Optimally Learning Social Networks with Activations and Suppressions

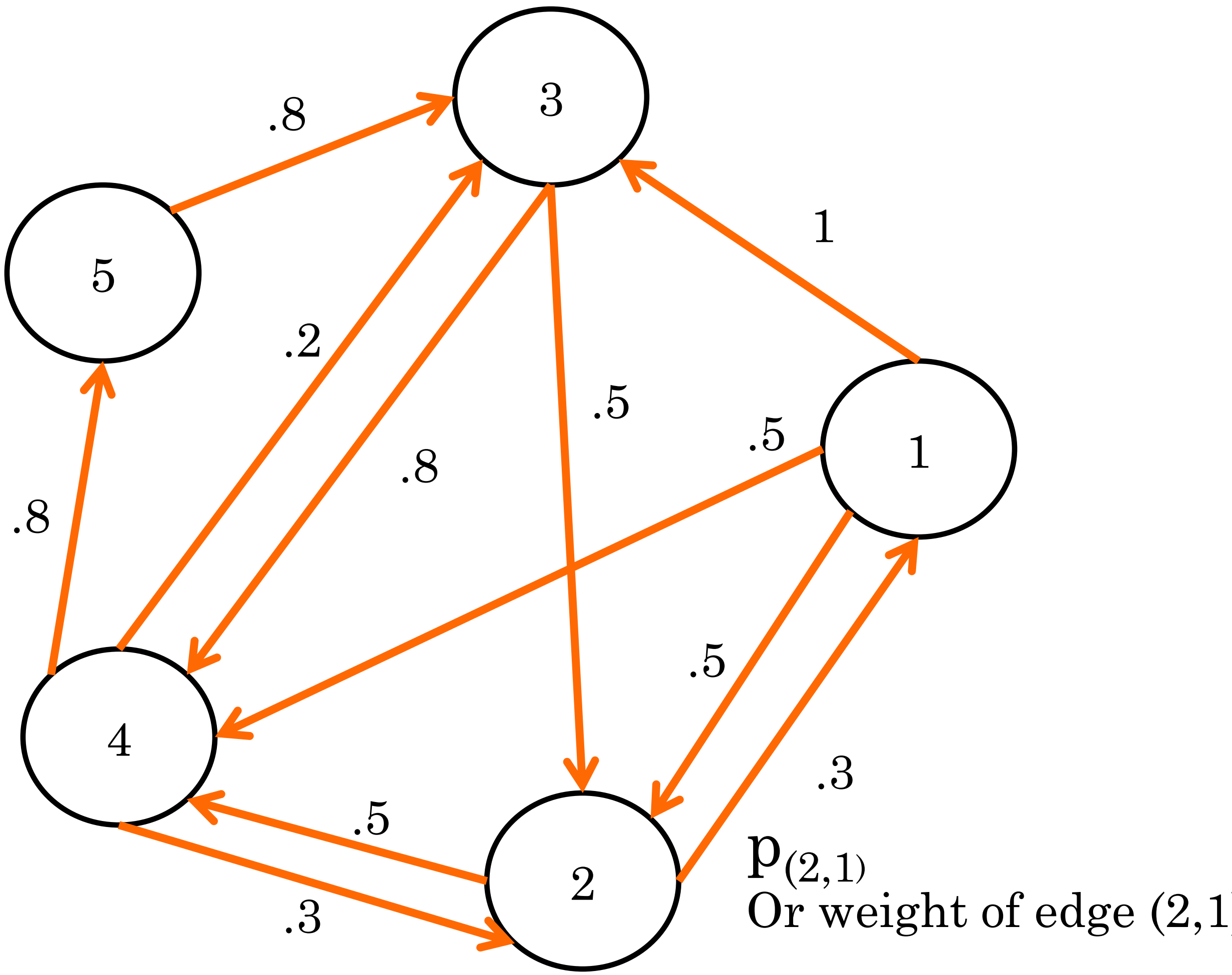
ALT 2008

James Aspnes, Dana Angluin, **Lev Reyzin**
Yale University

Talk Outline

- Independent Cascade Social Networks (SNs)
- Introduce Value Injection Queries (VIQs) on SNs
  - VIQs override the values of node (or gates in the case of circuits)
  - These were introduced by AACW (STOC '06), extended in AACR (COLT '07) and AACER (COLT '08)
- Show lower bound for learning SNs with VIQs
- Give an idea of the algorithm for learning SNs
- Other results
  - Approximation algorithm
  - Special case of trees
  - Exponential attenuation of paths

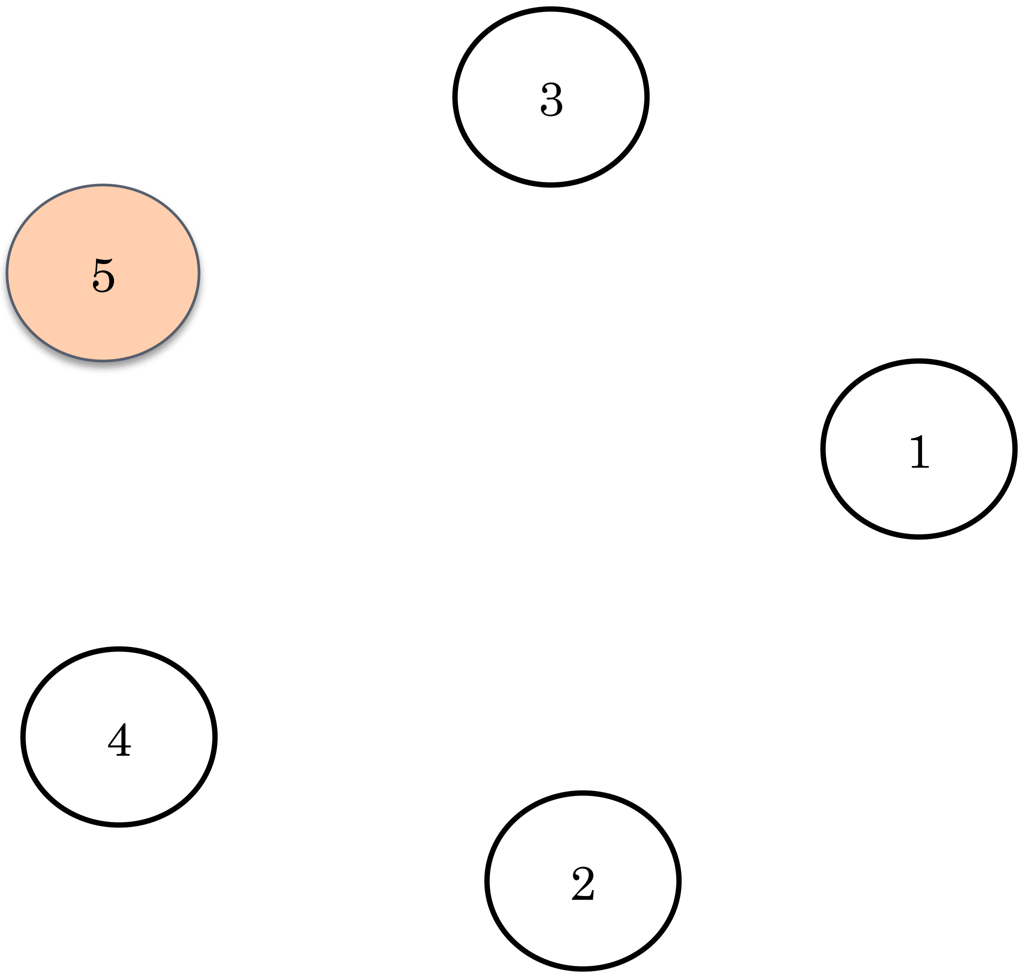
Independent Cascade Social Networks



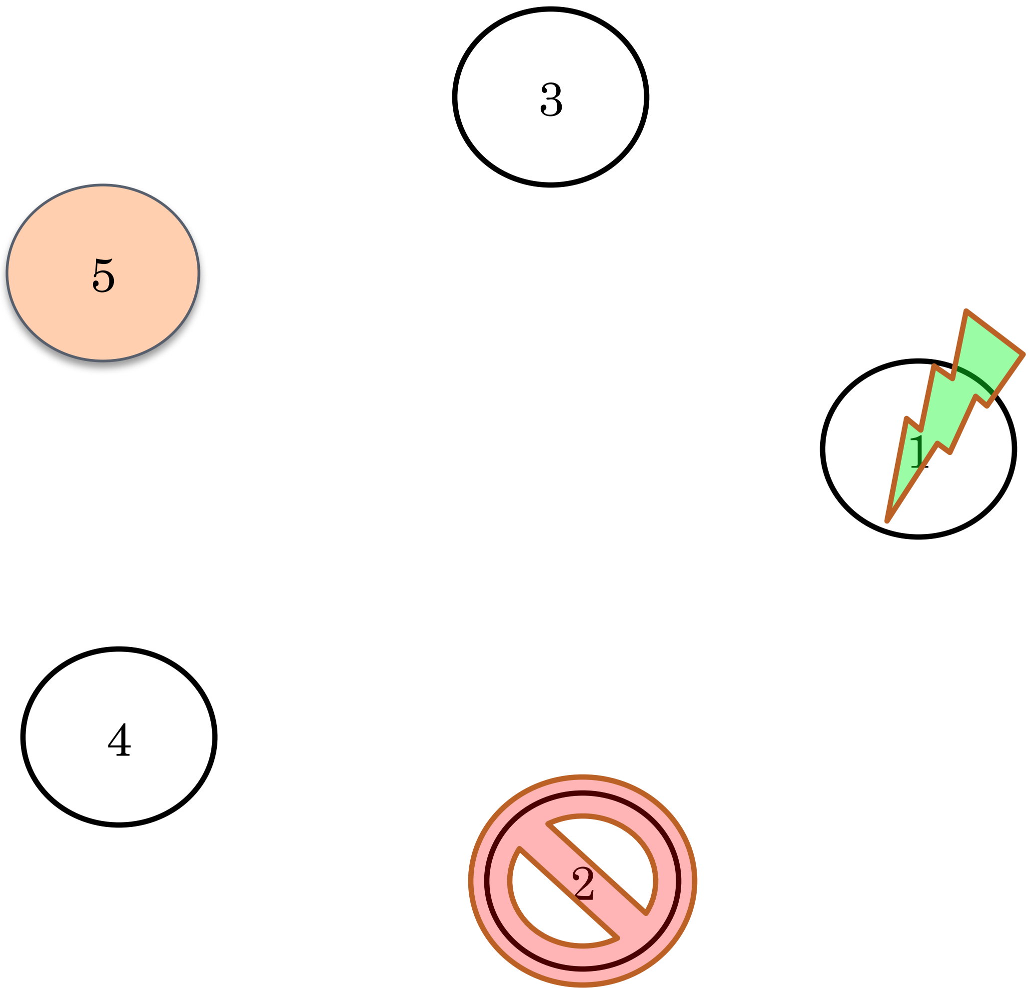
Talk Outline

- Independent Cascade Social Networks (SNs)
- Introduce Value Injection Queries (VIQs) on SNs
  - These were introduced by AACW (STOC '06), extended in AACR (COLT '07) and AACER (COLT '08)
- Show lower bound for learning SNs with VIQs
- Give an idea of the algorithm for learning SNs
- Other results
  - Approximation algorithm
  - Special case of trees
  - Exponential attenuation of paths

