactive Node
Active Node
Threshold
Active neighbors
0.6
0.3
0.2
0.2
X
0.1
0.4
U
0.5
0.3
0.2
0.5
w
v

Example

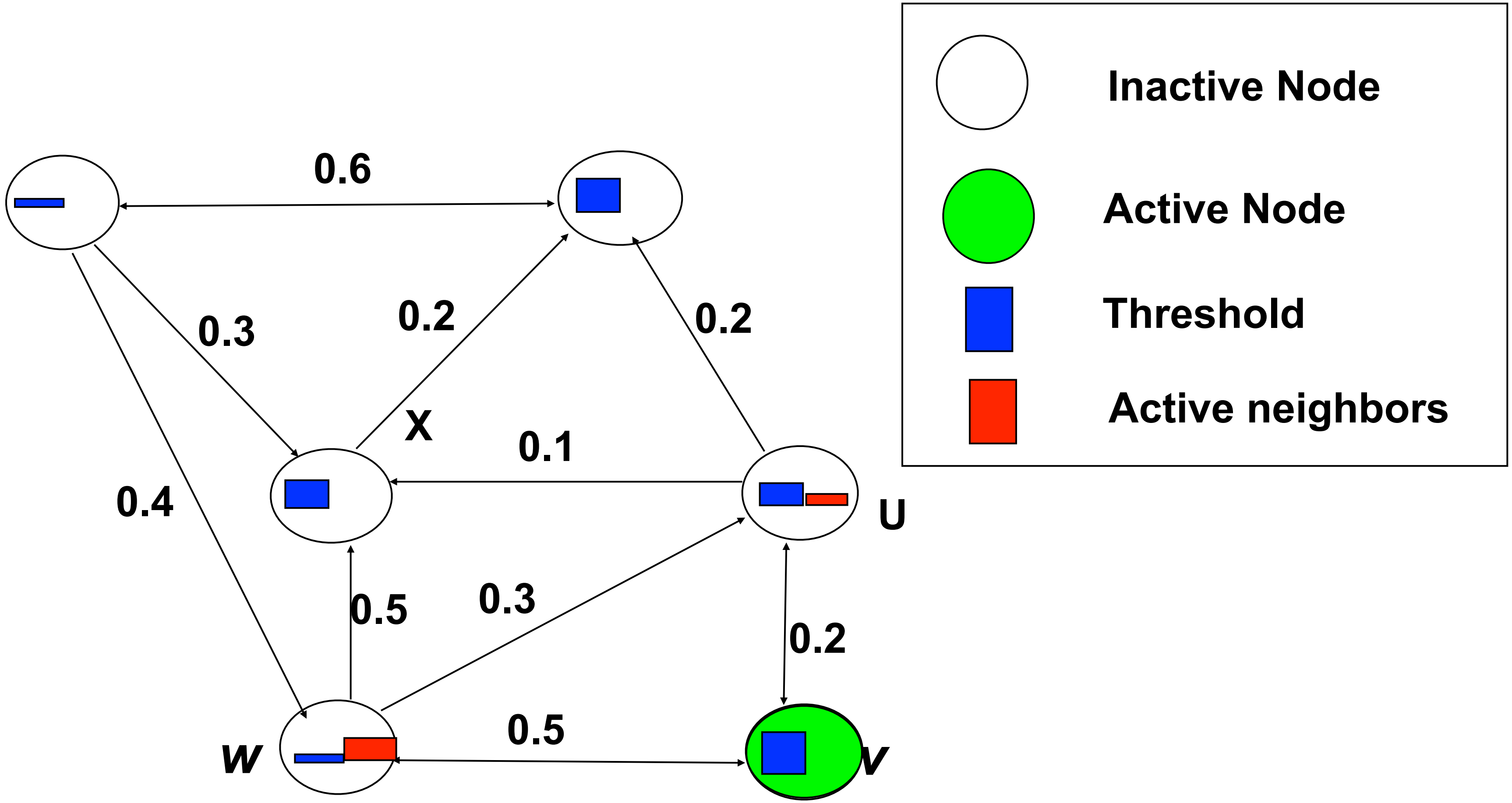

0.6
0.3
0.2
0.2
X
0.1
0.4
U
0.5
0.3
0.2
w
0.5
v
Inactive Node
Active Node
Threshold
Active neighbors

Example



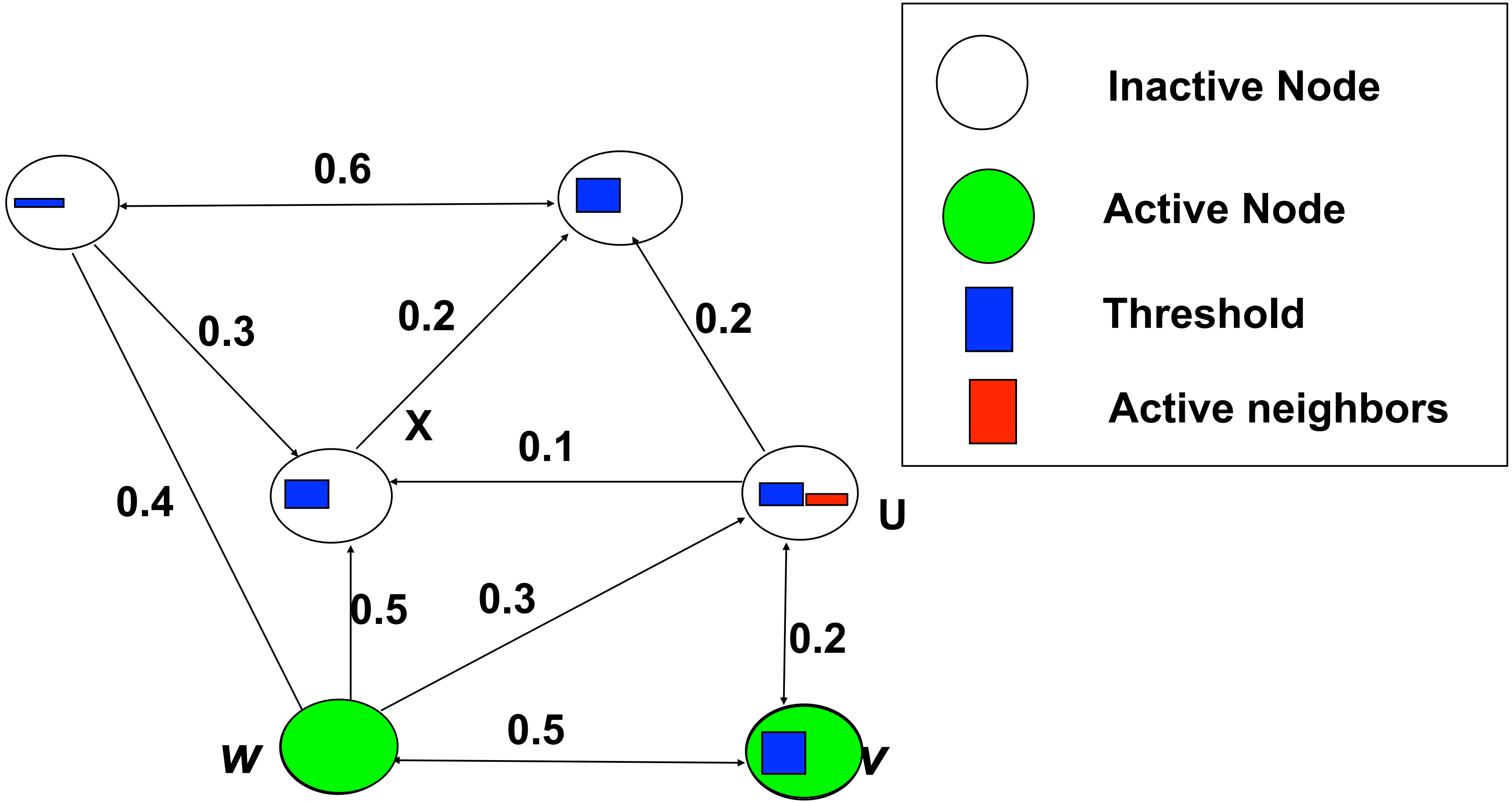
0.6
0.3
0.2
0.2
X
0.1
0.4
U
0.5
0.3
0.2
w
0.5
v
Inactive Node
Active Node
Threshold
Active neighbors

Example



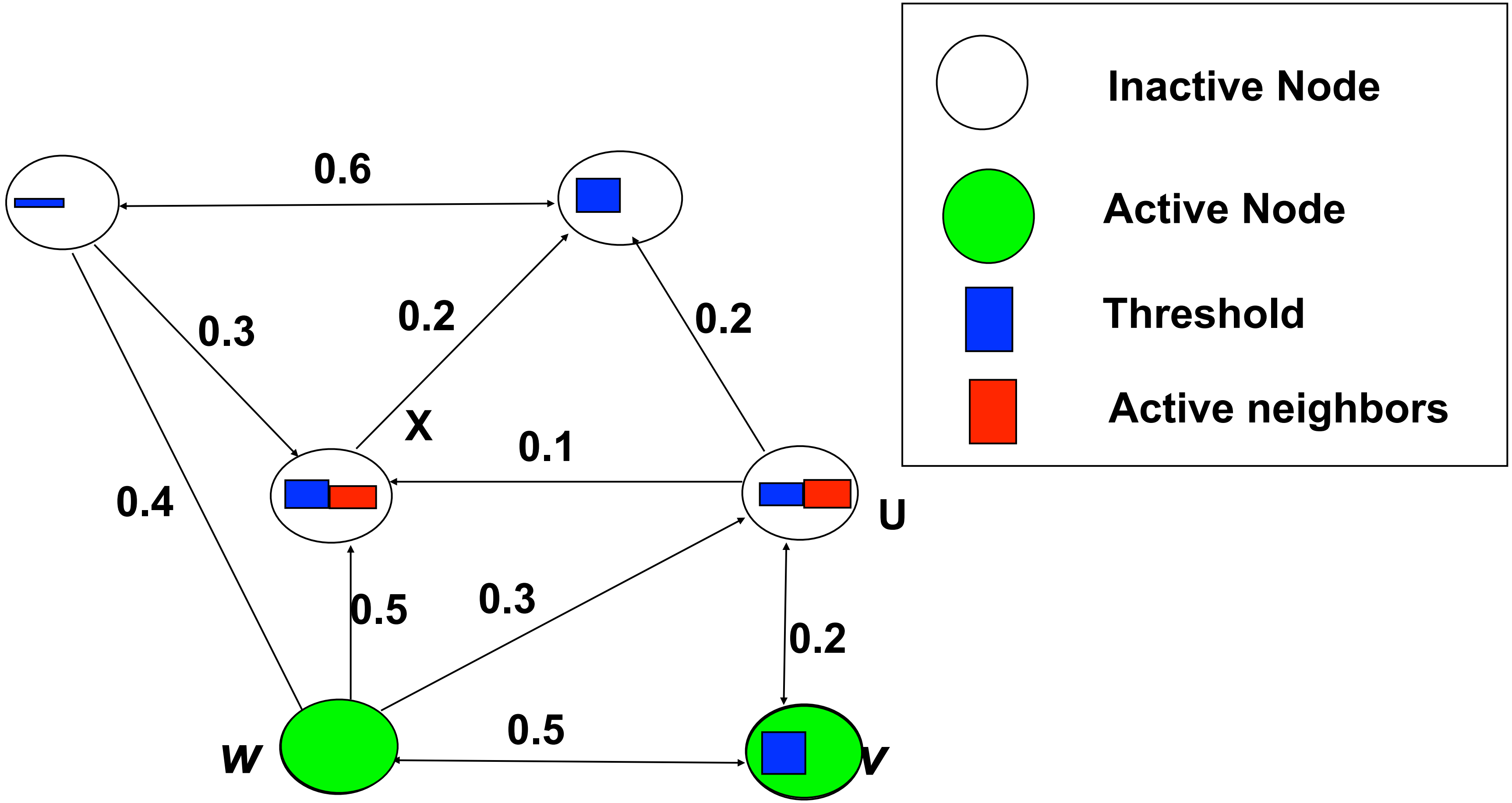
0.6
0.3
0.2
0.2
X
0.1
0.4
U
0.5
0.3
0.2
0.5
w
v
Inactive Node
Active Node
Threshold
Active neighbors

