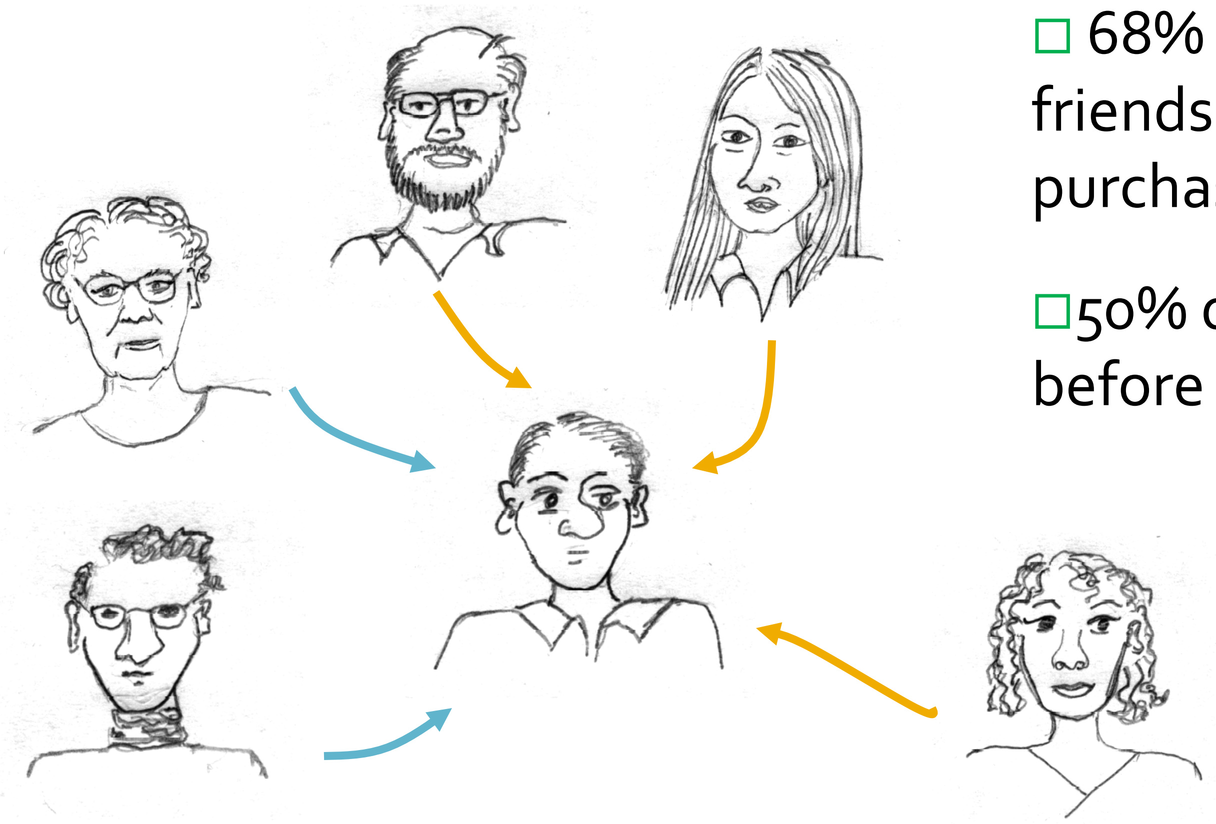
Influence Maximization in Networks

CS224W: Analysis of Networks
Jure Leskovec, Stanford University
http://cs224w.stanford.edu

Viral Marketing?

- **We are more influenced by our friends than strangers**



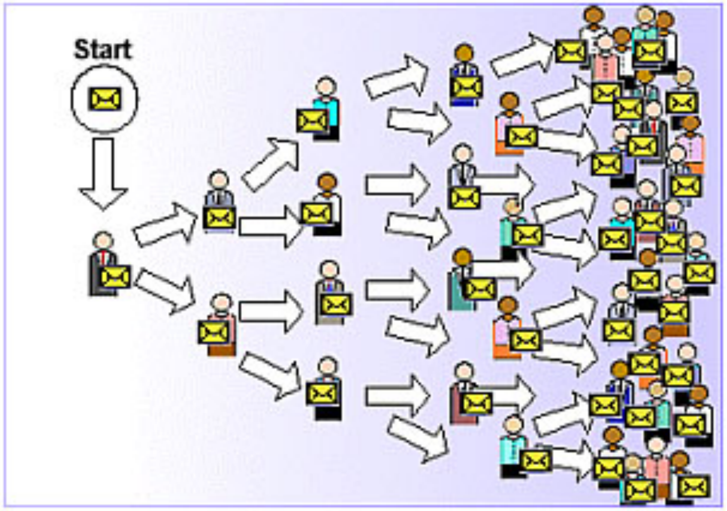
- 68% of consumers consult friends and family before purchasing home electronics
- 50% do research online before purchasing electronics

