Diffusion in Networks

Diffusion:

- Process by which a **contagion** (e.g. information, disease, fads) spreads through a social network.
- Also called **network dynamics**.

Diffusion: Early Empirical Work

Cultivation of Hybrid Seed Corn:

- Study by Bruce Ryan and Neal Gross in the 1920's at Iowa State University.
- **Goal:** To understand how the practice of cultivating hybrid seed corn spread among farmers in Iowa.
- This form of corn had a higher yield and was disease resistant.
- Yet, there was resistance to its use ("inertia").
- The practice didn't take off until 1934 when some elite farmers started cultivating it.
- Ryan/Gross analyzed surveys; they didn't construct social networks.

Diffusion: Early Empirical Work (continued)

Use of Tetracycline (an antibiotic):

- Study by James Coleman, Herbert Menzel and Elihu Katz in the 1960's at Columbia University.
- Tetracycline was a new drug marketed by Pfizer.
- Analyzed data from doctors who prescribed the medicine and pharmacists that filled the prescriptions.
- Constructed a social network of doctors and pharmacists.
- **Summary:**
  - A large fraction of the initial prescriptions were by a small number of doctors in large cities.
  - Doctors who had many physician friends started prescribing the medicine more quickly.

Diffusion: Early Empirical Work (continued)

Other studies:

- Use of telephones (Claude Fischer).
- Use of email (Lynne Markus).

Modeling diffusion through a network:

- Consider diffusion of new behavior.
- **Assumptions:**
  - People makes decisions about adopting a new behavior based on their friends.
  - Benefits of adopting a new behavior increase as more friends adopt that behavior.

    Example: It may be easier to collaborate with colleagues if compatible technologies are used.
- This "direct benefit" model is due to Stephen Morris (Princeton University).

A Coordination Game

Rules of the game:

- A social network (an undirected graph) is given.
- Each node has a choice between behaviors **A** and **B**.
- For each edge $\{x, y\}$, there is an **incentive** for the behaviors of nodes x and y to **match**, as given by the following **payoff matrix**.

		y: A	y: B
x	A	a, a	0, 0
	B	0, 0	b, b

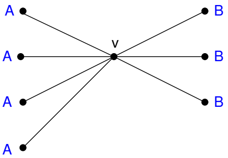
- If x and y both adopt **A**, they both get a benefit of a.
- If x and y both adopt **B**, they both get a benefit of b.
- If x and y **don't** adopt the same behavior, their benefit is **zero**.

A Coordination Game (continued)

Rules of the game (continued):

- Each node v plays this game with **each of its neighbors**.
- The payoff for a node v is the **sum** of the payoffs over all the edge incident on v.

Example:



- Let $a = 5$ and $b = 7$.
- If v adopts **A**, payoff $= 4 \times 5 = 20$.
- If v adopts **B**, payoff $= 3 \times 7 = 21$.
- So, v should adopt **B** (rational behavior).

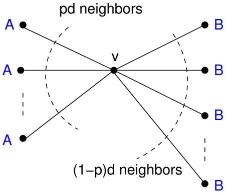
Note: The example points out that v's choice depends on the choices made by all its neighbors and the parameters a and b.

A Coordination Game (continued)

Question: In general, how should a node v choose its behavior, given the choices of its neighbors?

Analysis:

- Suppose the degree of v is d.
- Suppose a fraction p of v's neighbors have chosen **A** and the remaining fraction $(1 - p)$ have chosen **B**.
- So, pd neighbors have chosen **A** and $(1 - p)d$ neighbors have chosen **B**.



- If v chooses **A**, its payoff $= pda$.
- If v chooses **B**, its payoff $= (1 - p)db$.
- So, **A** is the better choice if

$$pda \geq (1 - p)db$$

that is, $p \geq b/(a + b)$.

A Coordination Game (continued)

Analysis (continued):

- Leads to a simple rule:
  - If a fraction of at least $b/(a+b)$ neighbors of v use **A**, then v must also use **A**.
  - Otherwise, v must use **B**.

- The rule is intuitive:
  1. If $b/(a+b)$ is small (say, 1/100):
     - Then b is small and **A** is the "more profitable" behavior.
     - So, a **small** fraction of neighbors adopting **A** is enough for v to change to **A**.
  2. If $b/(a+b)$ is large (say, 99/100):
     - Then b is large and **B** is the "more profitable" behavior.
     - So, a **large** fraction of neighbors adopting **A** is necessary for v to change to **A**.

A Coordination Game (continued)

Note: The quantity $b/(a+b)$ is called the **threshold** for a node to change from **B** to **A**.

Cascading behavior:

- The model has two situations that correspond to **equilibria**.
  - Every node uses **A**.
  - Every node uses **B**.

  In these situation no single node has an **incentive** to change to the other behavior.

Note: These situations are called **pure Nash equilibria** for the game.

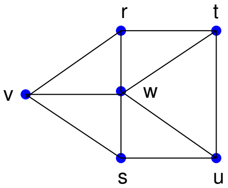
- What happens if some subset of nodes ("early adopters") decide to change their behavior (for reasons outside the definition of the game)?

Cascading Behavior (continued)

Assumptions:

- At the starting point, all nodes use **B**.
- Some nodes change to **A**.
- Other nodes evaluate their payoffs and switch to **A** if it is more profitable.
- For simplicity, the system is assumed to be **progressive**; that is, once a node switches to **A**, it won't switch back to **B**.

Equilibrium configuration:

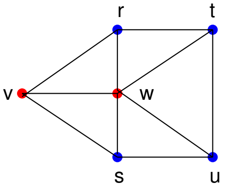


- Payoffs: $a = 3$ and $b = 2$.
- Threshold for switching from **B** to **A** $= b/(a+b) = 2/5$.
- **Notation:** **Blue** represents **B** and **red** represents **A**.

- At some time point ($t = 0$), suppose nodes v and w switch to **A**.

Cascading Behavior (continued)

Configuration at $t = 0$:



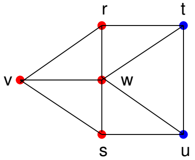
- **Note:** Threshold for switching from **B** to **A** $=$ $2/5$.

Analysis:

- Node r has $2/3$ of its neighbors using **A**. Since $2/3 > 2/5$, r will switch to **A**.
- Node s also has $2/3$ of its neighbors using **A**. So, s will also switch to **A**.
- Node t has $1/3$ of its neighbors using **A**. Since $1/3 < 2/5$, t **won't** switch to **A**.
- Node u also has $1/3$ of its neighbors using **A**. So, u **won't** switch to **A**.

