UNIVERSITÀ
DEGLI STUDI
DI PADOVA

elicsir
Elevating Italian Computer Science
Innovation and Research

Influence Maximization: Dai modelli di base all'approccio submodulare

Alberto Barbato

Mentore: Geppino Pucci

Motivation

Motivation

Maximizing the Spread of Influence through a Social Network

David Kempe* Jon Kleinberg† Éva Tardos‡

Received April 17, 2014; Revised September 7, 2014; Published April 22, 2015

More than 10000 citations!

Summary

- **Core Concepts**
  - Motivation and Influence Spread.

Summary

- **Core Concepts**
  - Motivation and Influence Spread.
- **Models Overview**:
  - General Threshold Model (**GTM**)
  - Linear Threshold Model (**LTM**)
  - Independent Cascade Model (**IC**)

Summary

- **Core Concepts**
  - Motivation and Influence Spread.
- **Models Overview**:
  - General Threshold Model (**GTM**)
  - Linear Threshold Model (**LTM**)
  - Independent Cascade Model (**IC**)
- **Submodularity**:
  - The Simple Greedy Algorithm
  - Approximation Guarantees
  - Submodular Threshold Model (**STM**)

Summary

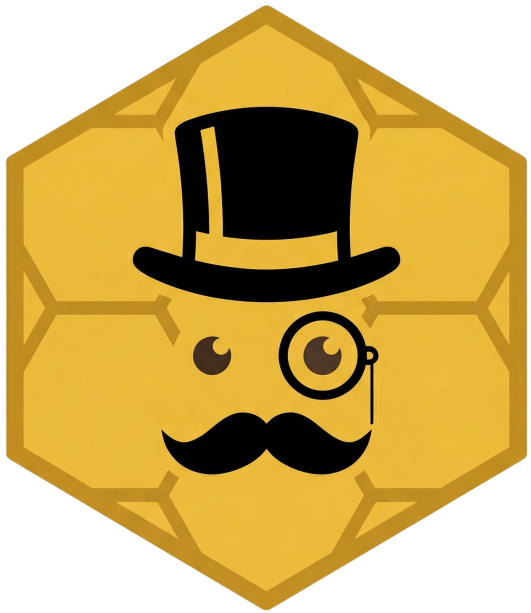
- **Core Concepts**
  - Motivation and Influence Spread.
- **Models Overview**:
  - General Threshold Model (**GTM**)
  - Linear Threshold Model (**LTM**)
  - Independent Cascade Model (**IC**)
- **Submodularity**:
  - The Simple Greedy Algorithm
  - Approximation Guarantees
  - Submodular Threshold Model (**STM**)
- **Conclusions**

Influence?

Let's start with a practical example:

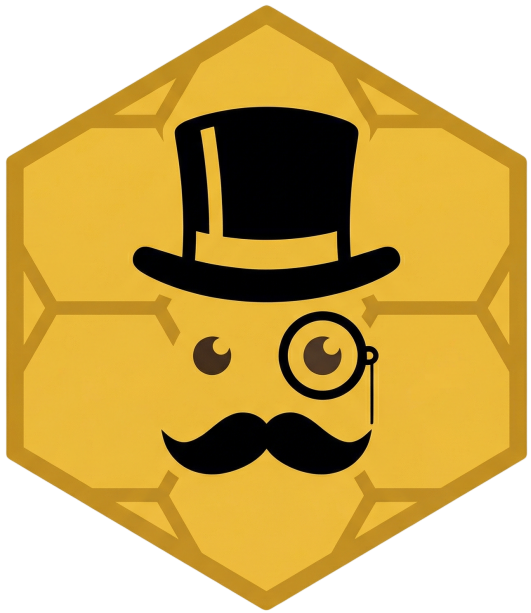
Influence?

Let's start with a practical example:
We want to promote on the social network "**NerdBook**" our new software: a beautiful hive engine!



Influence?

Let's start with a practical example:
We want to promote on the social network "**NerdBook**" our new software: a beautiful hive engine!



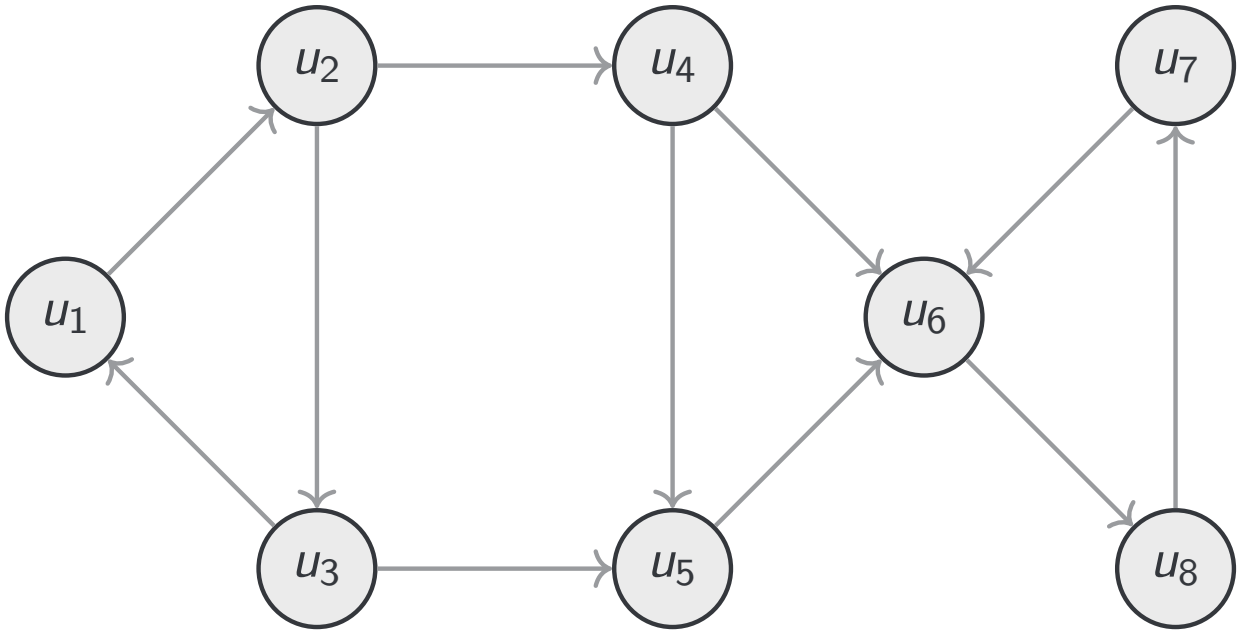
MrHive: `play.mrhive.dev`

Influence?

We will model the network of relationships of NerdBook between users as a graph.

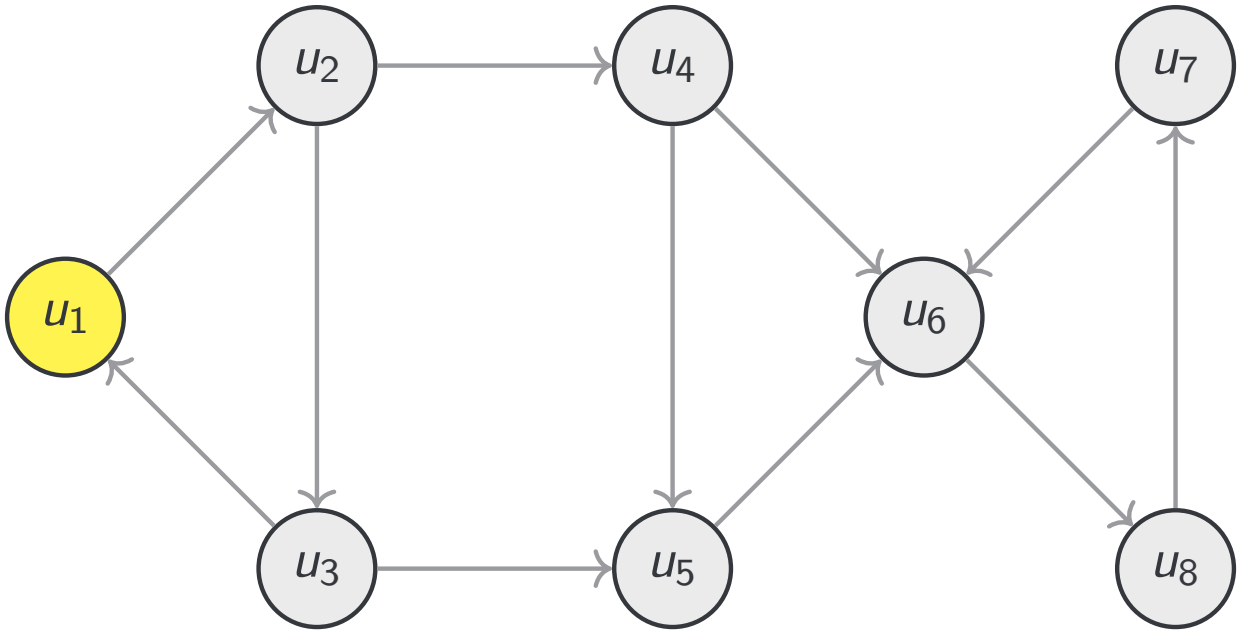
Influence?

We will model the network of relationships of NerdBook between users as a graph.



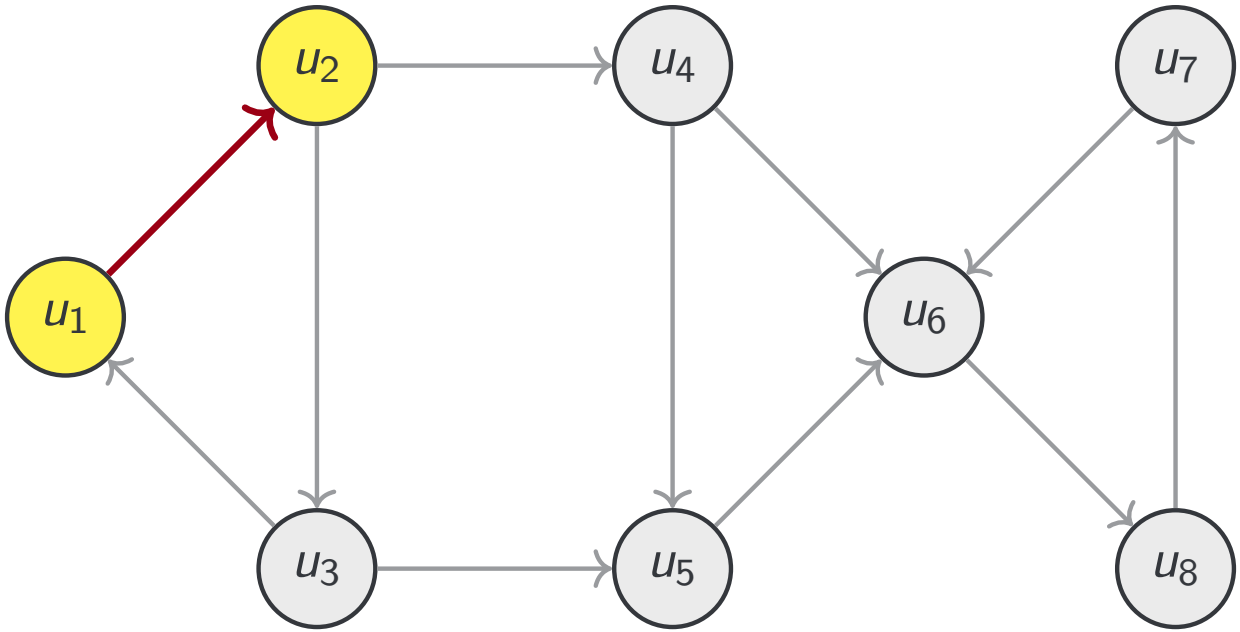
Influence?