Viral Marketing

Identify influential customers

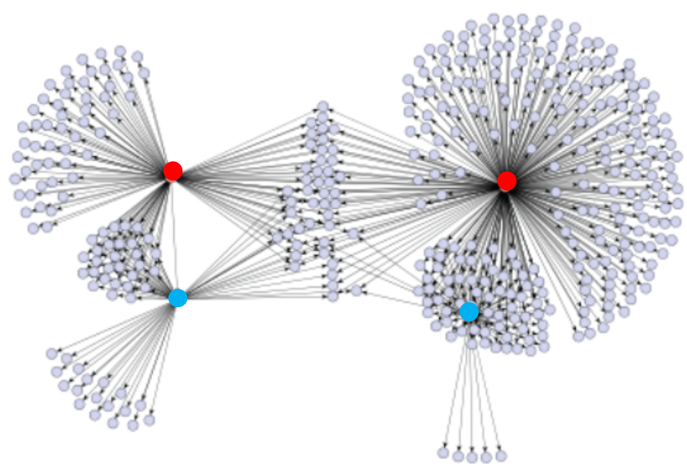
Convince them to adopt the product – Offer discount or free samples

These customers endorse the product among their friends



How to Create Big Cascades?

- **Information epidemics:**
  - Which are the influential users?
  - Which news sites create big cascades?
  - Where should we advertise?



Which node shall we target?

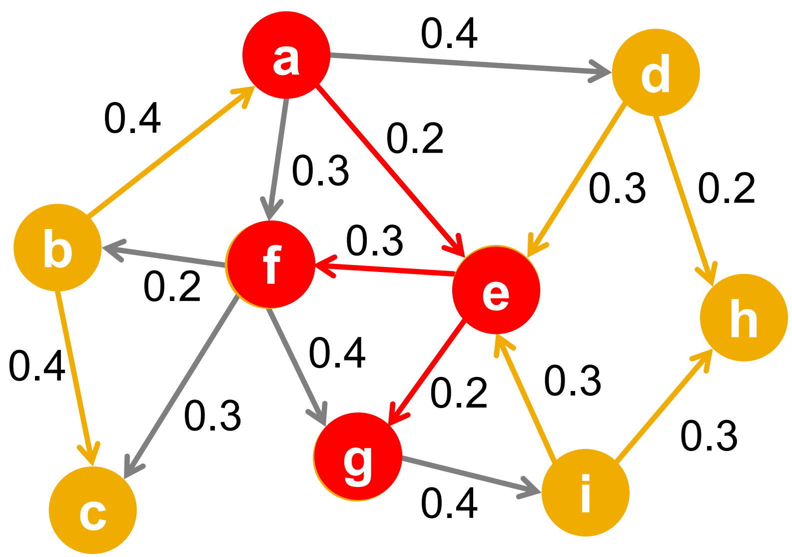
vs.

Probabilistic Contagion

- **Independent Cascade Model**
  - Directed finite $G = (V, E)$
  - Set S starts out with new behavior
    - Say nodes with this behavior are "**active**"
  - Each edge (v, w) has a probability p_{vw}
  - If node v is active, it gets **one** chance to make w active, with probability p_{vw}
    - Each edge fires at most once
- **Does scheduling matter? No**
    - If u, v are both active at the same time, it doesn't matter which tries to activate w first
  - **But the time moves in discrete steps**

Independent Cascade Model

- **Initially some nodes *S* are active**
- Each edge $(\boldsymbol{v}, \boldsymbol{w})$ has probability (weight) $\boldsymbol{p_{vw}}$

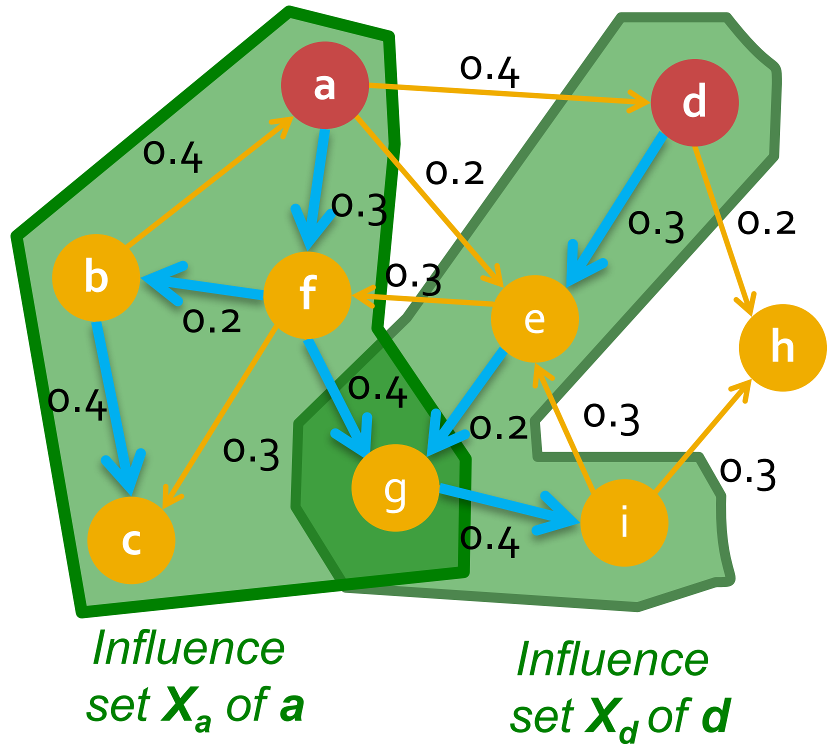


- **When node *v* becomes active:**
  - It activates each out-neighbor $\boldsymbol{w}$ with prob. $\boldsymbol{p_{vw}}$
- **Activations spread through the network**