Example

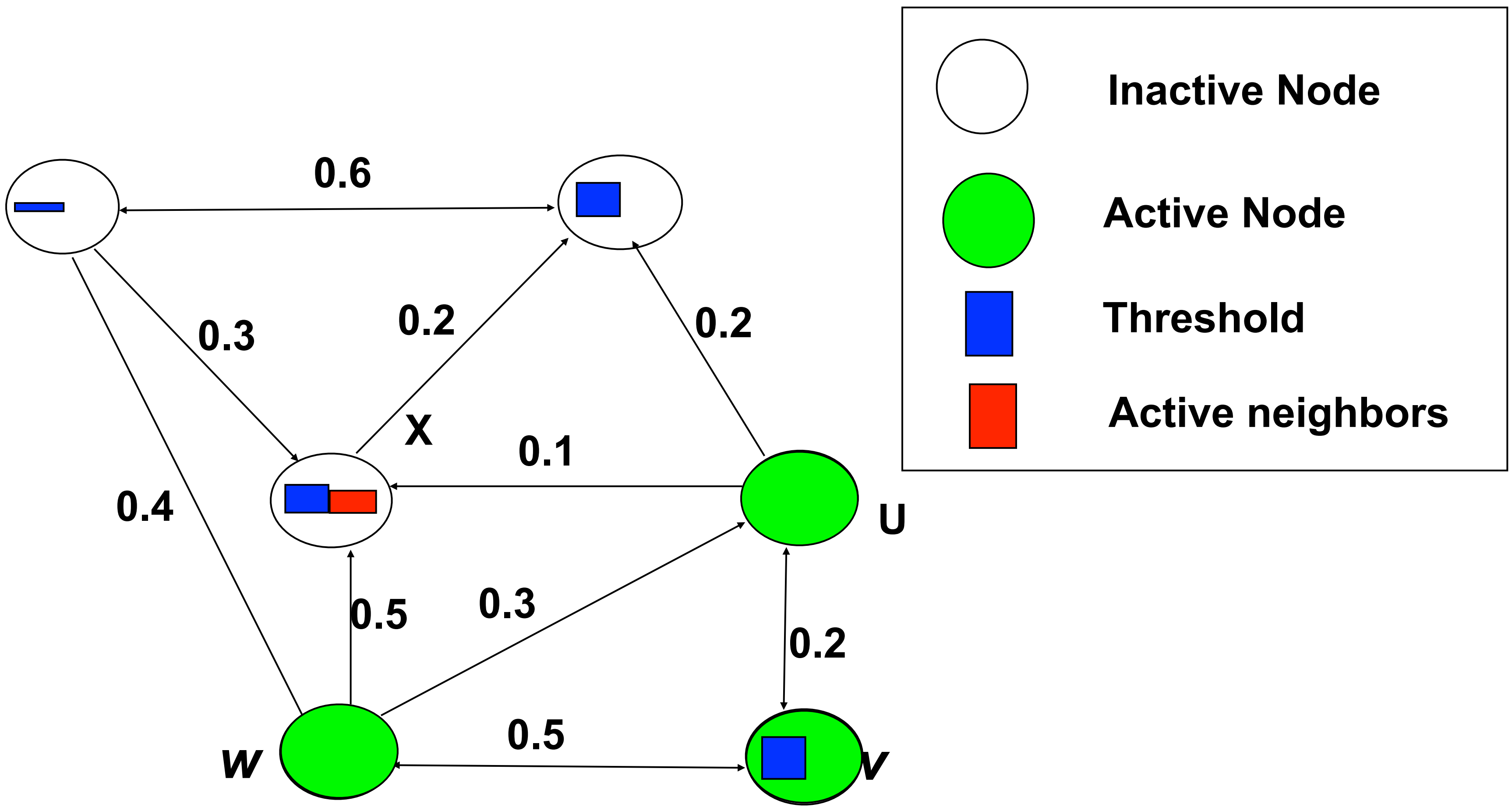


Inactive Node
Active Node
Threshold
Active neighbors
0.6
0.3
0.2
0.2
X
0.1
0.4
U
0.5
0.3
0.2
w
0.5
v

Example

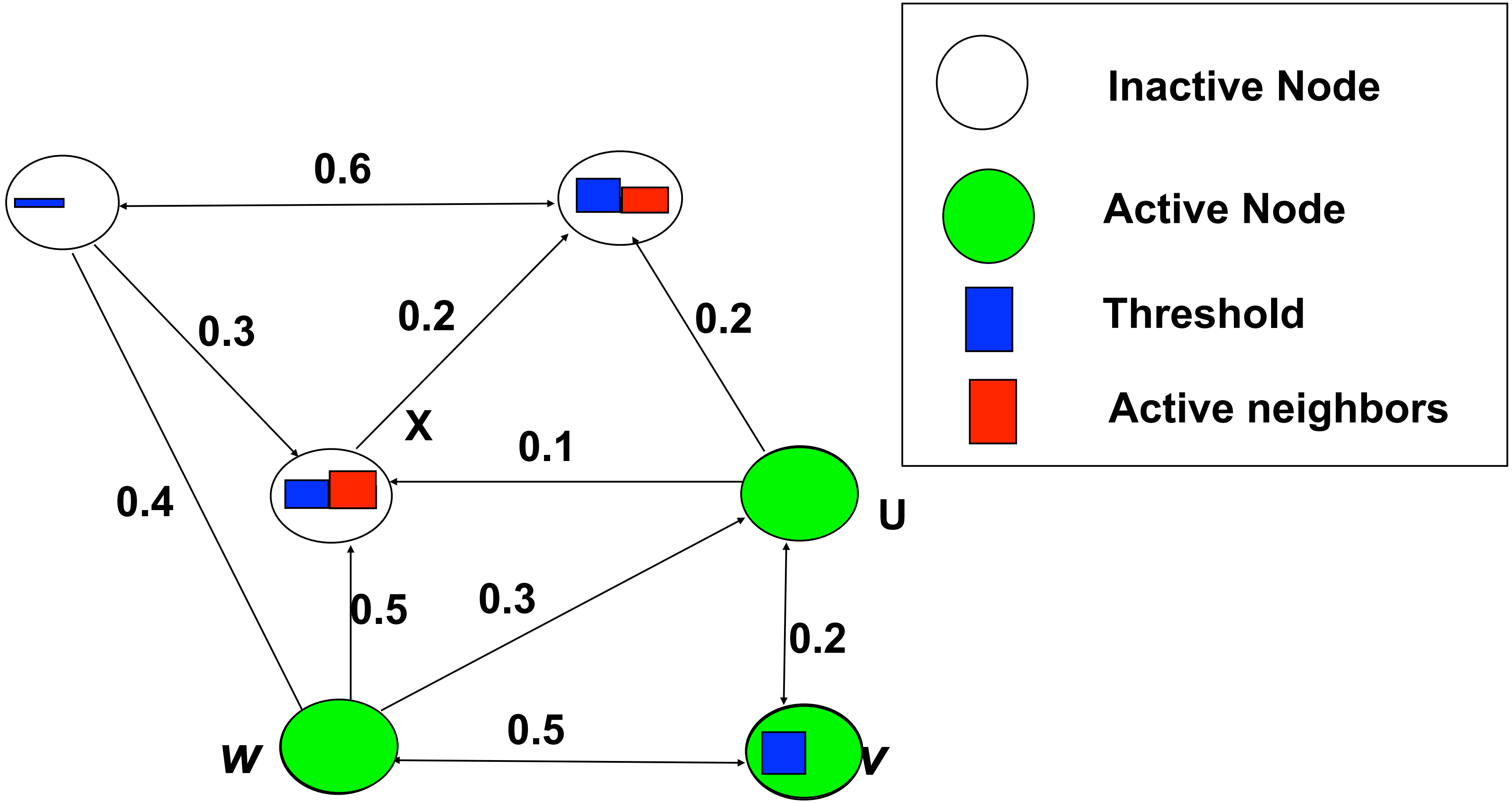

Inactive Node
Active Node
Threshold
Active neighbors
0.6
0.3
0.2
0.2
X
0.1
0.4
U
0.5
0.3
0.2
w
0.5
v

Example

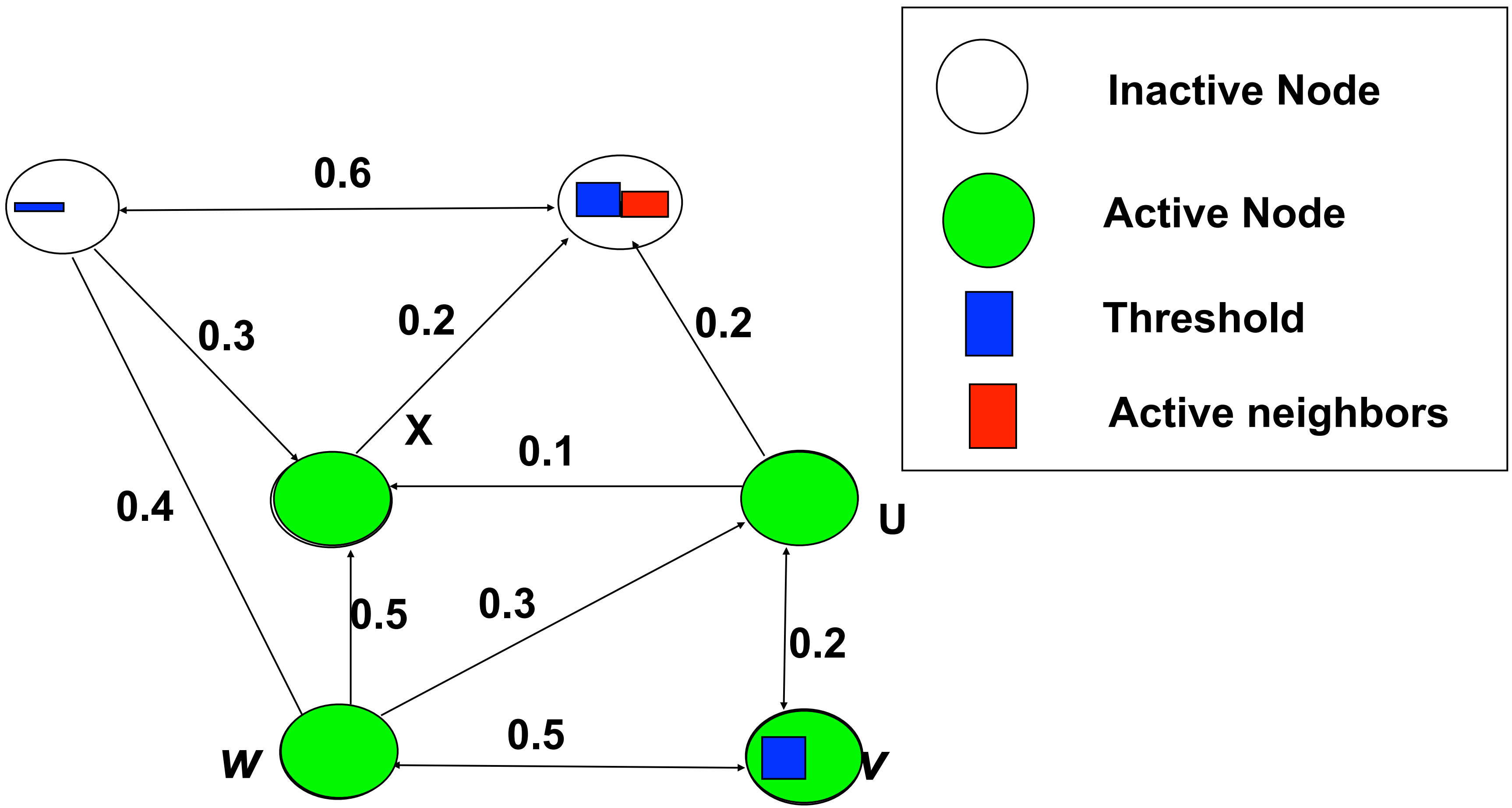


0.6
0.3
0.2
0.2
X
0.1
0.4
U
0.5
0.3
0.2
0.5
w
v
Inactive Node
Active Node
Threshold
Active neighbors