Cascading Behavior (continued)

Configuration at $t = 1$:

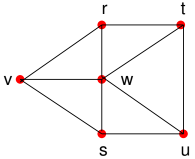


- **Note:** Threshold for switching from **B** to **A** $=$ 2/5.

Analysis:

- Now, node t has 2/3 of its neighbors using **A**. Since $2/3 > 2/5$, t will switch to **A**.
- Node u also has 2/3 of its neighbors using **A**. So, u will also switch to **A**.

Configuration at $t = 2$:



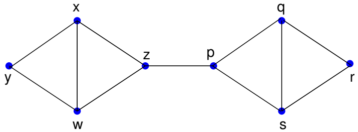
- The system has reached the other equilibrium.

Cascading Behavior (continued)

Notes:

- In the example, there was a **cascade** of switches that resulted in all nodes switching to **A**.
- The example shows **complete cascade**.
- Cascades may also be **partial** as shown by the following example.

Equilibrium configuration:

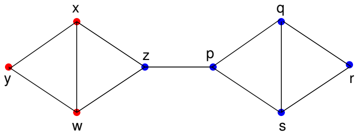


- Payoffs: $a = 3$, $b = 2$.
- Threshold for switching from **B** to **A** $= 2/5$.

- At some time point ($t = 0$), suppose nodes x, y and w switch to **A**.

Cascading Behavior (continued)

Configuration at $t = 0$:



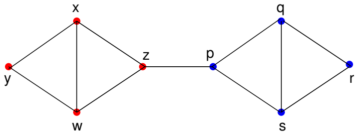
- **Note:** Threshold for switching from **B** to **A** $= 2/5$.

Analysis:

- Node z has $2/3$ of its neighbors using **A**. Since $2/3 > 2/5$, z will switch to **A**.
- Nodes p, q, r and s have **zero** neighbors using **A**. So, **none** of them will switch to **A**.

Cascading Behavior (continued)

Configuration at $t = 1$:



- **Note:** Threshold for switching from **B** to **A** $= 2/5$.

Analysis:

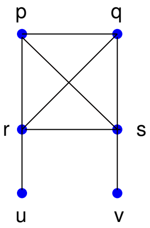
- Node p has $1/3$ of its neighbors using **A**. Since $1/3 < 2/5$, p **won't** switch to **A**.
- Nodes q, r and s have **zero** neighbors using **A**. So, **none** of them will switch to **A**.
- Thus, the configuration shown above is another **equilibrium** for the system.
- Here, the cascade is **partial**.

Cascading Behavior (continued)

Brief digression – A non-progressive system:

- A node may switch from **A** to **B** or vice versa.

Example – Equilibrium configuration:

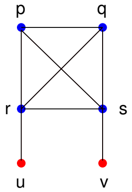


- Payoffs: $a = 3$ and $b = 2$.
- Threshold for switching from **B** to **A** $= 2/5$.

- At some time point ($t = 0$), suppose nodes u and v switch to **A**.

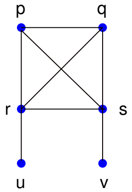
A Non-progressive System (continued)

Configuration at $t = 0$:



- Nodes p and q have **zero** neighbors using **A**. So, they **won't** switch to **A**.
- Nodes r and s have only $1/4$ of their neighbors using **A**. So, they **won't** switch to **A**.

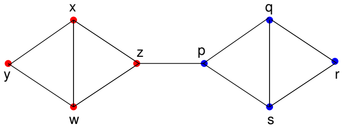
- The only neighbor of node u uses **B**. So, it is more profitable for u to switch back to **B**.
- For the same reason, it is more profitable for v to switch back to **B**.



- So, the system switches back to the previous equilibrium configuration.
- There is no cascade here.

Obstacles to Cascades (Progressive Systems)

Example: The cascade stopped in the following network.



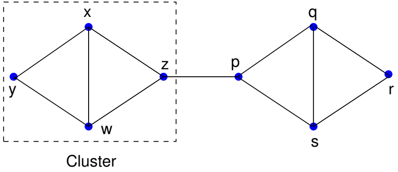
- Threshold for switching from **B** to **A** $= 2/5$.

- The cascade didn't spread to nodes p, q, r and s.
- The situation can be explained formally.

Definition: Given an undirected graph $G(V, E)$, a subset $V_1 \subseteq V$ of nodes forms a **cluster** of **density** α if for every node $v \in V_1$, at least a fraction α of the neighbors of v in G are in V_1.

Obstacles to Cascades (continued)

Example: (Density of a cluster)



- Let $V_1 = \{x, y, z, w\}$.
- For x, y and w, all their neighbors are in V_1. (So, fraction of neighbors in $V_1 = 1$.)
- For z, a fraction $2/3$ of its neighbors are in V_1.
- So, density of the cluster formed by $V_1 = 2/3$.

Note: Density of a cluster is determined by the **smallest** fractional value among the nodes in the cluster.

Obstacles to Cascades (continued)

Brief discussion on clusters and their densities:

- The notion of clusters suggests some level of internal "cohesion"; that is, for all the nodes in the cluster, a specified fraction of their neighbors are also in the cluster.
- However, high cluster density **doesn't** mean that two nodes in the same cluster have much in common.

  Reason: If we consider the whole graph, it forms a cluster of density 1. (This holds even when the graph is disconnected.)

Obstacles to Cascades (continued)

Theorem: [due to Stephen Morris]

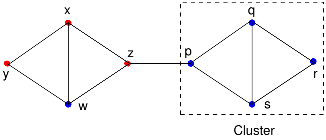
Suppose $G(V, E)$ is a network where each node is using behavior **B**. Let $V' \subseteq V$ be a subset of "early adopters" of behavior **A**. Further, let α be threshold for the other nodes to switch from **B** to **A**.

1. If the subnetwork of G formed on the remaining nodes (i.e., $V - V'$) has a cluster of density $> (1 - \alpha)$, then V' **won't** cause a complete cascade.
2. If V' does not cause a complete cascade, then the subnetwork on the remaining nodes **must** contain a cluster of density $> (1 - \alpha)$.

Interpretation:

- Part 1: Clusters of density $> (1 - \alpha)$ act as "obstacles" to a complete cascade.
- Part 2: Clusters of density $> (1 - \alpha)$ are the **only** "obstacles" to a complete cascade.