Most Influential Set

Problem: **(*k* is a user-specified parameter)**

- **Most influential set of size *k*:** set ***S*** of ***k*** nodes producing largest **expected** cascade size ***f(S)*** if activated [Domingos-Richardson '01]
- **Optimization problem:** $\max_{S \text{ of size } k} f(S)$

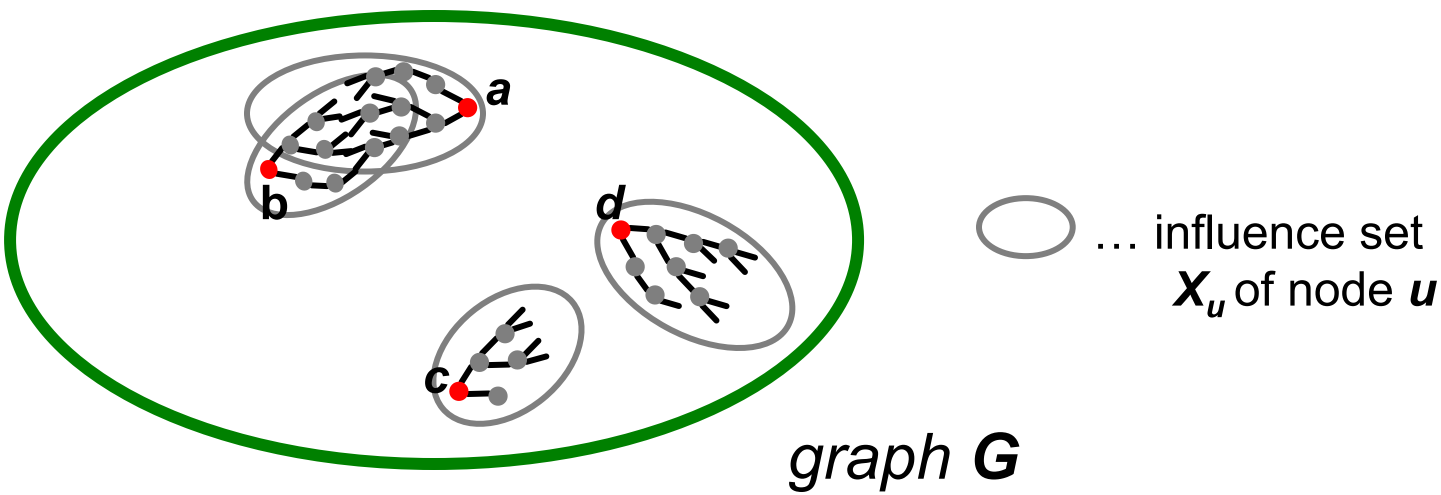


Why "expected cascade size"? X_a is a result of a random process. So in practice we would want to compute X_a for many random realizations and then maximize the "average" value *f(S)*. For now let's ignore this nuisance and simply assume that each node *u* influences *a* set of nodes X_u

$$f(S) = \frac{1}{|I|} \sum_{\text{Random realizations } i} f_i(S)$$

Most Influential Set of Nodes

- ***S***: is initial active set
- ***f(S)***: **The expected size of final active set**
  - ***f(S)*** is the size of the union of X_u: $\boldsymbol{f(S)} = |\cup_{\boldsymbol{u \in S}} \boldsymbol{X_u}|$



- **Set *S* is more influential if *f(S)* is larger**

$$\boldsymbol{f(\{a,b\}) < f(\{a,c\}) < f(\{a,d\})}$$

How hard is influence maximization?

Most Influential Subset of Nodes

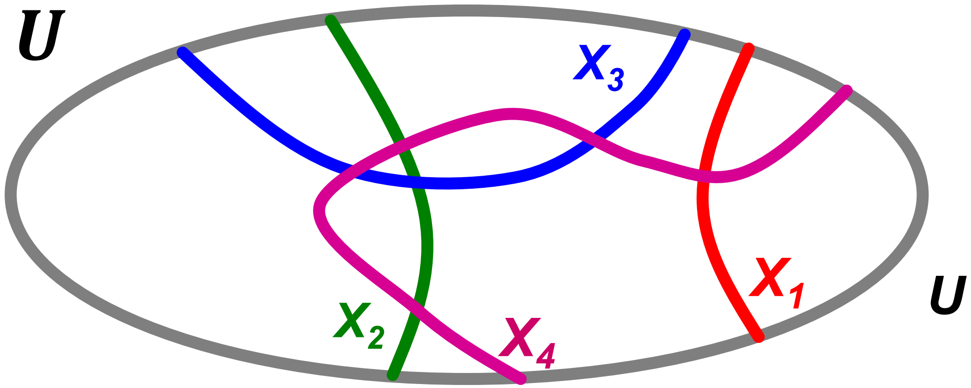
- **Problem: Most influential set of *k* nodes:** set ***S*** on ***k*** nodes producing largest expected cascade size ***f(S)*** if activated
- **The optimization problem:**

$$\max_{\text{S of size k}} f(S)$$

- **How hard is this problem?**
  - **NP-COMPLETE!**
    - Show that finding most influential set is at least as hard as a **set cover problem**