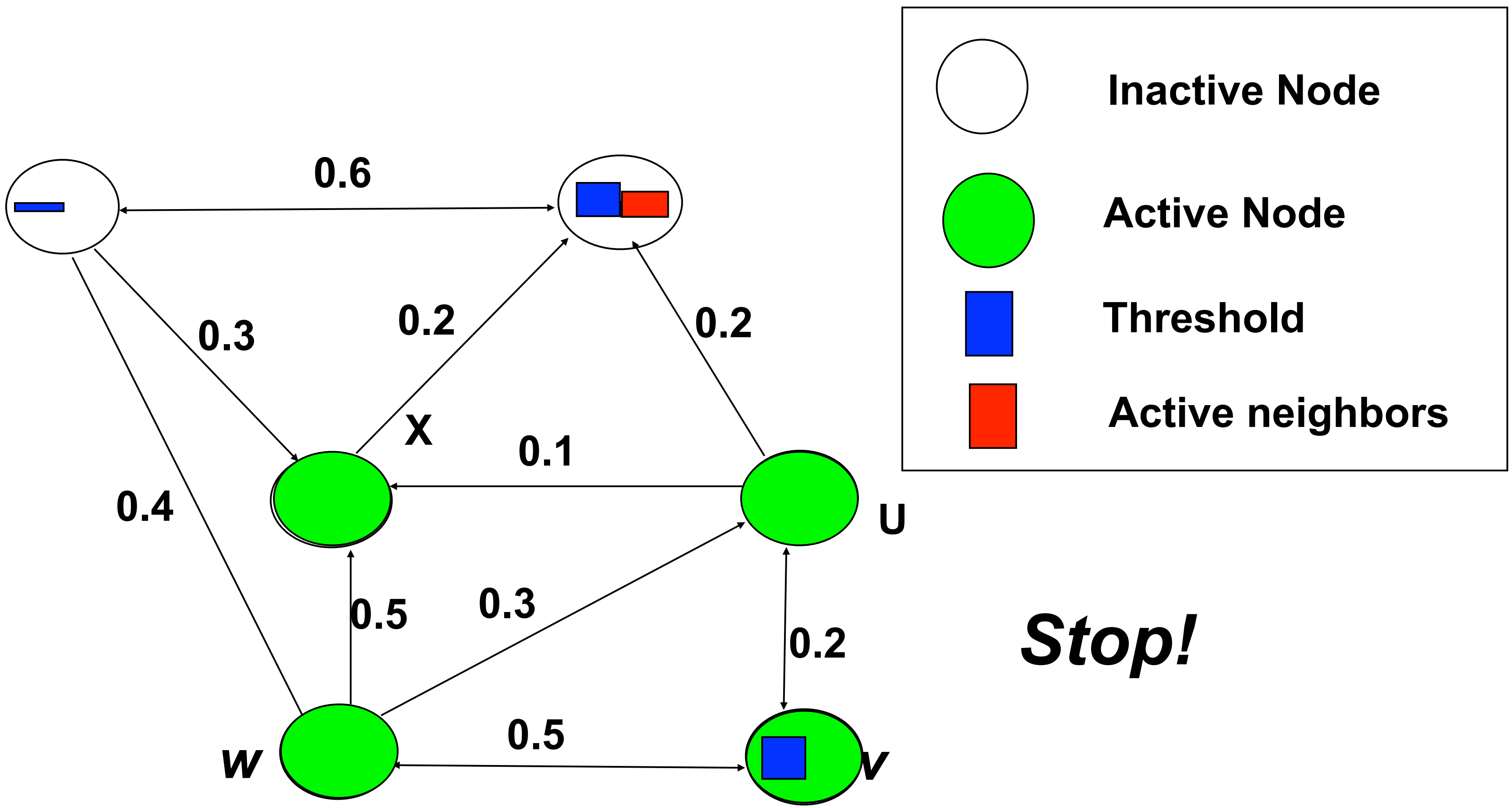
Example


0.6
0.3
0.2
0.2
X
0.1
0.4
U
0.5
0.3
0.2
0.5
w
v
Inactive Node
Active Node
Threshold
Active neighbors

Example


0.6
0.3
0.2
0.2
X
0.1
0.4
U
0.5
0.3
0.2
w
0.5
v
Inactive Node
Active Node
Threshold
Active neighbors

Example



Stop!

Independent Cascade Model

Independent Cascade Model

- When node *v* becomes active, it has a single chance of activating each currently inactive neighbor *w.*

Independent Cascade Model

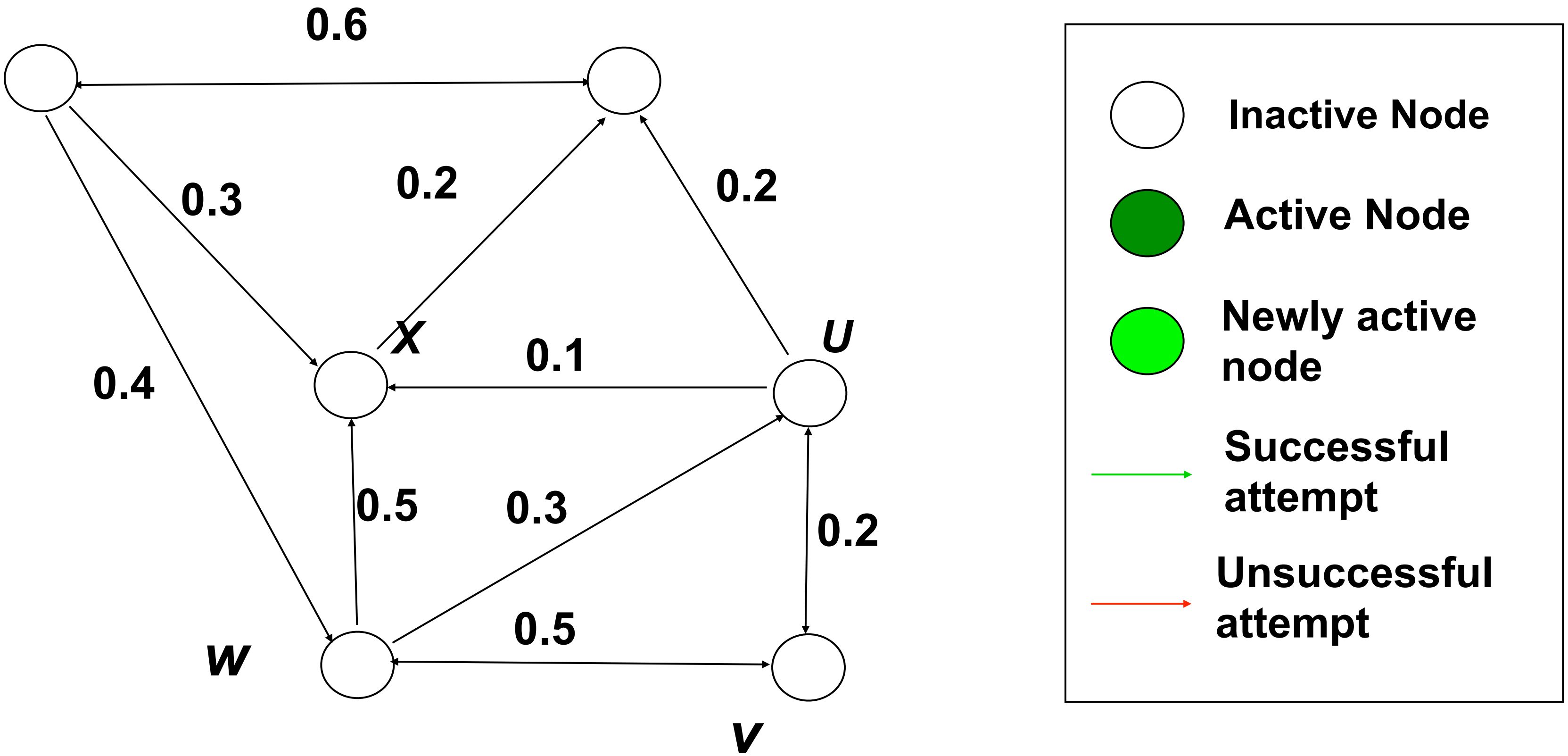
- When node *v* becomes active, it has a single chance of activating each currently inactive neighbor *w.*

Independent Cascade Model

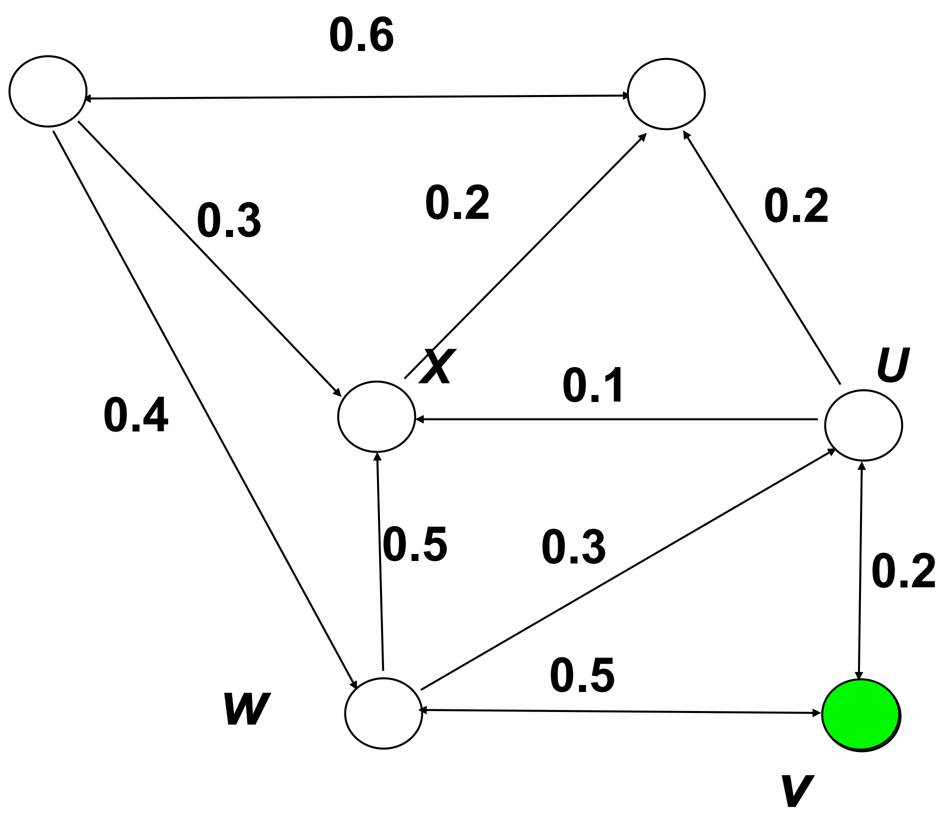
- When node *v* becomes active, it has a single chance of activating each currently inactive neighbor *w.*