Suppose u_1 knows MrHive (i.e. u_1 is **Active**), how could the information spread across the network?



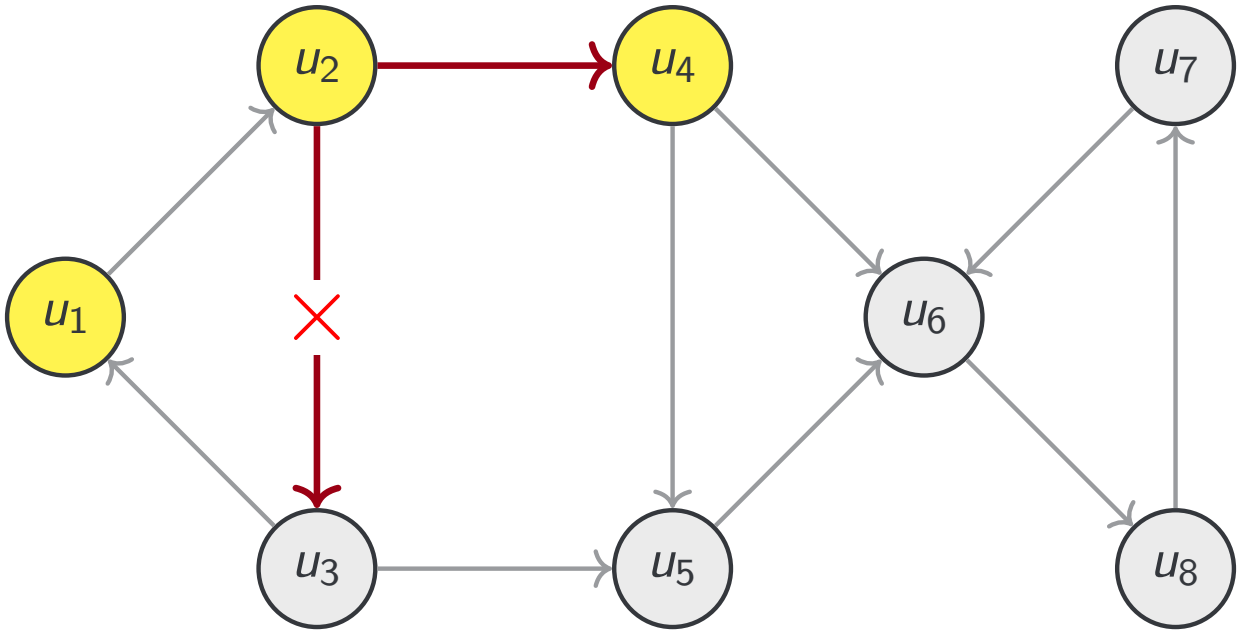
Influence?

Suppose u_1 knows MrHive (i.e. u_1 is **Active**), how could the information spread across the network?



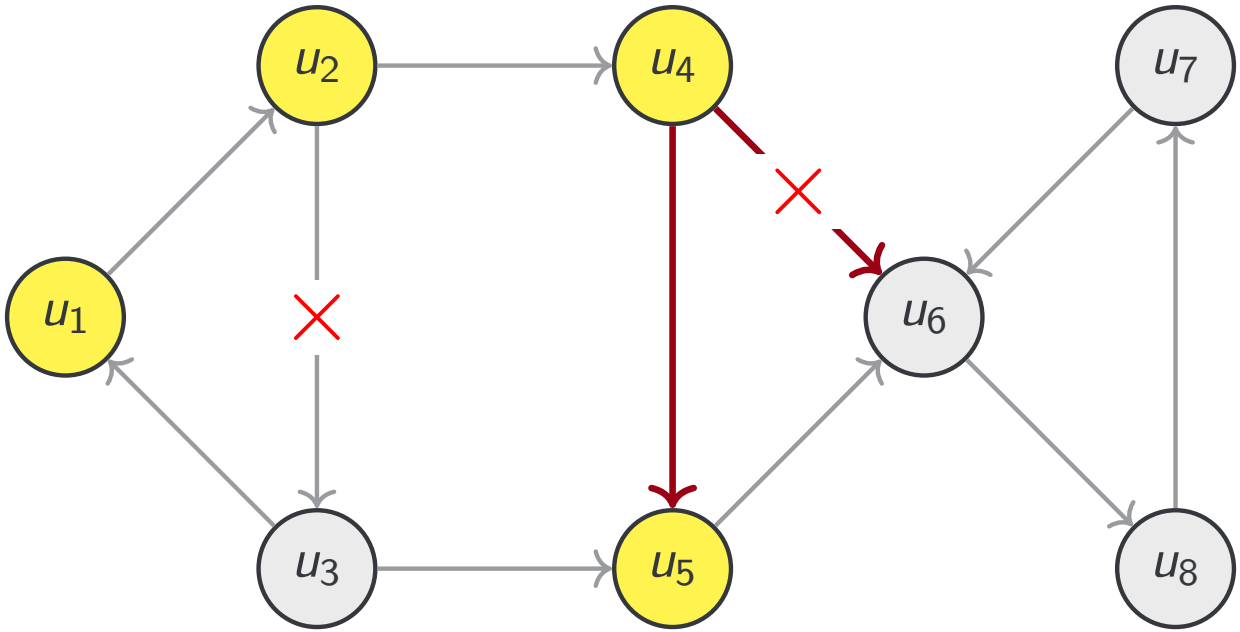
Influence?

Suppose u_1 knows MrHive (i.e. u_1 is **Active**), how could the information spread across the network?



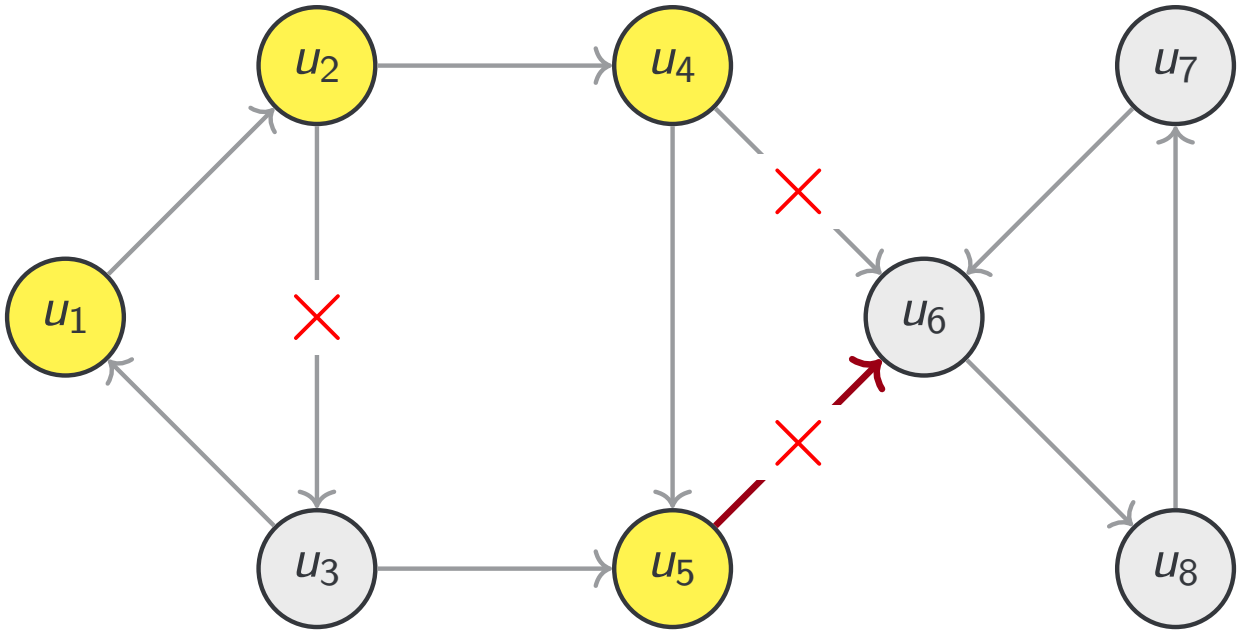
Influence?

Suppose u_1 knows MrHive (i.e. u_1 is **Active**), how could the information spread across the network?



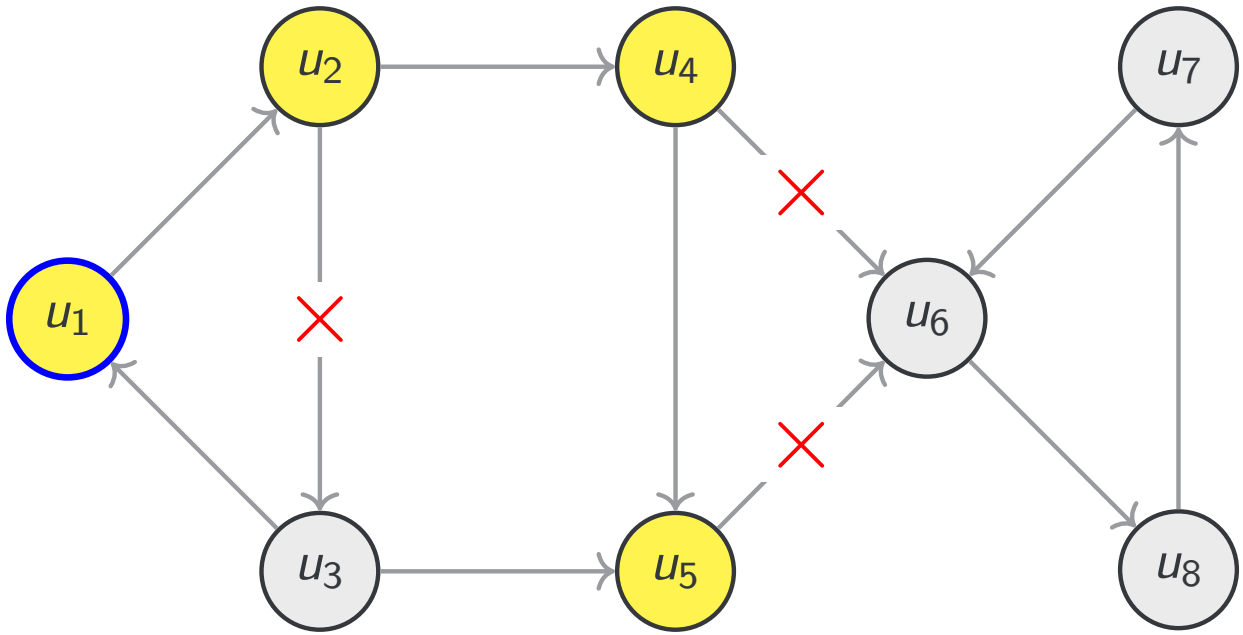
Influence?