An Example for Morris's Theorem



- Recall: Threshold α for **B** to **A** switch $= 2/5$.
- Let $V' = \{x, y, z\}$ be the "early adopters".

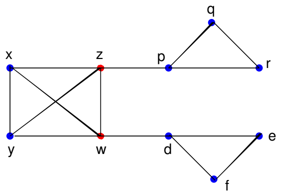
- Consider $V_1 = \{p, q, r, s\}$.
- For q, r and s, all their neighbors are in V_1. (So, fraction of neighbors in $V_1 = 1$.)
- For p, a fraction $2/3$ of its neighbors are in V_1.
- So, density of the cluster formed by $V_1 = 2/3$.
- Note that $1 - (2/5) = 3/5$ and $2/3 > 3/5$.
- So, the cascade cannot be complete.

Diffusion and Weak Ties

Recall:

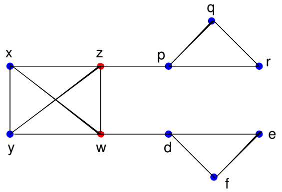
- A **local bridge** is an edge $\{x, y\}$ such that x and y don't have any neighbor in common.
- Local bridges are weak ties but enable nodes to get information from other parts of the network ("strength of weak ties").

Do local bridges help in the diffusion of behavior?



- Edges $\{z, p\}$ and $\{w, d\}$ are local bridges.
- Let threshold for switching be 2/5.
- Let z and w be the "early adopters".

Diffusion and Weak Ties (continued)



- Nodes x and y will switch to **A**.
- However, none of the other nodes will switch.

- Local bridges are "too weak" to propagate behaviors that require higher thresholds.
- If threshold for each node v is set to $1/\mathrm{degree}(v)$, then there will be a complete cascade (**low threshold**).
- The concept of thresholds provides one way to explain why **information** (e.g. jokes, link to videos, news) spreads to a much larger population compared to behaviors such as **political mobilization**.

Homogeneous and Heterogeneous Thresholds

- In the coordination game, all the nodes had the same threshold value (**homogeneous thresholds**).
- In the context of weak ties, using a different threshold for each node can cause a complete cascade (**heterogeneous thresholds**).
- Heterogeneous thresholds can also arise in the coordination game: choose a different payoff for each node.

x \ y	A	B
A	a_x, a_y	0, 0
B	0, 0	b_x, b_y

- If x and y both adopt **A**, x gets a_x and y gets a_y.
- If x and y both adopt **B**, x gets b_x and y gets b_y.
- If x and y **don't** adopt the same behavior, their benefit is **zero**.

Homogeneous and Heterogeneous Thresholds (continued)

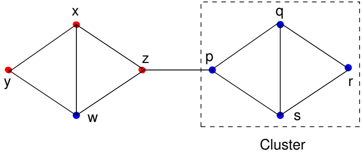
- The threshold for any node v (to switch from **B** to **A**) is $b_v/(a_v + b_v)$. (Thus, each node may have a different threshold.)

Definition: **(Blocking Cluster)**

Consider a network $G(V, E)$ where each node v has a threshold α_v. A subset $V_1 \subseteq V$ of nodes is a **blocking cluster** if for every node $v \in V_1$, **more than** $1 - \alpha_v$ fraction of the neighbors of v are in V_1.

Note: This generalizes the notion of a cluster defined in the homogeneous case.

Example 1: (Blocking Cluster)

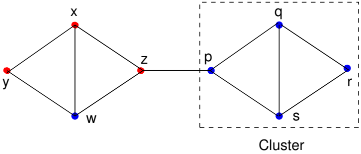


- Let $\alpha_p = 1/2$ and $\alpha_q = \alpha_r = \alpha_s = 2/5$.

- Consider the cluster $V_1 = \{p, q, r, s\}$.
  - For p, $1 - \alpha_p = 1/2$, the fraction of neighbors in $V_1 = 2/3$ and $2/3 > 1/2$.
  - For the nodes q, r and s, all their neighbors are in V_1.
  - So, V_1 **is** a blocking cluster.

Homogeneous and Heterogeneous Thresholds (continued)

Example: (continued)



- Let $\alpha_p = 1/6$ and $\alpha_q = \alpha_r = \alpha_s = 2/5$.

- The only change is that $\alpha_p = 1/6$ (instead of $1/2$).
  - For p, $1 - \alpha_p = 5/6$ and the fraction of neighbors in $V_1 = 2/3$. However, $2/3 < 5/6$.
  - So, V_1 is **not** a blocking cluster with the new threshold value for p.
  - Easy to verify that $V_2 = \{q, r, s\}$ is still a blocking cluster.

Generalization of Morris's Theorem:

Suppose $G(V, E)$ is a network where each node v has a threshold α_v. Let $V' \subseteq V$ be the "early adopters".

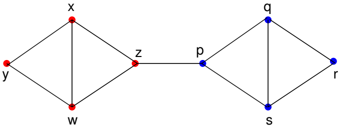
1. If the subnetwork of G formed on the remaining nodes (i.e., $V - V'$) has a blocking cluster, then V' **won't** cause a complete cascade.

2. If V' does not cause a complete cascade, then the subnetwork on the remaining nodes **must** contain a blocking cluster.

Cascades and Viral Marketing

Note: Think of **A** and **B** as competing products.

Example with a partial cascade:



- Threshold for switching from **B** to **A** $= 2/5$.

- **A** didn't propagate to the cluster $\{p, q, r, s\}$ at the threshold value of $2/5$.

- What can the marketing agency for **A** do?

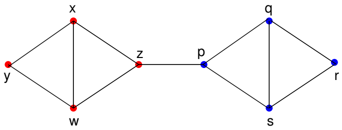
  1. Try to decrease the threshold.
  2. Try to choose the early adopters carefully.

Cascades and Viral Marketing (continued)

1. Decreasing the threshold:
   - Formula for threshold $= b/(a+b)$.
   - With $a = 3$ and $b = 2$, threshold $= 2/5$.
   - The threshold can be decreased by **increasing** a; that is, by improving the quality of **A**.
   - **Example:** Let $a = 4$ while b remains at 2.
   - New threshold $= 2/(4+2) = 1/3$.
   - This threshold causes a complete cascade.
     (See the next two slides).