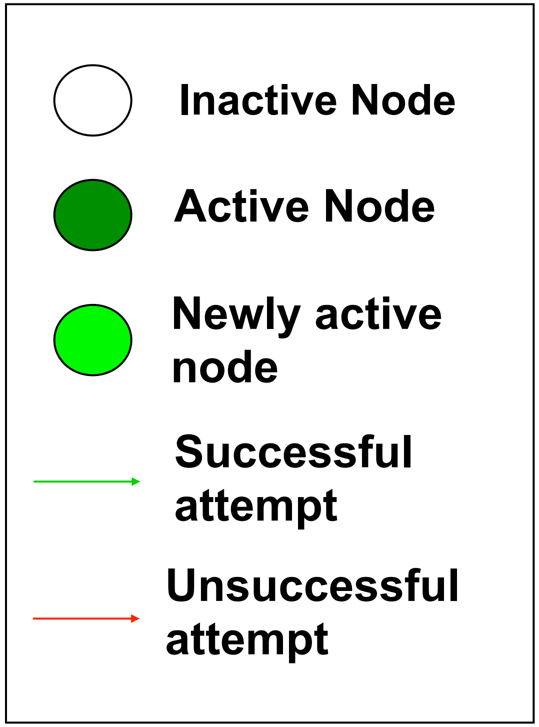
- The activation attempt succeeds with probability p_{vw} .

Example

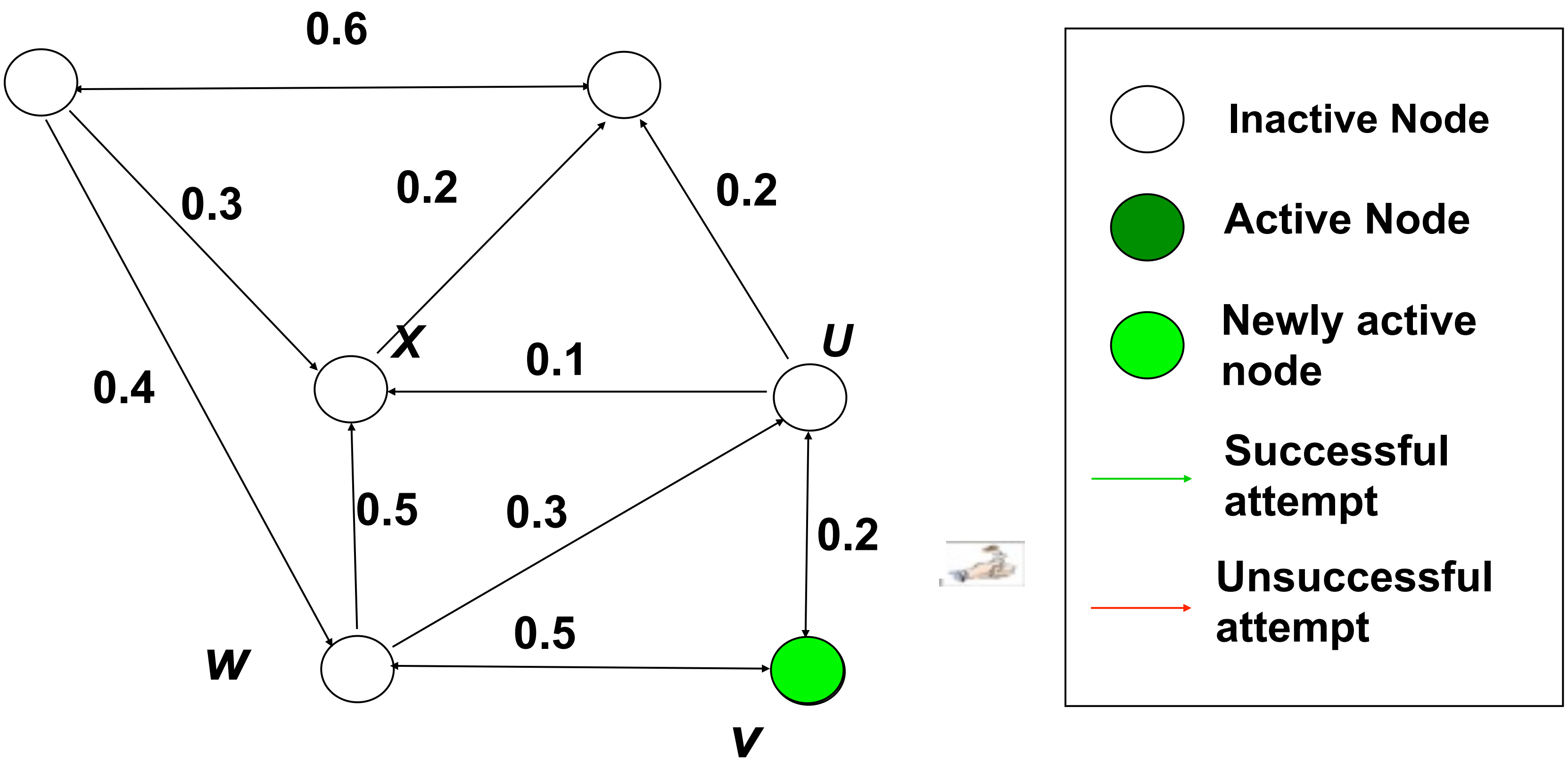

0.6
0.3
0.2
0.2
X
0.1
U
0.4
0.5
0.3
0.2
w
0.5
v
Inactive Node
Active Node
Newly active node
Successful attempt
Unsuccessful attempt

Example

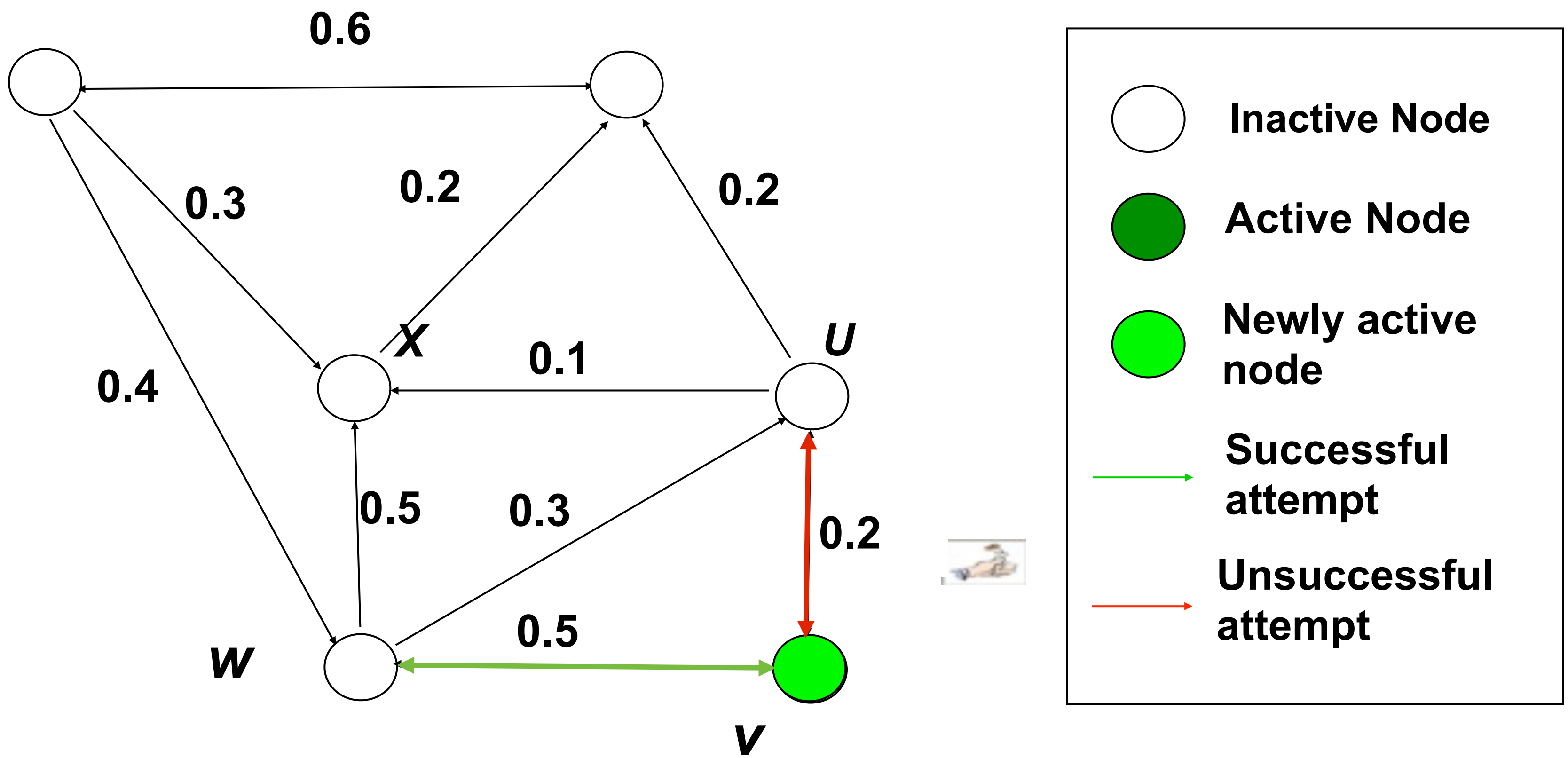

0.6
0.3
0.2
0.2
0.4
X
0.1
U
0.5
0.3
0.2
w
0.5
v


Inactive Node
Active Node
Newly active node
Successful attempt
Unsuccessful attempt