Background: Set Cover

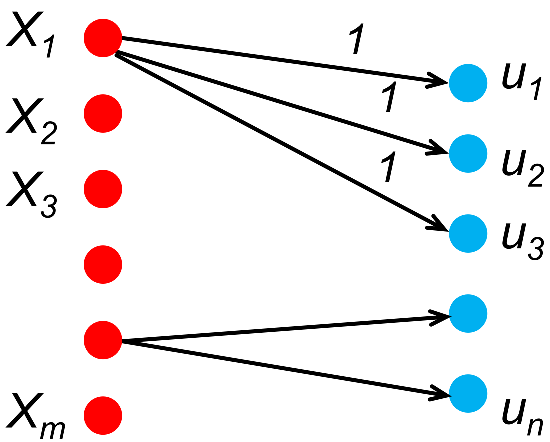
- **Set cover problem** (a known NP-complete problem):
  - Given universe of elements $\boldsymbol{U} = \{\boldsymbol{u_1}, \dots, \boldsymbol{u_n}\}$ and sets $\boldsymbol{X_1}, \dots, \boldsymbol{X_m} \subseteq \boldsymbol{U}$



  - **Q: Are there *k* sets among *X_1*,…, *X_m* such that their union is *U*?**
- **Goal:**
  Encode set cover as an instance of $\max_{S \text{ of size } k} f(S)$

Influence Maximization is NP-hard

- **Given a set cover instance with sets $X_1, \dots, X_m$**
- **Build a bipartite "X-to-U" graph:**



e.g.:
$X_1 = \{u_1, u_2, u_3\}$

Construction:
- Create edge (X_i, u) $\forall X_i$ $\forall u \in X_i$ -- directed edge from sets to their elements
- Put weight 1 on each edge (the activation is deterministic)

- **Set Cover as Influence Maximization in X-to-U graph: There exists a set S of size k with $f(S)=k+n$ iff there exists a size k set cover**

Note: Optimal solution is always a set of nodes X_i (we never influence nodes "u")
This problem is hard in general, but there could be special cases that are easier.

Summary so Far

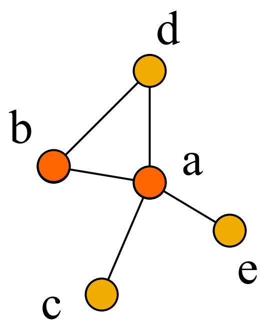
- **Extremely bad news:**
  - **Influence maximization is NP-complete**
- **Next, good news:**
  - **There exists an approximation algorithm!**
    - For some inputs the algorithm won't find globally optimal solution/set ***OPT***
    - But we will also prove that the algorithm will never do too badly either. More precisely, the algorithm will find a set ***S*** that where ***f(S) > 0.63*g(OPT)***, where *OPT* is the globally optimal set.

The Approximation Algorithm

- **Consider a Greedy Hill Climbing algorithm to find *S*:**
  - **Input:**
    Influence set X_u of each node u: $X_u = \{v_1, v_2, \dots\}$
    - That is, if we activate u, nodes $\{v_1, v_2, \dots\}$ will eventually get active
  - **Algorithm:** At each iteration i activate the node u that gives largest marginal gain: $\max_u f(S_{i-1} \cup \{u\})$

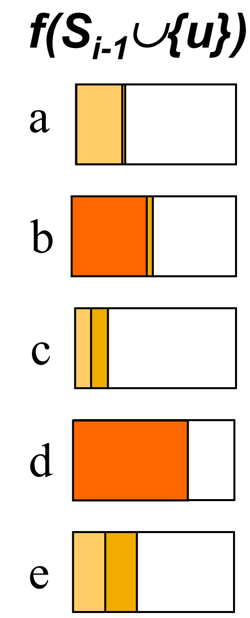
S_i … Initially active set
$f(S_i)$ … Size of the union of X_u, $u \in S_i$

(Greedy) Hill Climbing



Algorithm:

- Start with $\boldsymbol{S_0} = \{\,\}$
- For $\boldsymbol{i = 1 \ldots k}$
  - Activate node $\boldsymbol{u}$ that max $\boldsymbol{f(S_{i-1} \cup \{u\})}$
  - Let $\boldsymbol{S_i = S_{i-1} \cup \{u\}}$



- **Example:**
  - Eval. $f(\{a\}), \ldots, f(\{e\})$, pick argmax of them
  - Eval. $f(\{\boldsymbol{d}, a\}), \ldots, f(\{\boldsymbol{d}, e\})$, pick argmax
  - Eval. $f(\boldsymbol{d}, \boldsymbol{b}, a\}), \ldots, f(\{\boldsymbol{d}, \boldsymbol{b}, e\})$, pick argmax