Suppose u_1 knows MrHive (i.e. u_1 is **Active**), how could the information spread across the network?



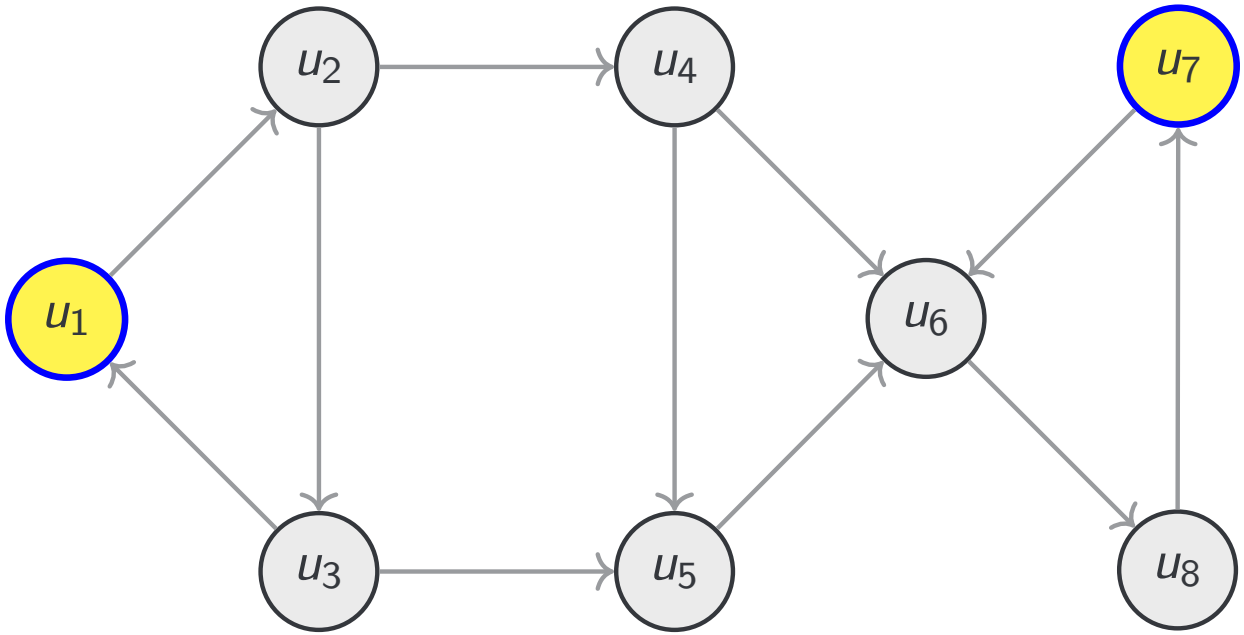
Influence?

When no new nodes activate, the process ends (**quiescence**).
In this example, with **seed set** $S = \{u_1\}$, the **influence spread** is 4.



Influence?

What if we had chosen a different seed set? Here $S = \{u_1, u_7\}$.



Influence

We will consider only **progressive** models, where once a node becomes active, it remains active for the rest of the process.

Influence

We will consider only **progressive** models, where once a node becomes active, it remains active for the rest of the process.

We can define the process more formally as follows:

Influence

We will consider only **progressive** models, where once a node becomes active, it remains active for the rest of the process.

We can define the process more formally as follows:

1. We start with a seed set $A_0 = S$ of active nodes at time $t = 0$.

Influence

We will consider only **progressive** models, where once a node becomes active, it remains active for the rest of the process.

We can define the process more formally as follows:

1. We start with a seed set $A_0 = S$ of active nodes at time $t = 0$.
2. At each time step t, A_{t-1} may activate their neighbors according **to some model**, giving us A_t.
   Note that $A_{t-1} \subseteq A_t$.

Influence

We will consider only **progressive** models, where once a node becomes active, it remains active for the rest of the process.

We can define the process more formally as follows:

1. We start with a seed set $A_0 = S$ of active nodes at time $t = 0$.
2. At each time step t, A_{t-1} may activate their neighbors according **to some model**, giving us A_t.
   Note that $A_{t-1} \subseteq A_t$.
3. If $A_t = A_{t-1}$, the process ends (quiescence).

Influence & Problem

Definition (Influence Spread)

Given a seed set S, the influence spread $\sigma(S)$ is the *expected number of active nodes* at the end of the diffusion process (time t).

$$\sigma(S) = \mathbb{E}[|A_t|]$$

Influence & Problem

Definition (Influence Spread)

Given a seed set S, the influence spread $\sigma(S)$ is the *expected number of active nodes* at the end of the diffusion process (time t).

$$\sigma(S) = \mathbb{E}[|A_t|]$$

Definition (Influence Maximization Problem)

Given a graph G and a budget k, find a seed set S of size at most k that maximizes $\sigma(S)$.

Models

Summary

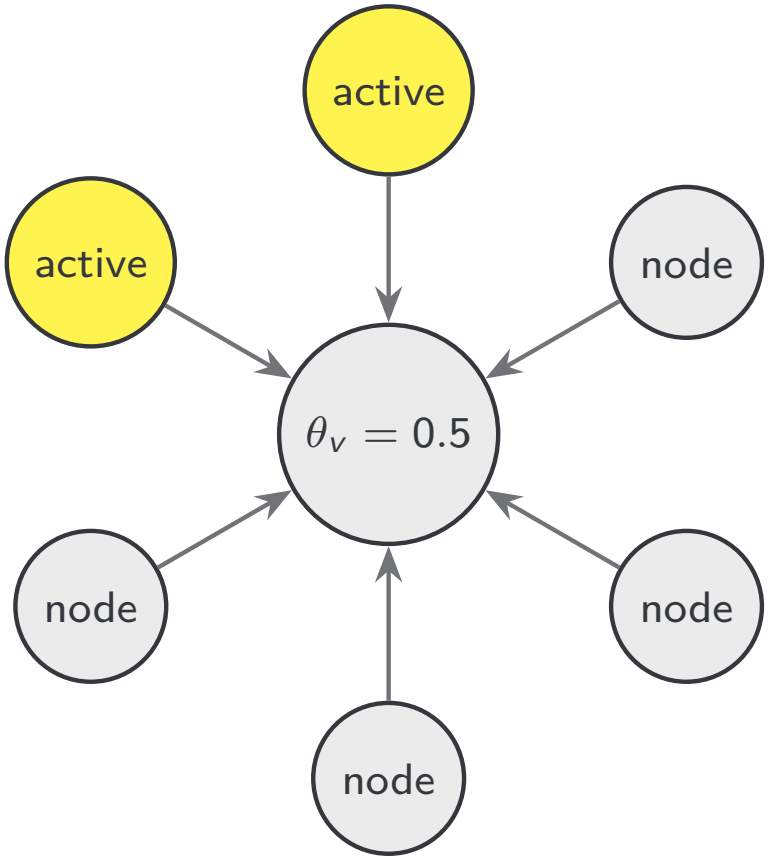
1. **Threshold Models**: GTM, LTM
2. **Cascade Models**: IC

Threshold Models

- Each node has a threshold $\theta_v \in [0, 1]$.
- A node v activates if:

$$f_v(S_v) \geq \theta_v$$

where $f_v(S_v)$ is a function of the active neighbors of v at time $t - 1$.



General Threshold Model

The **General Threshold Model (GTM)** is defined by letting $f_v(S_v)$ be any monotone function.

General Threshold Model

The **General Threshold Model (GTM)** is defined by letting $f_v(S_v)$ be any monotone function.

Theorem

In general, it is NP-hard to approximate the influence maximization problem to within a factor of $n^{1-\varepsilon}$ for any $\varepsilon > 0$.

This motivates the study of simpler models...

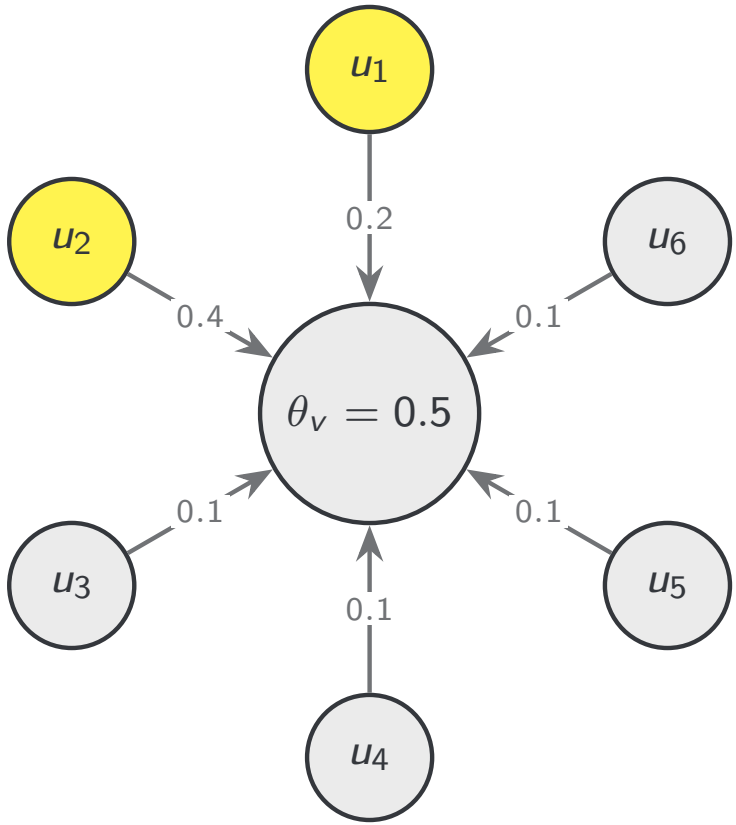
Model: LTM

Linear Threshold Model (LTM):