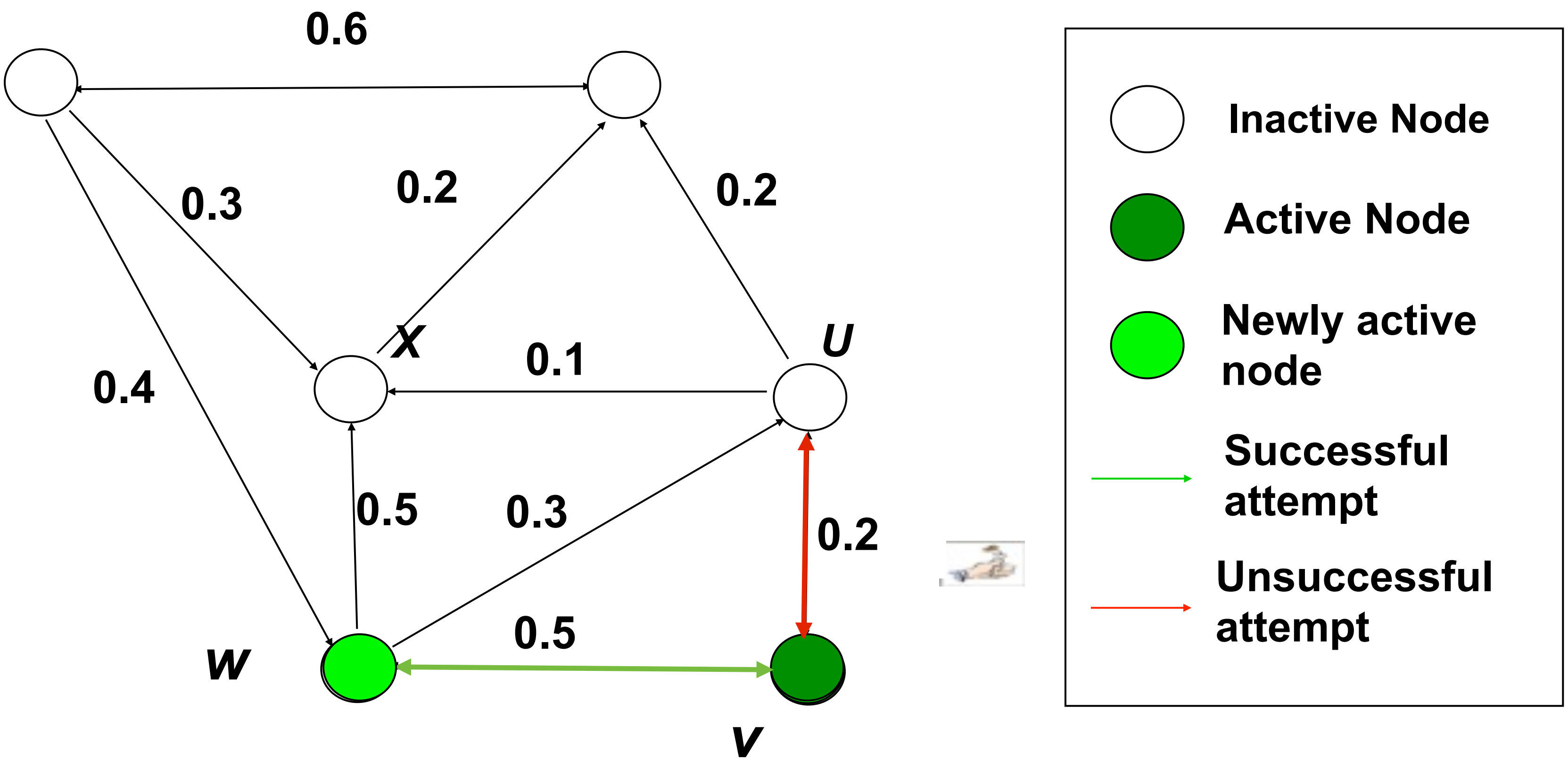
Example


0.6
0.3
0.2
0.2
0.4
X
0.1
U
0.5
0.3
0.2
0.5
W
V
Inactive Node
Active Node
Newly active node
Successful attempt
Unsuccessful attempt

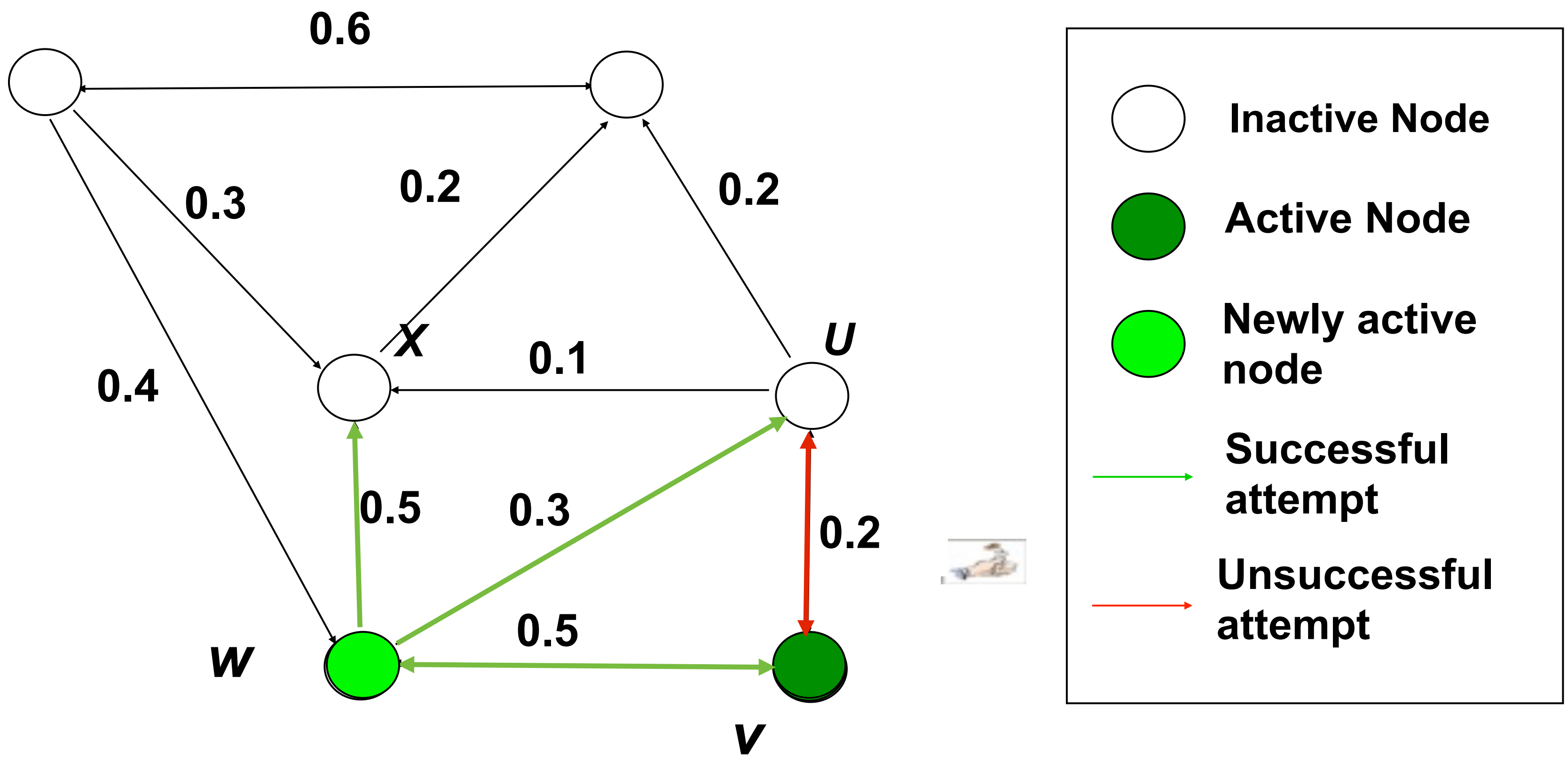
Example


0.6
0.3
0.2
0.2
0.4
X
0.1
U
0.5
0.3
0.2
W
0.5
V
Inactive Node
Active Node
Newly active node
Successful attempt
Unsuccessful attempt

Example

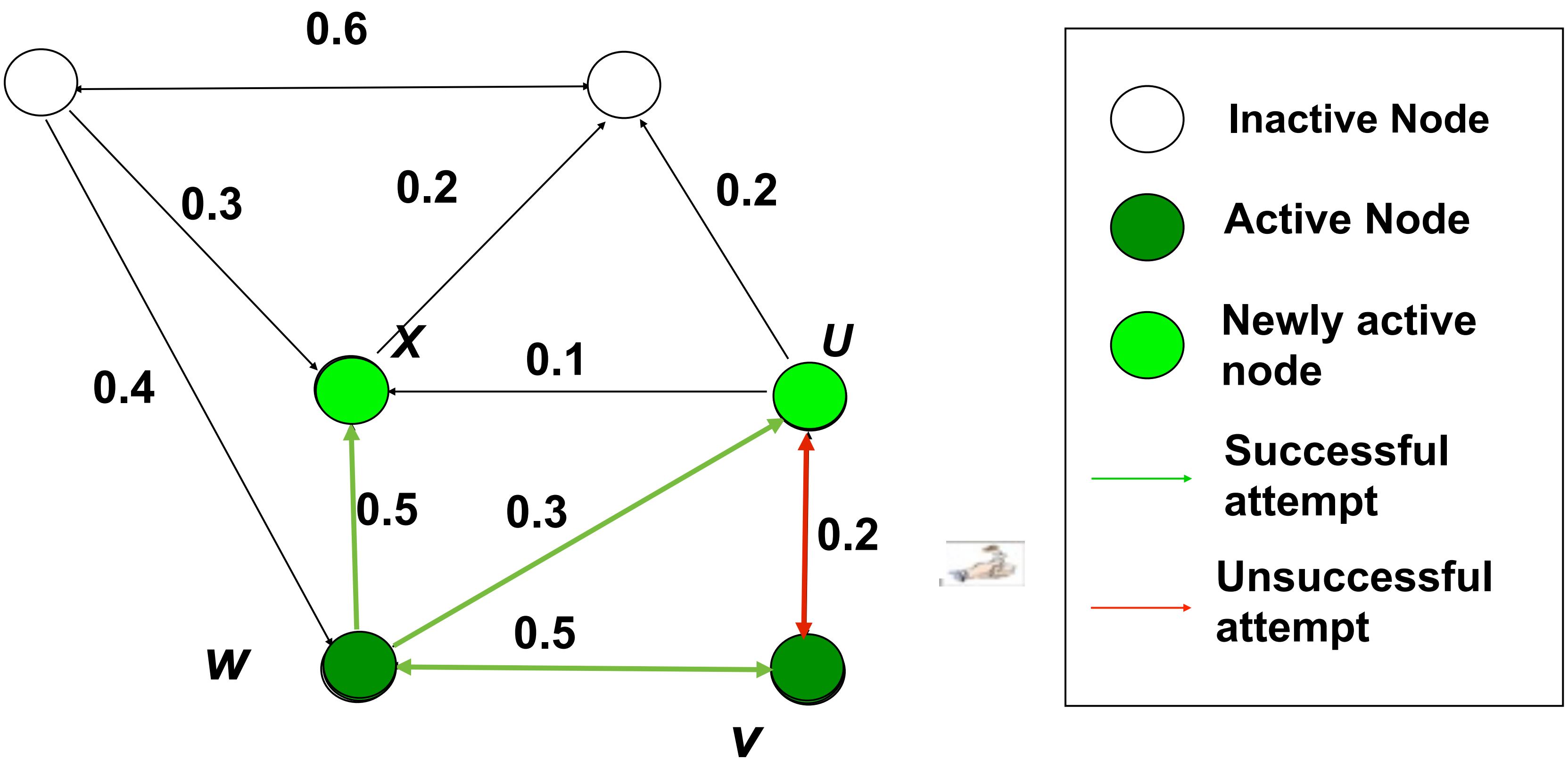

0.6
0.3
0.2
0.2
0.4
X
0.1
U
0.5
0.3
0.2
0.5
w
v
Inactive Node
Active Node
Newly active node
Successful attempt
Unsuccessful attempt