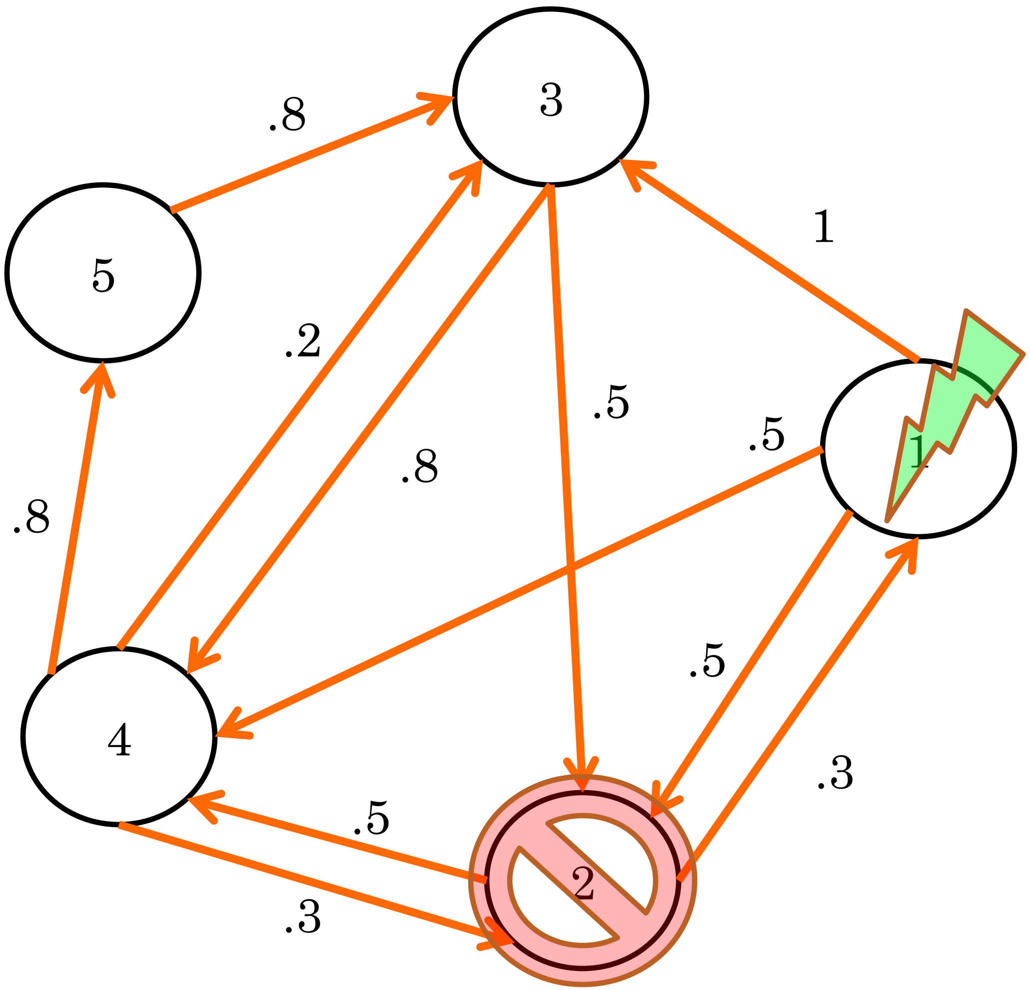
What the Learner Sees



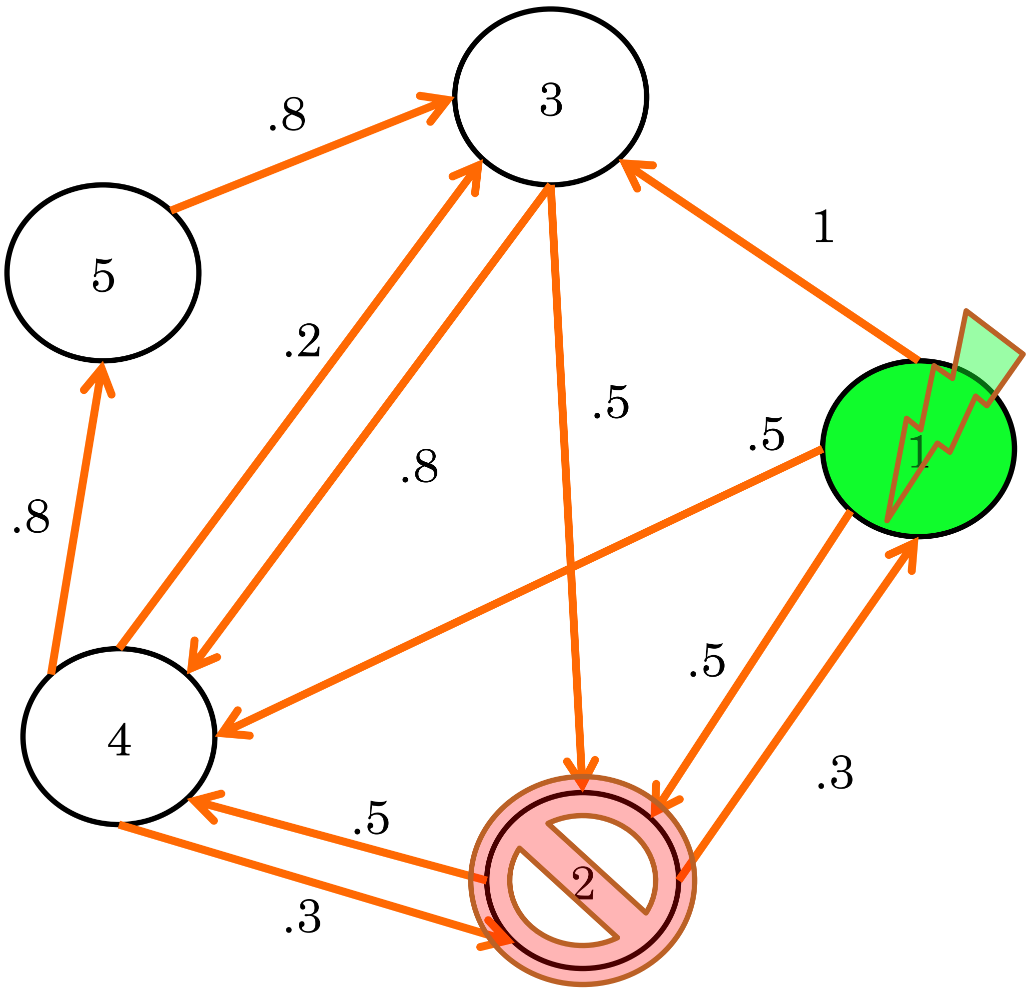
Activations and Suppressions



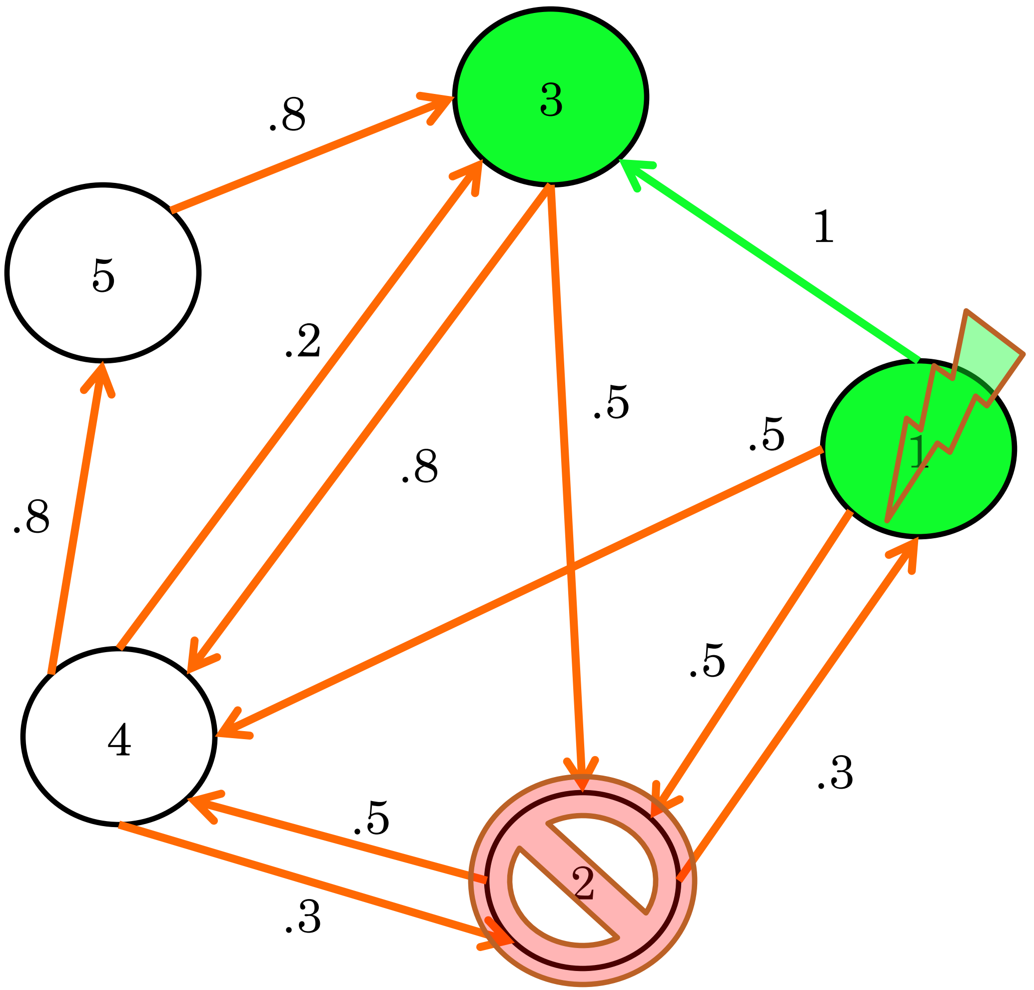
Activations and Suppressions



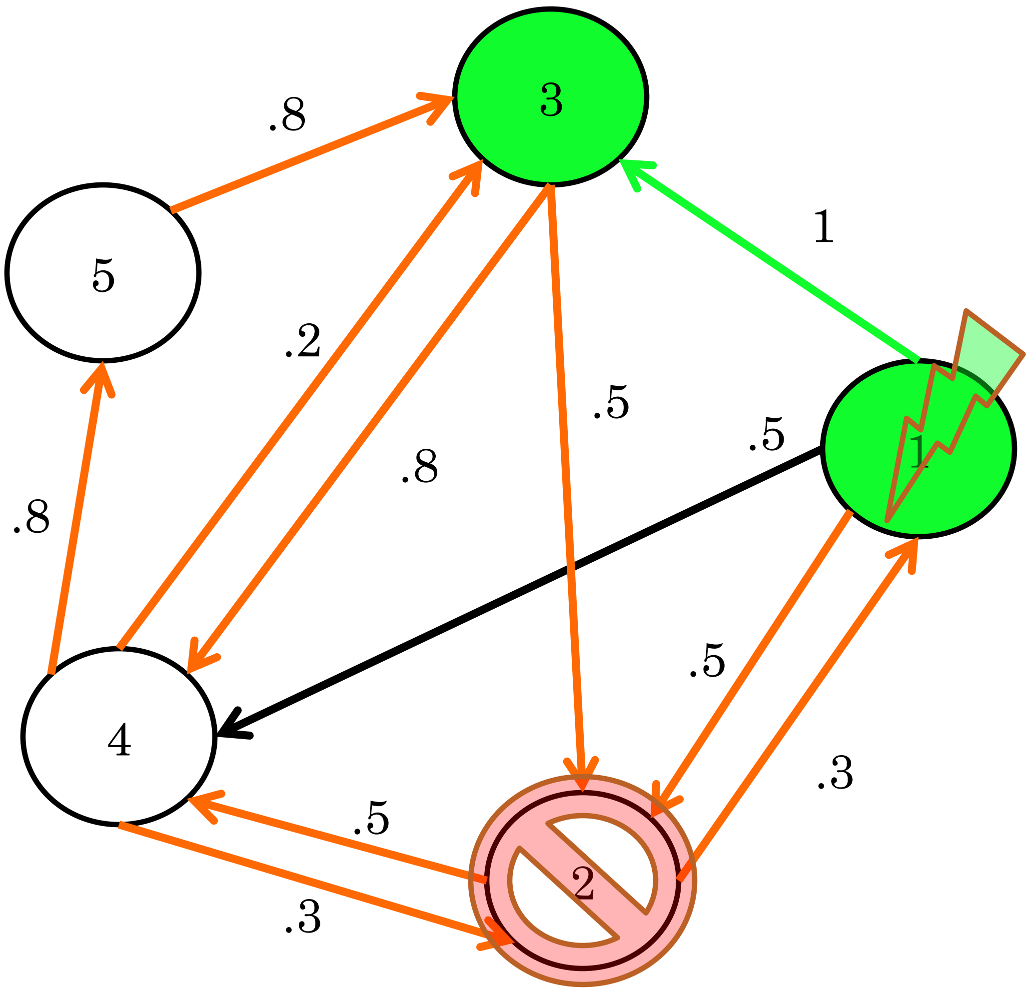
Activations and Suppressions



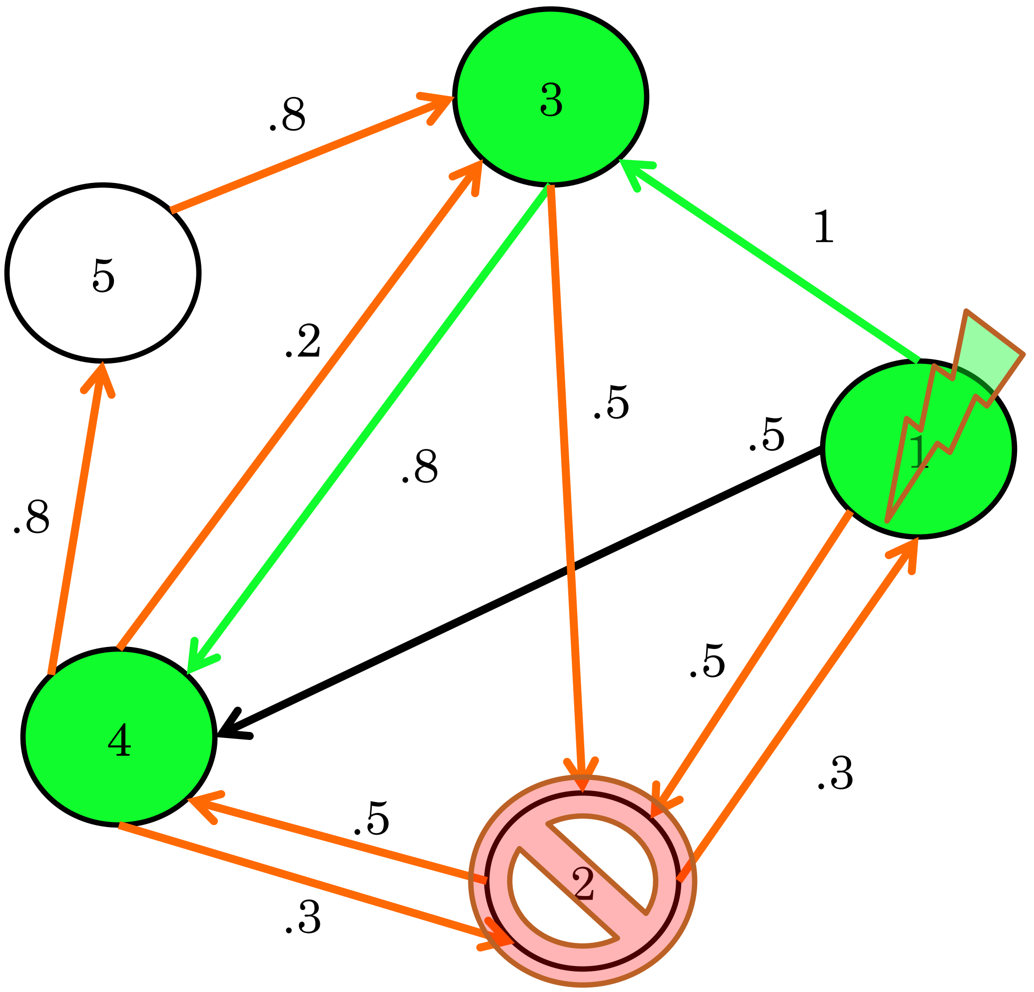
Activations and Suppressions



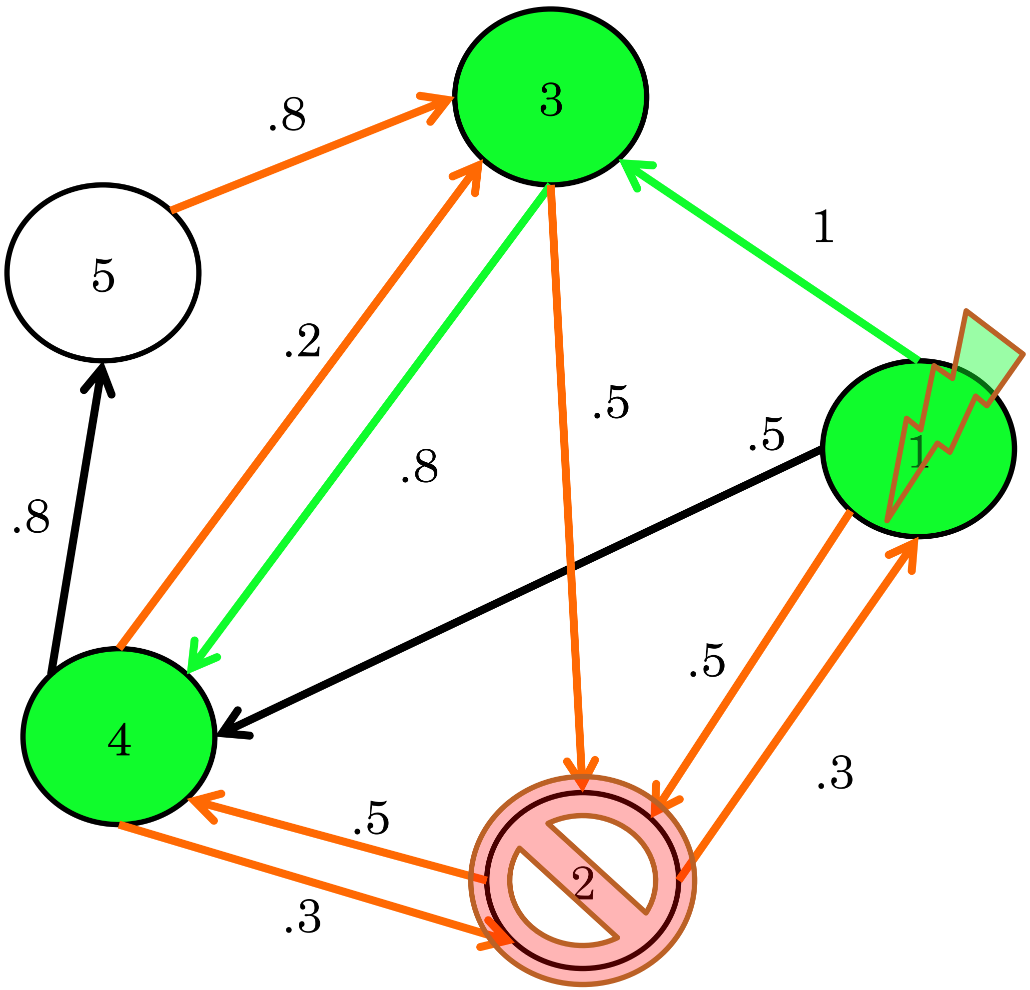
Activations and Suppressions



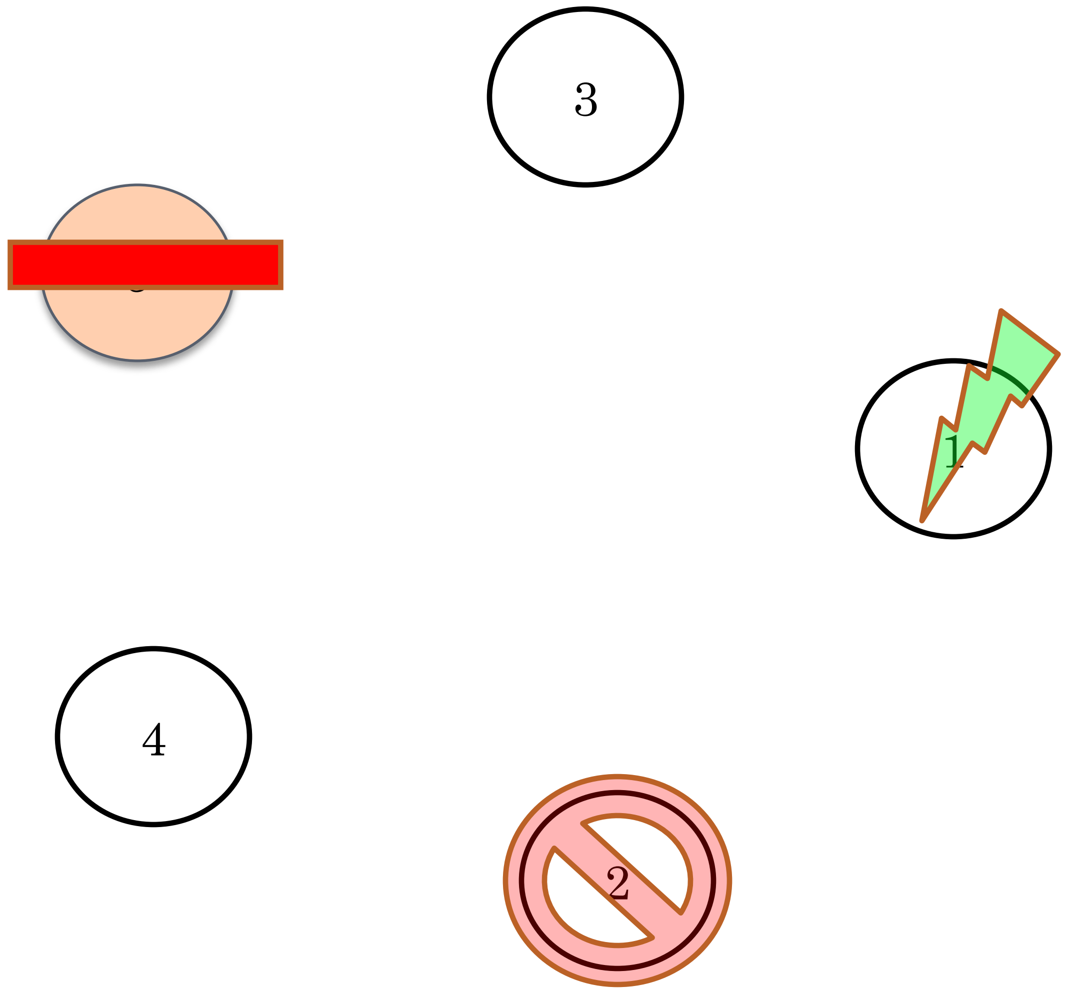
Activations and Suppressions



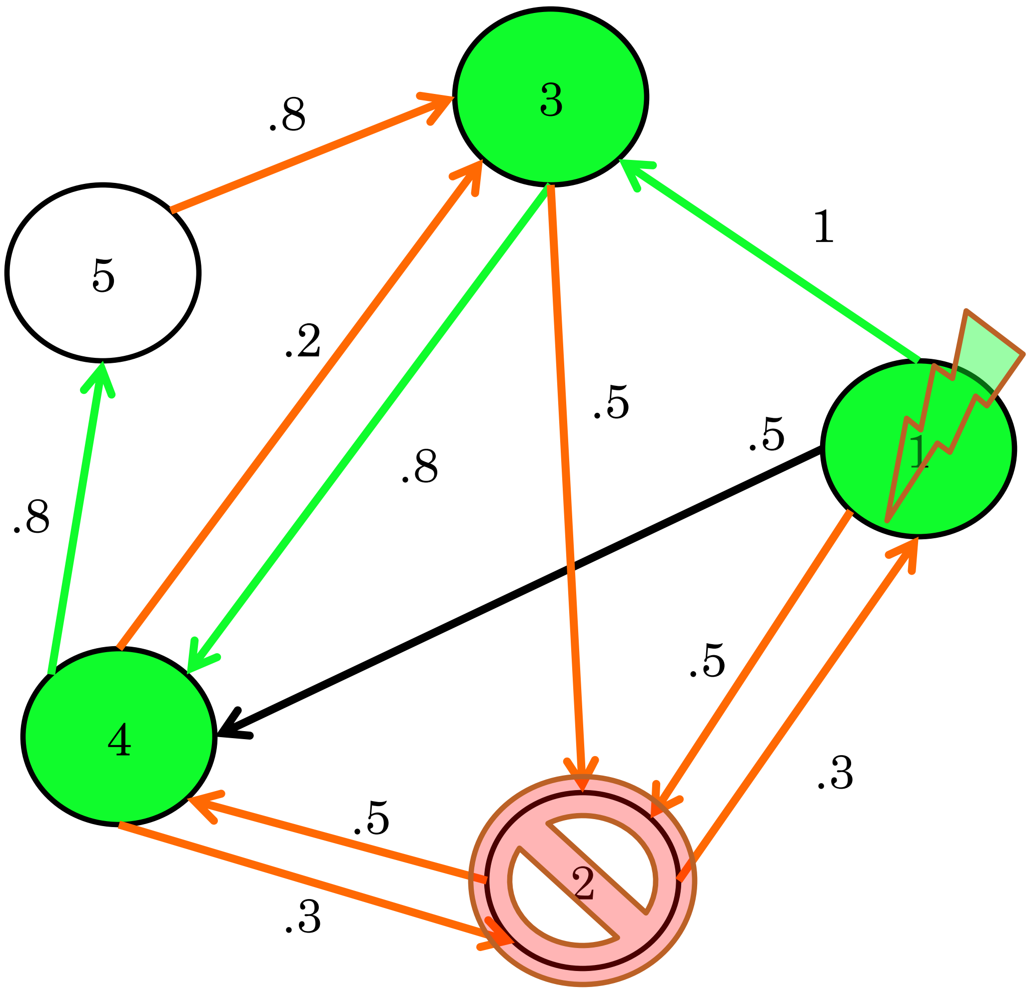