- Each node has a random threshold $\theta_v \sim U[0,1]$.
- Each edge (u, v) has a weight w_{uv}, such that

$$\sum_{u \in In(v)} w_{uv} \leq 1$$

- $$f_v(S_v) = \sum_{u \in S_v} w_{uv}$$



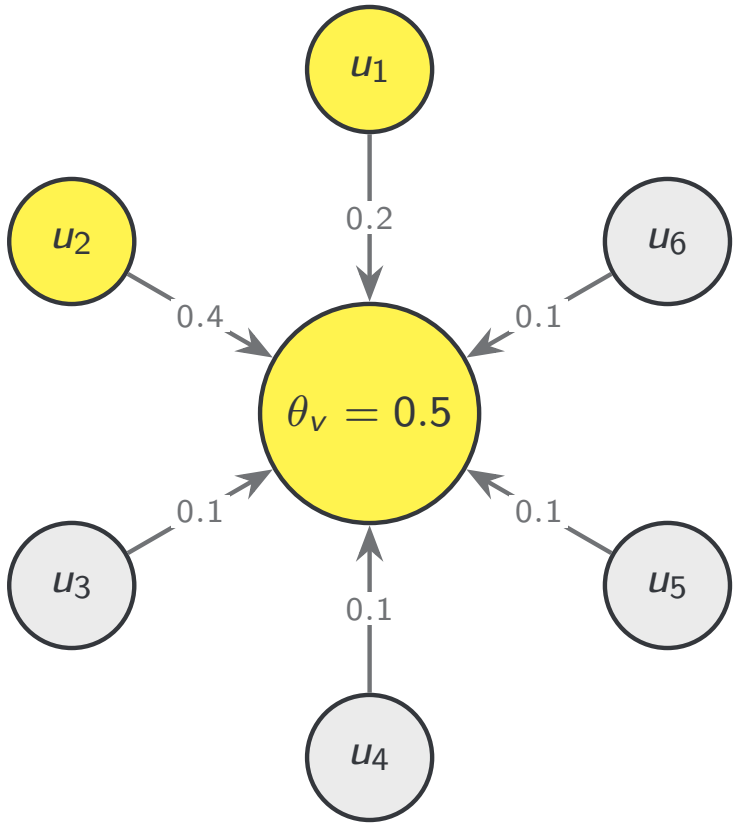
Model: LTM

Linear Threshold Model (LTM):

- Each node has a random threshold $\theta_v \sim U[0,1]$.
- Each edge (u, v) has a weight w_{uv}, such that

$$\sum_{u \in In(v)} w_{uv} \leq 1$$

- $$f_v(S_v) = \sum_{u \in S_v} w_{uv}$$



Complexity of LTM

Theorem

For the Linear Threshold Model, the influence maximization problem is NP-hard.

Complexity of LTM

Theorem

For the Linear Threshold Model, the influence maximization problem is NP-hard.

- What if we look for approximation algorithms?

Complexity of LTM

Theorem

For the Linear Threshold Model, the influence maximization problem is NP-hard.

- What if we look for approximation algorithms?
- Could we also improve the model to capture more complex phenomena?

Summary

1. **Threshold Models**: GTM, LTM
2. **Cascade Models**: IC

Independent Cascade (IC)

The **Independent Cascade Model (IC)** is a stochastic model where the activation process unfolds as follows:

Independent Cascade (IC)

The **Independent Cascade Model (IC)** is a stochastic model where the activation process unfolds as follows:

- Each edge (u, v) has an associated probability p_{uv}.

Independent Cascade (IC)

The **Independent Cascade Model (IC)** is a stochastic model where the activation process unfolds as follows:

- Each edge (u, v) has an associated probability p_{uv}.
- When node u becomes active at time t, it has a single opportunity to activate each of its neighbors v at time $t + 1$ with probability p_{uv}.

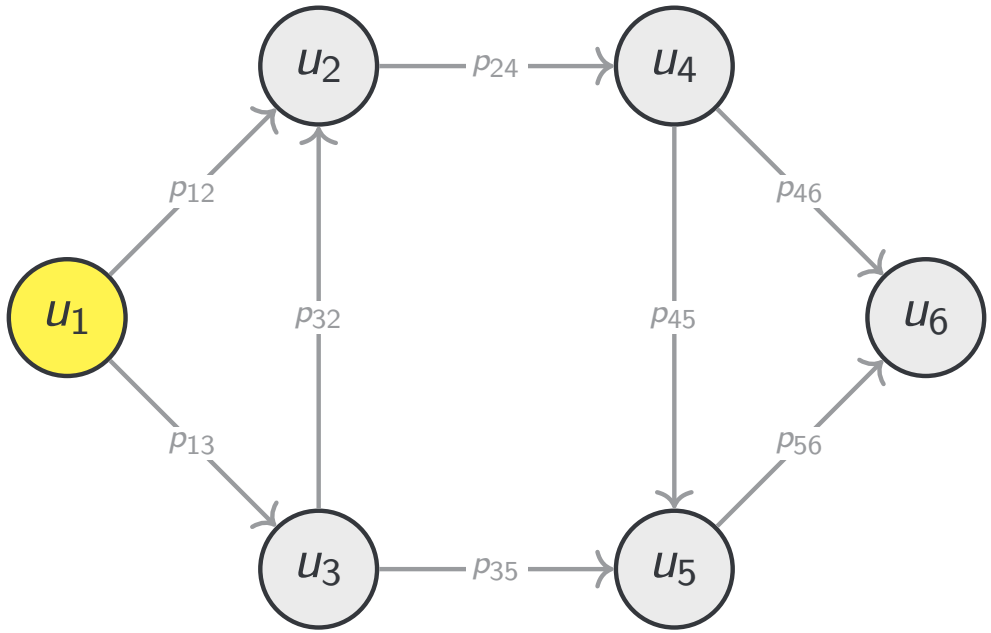
Independent Cascade (IC)

The **Independent Cascade Model (IC)** is a stochastic model where the activation process unfolds as follows:

- Each edge (u, v) has an associated probability p_{uv}.
- When node u becomes active at time t, it has a single opportunity to activate each of its neighbors v at time $t + 1$ with probability p_{uv}.

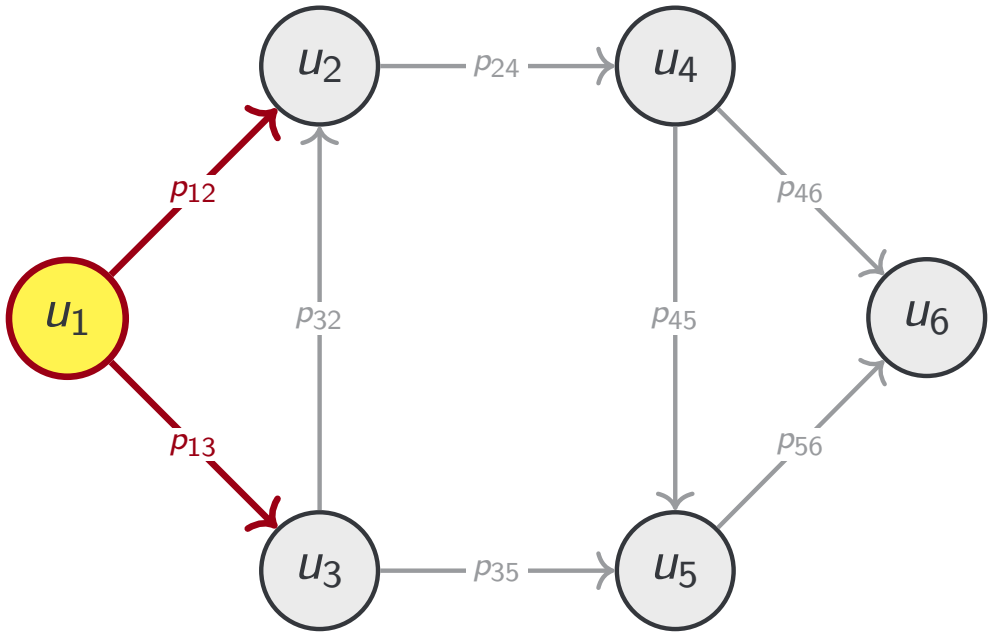
Let's trace the process in a simple example...

IC Example



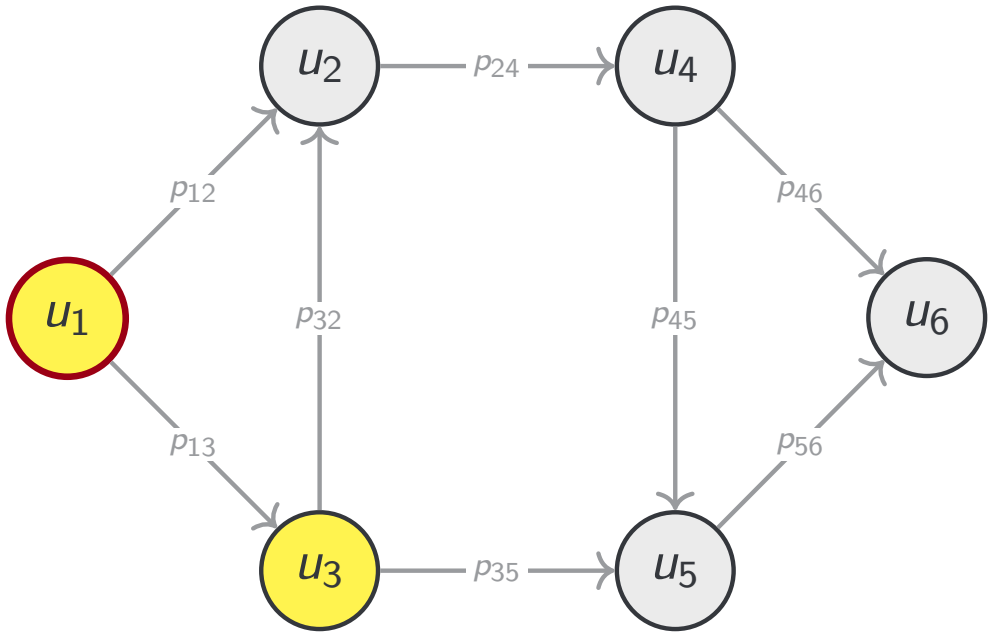
$t = 0$: u_1 is active.