Example

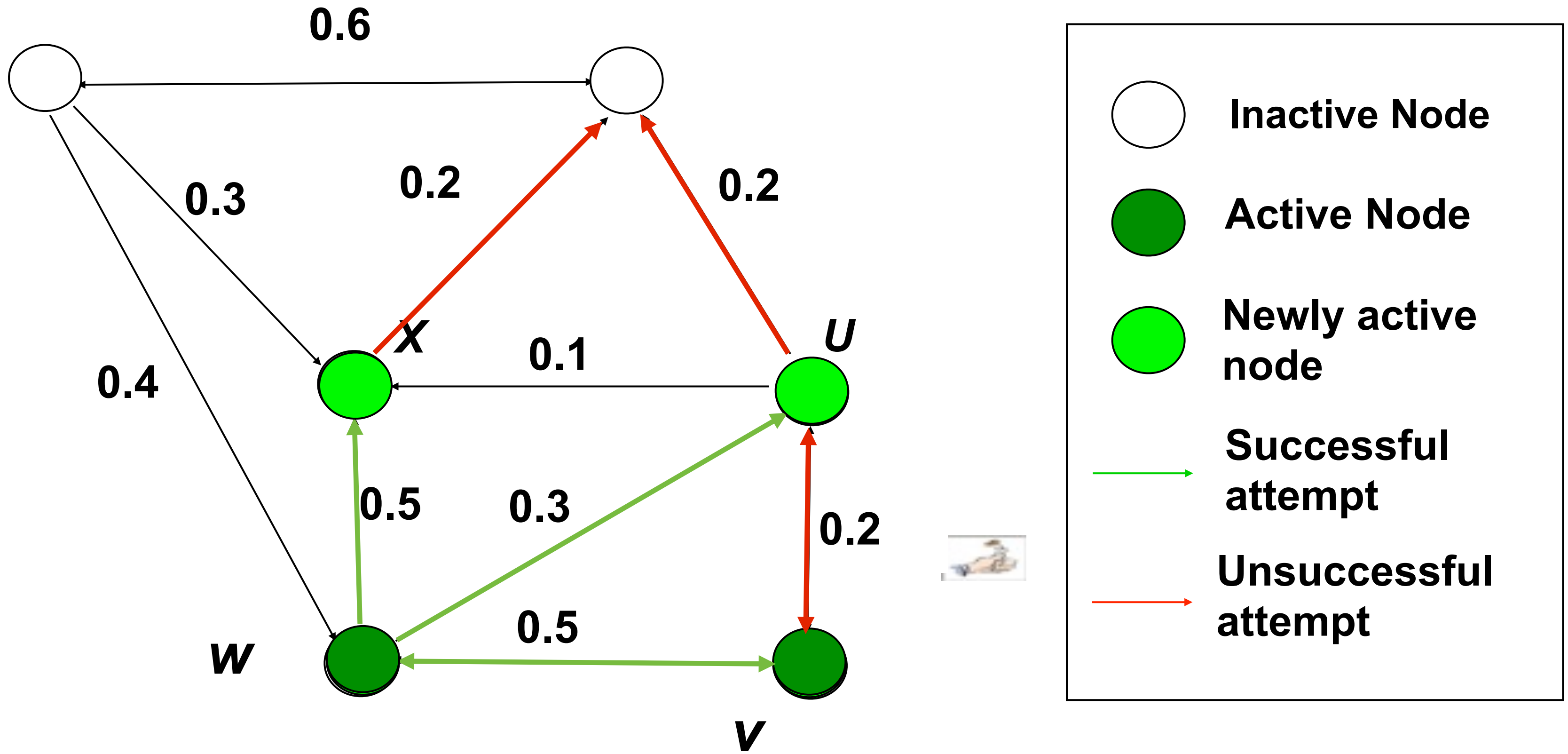


0.6
0.3
0.2
0.2
0.4
X
0.1
U
0.5
0.3
0.2
0.5
W
V
Inactive Node
Active Node
Newly active node
Successful attempt
Unsuccessful attempt

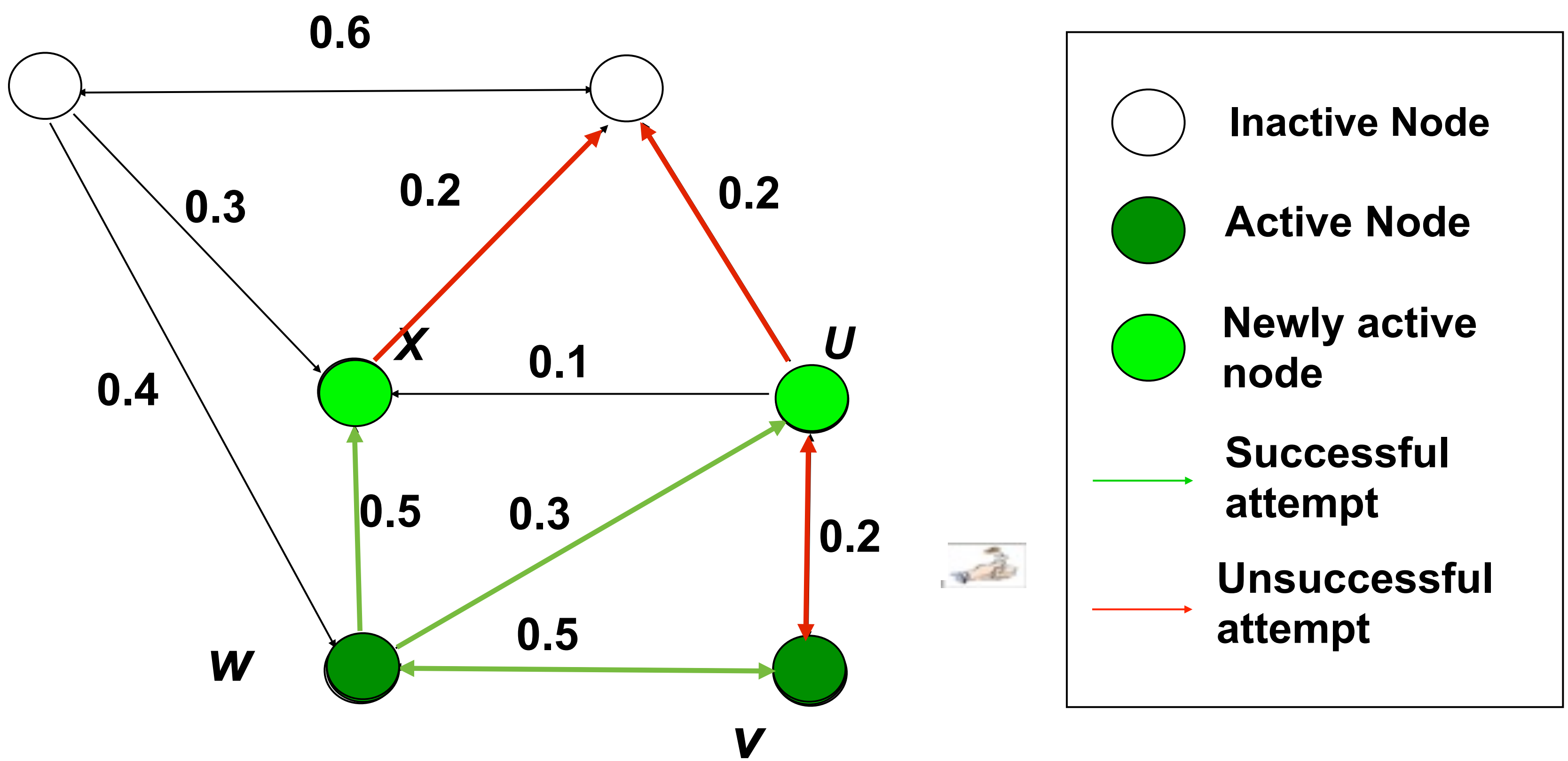
Example


0.6
0.3
0.2
0.2
0.4
X
0.1
U
0.5
0.3
0.2
W
0.5
V
Inactive Node
Active Node
Newly active node
Successful attempt
Unsuccessful attempt

Example


0.6
0.3
0.2
0.2
0.4
X
0.1
U
0.5
0.3
0.2
W
0.5
V
Inactive Node
Active Node
Newly active node
Successful attempt
Unsuccessful attempt

Example



Stop!

Optimization problems

- Given a particular model, there are some natural optimization problems.

    1. How do I select a set of users to give coupons to in order to maximize the total number of users infected?
    2. How do I select a set of people to vaccinate in order to minimize influence/infection?
    3. If I have some sensors, where do I place them to detect an epidemic ASAP?

Influence Maximization Problem

Influence Maximization Problem

- Influence of node set S: f(S)
  - expected number of active nodes at the end, if set S is the initial active set

Influence Maximization Problem

- Influence of node set S: f(S)
  - expected number of active nodes at the end, if set S is the initial active set
- Problem:
  - Given a parameter *k* (budget), find a *k*-node set *S* to maximize f(S)
  - Constrained optimization problem with f(S) as the objective function

f(S): properties (to be demonstrated)

- Non-negative (obviously)
- Monotone: $f(S \cup \{v\}) \geq f(S)$
- Submodular:
  - Let *N* be a finite set
  - A set function $f : 2^N \to \mathbb{R}$ is submodular *iff*

$$\forall S \subset T \subset N, \forall v \in N \setminus T$$

$$f(S \cup \{v\}) - f(S) \geq f(T \cup \{v\}) - f(T)$$

(diminishing returns)

Bad News

Bad News

- For a submodular function *f*, if *f* only takes non-negative value, and is monotone, finding a *k*-element set *S* for which *f(S)* is maximized is an NP-hard optimization problem[GFN77, NWF78].

Bad News

- For a submodular function *f*, if *f* only takes non-negative value, and is monotone, finding a *k*-element set *S* for which *f(S)* is maximized is an NP-hard optimization problem[GFN77, NWF78].
- It is NP-hard to determine the optimum for influence maximization for both independent cascade model and linear threshold model.

Good News

$$f(S \cup \{v\}) - f(S)$$

Good News

- We can use Greedy Algorithm!
  - Start with an empty set S
  - For k iterations:

    Add node v to S that maximizes $f(S \cup \{v\}) - f(S)$

Good News

- We can use Greedy Algorithm!
  - Start with an empty set S
  - For k iterations:

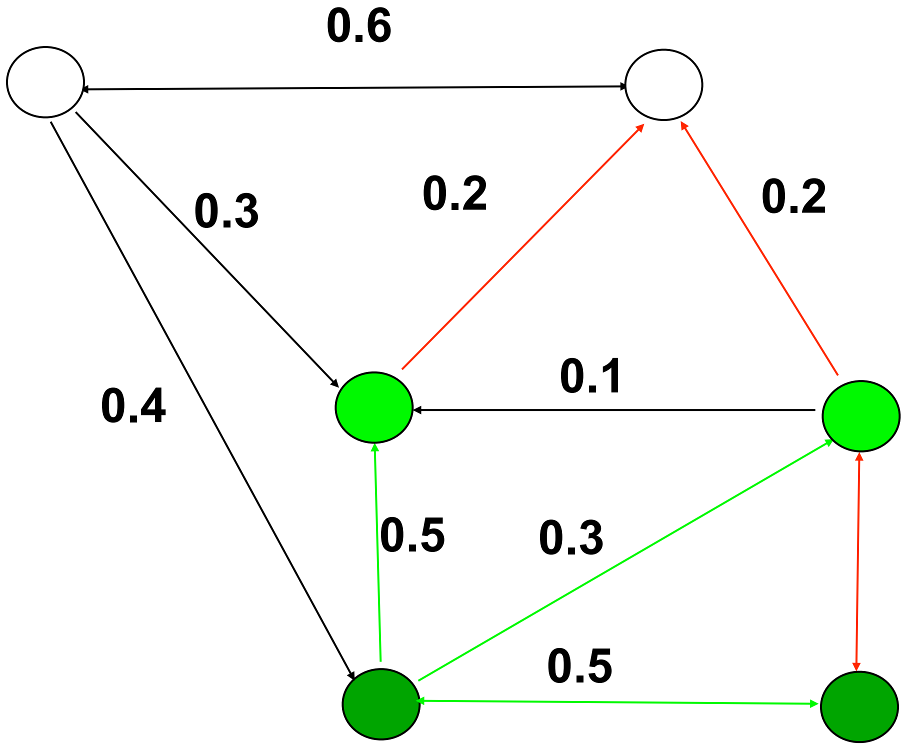
    Add node v to S that maximizes $f(S \cup \{v\}) - f(S)$
- How good (bad) it is?
  - Theorem: The greedy algorithm is a (1 – 1/*e*) approximation.
  - The resulting set S activates at least (1- 1/*e*) > 63% of the number of nodes that any size-k set S could activate.