Activations and Suppressions



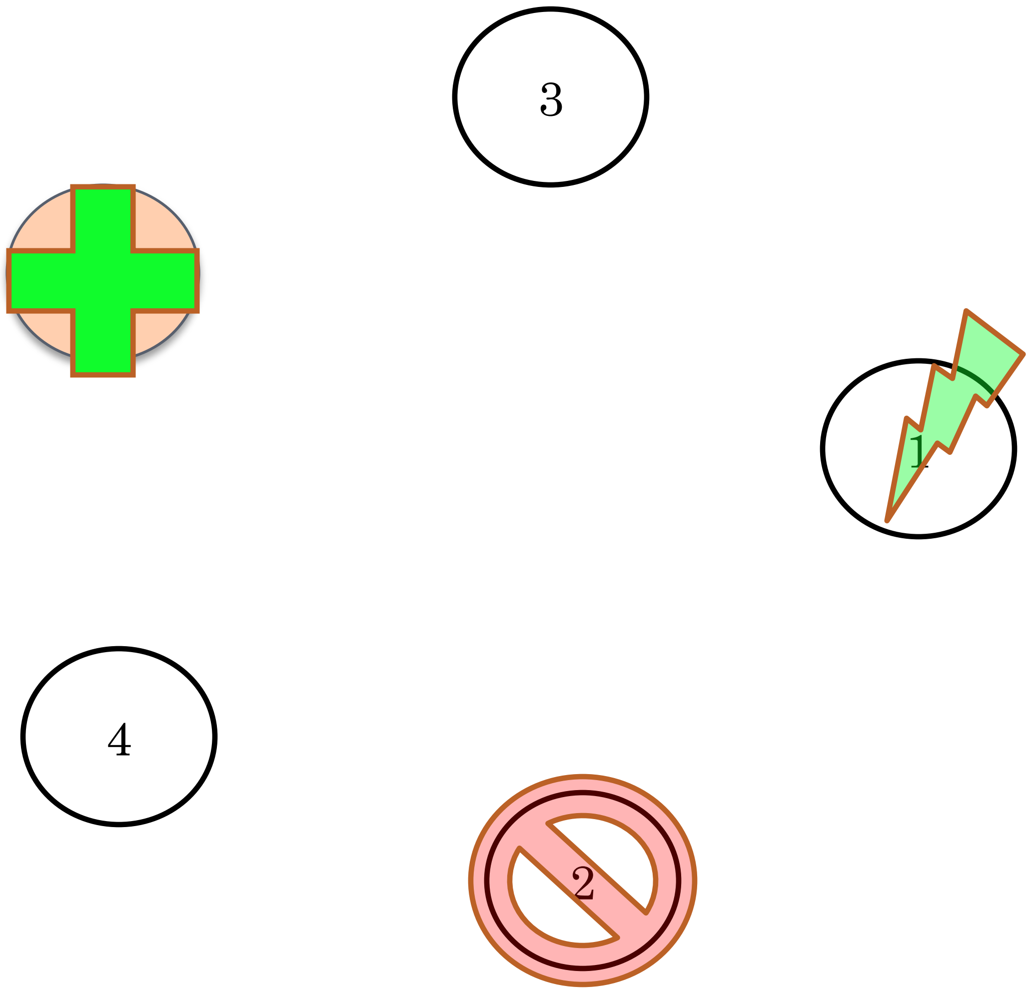
Activations and Suppressions



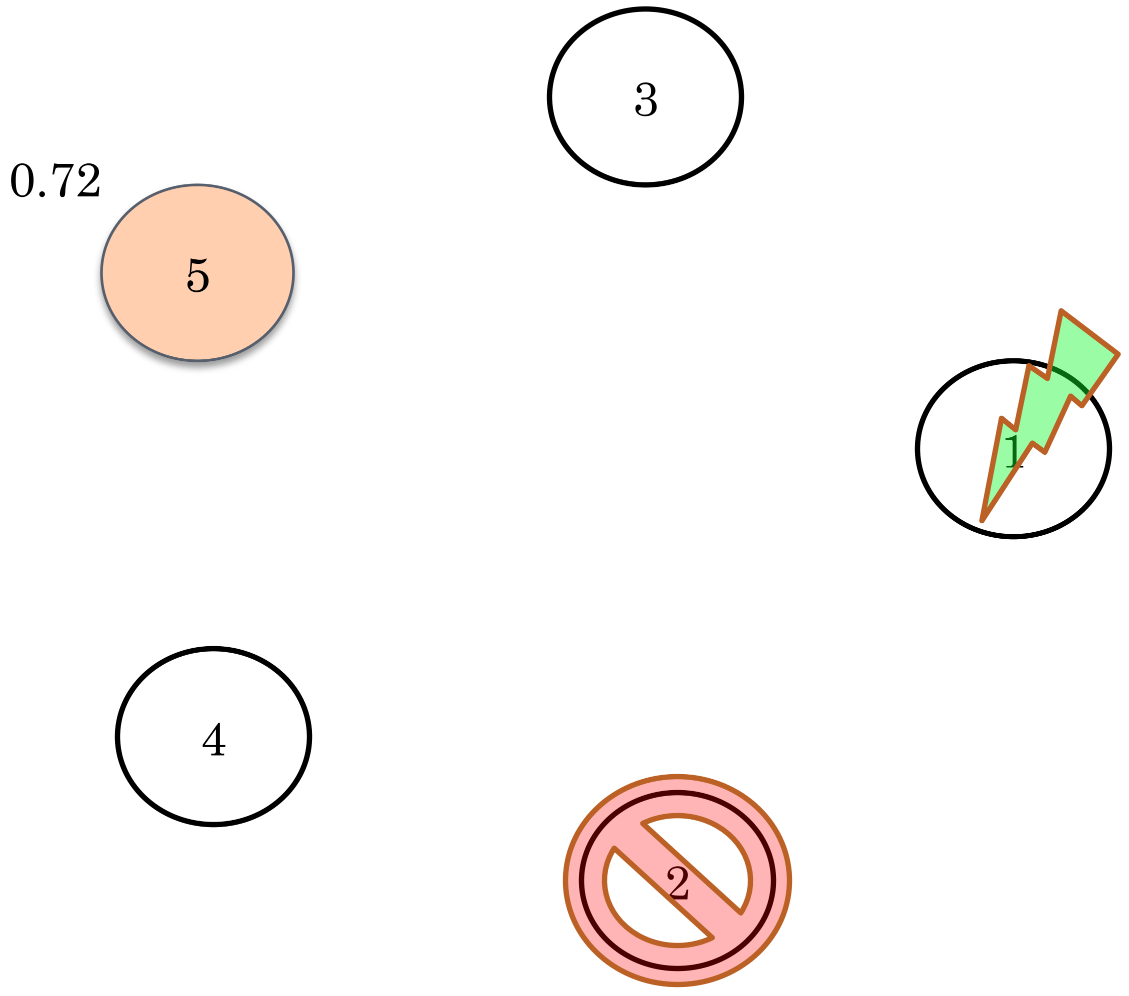
Activations and Suppressions



Activations and Suppressions



Exact Value Injection Queries



The Learning Task

- Two social networks S and S' are behaviorally equivalent if for any experiment e, S(e) = S'(e)
- Give access to a hidden social network S*, the learning problem is to find a social network S behaviorally equivalent to S* using value injection queries.