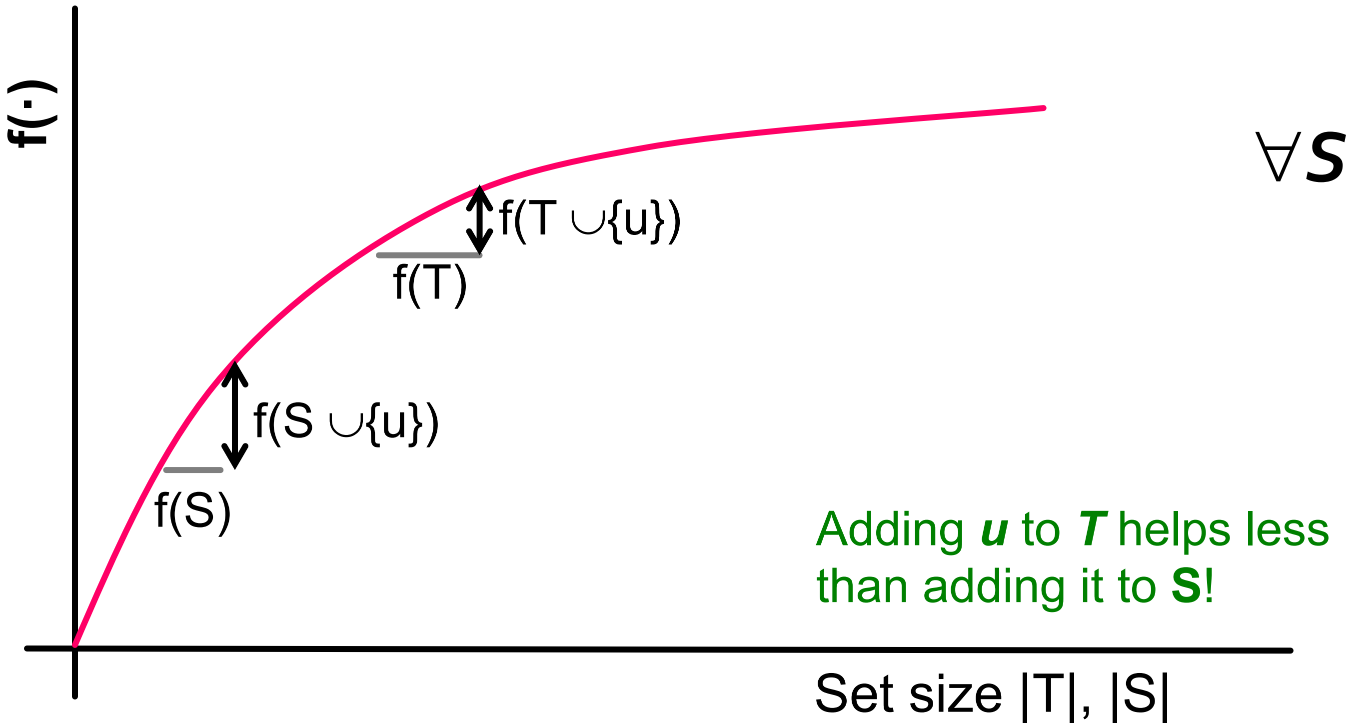
Approximation Guarantee

- **Claim: Hill climbing produces a solution S where:** $f(S) \geq (1-1/e) * f(OPT)$ $\quad (f(S) > 0.63 * f(OPT))$
  [Nemhauser, Fisher, Wolsey '78, Kempe, Kleinberg, Tardos '03]
- **Claim holds for functions $f(\cdot)$ with 2 properties:**
  - ***f* is monotone:** (activating more nodes doesn't hurt)
    if $S \subseteq T$ then $f(S) \leq f(T)$ and $f(\{\}) = 0$
  - ***f* is submodular:** (activating each additional node helps less)
    adding an element to a set gives less improvement than adding it to one of its subsets: $\forall S \subseteq T$

$$\underbrace{f(S \cup \{u\}) - f(S)}_{\text{Gain of adding a node to a small set}} \geq \underbrace{f(T \cup \{u\}) - f(T)}_{\text{Gain of adding a node to a large set}}$$

Submodularity– Diminishing returns

- **Diminishing returns:**



$$\forall S \subseteq T$$

$$\underbrace{f(S \cup \{u\}) - f(S)}_{\text{Gain of adding a node to a small set}} \geq \underbrace{f(T \cup \{u\}) - f(T)}_{\text{Gain of adding a node to a large set}}$$

Plan: Prove 2 things
(1) Our f(S) is submodular
(2) Hill Climbing gives near-optimal solutions
(for monotone submodular functions)

Also see the hangout posted on the course website.