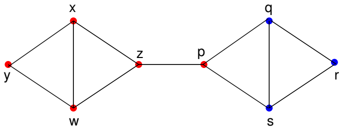
Cascades and Viral Marketing (continued)

Configuration at $t = 0$:



- Threshold for switching from **B** to **A** $= 1/3$.

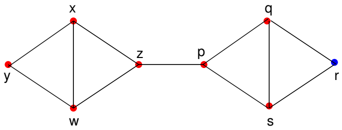
Configuration at $t = 1$:



- Node p switched from **B** to **A**.

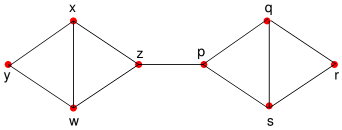
Cascades and Viral Marketing (continued)

Configuration at $t = 2$:



- Nodes q and s switched from **B** to **A**.

Configuration at $t = 3$:



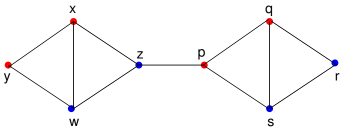
- Node r switched from **B** to **A**.
- The cascade is complete.

Cascades and Viral Marketing (continued)

2. Choose early adopters carefully.

   - With $\{x, y, z\}$ as the early adopters, the cascade is partial.
   - Suppose the early adopters are $\{x, y, p, q\}$.

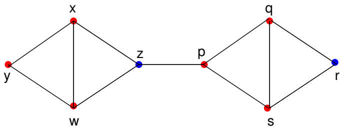
Configuration at $t = 0$:



- Threshold for switching from **B** to **A** $= 2/5$.
- This set of early adopters will cause a complete cascade. (See the next slide.)

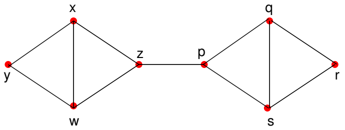
Cascades and Viral Marketing (continued)

Configuration at $t = 1$:



- Nodes w and s switched from **B** to **A**.

Configuration at $t = 2$:



- Nodes z and t switched from **B** to **A**.
- The cascade is complete.

Cascades and Viral Marketing (continued)

Notes on Viral Marketing:

- Marketing units can only choose a limited number of early adopters due to budget constraints.
- **Influence Maximization Problem:**
  - **Given:** A social network $G(V, E)$, a threshold value α and a budget on the number of early adopters N.
  - **Required:** Find a subset of V with at most N nodes (the early adopters) so that a maximum number of nodes change to **A**.
- The problem is known to be computationally difficult (**NP**-hard).
- The problem has also been studied under other models (e.g. probabilistic switches).

Towards a More General Model for Diffusion

Features of the current model:

1. A social network where the interaction is between a node and its neighbors (**local interactions**).
2. The current configuration of the system (i.e., the current behavior of each node).
3. A threshold value. (This was chosen based on the coordination game.)
4. A scheme for nodes to evaluate their payoffs and decide whether or not to switch behaviors (**synchronous evaluation and update**).

Towards a More General Model for Diffusion

Why generalization is useful:

- There are several diffusion phenomena (e.g. disease propagation) where there is no underlying game with payoffs.
- The decision to switch may involve more complex computations.

  Example: Most disease propagation models are probabilistic.

- The generalization also allows precise formulations of several other problems related to diffusion.

Note: The generalized model is called a **Synchronous Dynamical System** (or SyDS).

Components of a Synchronous Dynamical System

1. An undirected graph $G(V, E)$. (In most applications, this graph represents a **social contact network**.)

2. Each node v has **state** value, denoted by $s(v)$.

   - The state value is from a specified set (**domain**).
   - A typical example is the Boolean domain $\{0, 1\}$.
   - In some disease models, the domain is larger.
   - The interpretation of the state value depends on the application.

Components of a SyDS (continued)

Interpretation of state values in some applications:

(a) Coordination game: Values 0 and 1 represent behaviors **A** and **B** respectively.

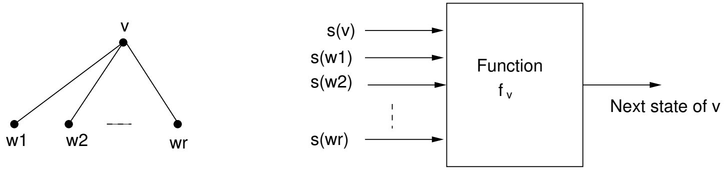
(b) Simple disease models: Value 0 $\Rightarrow$ node is **uninfected** and 1 $\Rightarrow$ node is **infected**.

(c) Information propagation: Value 0 $\Rightarrow$ node **does not have** the information and 1 $\Rightarrow$ node **has** the information.

(d) Complex disease models: State values represent different **levels of infection**.

Components of a SyDS (continued)

3. A **local function** f_v for each node v of the graph. (This function captures the local interactions between a node and its neighbors.)

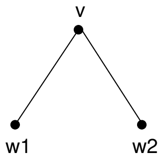


Notes:

- The inputs to the function f_v are the current state of node v and those of its neighbors.
- The value computed by the function f_v gives the state value of v for the **next** time instant.

Components of a SyDS (continued)

Example of a local function: Assume that the domain is $\{0, 1\}$.



$s(v)$	$s(w_1)$	$s(w_2)$	f_v
0	0	0	0
0	0	1	0
0	1	0	0
0	1	1	1
1	0	0	0
1	0	1	1
1	1	0	1
1	1	1	1

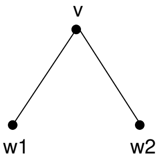
Notes:

- The above specification is a **truth table** for f_v.
- When a node has degree r, the truth table specifying f_v will have 2^{r+1} rows. (This is **exponential** in the degree of node v.)
- This is not practical for nodes of large degree.

Components of a SyDS (continued)

A more common local function: The domain is $\{0, 1\}$.

- For each node v, an integer **threshold** value τ is specified. (The value of τ may vary from node to node.)
- The function f_v has the value 1 if the number of 1's in the input is at least τ; it is 0 otherwise.
- This function is called the τ-**threshold** function.
- If v has degree d, then the τ-threshold function can be represented using a table with $d + 2$ rows.



No. of 1's	Value of f_v
0	0
1	0
2	1
3	1

A 2-threshold function

Absolute and Relative Thresholds

- In the definition of τ-threshold functions, the value τ specifies an **absolute threshold**.
- The threshold value specified in the coordination game is called a **relative threshold**; this is a fraction relative to the degree of the node.
- Any relative threshold can be converted into a corresponding absolute threshold and vice versa.