The Percolation Model

Given a network S and a VIQ

- All edges entering or leaving a suppressed node are automatically "closed."
- Each remaining edge (u,v) is "open" with probability $p_{(u,v)}$ and "closed" with probability (1- $p_{(u,v)}$)
- The result of a VIQ is the probability there is a path from a fired node to the output via open edges in S

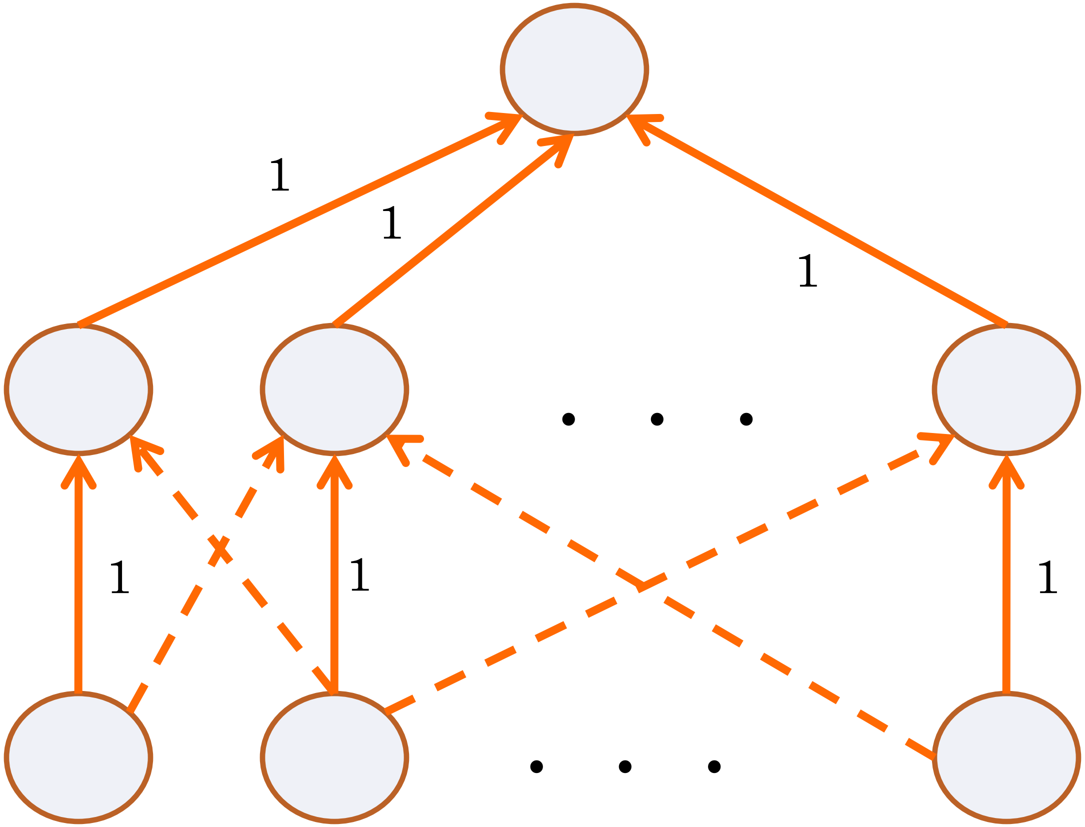
Discoverable Edges

- Let S be a social network and S' be another social network that differs from S only in edge (u,v).
- We say edge (u,v) is discoverable if there is an experiment e such that $S(e) \neq S'(e)$.
- We can view the learning problem as having to find all discoverable edges.

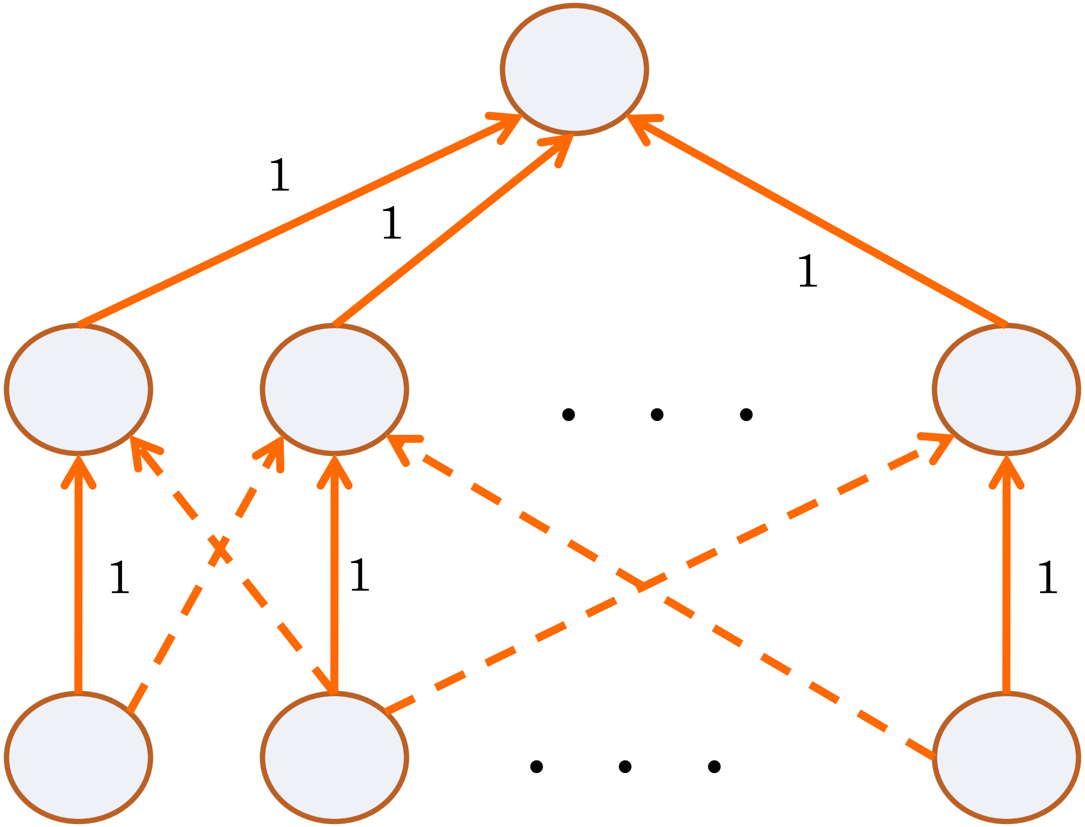
Talk Outline

- Independent Cascade Social Networks (SNs)
- Introduce Value Injection Queries (VIQs) on SNs
  - These were introduced by AACW (STOC '06), extended in AACR (COLT '07) and AACER (COLT '08)
- Show lower bound for learning SNs with VIQs
- Give an idea of the algorithm for learning SNs
- Other results
  - Approximation algorithm
  - Special case of trees
  - Exponential attenuation of paths

A Lower Bound

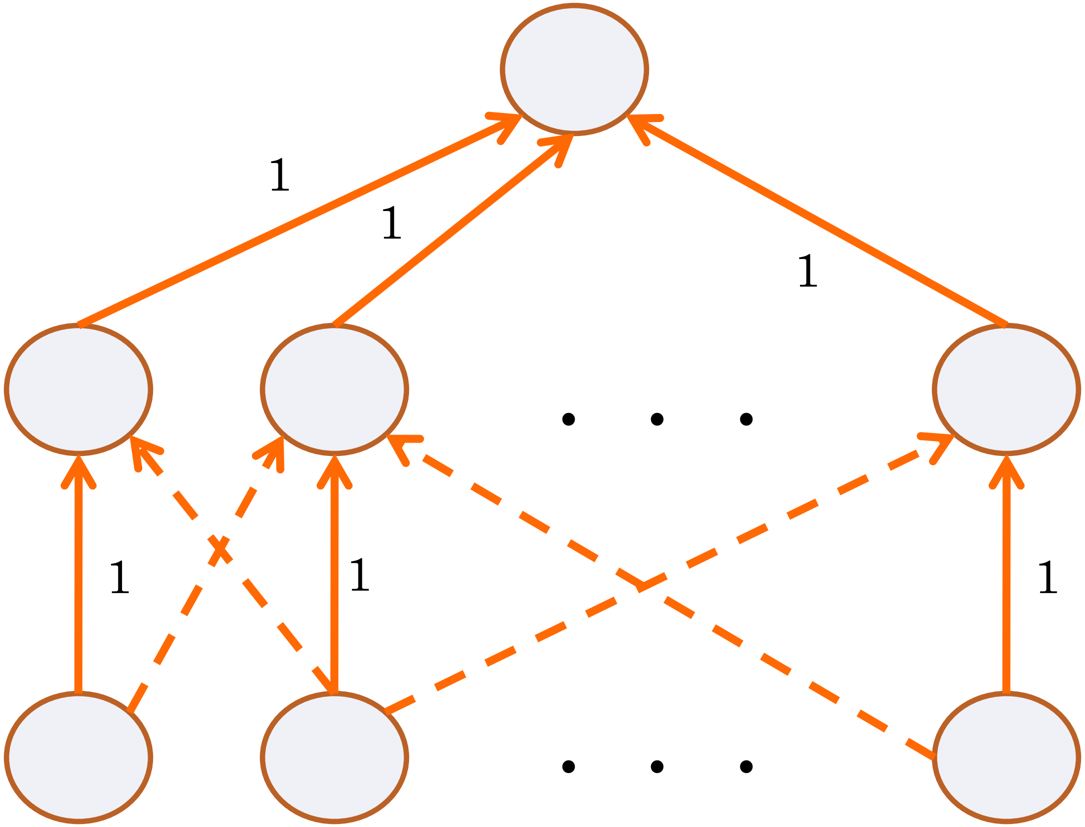


A Lower Bound



All queries give 1-bit answers

A Lower Bound



$2^{\Omega(n^2)}$ such graphs, $\Omega(n^2)$ l.b.

Talk Outline

- Independent Cascade Social Networks (SNs)
- Introduce Value Injection Queries (VIQs) on SNs
  - These were introduced by AACW (STOC '06), extended in AACR (COLT '07) and AACER (COLT '08)
- Show lower bound for learning SNs with VIQs
- Give an idea of the algorithm for learning SNs
- Other results
  - Approximation algorithm
  - Special case of trees
  - Exponential attenuation of paths

First Some Definitions

- The depth of a node is its distance to the root
- An Up edge is an edge from a node of larger depth to a node of smaller depth
- A Level edge is an edge between two nodes of same depth
- A Down edge is an edge from a node at smaller depth to a node at higher depth
- A leveled graph of a social network is the graph of Up edges

Excitation Paths

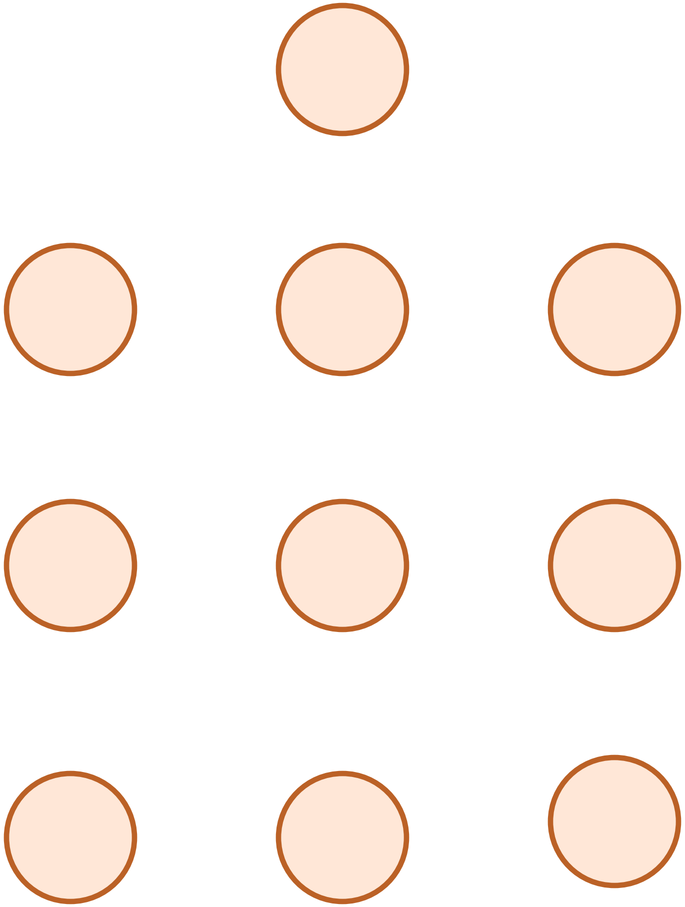
- An excitation path for a node n is a VIQ in which a subset of the free agents form a simple directed path from n to the output. All agents not on the path with inputs into the path are suppressed.