IC Example



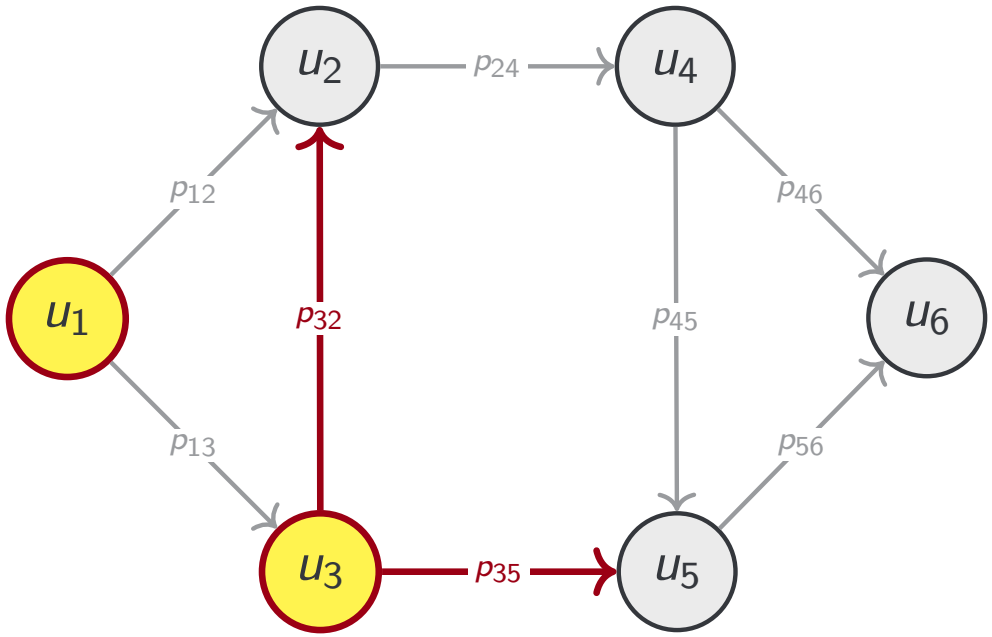
$t = 0$

IC Example



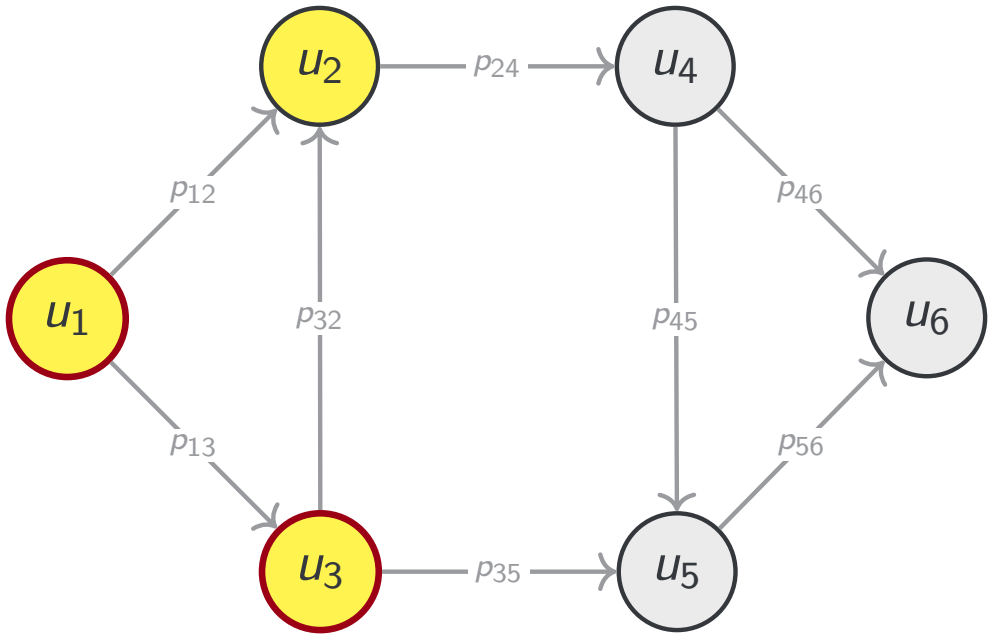
$$t = 1$$

IC Example



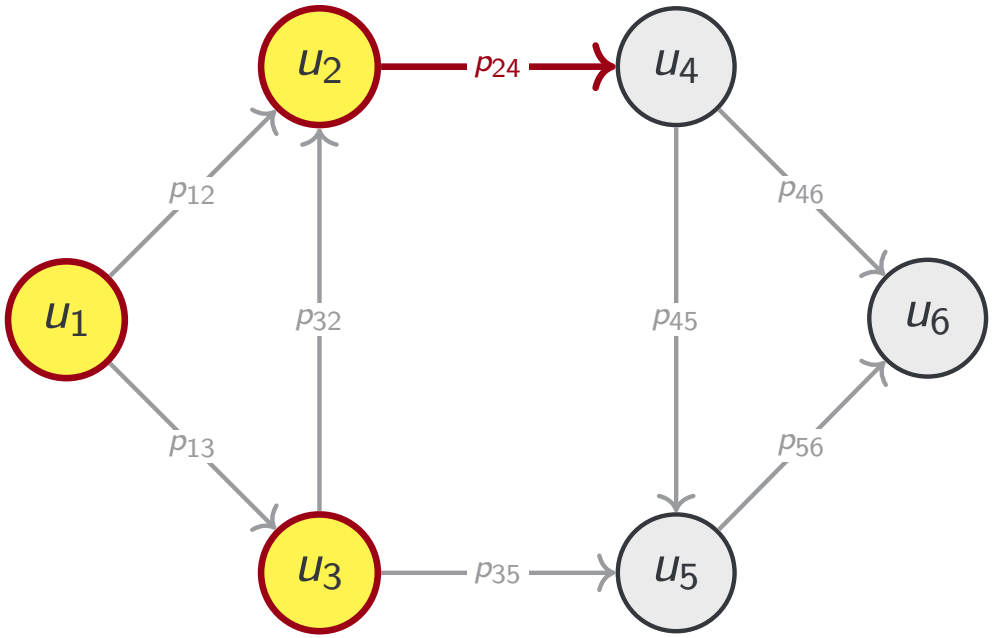
$t = 1$

IC Example



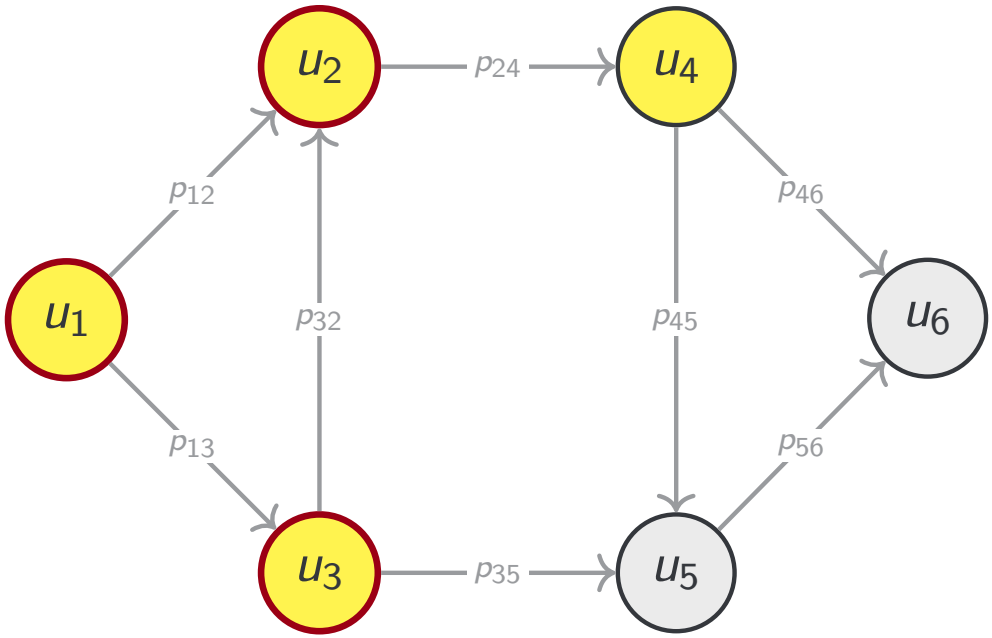
$t = 2$

IC Example



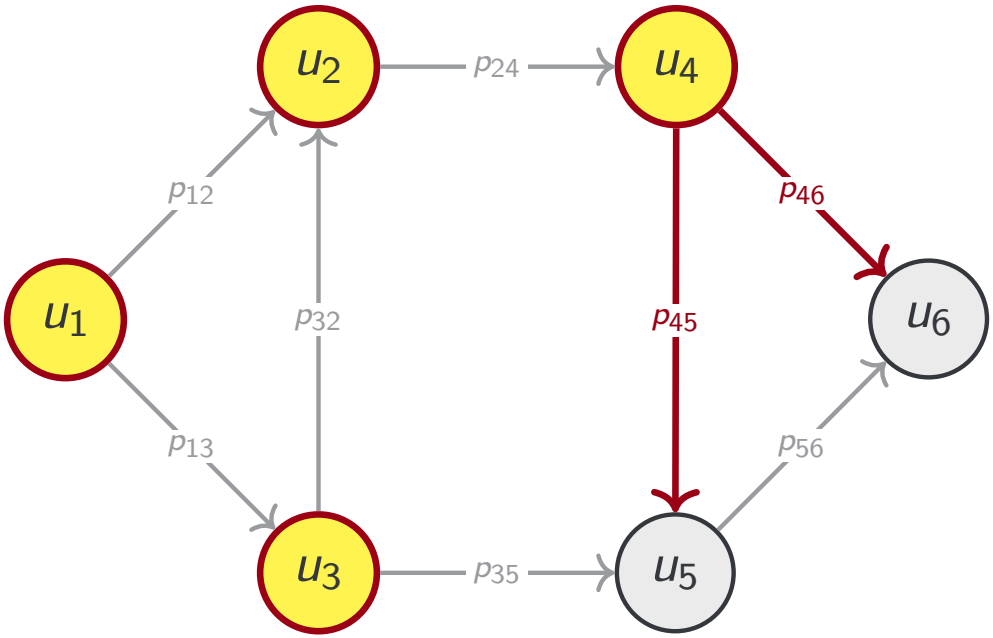
$t = 2$

IC Example



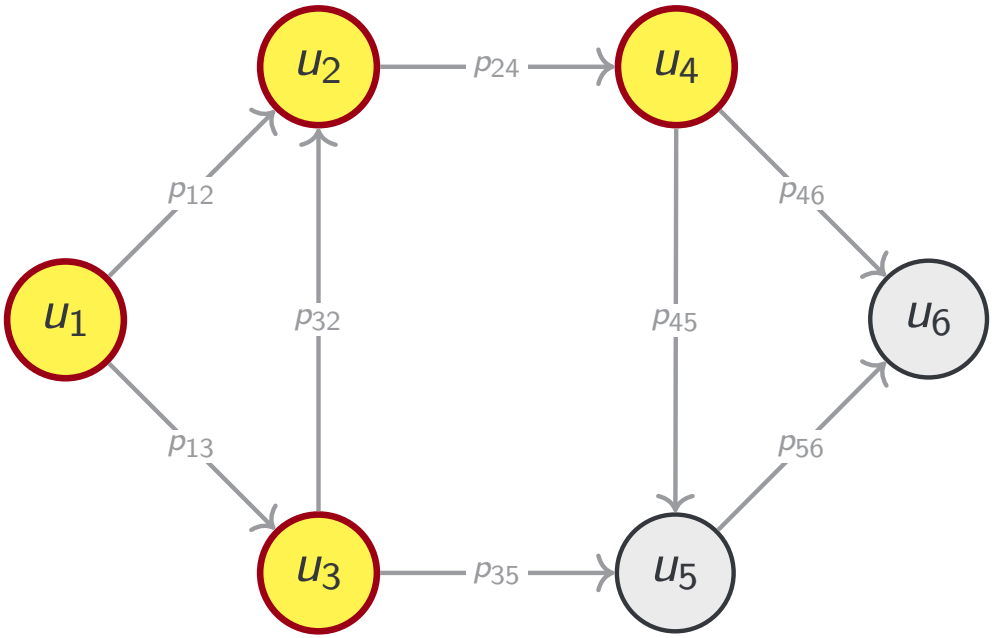
$t = 3$

IC Example



$t = 3$

IC Example



$t = 4$: Quiescence is reached as all activation attempts have concluded.