Example: Suppose a node v has a degree of 9. (So, the number of inputs to the function $f_v = 10$.)

- If f_v is specified by the absolute threshold value 3, then the relative threshold value is $3/10 = 0.3$.
- If f_v is specified using the relative threshold value $1/3$, the absolute threshold value is $\lceil 10 \times (1/3) \rceil = 4$.

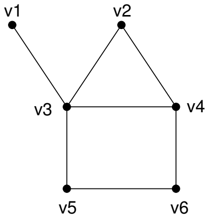
Other Definitions and Conventions in SyDSs

- A SyDS uses **synchronous computation and update**.
  - All nodes compute the values of their local functions **synchronously** (i.e., in parallel).
  - After all the computations are finished, all the nodes update their state values synchronously.
- The synchronous computation and update proceeds until the system reaches an **equilibrium**, where no further state changes occur.
- In a **progressive** SyDS over the Boolean domain, states of nodes may be change from 0 to 1; however, the states **cannot** change from 1 to 0.

  Consequence: In a progressive SyDS, once the state of node becomes 1, it remains at 1 for ever.
- In the discussion on SyDSs, local functions will be specified using **absolute** thresholds.

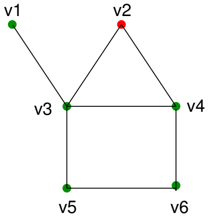
An Example of a SyDS

Example 1:



- Domain $= \{0, 1\}$.
- Each local function is the 1-threshold function (**simple contagion**).
- Note that the state of a node can't change from 1 to 0; the system is **progressive**.

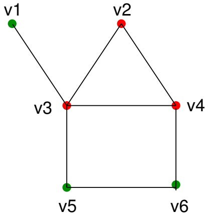
Configuration at $t = 0$:



- **Green** indicates state value 0.
- **Red** indicates state value 1.
- The configuration at $t = 0$ can also be represented as $(0,\ 1,\ 0,\ 0,\ 0,\ 0)$.

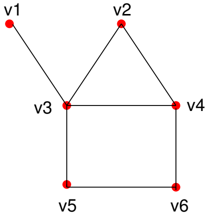
An Example of a SyDS (continued)

Configuration at $t = 1$:



- Nodes v_3 and v_4 switched from 0 to 1.
- The configuration at $t = 1$: $(0,\ 1,\ 1,\ 1,\ 0,\ 0)$.

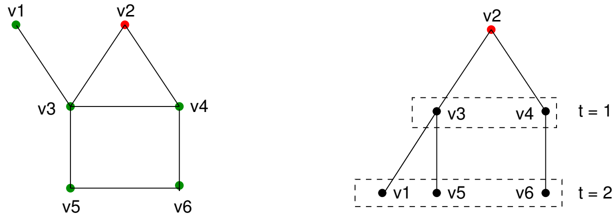
Configuration at $t = 2$:



- Nodes v_1, v_5 and v_6 switched from 0 to 1.
- The configuration at $t = 2$: $(1,\ 1,\ 1,\ 1,\ 1,\ 1)$.
- The cascade is complete.

Why did we get a complete cascade?

Explanation 1:



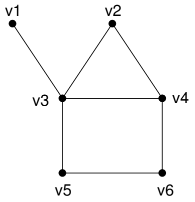
- Since the graph is connected, there is a path from node v_2 (the "early adopter") to every other node.

- So, if the interaction graph is connected, a simple contagion always results in a complete cascade.

Note: The order in which nodes change to state 1 is given by breadth-first search (BFS) starting from the set of early adopters.

Another Example of a SyDS

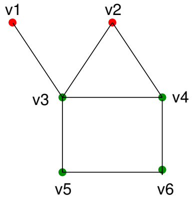
Example 2:



- Domain $= \{0, 1\}$.
- Each local function is the 2-threshold function.
- We will assume that the system is **progressive** (i.e., the state of a node **can't** change from 1 to 0).

Note: If at least one of the thresholds is > 1, the system models a **complex contagion**.

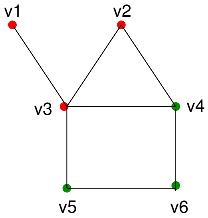
Configuration at $t = 0$:



- The configuration at $t = 0$ is (1, 1, 0, 0, 0, 0).

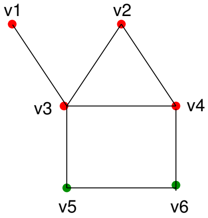
A Second Example of a SyDS (continued)

Configuration at $t = 1$:



- Node v_3 switched from 0 to 1.
- The configuration at $t = 1$: $(1, 1, 1, 0, 0, 0)$.

Configuration at $t = 2$:

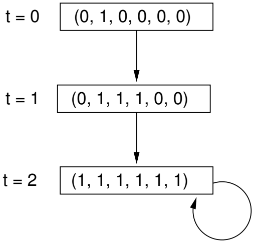


- Node v_4 switched from 0 to 1.
- The configuration at $t = 2$: $(1, 1, 1, 1, 0, 0)$.
- No further state changes can occur; the system has reached an **equilibrium** (**fixed point**).
- The cascade is **partial**.

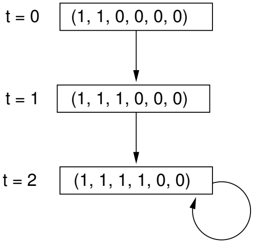
Phase Space of a SyDS

Sequences of configurations:

Example 1



Example 2



- For any SyDS, we can construct these sequences starting from any initial configuration.
- The collection of all such sequences forms the **phase space** of a SyDS.

Phase Space of a SyDS (continued)

Definition: The **phase space** of a SyDS is a **directed** graph where

- each node represents a configuration and
- for any two nodes x and y, there is a directed edge (x, y) if the configuration represented by x changes to that represented by y in **one time step**.

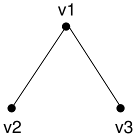
Comment: The phase space may have self-loops.

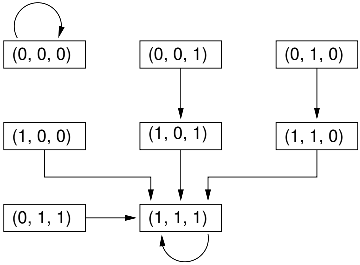
How Large is the Phase Space? (Assume that the Domain is $\{0, 1\}$.)