- We also have a shortest excitation path

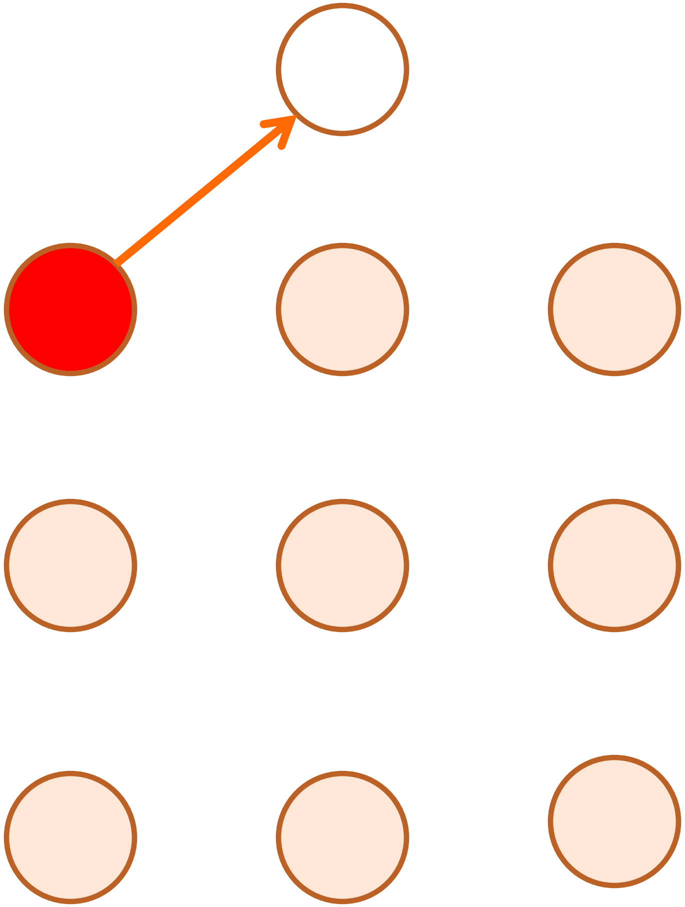
The Learning Algorithm For Networks Without 1 Edges

- First Find-Up-Edges to learn the leveled graph of S
- For each level, Find-Level-Edges
- For each level, bottom-down, Find-Down-Edges