IC Complexity

As with threshold models, finding an exact solution for IC is intractable:

Theorem

For the Independent Cascade Model, the influence maximization problem is NP-hard.

Approximation Limits

There are also fundamental limits on approximation:

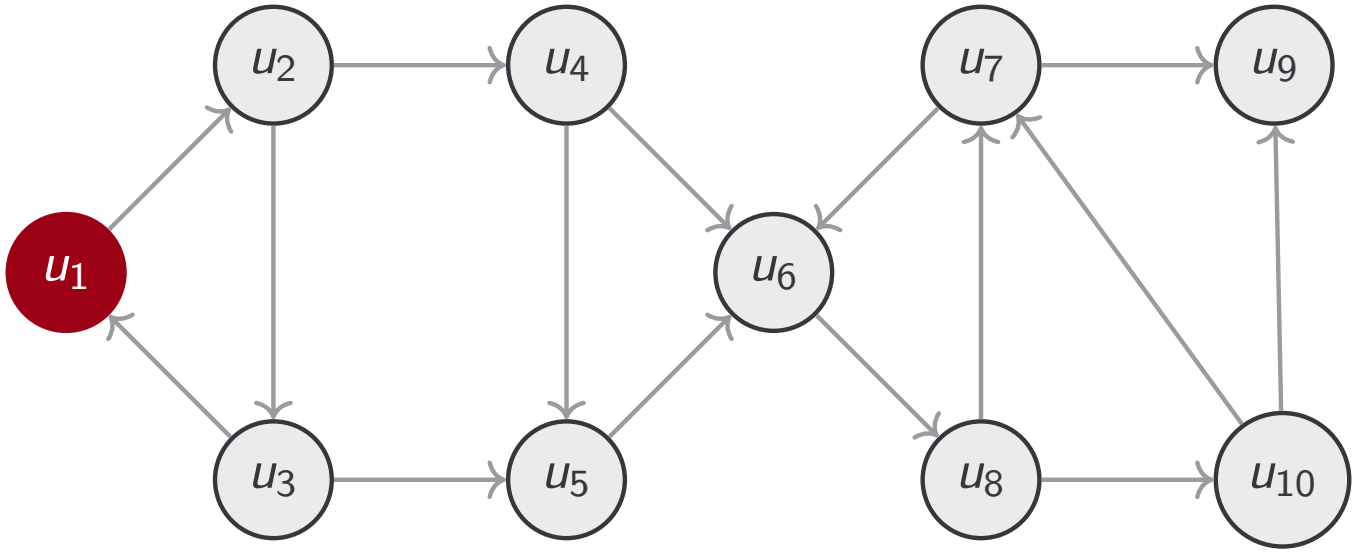
Theorem

For the Independent Cascade Model, it is NP-hard to approximate the influence maximization problem within a factor better than $1 - \frac{1}{e}$.

This $(1 - 1/e)$ bound serves as our primary goal for approximation algorithms.