- If the underlying network of the SyDS has n nodes, then the number of nodes in the phase space $= 2^n$; that is, the size of the phase space is **exponential** in the number of nodes.
- For the SyDSs considered so far (**deterministic** SyDSs), each node in the phase space has an outdegree of 1. (So, the number of edges in the phase space is also 2^n.)

Phase Space of a SyDS (continued)

Example – A SyDS and its Phase Space: The domain is $\{0, 1\}$ and each node has a 1-threshold function.

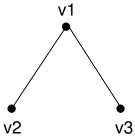


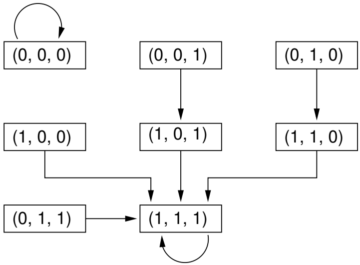


Notes:

- **Fixed points:** (0, 0, 0) and (1, 1, 1).
- The configuration (1, 1, 0) is the **successor** of (0, 1, 0). (Each configuration has a **unique** successor.)

Phase Space of a SyDS (continued)





Notes (continued):

- The configuration (1, 1, 0) is a **predecessor** of (1, 1, 1). (A configuration may have **zero or more** predecessors.)
- The configuration (1, 0, 0) **doesn't** have a predecessor. It is a **Garden of Eden** configuration.

Some Known Results Regarding SyDSs

- Every progressive SyDS has a fixed point. (If the underlying network has n nodes, the system reaches a fixed point in at most n time steps.)

- In general, the following problems for SyDSs are computationally intractable:

  - **(Fixed Point Existence)** Given a SyDS $\mathcal{S}$, does $\mathcal{S}$ have a fixed point?

  - **(Predecessor Existence)** Given a SyDS $\mathcal{S}$ and a configuration $\mathcal{C}$, does $\mathcal{C}$ have a predecessor?
  - **(Garden of Eden Existence)** Given a SyDS $\mathcal{S}$, does $\mathcal{S}$ have a Garden of Eden configuration?
  - **(Reachability)** Given a SyDS $\mathcal{S}$ and two configurations $\mathcal{C}_1$ and $\mathcal{C}_2$, does $\mathcal{S}$ starting from $\mathcal{C}_1$ reach $\mathcal{C}_2$?

Zero and Infinite Threshold Values

Assumption: The domain is $\{0, 1\}$.

Zero Threshold:

- A node with **zero** threshold changes from 0 to 1 at the first possible opportunity; it **won't** change back to 0.
- Useful in modeling early adopters.

Infinite Threshold:

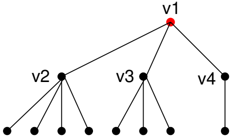
- A node with **infinite** threshold will stay at 0.
- For a node of degree d, setting its threshold to $d + 2$ will ensure that property.
- Useful in several applications.
  - **Opinion propagation:** Nodes with infinite thresholds model "stubborn" people.
  - **Disease propagation:** Nodes with infinite thresholds model nodes which have been vaccinated (so that they will never get infected).

Some Applications of the Model

Blocking Disease Propagation:

- **Given:** A social network, local functions that model disease propagation, the set of initially infected nodes and a budget β on the number of people who can be vaccinated.
- **Goal:** Vaccinate at most β nodes of the network so that the number of new infections is **minimized**.

Example:



- Assume that threshold for each node is 1.
- If the vaccination budget is 2, then nodes v_2 and v_3 should be chosen.

Some Applications of the Model (continued)

Some Results on Blocking Disease Propagation:

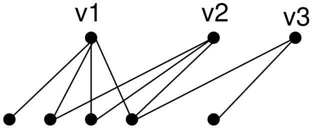
- For simple contagions (or when the graph has some special properties), the blocking problem can be solved efficiently.
- For complex contagions, the blocking problem is computationally intractable. (Even obtaining near-optimal solutions is computationally intractable.)
- Many algorithms that work well on large networks are available.
- The problem has also been investigated under probabilistic disease transmission models.

Some Applications of the Model

Viral Marketing:

- **Given:** A social network, local functions that model propagation of behavior and a budget β on the number of initial adopters.
- **Goal:** Choose a subset of at most β initial adopters so that the number of nodes to which the behavior propagates is **maximized**.

Example:



- Suppose $\beta = 2$.
- If the threshold for each node is 1, the solution is $\{v_1, v_3\}$.
- If the threshold for each node is 2, the solution is $\{v_1, v_2\}$.

Some Applications of the Model (continued)

Some Results on Viral Marketing:

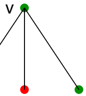
- For simple contagions (or when the graph has some special properties), the viral marketing problem can be solved efficiently.
- For complex contagions, the problem is computationally intractable. (However, near-optimal solutions can be obtained efficiently.)
- The problem has been studied extensively under various propagation models (including probabilistic models).

A Bi-threshold Model

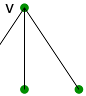
- Models for some social phenomena require "back and forth" state changes (i.e., changes from 0 to 1 as well as 1 to 0).
- **Examples:** Smoking, Drinking, Dieting.
- The **bi-threshold** model was proposed to address such behaviors.
- Each node v has **two** threshold values, denoted by T_v^1 (the **up threshold**) and T_v^0 (the **down threshold**).
  - If the current state of v is 0 and at least T_v^1 neighbors of v are in state 1, then the next state of v is 1; otherwise, the next state of v is 0.
  - If the current state of v is 1 and at least T_v^0 neighbors of v are in state 0, then the next state of v is 0. otherwise, the next state of v is 1.

A Bi-threshold Model (continued)

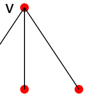
Examples: Assume that T_v^1 (the up threshold) is 2 and T_v^0 (the down threshold) is 1. (Also, **green** and **red** represent states 0 and 1 respectively.)



- The state of v will change to 1.



- The next state of v is also 0.



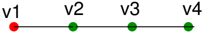
- The state of v will change to 0.

A Bi-threshold Model (continued)

Example – A bi-threshold SyDS:

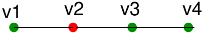
- For each node, the up and down threshold values are 1.

Configuration at $t = 0$:



- States of v_1 and v_2 will change.

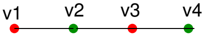
Configuration at $t = 1$:



- States of v_1, v_2 and v_3 will change.