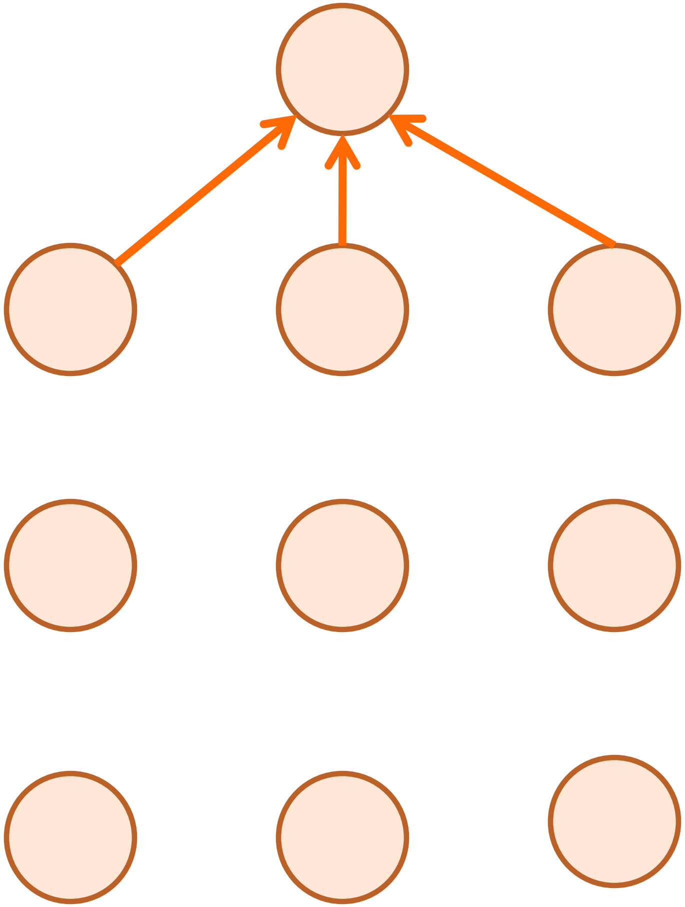
FIND-UP-EDGES



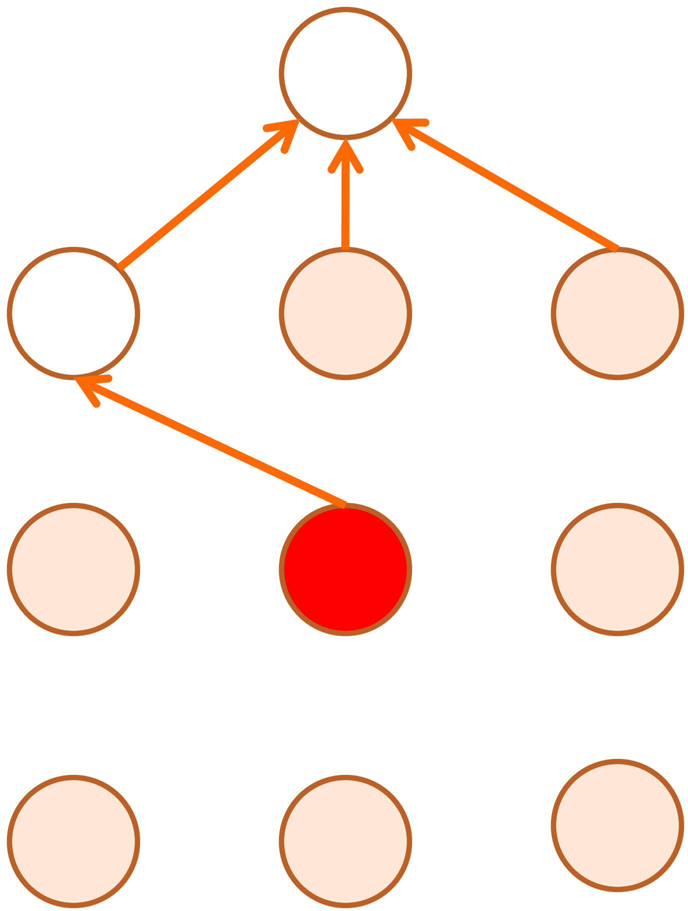
FIND-UP-EDGES



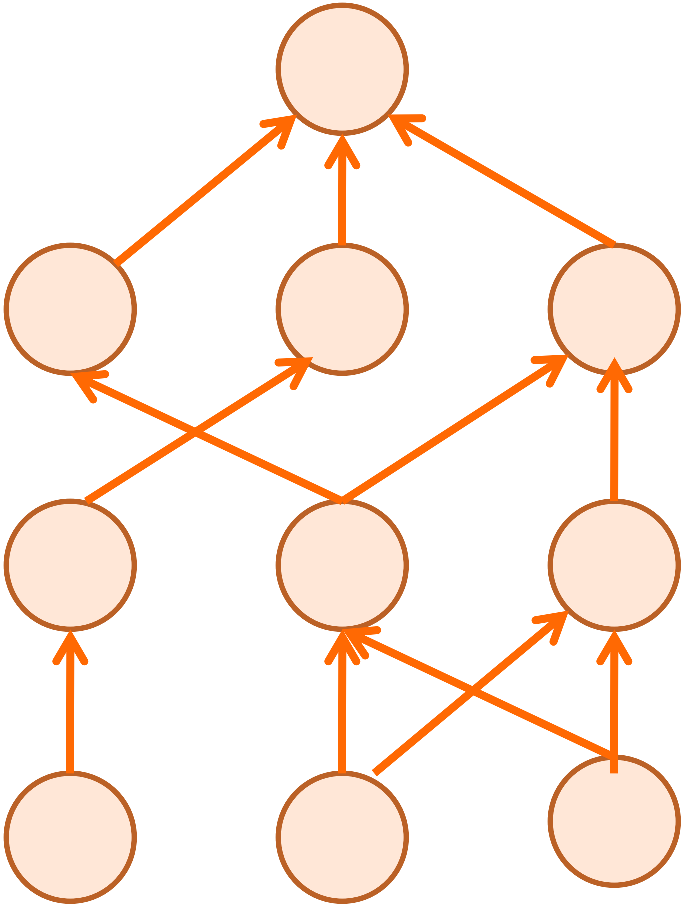
Find-Up-Edges



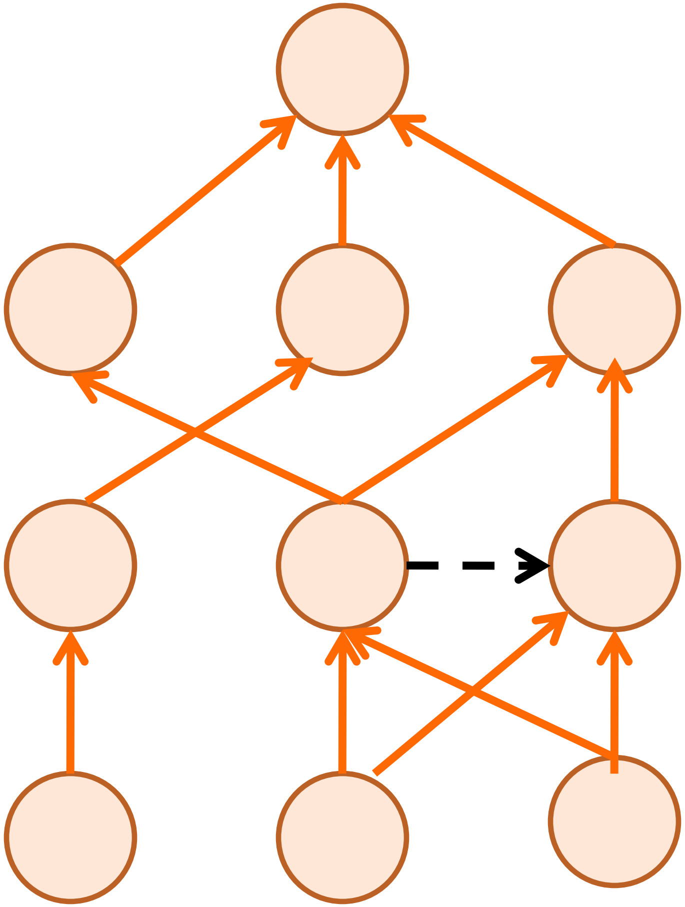
Find-Up-Edges



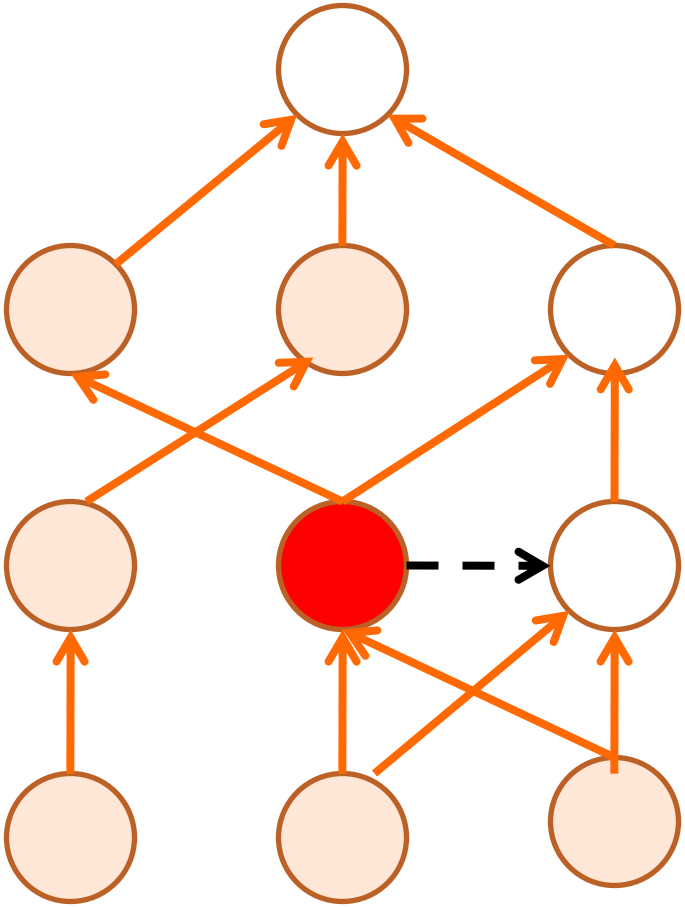
FIND-UP-EDGES



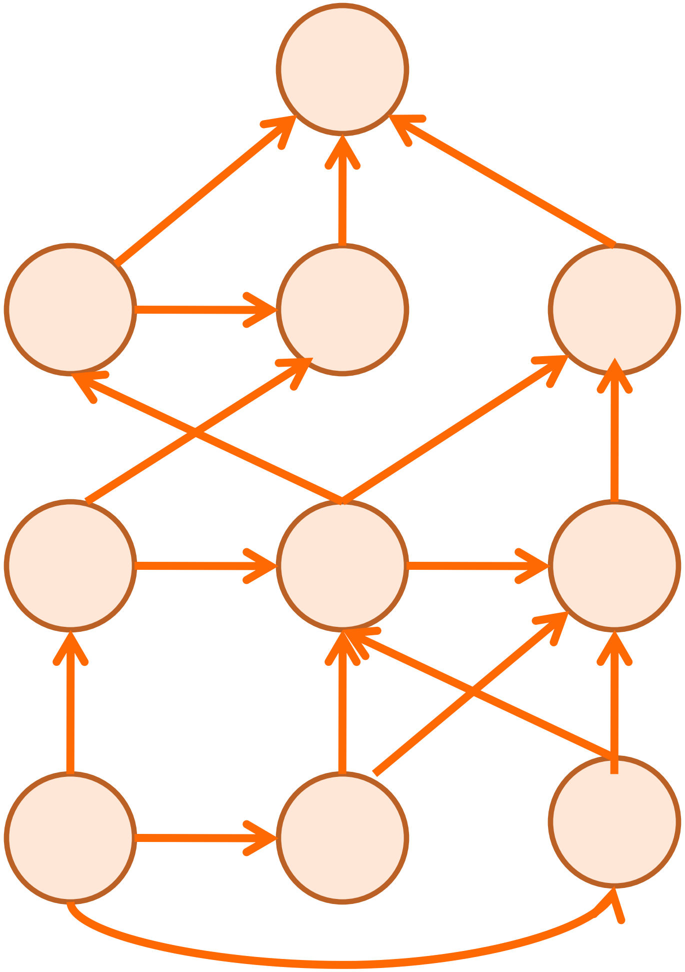
Find-Level-Edges



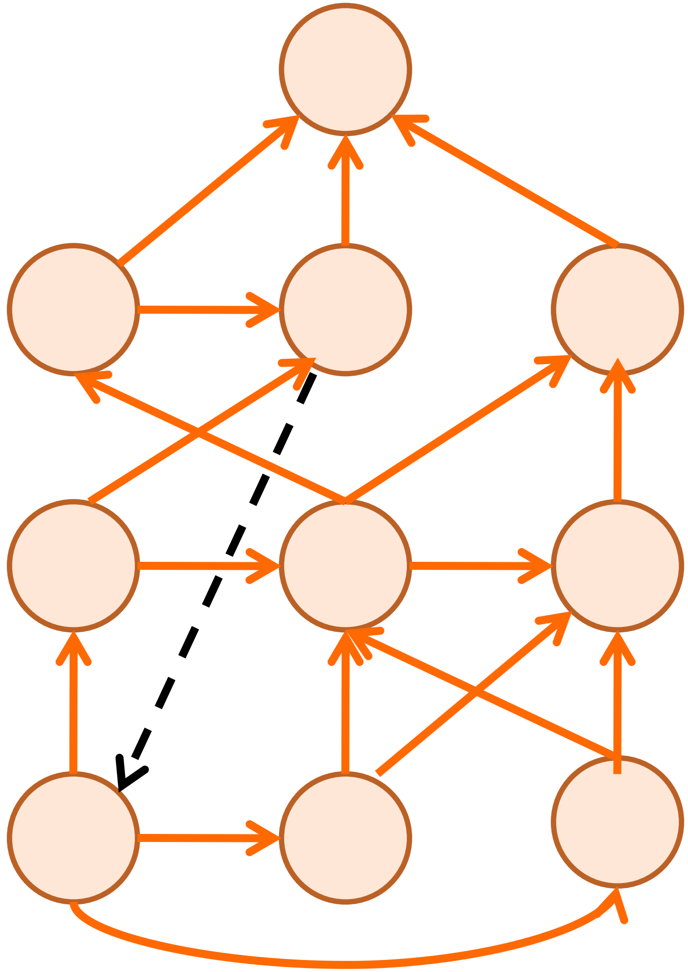
Find-Level-Edges



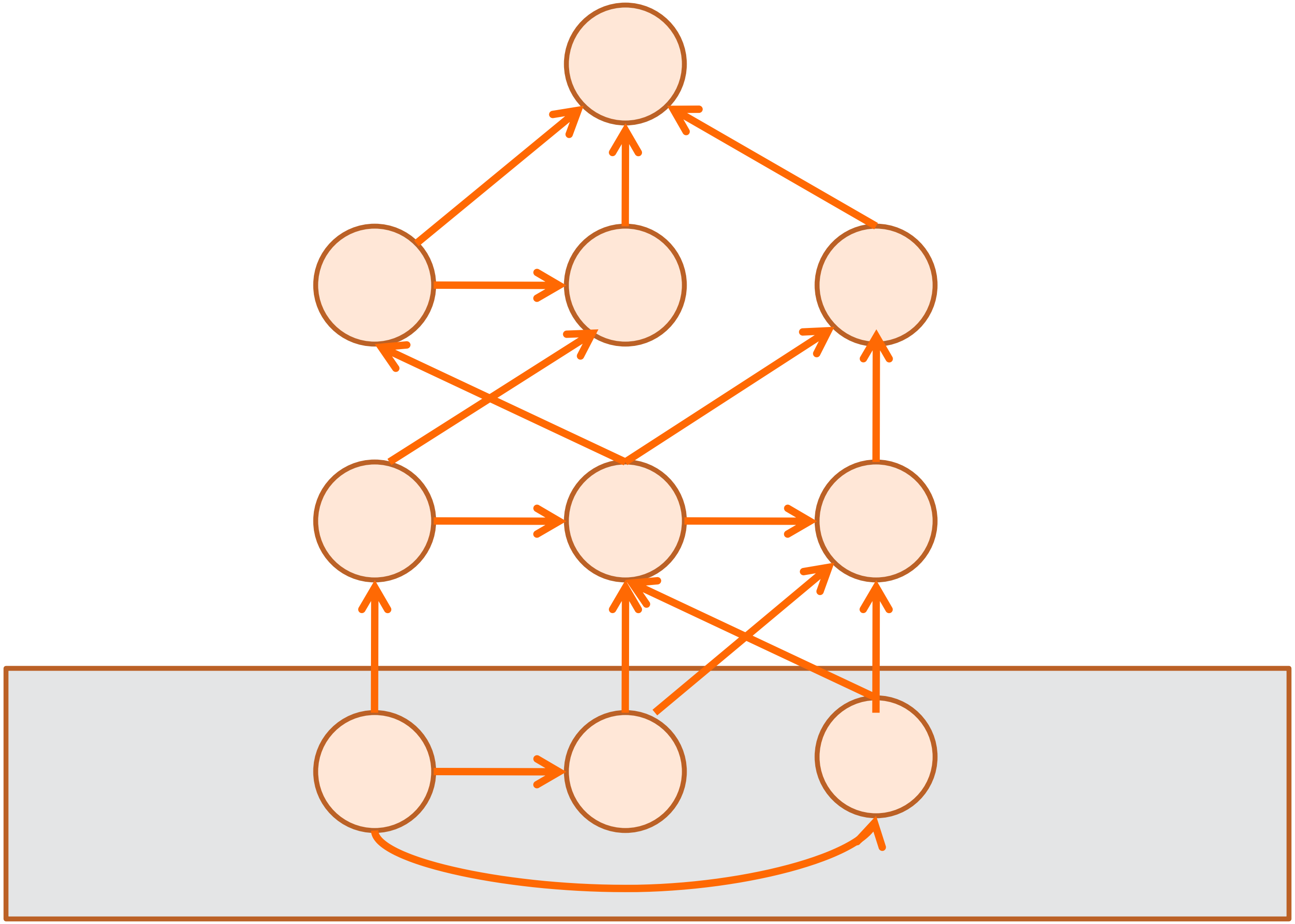
Find-Level-Edges



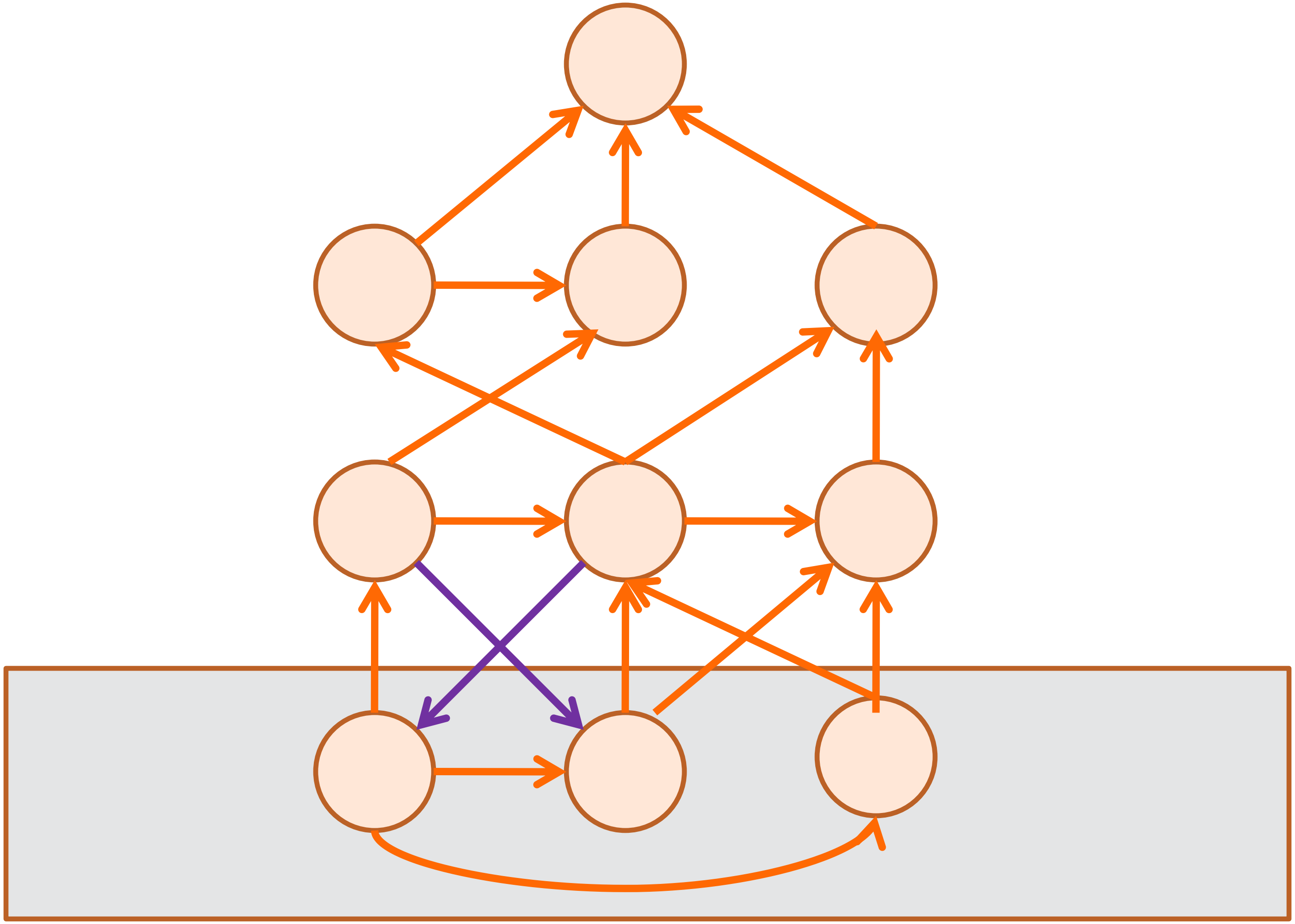
Find-Down-Edges



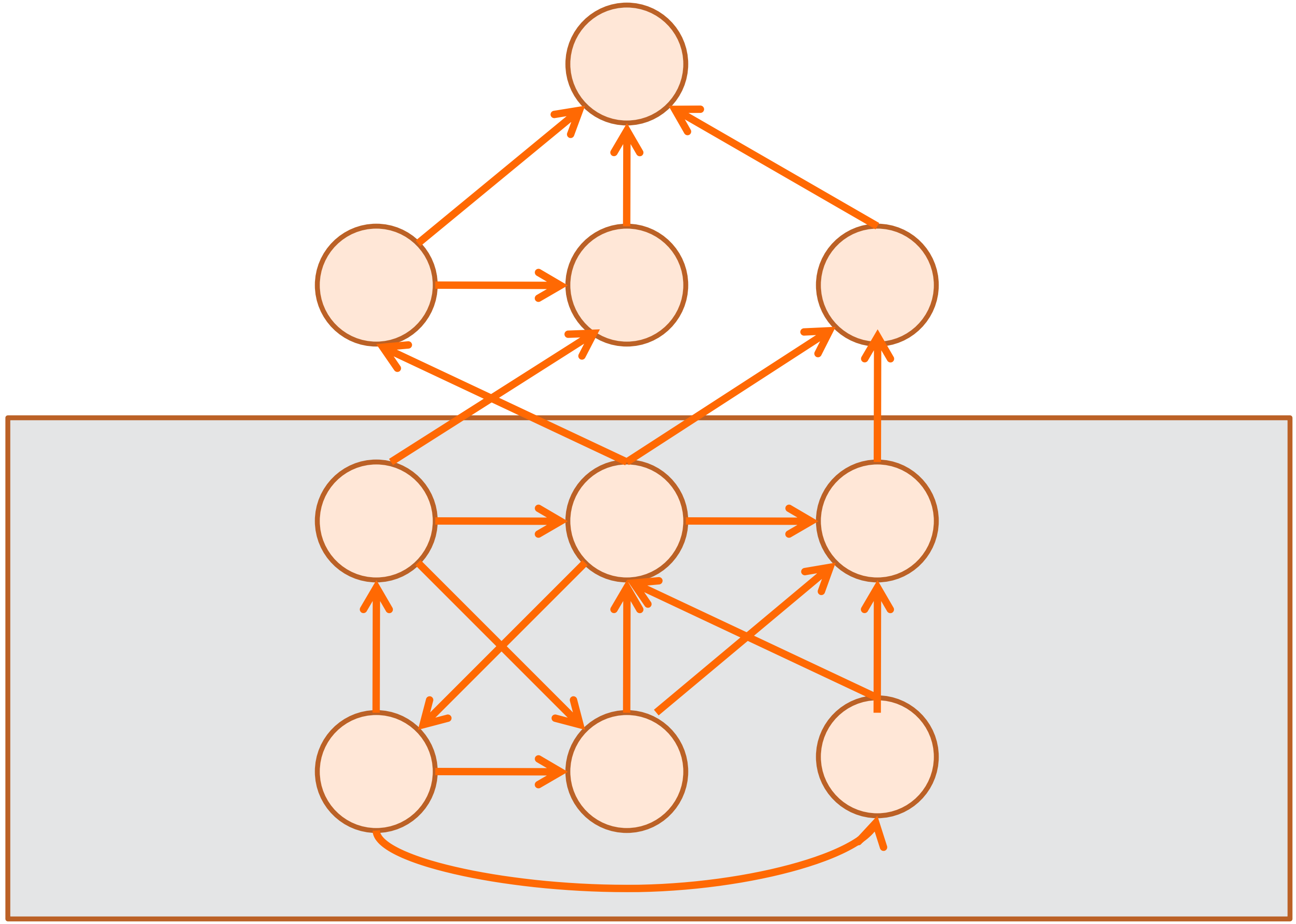
Find-Down-Edges



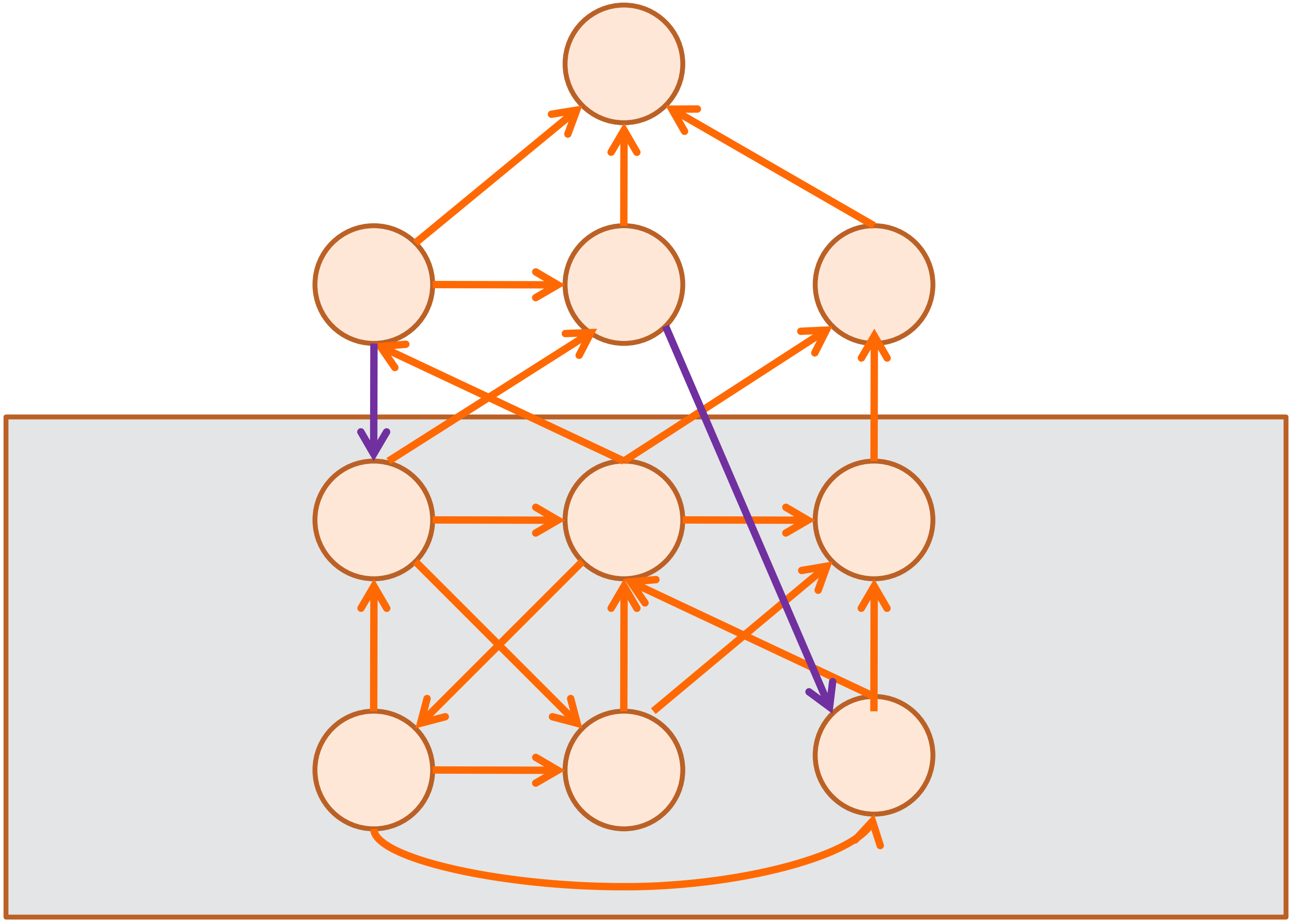