Key 1: Prove submodularity

$$\forall S \subset T \subset N, \forall v \in N \setminus T$$
$$f(S \cup \{v\}) - f(S) \geq f(T \cup \{v\}) - f(T)$$

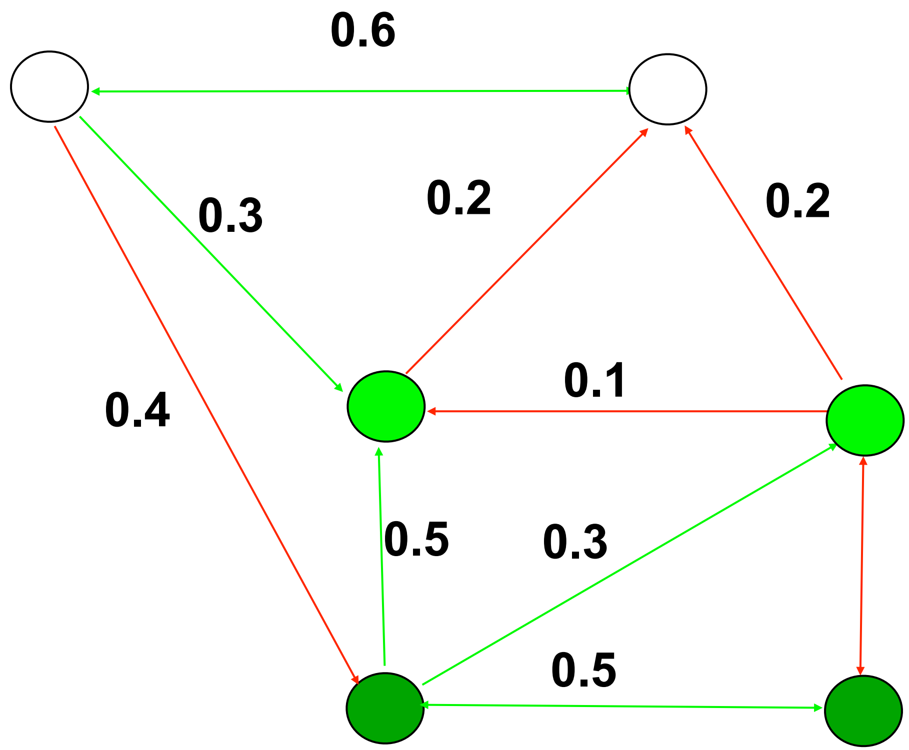
Submodularity for Independent Cascade

- Coins for edges are flipped during activation attempts.



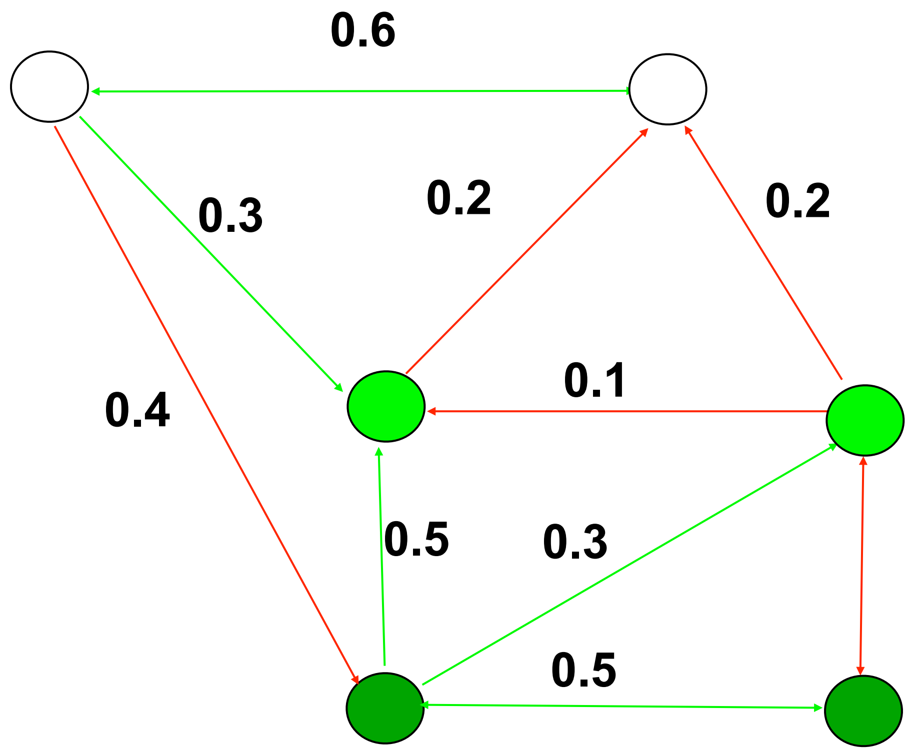
Submodularity for Independent Cascade

- Coins for edges are flipped during activation attempts.
- Can pre-flip all coins and reveal results immediately.



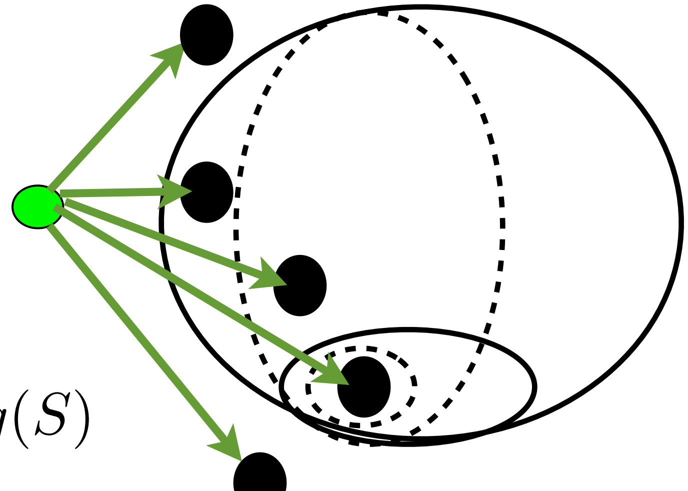
Submodularity for Independent Cascade

- Coins for edges are flipped during activation attempts.
- Can pre-flip all coins and reveal results immediately.



- Active nodes in the end are reachable via green paths from initially targeted nodes.
- Study reachability in green graphs

Submodularity, Fixed Graph



$$S \subseteq T$$

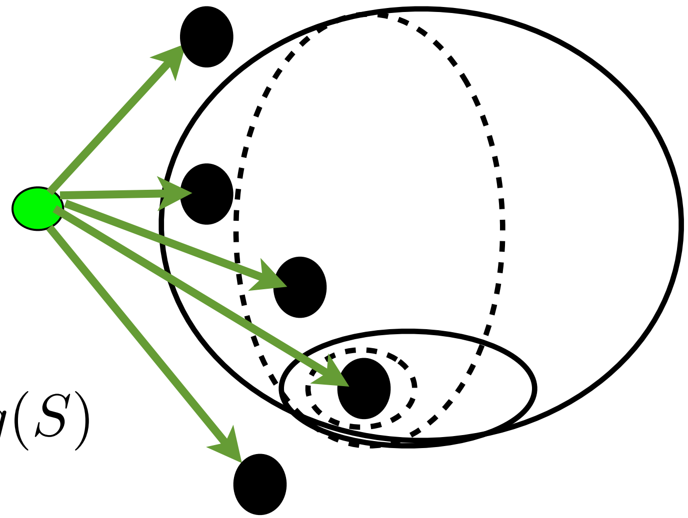
$$g(T \cup \{v\}) - g(T) \leq g(S \cup \{v\}) - g(S)$$

$$g(S \cup \{v\}) - g(S)$$

$(S \cup \{v\})$

Submodularity, Fixed Graph

- Fix "green graph" *G; g(S)* are nodes reachable from *S* in *G*.



$$S \subseteq T$$

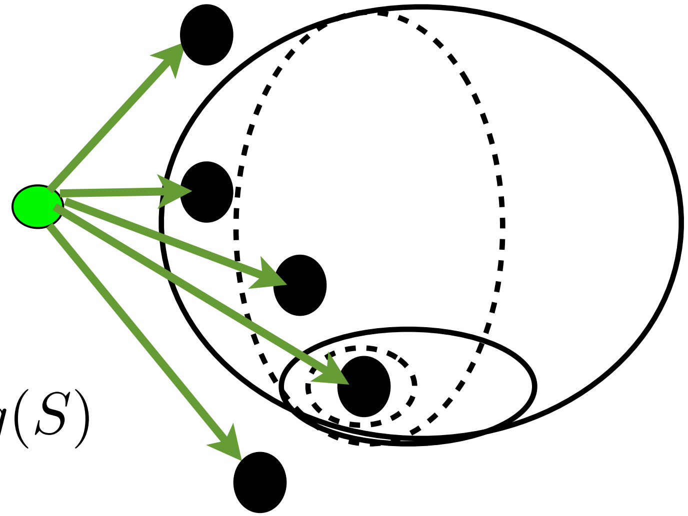
$$g(T \cup \{v\}) - g(T) \leq g(S \cup \{v\}) - g(S)$$

$$g(S \cup \{v\}) - g(S)$$

$(S \cup \{v\})$

Submodularity, Fixed Graph

- Fix "green graph" *G; g(S)* are nodes reachable from *S* in *G*.



$$(S \cup \{v\})$$

$$S \subseteq T$$

$$g(T \cup \{v\}) - g(T) \leq g(S \cup \{v\}) - g(S)$$

$$g(S \cup \{v\}) - g(S)$$

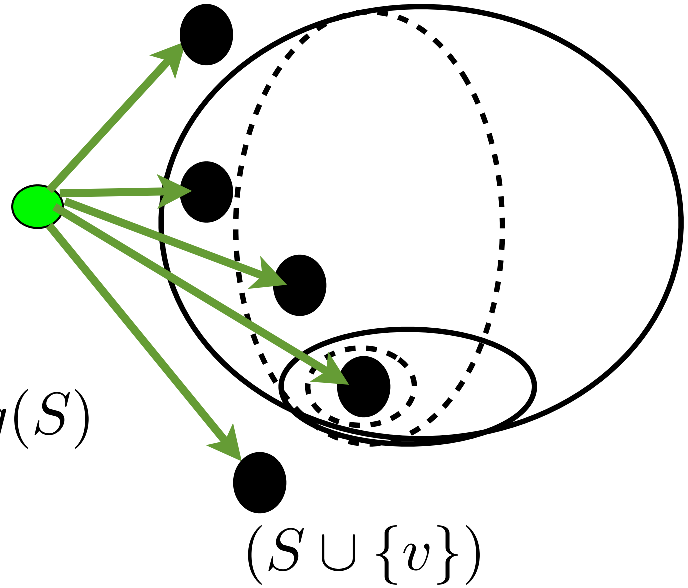
Submodularity, Fixed Graph

- Fix "green graph" *G; g(S)* are nodes reachable from *S* in *G*.