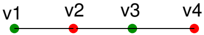
A Bi-threshold Model (continued)

Configuration at $t = 2$:



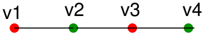
- States of v_1, v_2 and v_3 will change.

Configuration at $t = 3$:



- States of all the nodes will change.

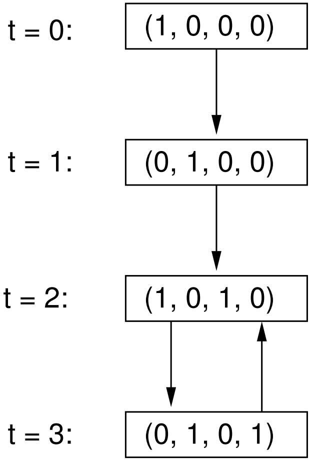
Configuration at $t = 4$:



- States of all the nodes will change.

Note: From this point on, the system goes back and forth between the two configurations for $t = 2$ and $t = 3$.

Bi-threshold System: Partial Phase Space



Note: The phase space contains a (directed) cycle of length 2.

SyDSs with Probabilistic Threshold Functions

- In general, diffusion is a probabilistic phenomenon.
- Even if the threshold is met, a person may decide not to change his/her behavior.
- Probabilistic threshold functions provide a way to model this uncertainty.

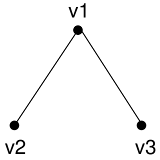
Probabilistic Thresholds: [Barrett et al. 2011]

- Domain = $\{0, 1\}$.
- For each node v, a threshold τ_v and a probability p_v are given.
- If the number of 1's in the input to f_v is $< \tau_v$, the next state of $v = 0$.
- If the number of 1's in the input to f_v is $\geq \tau_v$:
  - The next state of v is 1 with probability p_v and 0 with probability $1 - p_v$.
- This generalizes the deterministic case (where $p_v = 1$).

SyDSs with Probabilistic ... (continued)

Assumption: Nodes make **independent** choices.

Example:

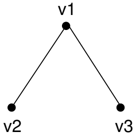


- Assume that each node has a threshold of 1 and probability of 3/4.

Table specifying local function f_1 (for v_1):

No. of 1's in the input	$\Pr\{s(v_1) = 1\}$
0	0
1	3/4
2	3/4
3	3/4

Computing the transition probability – Example 1:



- Each node has a threshold of 1 and probability of 3/4.
- Let the current configuration $\mathcal{C}_1$ be $(1, 0, 0)$.
- **Goal:** To compute the probability that the next configuration is $\mathcal{C}_2 = (1, 0, 1)$.

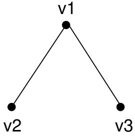
Steps: Note that in $\mathcal{C}_1$, the thresholds for all three nodes are satisfied.

- The probability that v_1 remains 1 is 3/4.
- The probability that v_2 remains 0 is 1/4.
- The probability that v_3 changes to 1 is 3/4.
- So, the probability of transition from $\mathcal{C}_1$ to $\mathcal{C}_2$ is

$$(3/4) \times (1/4) \times (3/4) \ = \ 9/64.$$

SyDSs with Probabilistic ... (continued)

Computing the transition probability – Example 2:



- Each node has a threshold of 1 and probability of 3/4.
- Let the current configuration $\mathcal{C}_1$ be $(0, 0, 1)$.
- **Goal:** To compute the probability that the next configuration is $\mathcal{C}_2 = (0, 1, 1)$.

Steps:

- In $\mathcal{C}_1$, the thresholds are satisfied for v_1 and v_3 but **not** for v_2.
- Thus, the probability that v_2 changes to 1 is 0.
- So, the probability of transition from $\mathcal{C}_1$ to $\mathcal{C}_2$ is $= 0$.

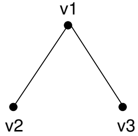
SyDSs with Probabilistic ... (continued)

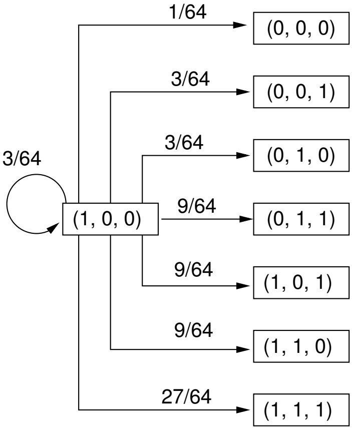
Phase Space with Probabilistic Transitions:

- There is a node for each configuration.
- The is a directed edge from node x to node y if the probability of transition from x to y (in one step) is **positive**.
- The probability value is indicated on the edge.
- The outdegree of each node may be (much) larger than 1.
- This represents the **Markov Chain** for the diffusion process.

SyDSs with Probabilistic ... (continued)

Example – A Part of the Phase Space:





Note: For each node,

- threshold = 1 and
- probability = 3/4.

Note: For each node, the sum of the probability values on the outgoing edges must be 1.

Some Known Results Regarding Probabilistic SyDSs

The following problems for probabilistic SyDSs are computationally intractable **[Barrett et al. 2011]**.

- **(Fixed Point Existence)** Given a probabilistic SyDS $\mathcal{S}$ and a probability value p, is there a configuration $\mathcal{C}$ such that $\mathcal{C}$ is its own successor with probability $\geq p$?
- **(Predecessor Existence)** Given a SyDS $\mathcal{S}$, a configuration $\mathcal{C}_1$ and a probability p, is there a configuration $\mathcal{C}_0$ such that the probability of transition from $\mathcal{C}_0$ to $\mathcal{C}_1$ is $\geq p$?
- **(Reachability)** Given a SyDS $\mathcal{S}$, two configurations $\mathcal{C}_1$ and $\mathcal{C}_2$ and a probability value p, does $\mathcal{S}$ starting from $\mathcal{C}_1$ reach $\mathcal{C}_2$ with probability $\geq p$?

The SIR Epidemic Model

Basics of the SIR Model:

- Proposed by William Kermack and Anderson McKendrick in 1927.
- Effective in the study of several diseases that affect humans.
- Each individual may be in one of the following three states:
  - **Susceptible** (denoted by $\mathbb{S}$),
  - **Infected** (denoted by $\mathbb{I}$) or
  - **Recovered** (denoted by $\mathbb{R}$).
- For any individual, the sequence of states is as follows:

$$\mathbb{S} \longrightarrow \mathbb{I} \longrightarrow \mathbb{R}$$

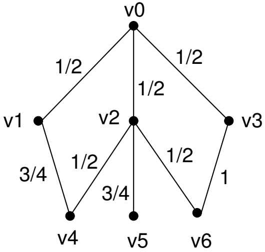
So, the system is **progressive**.