Background: Submodular Functions

- **We must show our *f(·)* is submodular:**
- $\forall S \subseteq T$

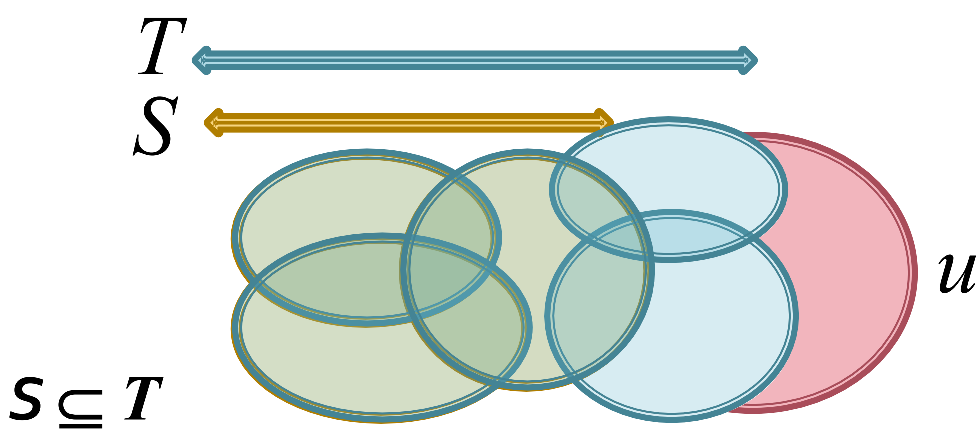
$$\underbrace{f(S \cup \{u\}) - f(S)}_{\text{Gain of adding a node to a small set}} \geq \underbrace{f(T \cup \{u\}) - f(T)}_{\text{Gain of adding a node to a large set}}$$

- **Basic fact 1:**
  - If $f_1(x), \dots, f_k(x)$ are **submodular**, and $c_1, \dots, c_k \geq 0$ then $F(x) = \sum_i c_i \cdot f_i(x)$ is also **submodular**

    (Non-negative combination of submodular functions is a submodular function)

Background: Submodular Functions

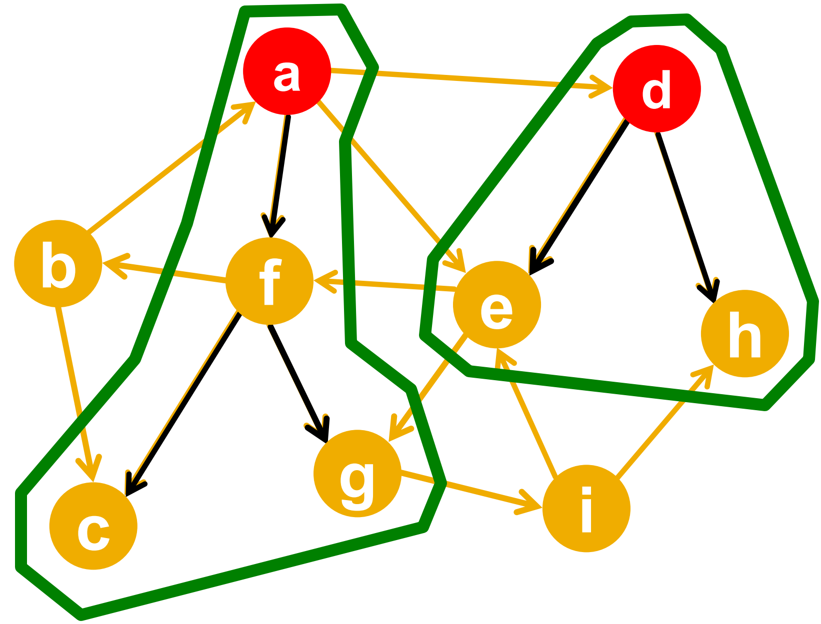
- $\forall \boldsymbol{S} \subseteq \boldsymbol{T}$: $\underbrace{f(S \cup \{u\}) - f(S)}_{\text{Gain of adding } u \text{ to a small set}} \geq \underbrace{f(T \cup \{u\}) - f(T)}_{\text{Gain of adding } u \text{ to a large set}}$
- **Basic fact 2: A simple submodular function**
  - Sets $\boldsymbol{X_1, \ldots, X_m}$
  - $\boldsymbol{f(S) = |\bigcup_{k \in S} X_k|}$ **(size of the union of sets $\boldsymbol{X_k}$, $\boldsymbol{k \in S}$)**
  - **Claim: $\boldsymbol{f(S)}$ is submodular!**



The more sets you already have the less new area a given set ***u*** will cover

Our *f(S)* is Submodular!

$$f(S) = \frac{1}{|I|}\sum_{\substack{\text{Random} \\ \text{realizations } i}} f_i(S)$$