- Submodularity: for $S \subseteq T$

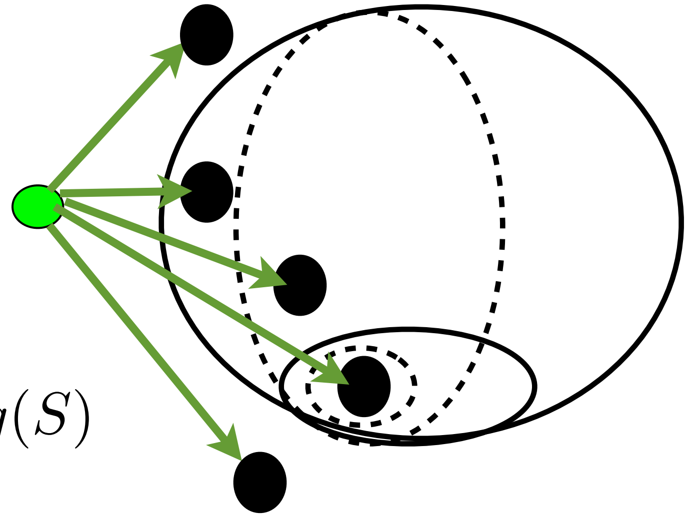
$$g(T \cup \{v\}) - g(T) \leq g(S \cup \{v\}) - g(S)$$

$$g(S \cup \{v\}) - g(S)$$



Submodularity, Fixed Graph

- Fix "green graph" *G; g(S)* are nodes reachable from *S* in *G*.



- Submodularity: for $S \subseteq T$

$$g(T \cup \{v\}) - g(T) \leq g(S \cup \{v\}) - g(S)$$

- $g(S \cup \{v\}) - g(S)$ nodes reachable from $(S \cup \{v\})$, but not from *S*.
- From the picture: g is submodular!

Submodularity of the Function

Fact: A non−negative linear combination of submodular functions is submodular

$$f(S) = \sum_{G} \mathrm{Prob}(G\ is\ green\ graph) \times g_G(S)$$

- $g_G(S)$: nodes reachable from S in G.
- Each $g_G(S)$: is submodular (previous slide).
- Probabilities are non-negative.

Submodularity for Linear Threshold

- Use similar “green graph” idea.
- Once a graph is fixed, “reachability” argument is identical.
- How do we fix a green graph now?
- Each node picks at most one incoming edge, with probabilities proportional to edge weights.
- Equivalent to linear threshold model (trickier proof).

Key 2: Evaluating $f(S)$

Evaluating $f(S)$

Evaluating $f(S)$

- How to evaluate $f(S)$?

Evaluating $f(S)$

- How to evaluate $f(S)$?

Evaluating $f(S)$

- How to evaluate $f(S)$?
- Still an open question of how to compute efficiently

Evaluating $f(S)$

- How to evaluate $f(S)$?
- Still an open question of how to compute efficiently

Evaluating $f(S)$

- How to evaluate $f(S)$?
- Still an open question of how to compute efficiently
- But: very good estimates by simulation
  - repeating the diffusion process enough times

Experiment Data

- A collaboration graph obtained from co-authorships in papers of the arXiv high-energy physics theory section
- co-authorship networks arguably capture many of the key features of social networks more generally
- Resulting graph: 10748 nodes, 53000 distinct edges

Experiment Settings

Experiment Settings

- Linear Threshold Model: multiplicity of edges as weights
  - weight(v→ω) = C_{vw} / dv, weight(ω→v) = C_{wv} / dw

Experiment Settings

- Linear Threshold Model: multiplicity of edges as weights
  - weight(v→ω) = C_{vw} / dv, weight(ω→v) = C_{wv} / dw
- Independent Cascade Model:
  - uniform probabilities p on each edge

Experiment Settings

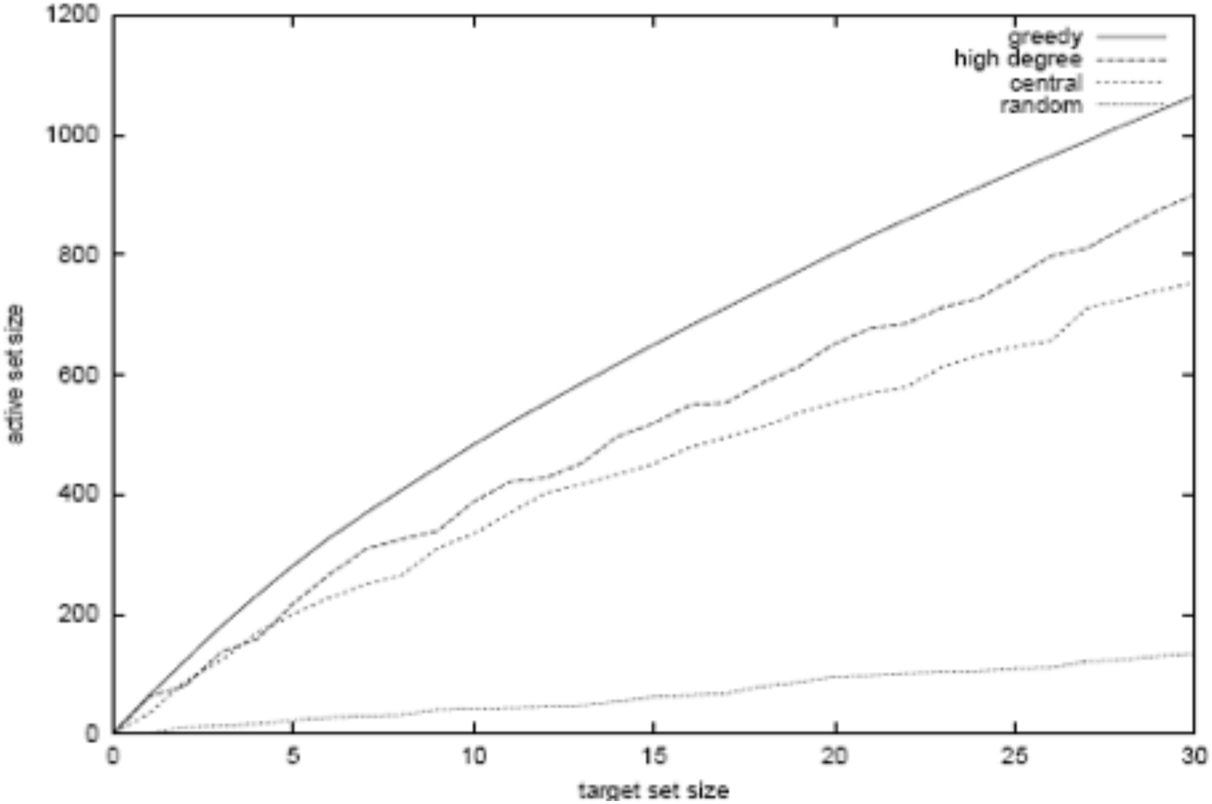
- Linear Threshold Model: multiplicity of edges as weights
  - weight(v→ω) = C_{vw} / dv, weight(ω→v) = C_{wv} / dw
- Independent Cascade Model:
  - uniform probabilities *p* on each edge
- Simulate the process 10000 times for each targeted set, re-choosing thresholds or edge outcomes pseudo-randomly from [0, 1] every time

Experiment Settings

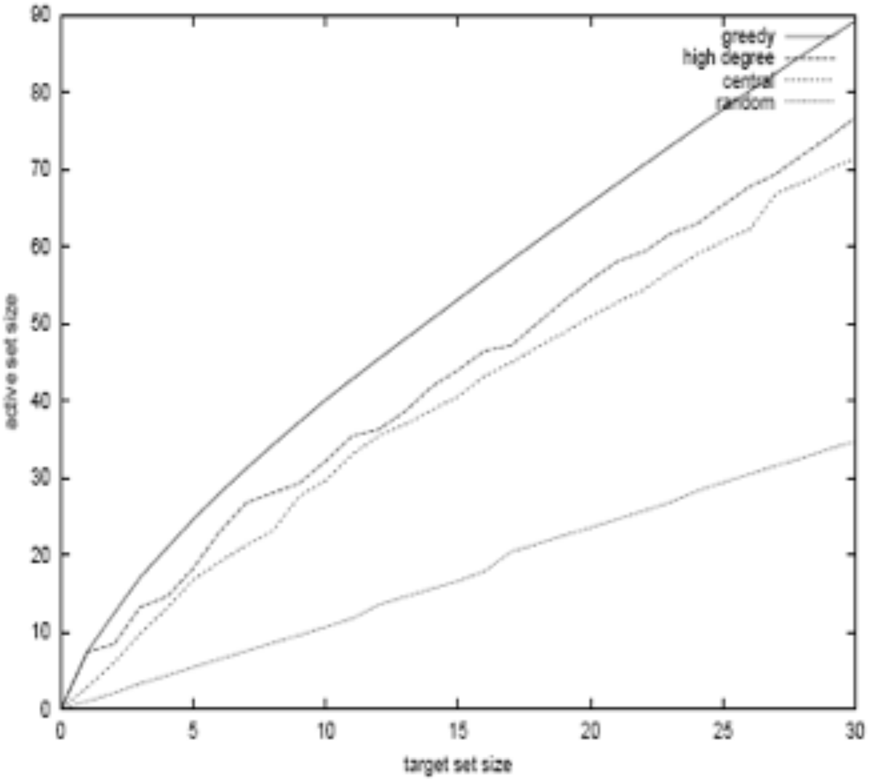
- Linear Threshold Model: multiplicity of edges as weights
  - weight(v→ω) = C_{vw} / dv, weight(ω→v) = C_{wv} / dw
- Independent Cascade Model:
  - uniform probabilities p on each edge

- Simulate the process 10000 times for each targeted set, re-choosing thresholds or edge outcomes pseudo-randomly from [0, 1] every time
- Compare with other 3 common heuristics
  - (in)degree centrality, distance centrality, random nodes.

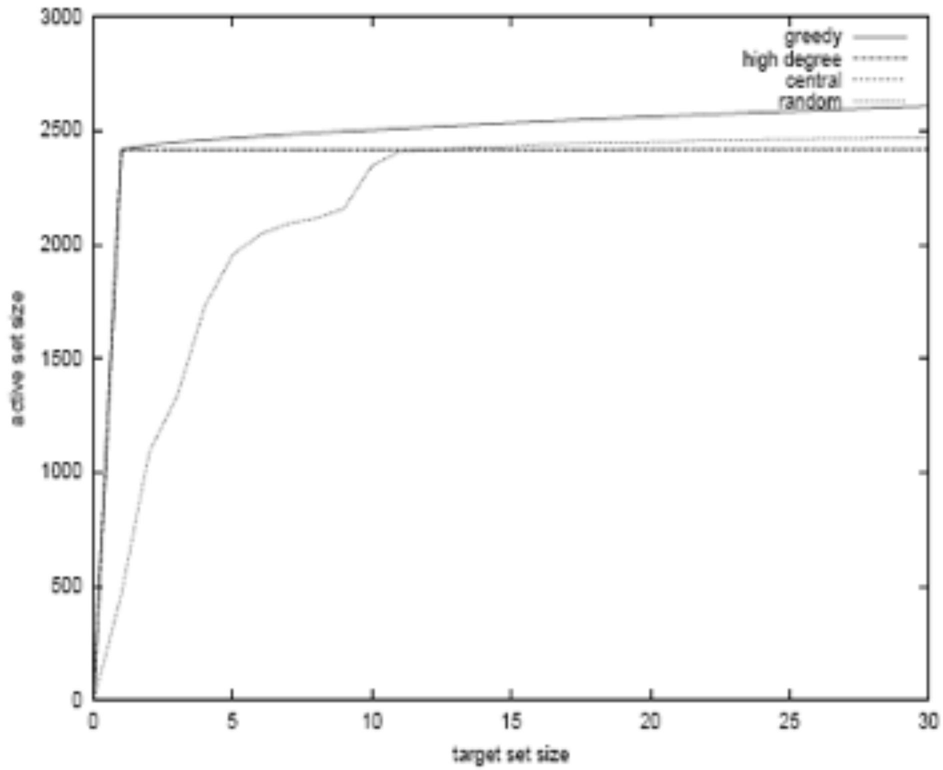
Results: linear threshold model



Independent Cascade Model



P = 1%



P = 10%