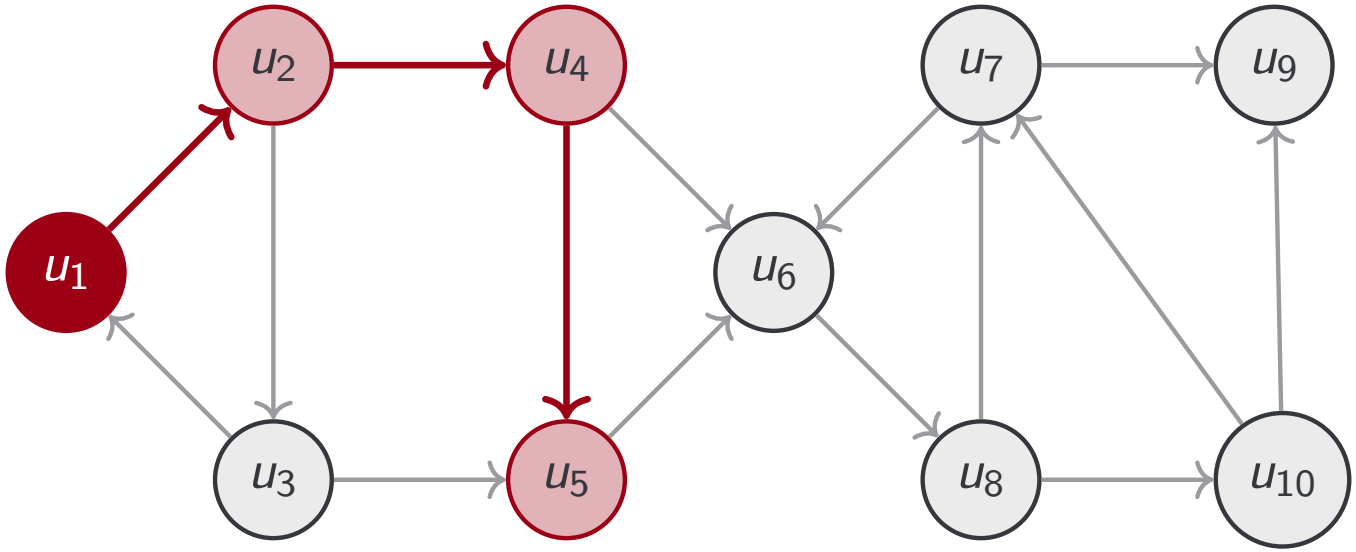
Submodularity

The idea



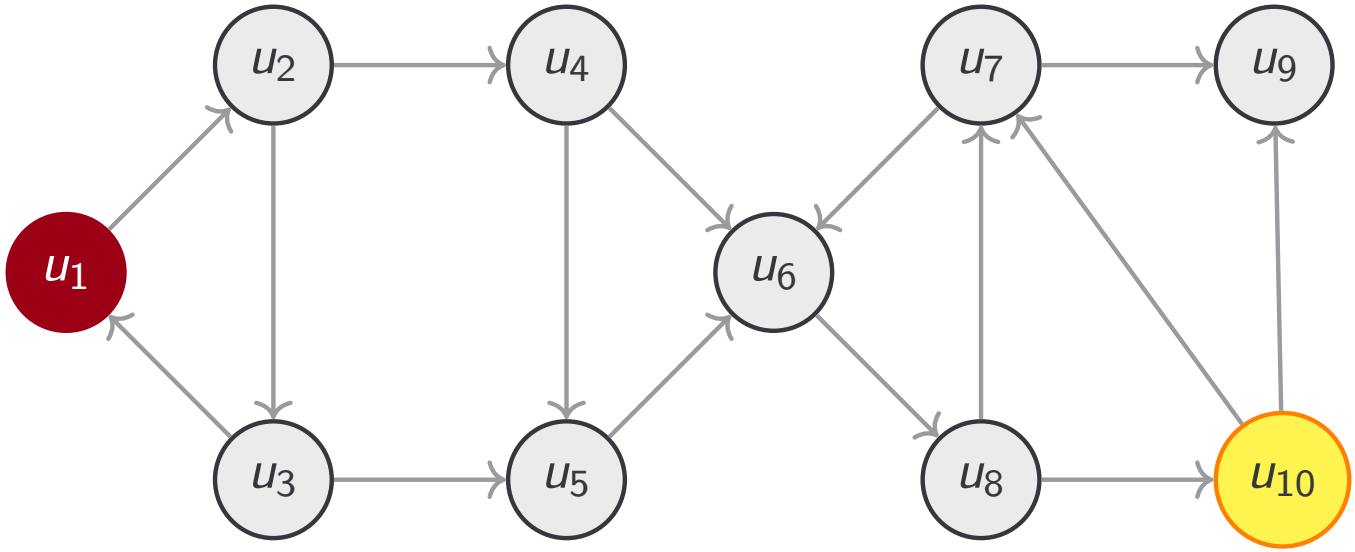
Set seed $S = \{u_1\}$

The idea



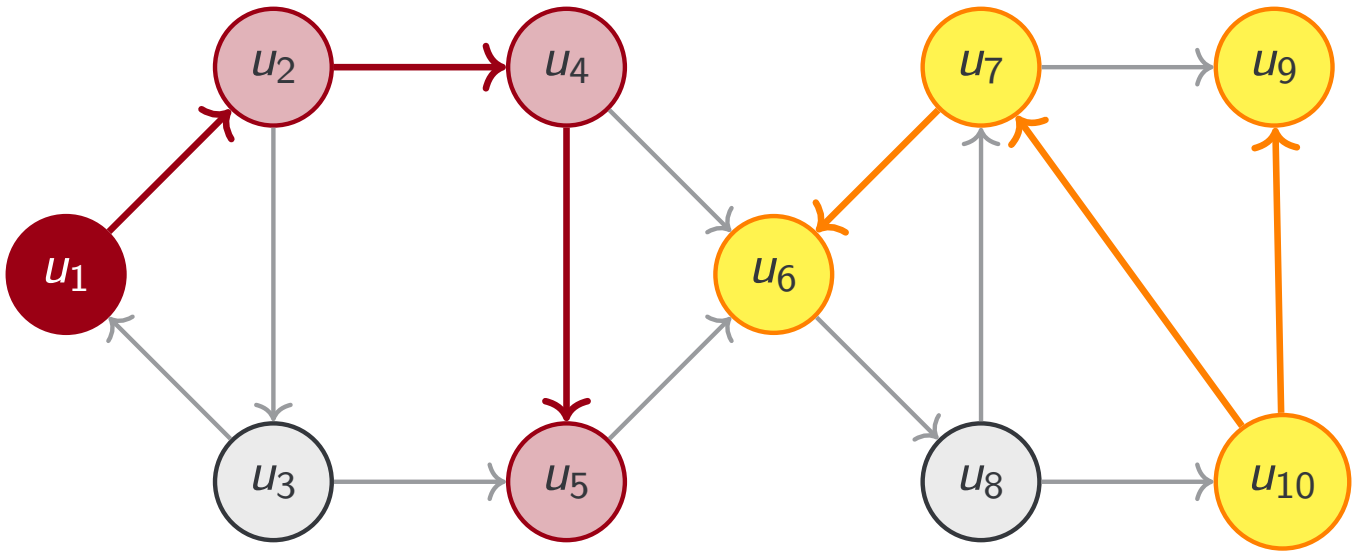
$$\sigma(S) = 4$$

Marginal Gain



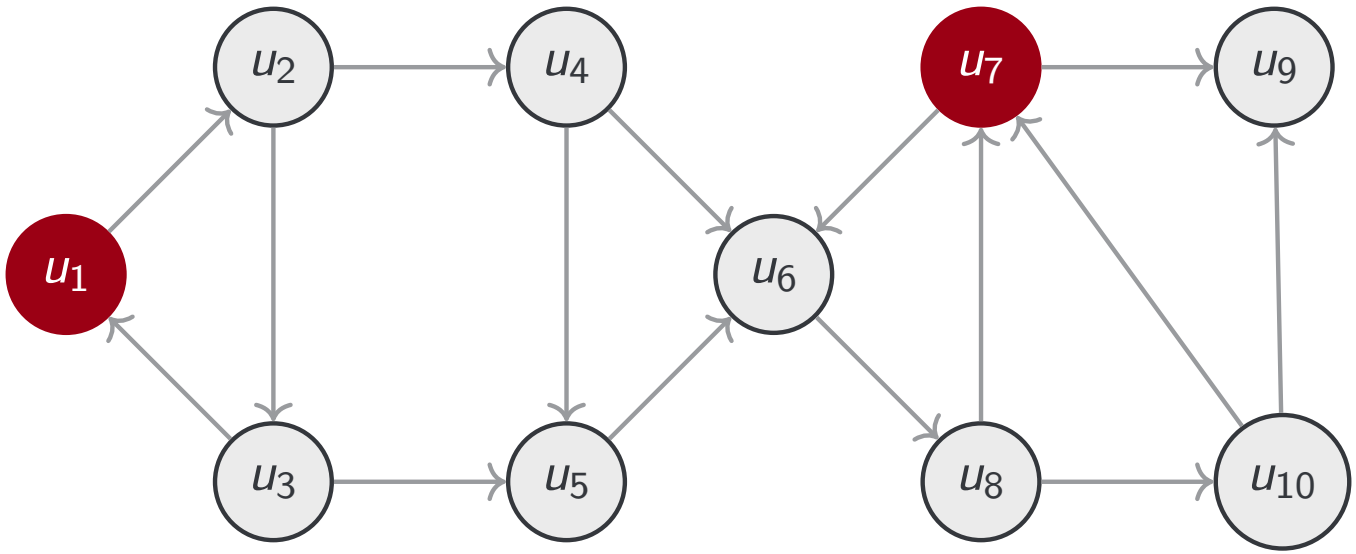
What if we add a new node to the seed set? How many new nodes will be activated?

Marginal Gain



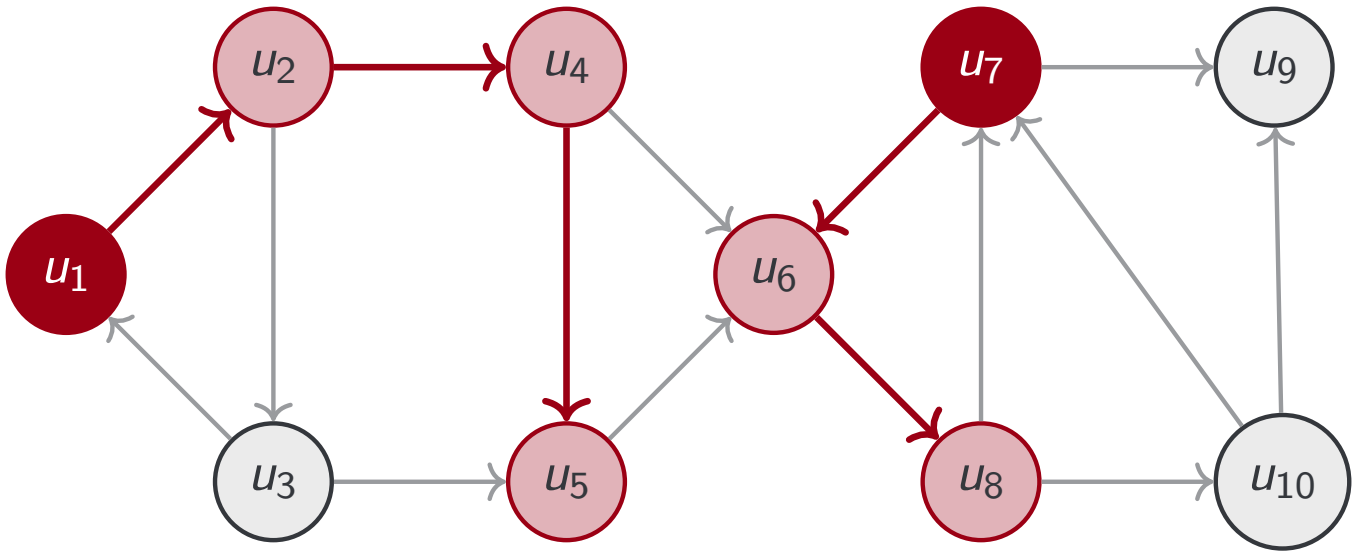
$$\sigma(S \cup \{u_{10}\}) - \sigma(S) = 4$$

The idea



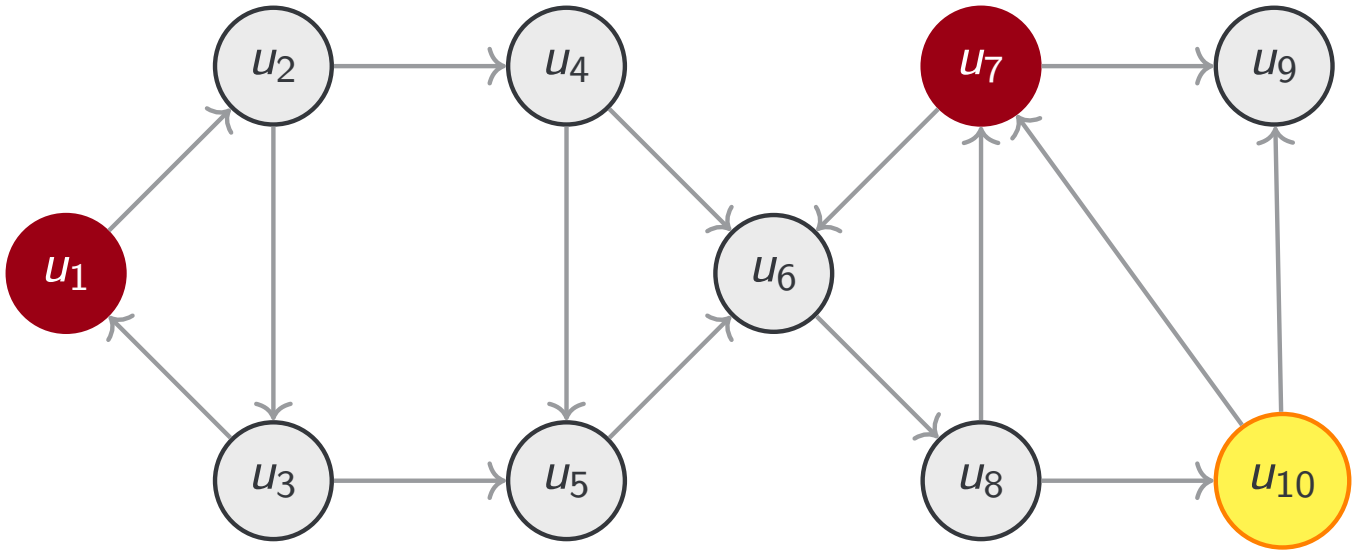
Now consider a larger seed set $T = \{u_1, u_7\} \supset S$

The idea



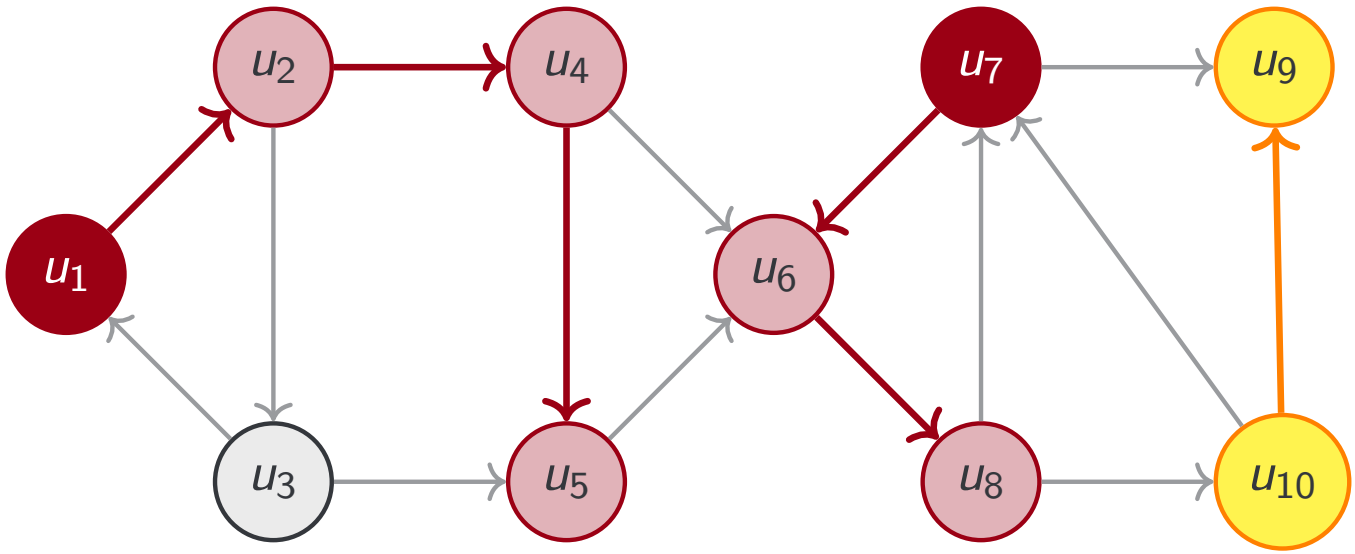
$$\sigma(T) = 7 > \sigma(S)$$

The idea



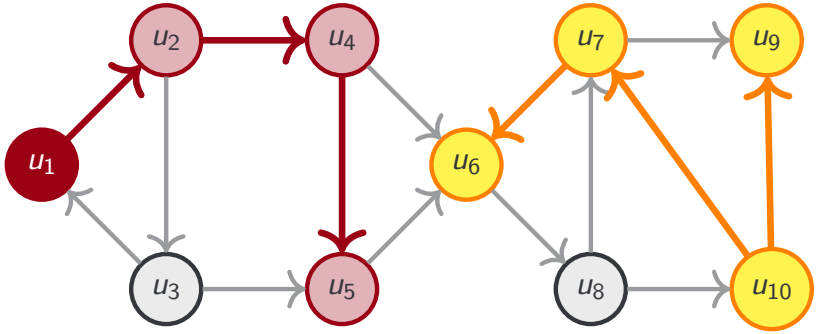
What is the marginal gain of adding u_{10} to T?