Find-Down-Edges



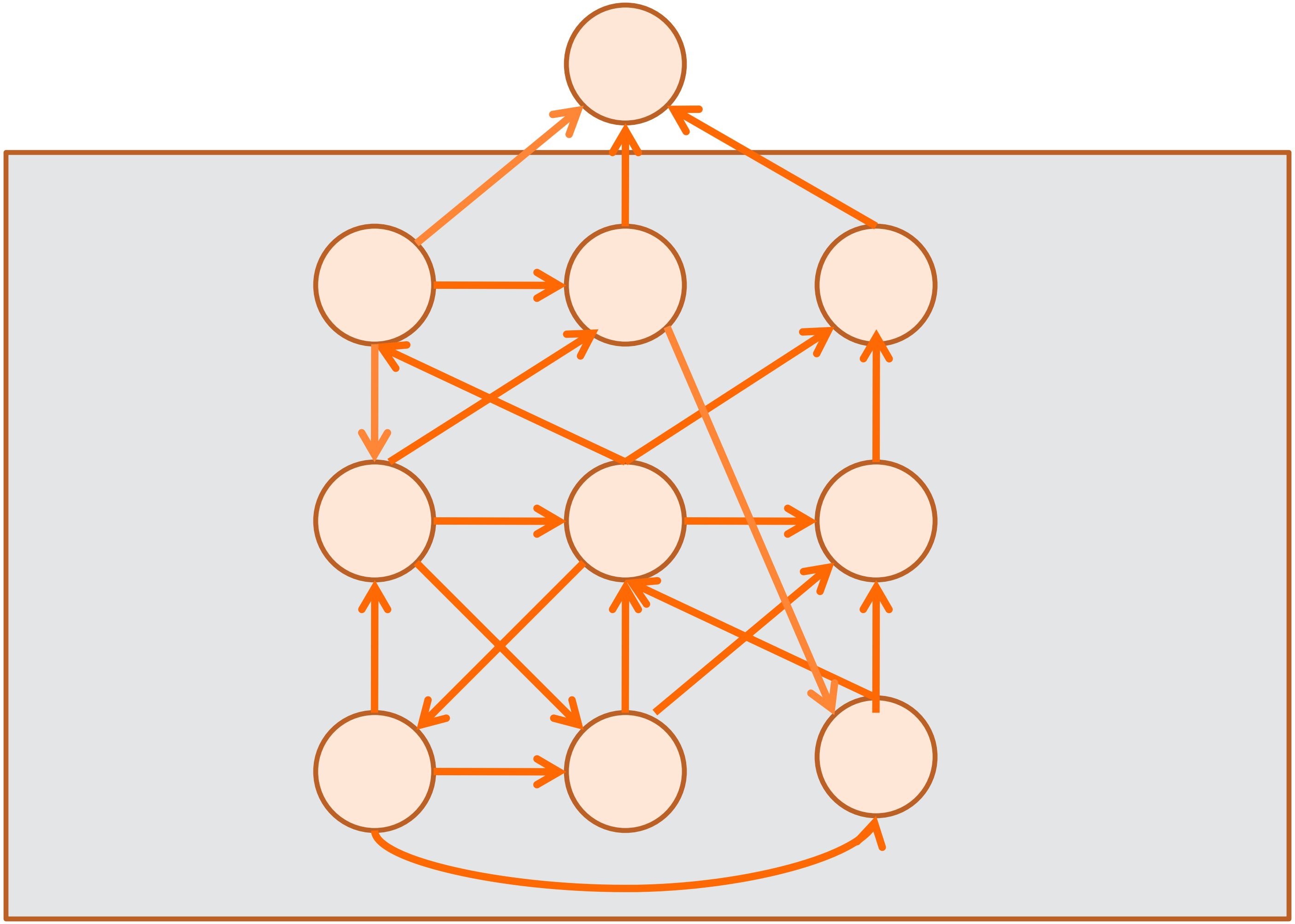
Find-Down-Edges



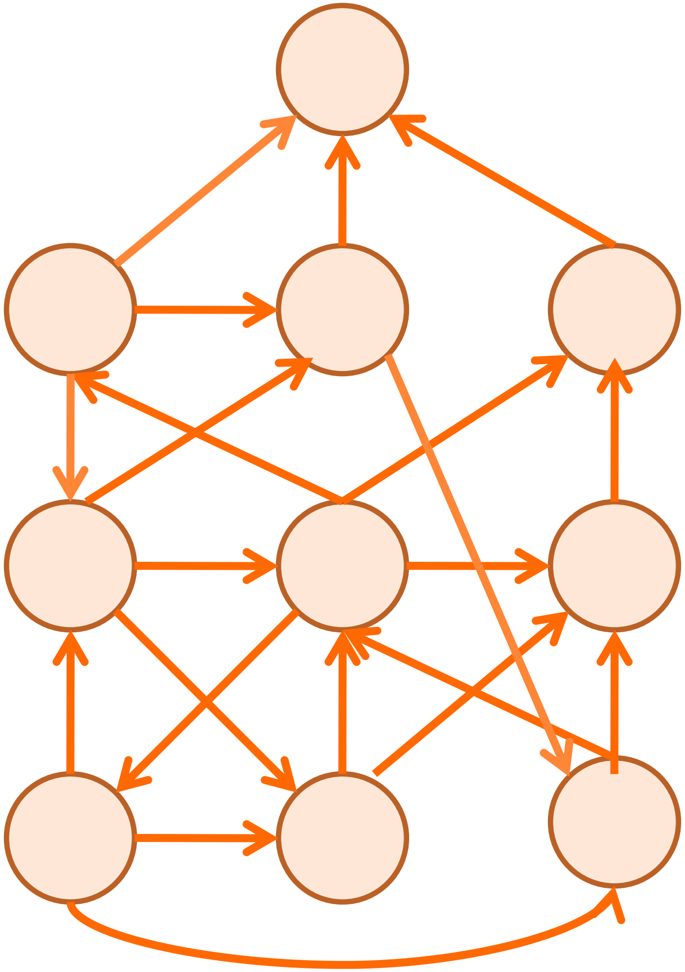
Find-Down-Edges



Find-Down-Edges



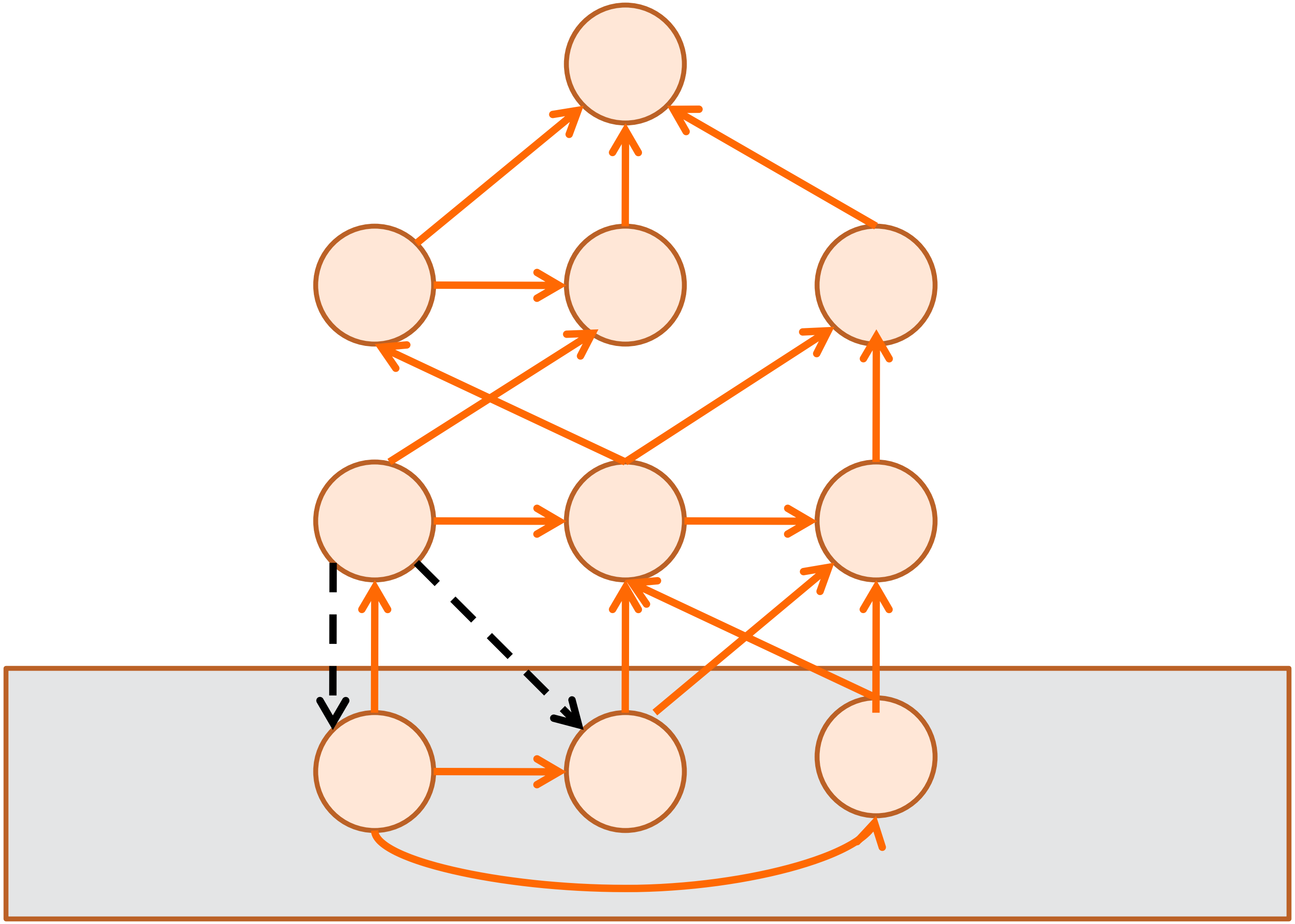
Find-Down-Edges



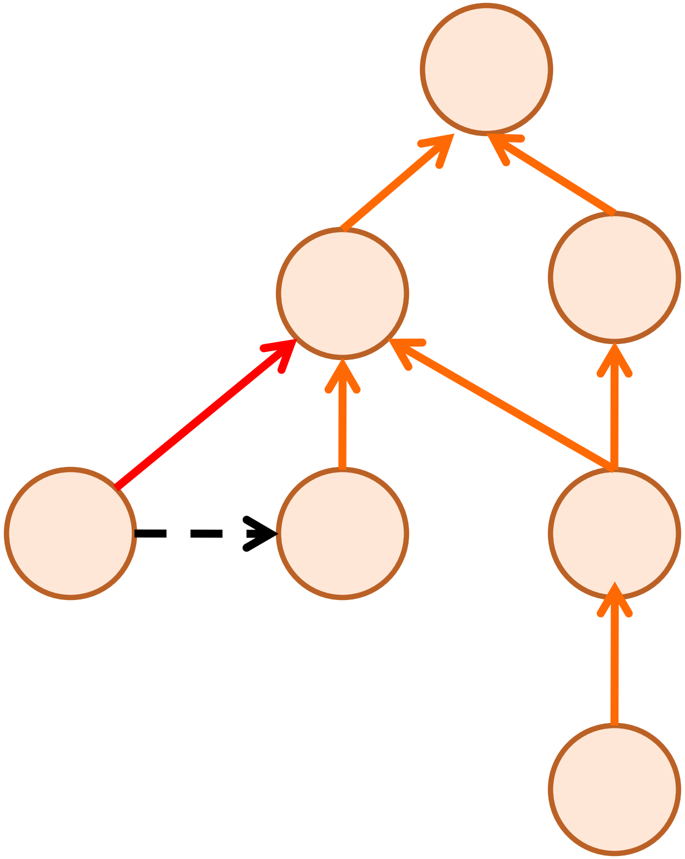
Find-Down-Edges

- For each node u at current level
  - Sort each node v_i in C (complete set) by distance to the root in $G – \{u\}$
  - Let $v_1 \dots v_k$ be the sorted v_is
  - Let $pi_1 \dots pi_k$ be their corresponding shortest paths to the root in $G – \{u\}$
  - For i from 1 to k
    - Do experiment of firing u, leaving pi_i free, and suppressing the rest of the nodes.