The SIR Epidemic Model (continued)

Basics of the SIR Model (continued):

- An individual remains in state $\mathbb{I}$ for a certain period (usually assumed to be 1) and changes to $\mathbb{R}$.
- Each edge of the network has a probability value (**transmission probability**).
- Nodes in state $\mathbb{R}$ play no further role in transmitting the disease.

Example:



The SIR Epidemic Model (continued)

Notation:

- For any edge $e = \{u, v\}$, the transmission probability of e is denoted by p_e (or $p_{\{u,v\}}$).
- For each node v_i, the set of neighbors of v_i is denoted by N_i.
- For any node v_i, $X_i(t) \subseteq N_i$ denotes the set of neighbors of v_i whose state at time t is $\mathbb{I}$.

Definition of the local function f_i at node v_i:

- If the state of v_i at time t is $\mathbb{R}$, then the state of v_i at time $t+1$ is also $\mathbb{R}$.
- If the state of v_i at time t is $\mathbb{I}$, then the state of v_i at time $t+1$ is $\mathbb{R}$.

The SIR Epidemic Model (continued)

Definition of the local function (continued):

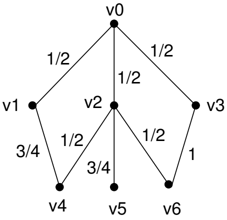
- If the state of v_i at time t is $\mathbb{S}$, then the the state of v_i at time $t+1$ is either $\mathbb{S}$ or $\mathbb{I}$ as determined by the following stochastic process.
  - Define $\pi(i,t)$ as follows:

$$\begin{aligned} \pi(i,t) &= 0 && \text{if } X_i(t) = \emptyset \\ &= 1 - \prod_{u \in X_i(t)} \left(1 - p_{\{u,v_i\}}\right) && \text{otherwise.} \end{aligned}$$

  - The state of v_i is $\mathbb{I}$ with probability $\pi(i,t)$ and $\mathbb{S}$ with probability $1 - \pi(i,t)$.

The SIR Epidemic Model (continued)

Example 1:

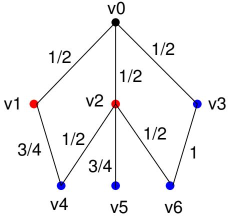


- At $t = 0$, let v_0 be the node in state $\mathbb{I}$. (All other nodes are in state $\mathbb{S}$.)
- **Goal:** To compute the probability that node v_1 gets infected.

- For v_1, the only infected neighbor at $t = 0$ is v_0.
- So, $\Pr\{v_1 \text{ gets infected}\} = 1/2$.
- Similarly, $\Pr\{v_2 \text{ gets infected}\} = 1/2$ and
- $\Pr\{v_3 \text{ gets infected}\} = 1/2$.

The SIR Epidemic Model (continued)

Example 2: System configuration at $t = 1$.



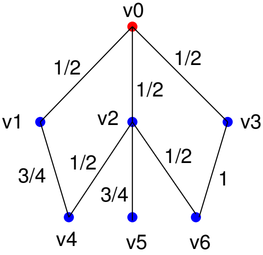
- **Notation:** **Blue**, **Red** and **Black** circles indicate states $\mathbb{S}$, $\mathbb{I}$ and $\mathbb{R}$ respectively.
- **Goal:** To compute the probability that node v_4 gets infected.

- For v_4, the infected neighbors are v_1 and v_2.
- $\Pr\{v_4 \text{ doesn't get infected by } v_1\} = 1 - (3/4) = 1/4$.
- $\Pr\{v_4 \text{ doesn't get infected by } v_2\} = 1 - (1/2) = 1/2$.
- Thus, $\Pr\{v_4 \text{ doesn't get infected}\} = (1/4) \times (1/2) = 1/8$.
- So, $\Pr\{v_4 \text{ gets infected}\} = 1 - (1/8) = 7/8$.

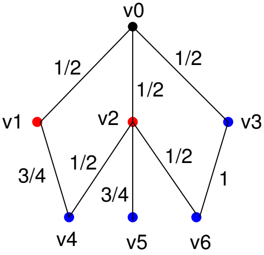
A Possible Sequence of Configurations

Note: **Blue**, **Red** and **Black** circles indicate states $\mathbb{S}$, $\mathbb{I}$ and $\mathbb{R}$ respectively.

Configuration at $t = 0$:



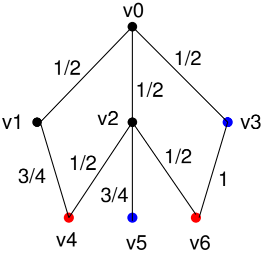
Configuration at $t = 1$:



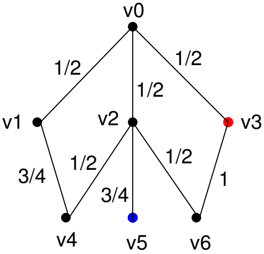
A Possible Sequence of Configurations (continued)

Note: **Blue**, **Red** and **Black** circles indicate states $\mathbb{S}$, $\mathbb{I}$ and $\mathbb{R}$ respectively.

Configuration at $t = 2$:



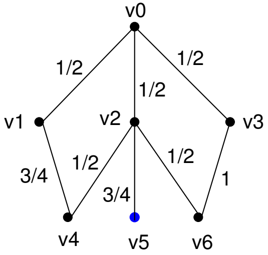
Configuration at $t = 3$:



A Possible Sequence of Configurations (continued)

Note: **Blue**, **Red** and **Black** circles indicate states $\mathbb{S}$, $\mathbb{I}$ and $\mathbb{R}$ respectively.

Configuration at $t = 4$:



- Node v_5 is in state $\mathbb{S}$ while all others are in state $\mathbb{R}$.
- This configuration is a **fixed point**.

SIR Model – Some Known Results

- Every SIR system has a fixed point. (If the underlying network has n nodes, the system reaches a fixed point in at most n time steps.)
- The following problems for the SIR model are computationally intractable:
  - **(Expected Number of Infections)** Given an SIR system and the set of initially infected nodes, compute the expected number of nodes that get infected.
  - **(Node Vulnerability)** Given an SIR system, the set of initially infected nodes and a node v, compute the probability that v gets infected.