The idea

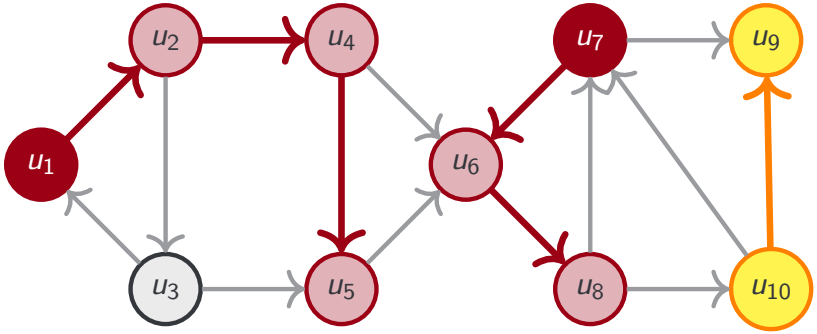


$$\sigma(T \cup \{u_{10}\}) - \sigma(T) = 2$$

The idea



Gain $= \mathbf{4}$



Gain $= \mathbf{2}$

$$\sigma(S \cup \{u_{10}\}) - \sigma(S) \geq \sigma(T \cup \{u_{10}\}) - \sigma(T)$$

$$S \subseteq T$$

The idea

Diminishing Returns: Effectiveness saturates as the active set grows in size. The influence function $\sigma(S)$ should reflect this property.

The idea

Diminishing Returns: Effectiveness saturates as the active set grows in size. The influence function $\sigma(S)$ should reflect this property.

Definition (Submodularity)

A set function $f : 2^V \to \mathbb{R}_+$ is **submodular** if for all $A \subseteq B \subseteq V$ and $x \in V \setminus B$:

$$f(A \cup \{x\}) - f(A) \geq f(B \cup \{x\}) - f(B)$$

The marginal gain of adding an element x to a set A is at least as large as adding it to any superset B.

The Greedy Algorithm

Theorem

If $\sigma(\cdot)$ is non-negative, monotone, and submodular, then the greedy algorithm that (for k iterations) adds an element with the largest marginal increase in $\sigma(\cdot)$ produces a k-element set S such that:

$$\sigma(S) \geq (1 - 1/e)\sigma(S^*)$$

where S^* is the optimal k-element set.

This provides a $(1 - 1/e) \approx 63\%$ approximation guarantee.

The Greedy Algorithm

Algorithm 1 Greedy Submodular Influence Maximization

1: $S \leftarrow \emptyset$
2: **while** $|S| < k$ **do**
3: $\quad v^* \leftarrow \text{argmax}_{v \in V \setminus S}(\sigma(S \cup \{v\}) - \sigma(S))$
4: $\quad S \leftarrow S \cup \{v^*\}$
5: **end while**

So simple! and yet, it guarantees a $(1 - 1/e)$ approximation to the optimal solution.

Proof Sketch (1/2)

The proof of the greedy approximation bound is a fundamental result in optimization.

Let S_i be the set selected after i iterations, and S^* the optimal set ($|S^*| = k$). By **monotonicity** and **submodularity**:

$$\sigma(S^*) \leq \sigma(S_i \cup S^*) \leq \sigma(S_i) + \sum_{x \in S^*} (\sigma(S_i \cup \{x\}) - \sigma(S_i))$$

Proof Sketch (1/2)

The proof of the greedy approximation bound is a fundamental result in optimization.

Let S_i be the set selected after i iterations, and S^* the optimal set ($|S^*| = k$). By **monotonicity** and **submodularity**:

$$\sigma(S^*) \leq \sigma(S_i \cup S^*) \leq \sigma(S_i) + \sum_{x \in S^*} (\sigma(S_i \cup \{x\}) - \sigma(S_i))$$

Since the greedy step chooses v_{i+1} to maximize gain:

$$\sigma(S_i \cup \{x\}) - \sigma(S_i) \leq \sigma(S_{i+1}) - \sigma(S_i)$$

Proof Sketch (1/2)

The proof of the greedy approximation bound is a fundamental result in optimization.

Let S_i be the set selected after i iterations, and S^* the optimal set ($|S^*| = k$). By **monotonicity** and **submodularity**:

$$\sigma(S^*) \leq \sigma(S_i \cup S^*) \leq \sigma(S_i) + \sum_{x \in S^*} (\sigma(S_i \cup \{x\}) - \sigma(S_i))$$

Since the greedy step chooses v_{i+1} to maximize gain:

$$\sigma(S_i \cup \{x\}) - \sigma(S_i) \leq \sigma(S_{i+1}) - \sigma(S_i)$$

Summing over S^* (k elements), we obtain:

$$\sigma(S^*) \leq \sigma(S_i) + k(\sigma(S_{i+1}) - \sigma(S_i))$$

Proof Sketch (2/2)

Let $\delta_i = \sigma(S^*) - \sigma(S_i)$ be the gap to optimality.

Gap Reduction: Rearranging the previous inequality:

$$\delta_{i+1} \leq (1 - 1/k)\delta_i$$

Proof Sketch (2/2)

Let $\delta_i = \sigma(S^*) - \sigma(S_i)$ be the gap to optimality.

Gap Reduction: Rearranging the previous inequality:

$$\delta_{i+1} \leq (1 - 1/k)\delta_i$$

By induction, after k iterations:

$$\delta_k \leq (1 - 1/k)^k \delta_0$$

Proof Sketch (2/2)

Let $\delta_i = \sigma(S^*) - \sigma(S_i)$ be the gap to optimality.

Gap Reduction: Rearranging the previous inequality:

$$\delta_{i+1} \leq (1 - 1/k)\delta_i$$

By induction, after k iterations:

$$\delta_k \leq (1 - 1/k)^k \delta_0$$

Using the inequality $(1 - 1/k)^k \leq 1/e$:

$$\sigma(S^*) - \sigma(S_k) \leq \frac{1}{e}\sigma(S^*)$$

Final Bound: $\sigma(S_k) \geq (1 - 1/e)\sigma(S^*) \approx 0.63 \cdot \text{OPT}$

Estimating $\sigma(S)$

The greedy algorithm assumes $\sigma(S)$ can be computed efficiently. However, this is not true for models like LTM and IC.

Estimating $\sigma(S)$

The greedy algorithm assumes $\sigma(S)$ can be computed efficiently. However, this is not true for models like LTM and IC.

Theorem

*For the LTM and IC models, computing $\sigma(S)$ exactly is **#P-hard**.*

Estimating $\sigma(S)$

The greedy algorithm assumes $\sigma(S)$ can be computed efficiently. However, this is not true for models like LTM and IC.

Theorem

*For the LTM and IC models, computing $\sigma(S)$ exactly is **#P-hard**.*

Fortunately, we can get arbitrary good approximations to $\sigma(S)$ by Monte Carlo simulations of the diffusion process.

Estimating $\sigma(S)$

Since $1 \leq \sigma(S) \leq n$, we can prove by the standard Chernoff bound that:

Theorem

If the diffusion process starting from seed set S is simulated independently at least

$$\Omega\left(\frac{n^2}{\varepsilon^2}\log\frac{1}{\delta}\right)$$

times, then the average number of active nodes at the end of the process is an $(1 \pm \varepsilon)$-approximation of $\sigma(S)$ with probability at least $1 - \delta$.

This allows the greedy algorithm to remain effective in practice.

Submodular Thresholds

Submodular Threshold Model (STM):

Nice immediate generalization of the LTM:

- Each node v has a random threshold $\theta_v \sim U[0, 1]$.
- Each node v has a **submodular** function $f_v(S_v)$ of its active neighbors.
- v activates if $f_v(S_v) \geq \theta_v$.

Submodular Thresholds

Submodular Threshold Model (STM):
Nice immediate generalization of the LTM:

- Each node v has a random threshold $\theta_v \sim U[0, 1]$.
- Each node v has a **submodular** function $f_v(S_v)$ of its active neighbors.
- v activates if $f_v(S_v) \geq \theta_v$.

Theorem (Mossel and Roch)

For the Submodular Threshold Model, the influence function $\sigma(S)$ is submodular.

Thus, the $(1 - 1/e)$ greedy approximation guarantee also applies to the STM.

Conclusions

- We formalized the **Influence Maximization** problem in social networks.

Conclusions

- We formalized the **Influence Maximization** problem in social networks.
- **Basic Models**: Reviewed **GTM**, **LTM**, and **IC**, and their limitations.

Conclusions

- We formalized the **Influence Maximization** problem in social networks.
- **Basic Models**: Reviewed **GTM**, **LTM**, and **IC**, and their limitations.
- **Submodularity**:
  We formalize the concept of **diminishing returns**

Conclusions

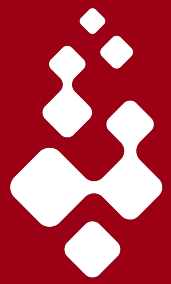
- We formalized the **Influence Maximization** problem in social networks.
- **Basic Models**: Reviewed **GTM**, **LTM**, and **IC**, and their limitations.
- **Submodularity**:
  We formalize the concept of **diminishing returns**
- **Greedy Algorithm**: A simple, efficient strategy that guarantees a $(1 - 1/e) \approx 63\%$ approximation.

Conclusions

- We formalized the **Influence Maximization** problem in social networks.
- **Basic Models**: Reviewed **GTM**, **LTM**, and **IC**, and their limitations.
- **Submodularity**:
  We formalize the concept of **diminishing returns**
- **Greedy Algorithm**: A simple, efficient strategy that guarantees a $(1 - 1/e) \approx 63\%$ approximation.
- **Generalization**: The **STM** extends these guarantees to a broader class of realistic influence functions.

Thank you for your attention!

UNIVERSITÀ
DEGLI STUDI
DI PADOVA



elicsir
Elevating Italian Computer Science
Innovation and Research