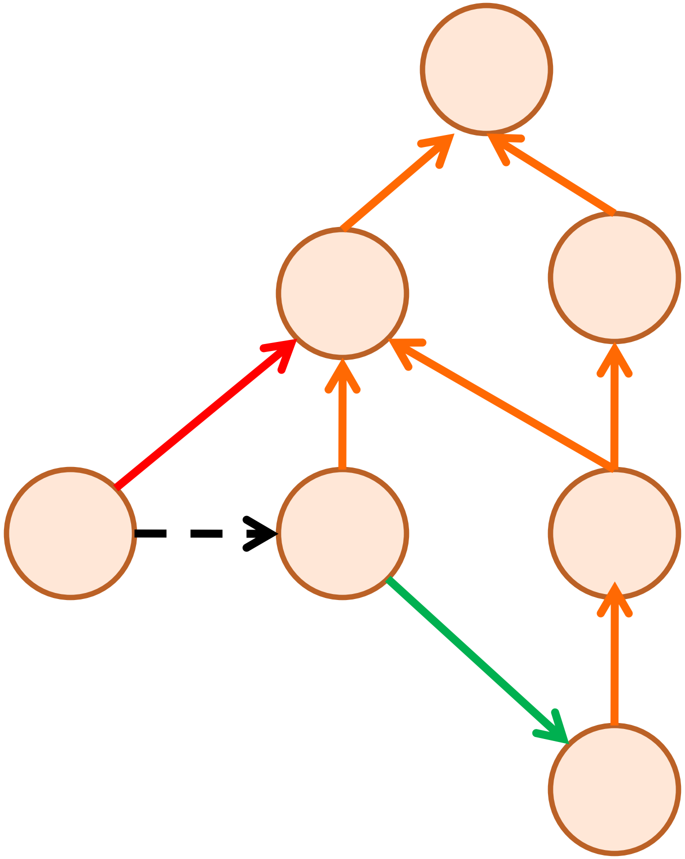
For Example



With Ones – a Problem



With Ones – a Problem



With Ones

- Algorithm gets more complicated
- Level edges and down edges are found in one subroutine
- In looking for down edges from u, need to avoid not just u, but also all nodes reachable from u by 1 edges
- There always exists some pair of nodes, with source in L (current level) and destination in C + L where you can look for an edge.

In the End

- We do 1 query per each possible edge, giving an $O(n^2)$ algorithm
- Matches the $\Omega(n^2)$ lower bound

Talk Outline

- Independent Cascade Social Networks (SNs)
- Introduce Value Injection Queries (VIQs) on SNs
  - These were introduced by AACW (STOC '06), extended in AACR (COLT '07) and AACER (COLT '08)
- Show lower bound for learning SNs with VIQs
- Give an idea of the algorithm for learning SNs
- Other results
  - Approximation algorithm
  - Special case of trees
  - Exponential attenuation of paths

Finding Influential Nodes

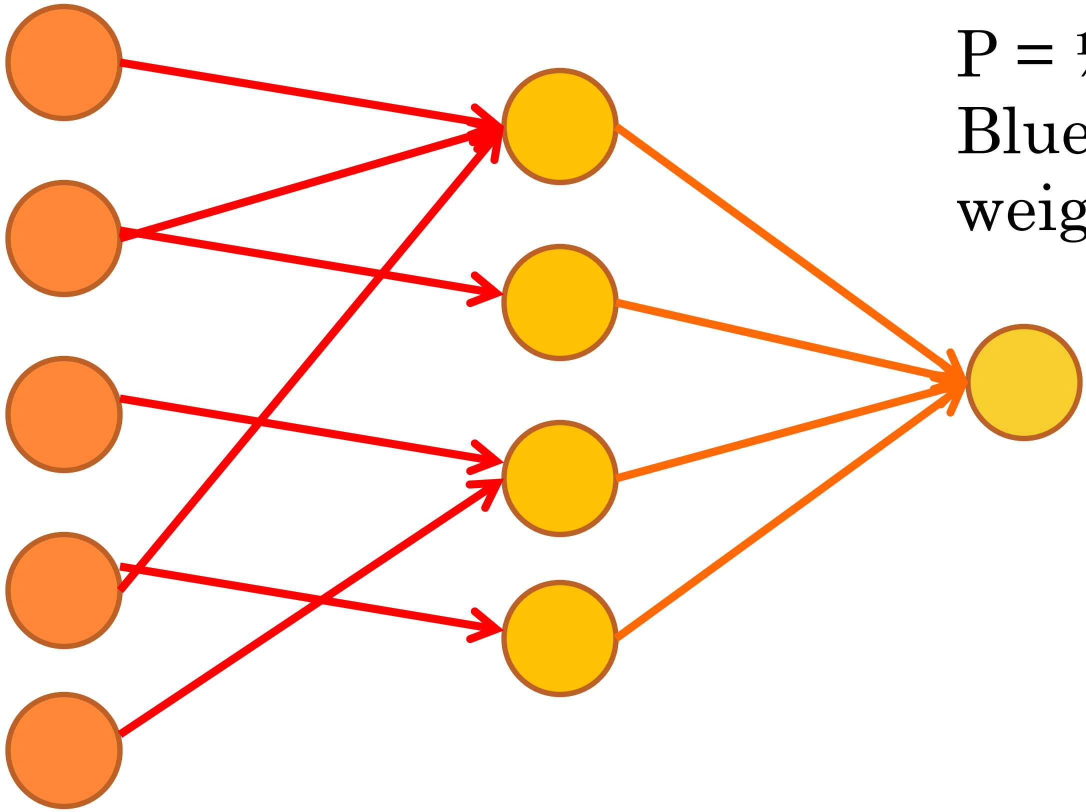
- Suppose instead of learning the social network, we wanted to find an influential set of nodes quickly.

- A set of nodes is influential if, when activated, activates the output with probability at least p

Finding Influential Nodes

- Suppose instead of learning the social network, we wanted to find an influential set of nodes quickly.
- A set of nodes is influential if, when activated, activates the output with probability at least p
- NP Hard to Approximate to log n, even if we know the structure of the network

Reduction from Set Cover



P = ½
Blue edges have weight (1- ½ ^(1/k))

sets

k elements

An Approximation Algorithm

- Say the optimal solution has m nodes
- Suppose we wanted to fire the output with probability (p – ε)
- Let I be the set of chosen influential nodes.
- Observation: at any point in the algorithm, greedily adding one more node w to I makes

$$S(e_{I \cup \{w\}}) \geq S(e_I) + \frac{p - S(e_I)}{m}$$

Analyzing Greedy

- Using a greedy algorithm, we let k be the number of rounds the algorithm is run

For

$$p\left(1-\frac{1}{m}\right)^k < \epsilon$$

it suffices that

$$e^{-\frac{k}{m}} < \frac{\epsilon}{p}$$

or

$$k > m\log\left(\frac{p}{\epsilon}\right).$$

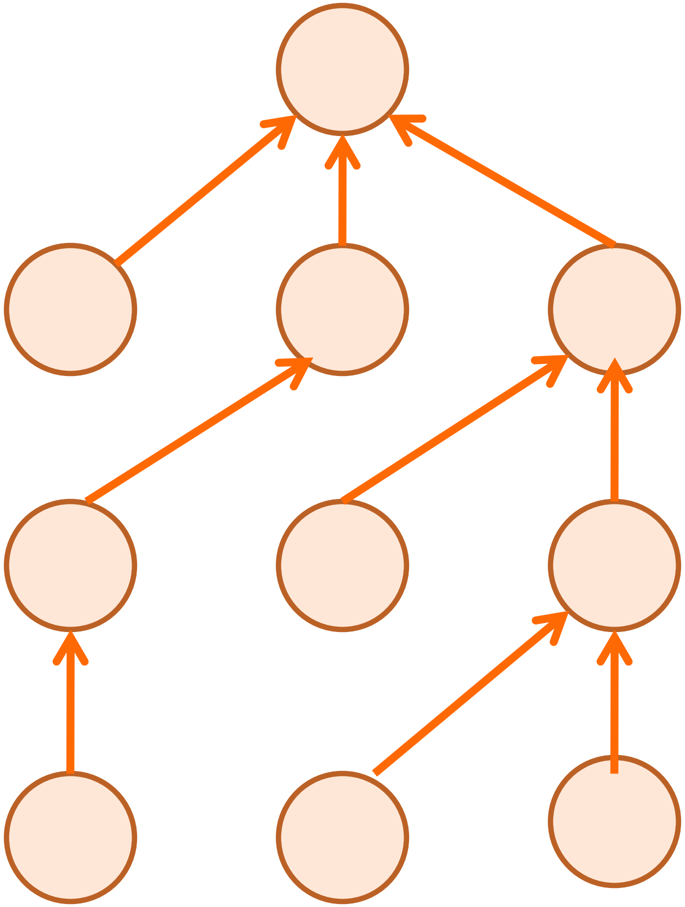
Reconciling Greedy

- Therefore after m $\log(p/\varepsilon)$ rounds, we get to within ε of p.

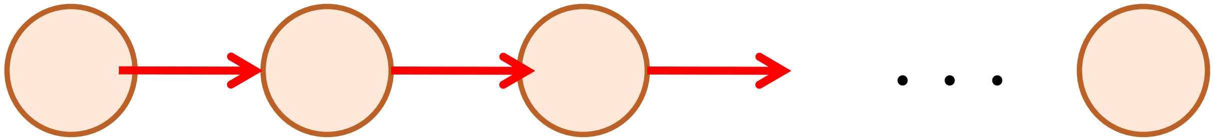
- To reconcile with the set cover reduction, if we set $\varepsilon = ½^{(1/n)} - ½^{(1/(n-1))} = \theta(1/n^2)$, this forces us to cover all the elements.

- Giving a m $\log (p\ n^2) = O(m \log (n))$ approximation. Matches the set cover lower bound.

Trees



Lower Bound



Like sorting with comparisons

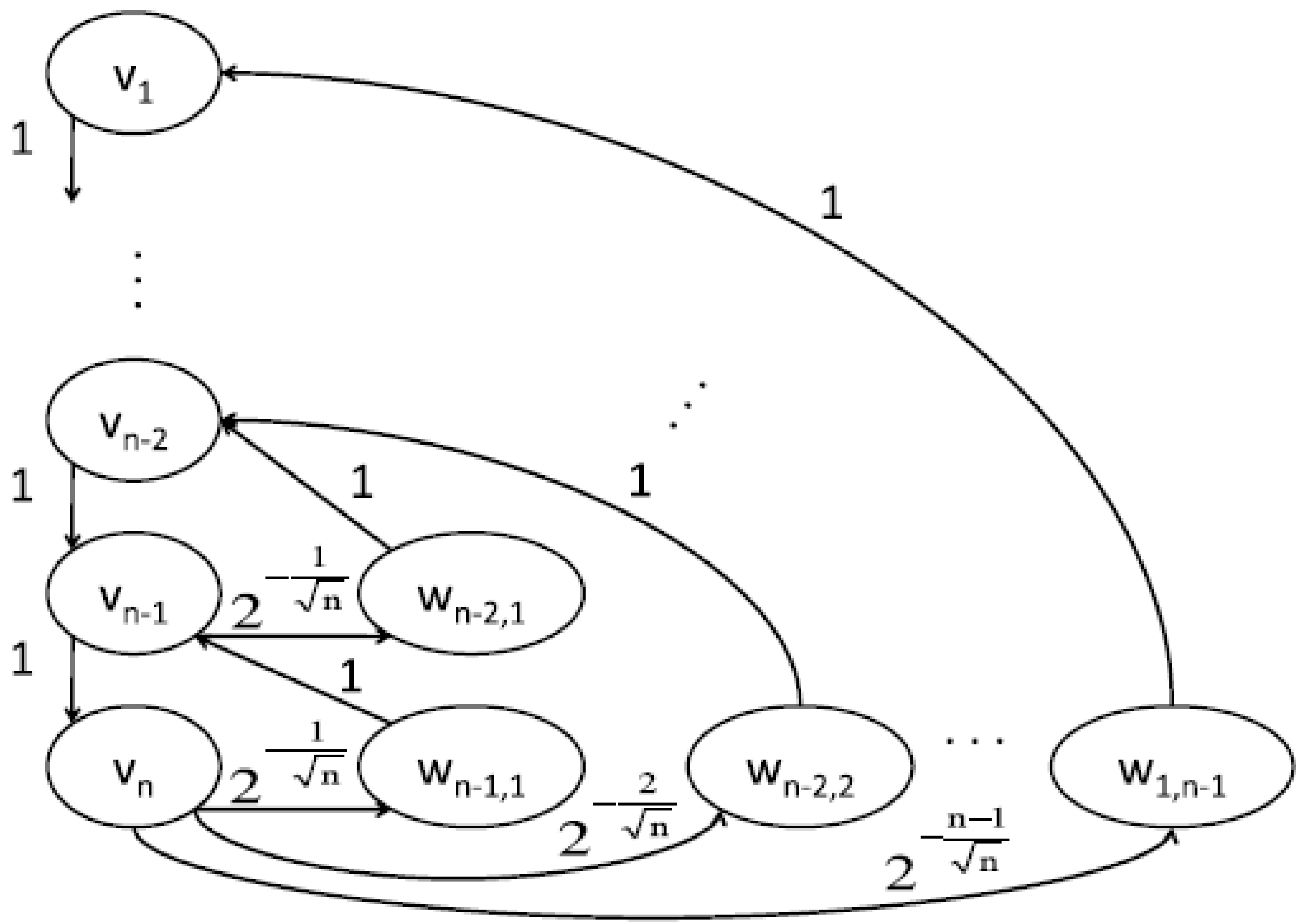
Gives a Ω(n log n) lower bound

The Idea for Algorithm

- Ancestor Test. To test if u has an ancestor in set S, fire u and suppress all nodes in S.

- We can build up a tree by adding one node at a time and binary searching where it belongs in the tree using ancestor tests.

If We Don't Have Exact VIQs



Discussions and Open Problems

- Interesting model to study various hidden structures.
- Finding non-path based methods for social networks. (Actually also an open question for learning circuits with VIQs)
- Reducing Query Size
- Other models.