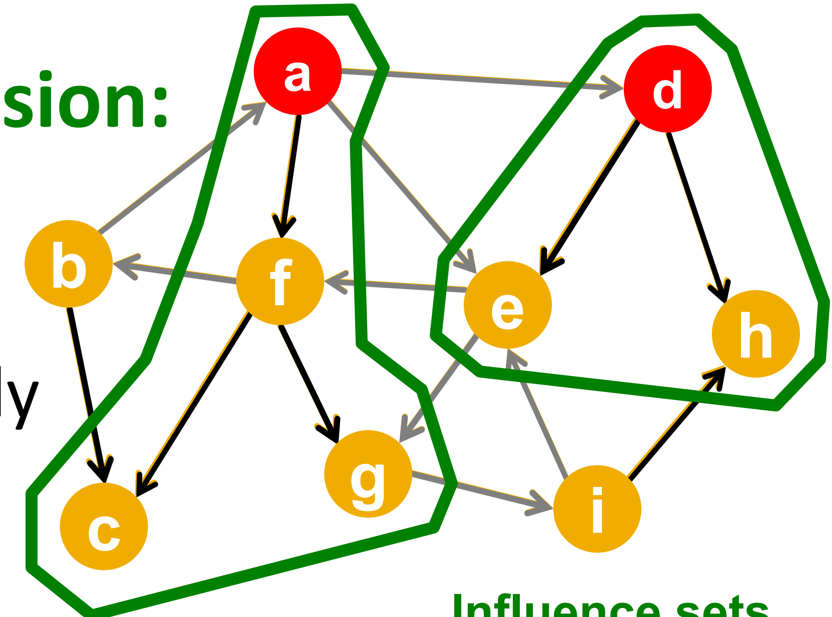


- **Proof strategy:**
  - We will argue that influence maximization is an instance of the **Set cover problem**:
    - ***Set cover problem:*** ***f(S)*** is the size of the union of nodes influenced by active set **S**
      - Note ***f(S)*** is "random" (a result of a random process) so we need to be a bit careful
    - **Principle of deferred decision to the rescue!**
- We will create many parallel universes and then average over them

Our $f(S)$ is Submodular!

$$f(S) = \frac{1}{|I|} \sum_{\substack{\text{Random} \\ \text{realizations } i}} f_i(S)$$

- **Principle of deferred decision:**
  - Flip all the coins at the beginning and record which edges fire successfully
  - **Now we have a deterministic graph!**
  - **Def:** Edge is <u>live</u> if it fired successfully
    - That is, we remove edges that did not fire



Influence sets for realization i:

X_a^i = {a,f,c,g}

X_b^i = {b,c},

X_c^i = {c}

X_d^i = {d,e,h}

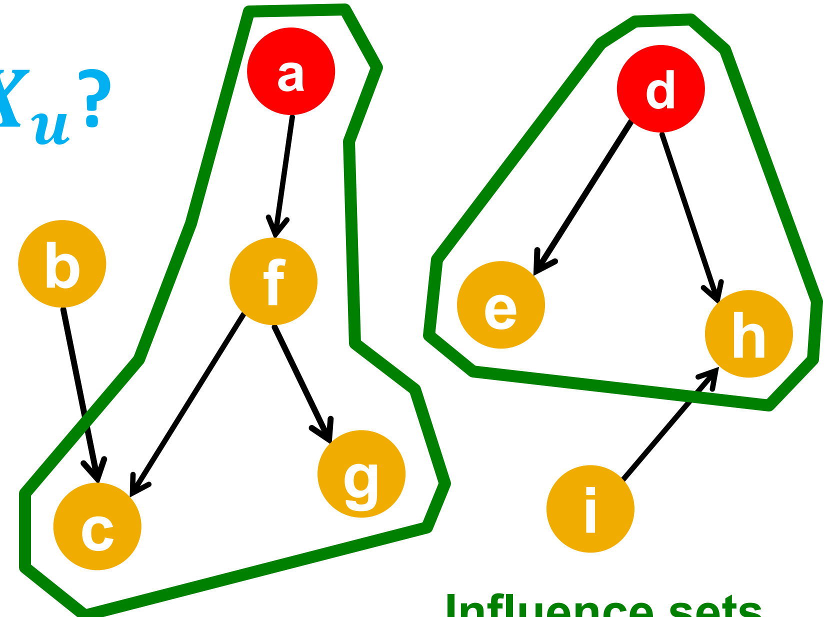
…

- **What is influence set X_u of node u?**
  - **The set reachable by live-edge paths from u**

Our $f(S)$ is Submodular!

$$f(S) = \frac{1}{|I|} \sum_{\substack{\text{Random} \\ \text{realizations } i}} f_i(S)$$

- What is an influence set X_u?
  - The set reachable by live-edge paths from u
- What is now $f(S)$?
  - $f_i(S)$ = size of the set reachable by live-edge paths from nodes in S



Influence sets for realization i:

X_a^i = {a,f,c,g}

X_b^i = {b,c},

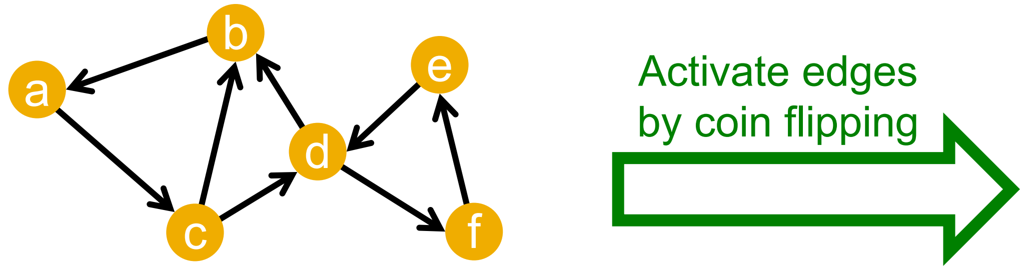
X_c^i = {c}

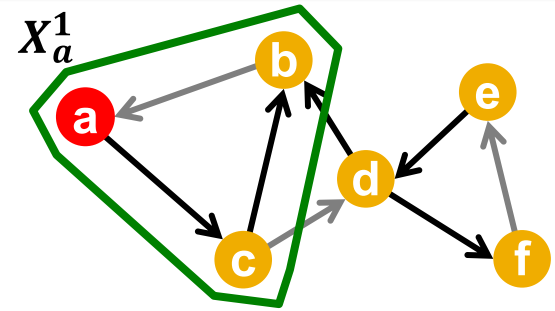
X_d^i = {d,e,h}

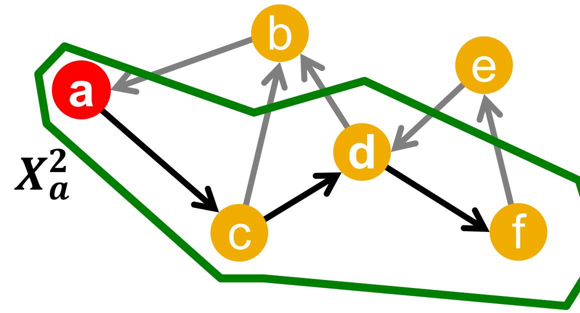
- For the i-th realization of coin flips
  - $f_i(S = \{a, b\}) = |\{a, f, c, g\} \cup \{b, c\}| = 5$
  - $f_i(S = \{a, d\}) = |\{a, f, c, g\} \cup \{d, e, h\}| = 7$

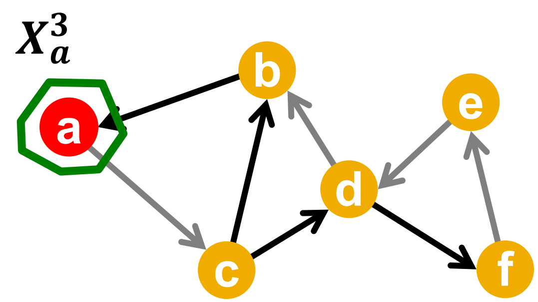
Our $f(S)$ is Submodular!

$$f(S) = \frac{1}{|I|} \sum_{\substack{\text{Random} \\ \text{realizations } i}} f_i(S)$$









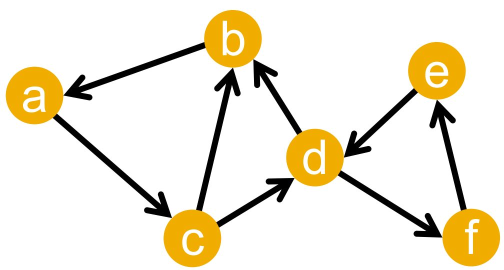
- **Fix outcome $i \in I$ of coin flips**
- X_v^i = set of nodes reachable from v on live-edge paths
- $f_i(S)$ = size of cascades from S given the coin flips i
- $f_i(S) = \left|\bigcup_{v \in S} X_v^i\right| \Rightarrow f_i(S)$ is **submodular!**
  - X_v^i are sets, $f_i(S)$ is the size of their union
- **Expected influence set size:**

$$f(S) = \frac{1}{|I|} \sum_{i \in I} f_i(S) \Rightarrow f(S)$$ is **submodular!**

  - $f(S)$ is a linear combination of submodular functions

RECAP: Influence Maximization

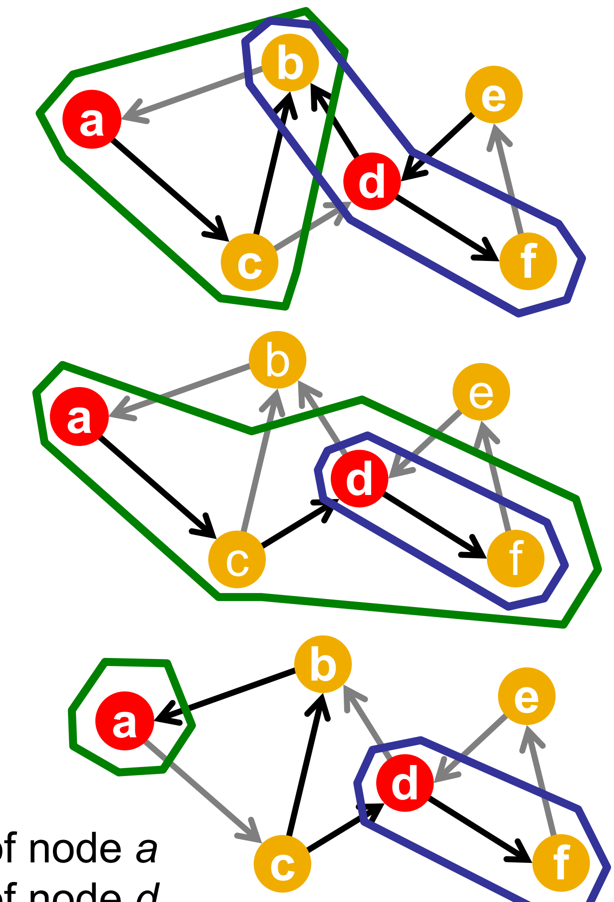
- **Find most influential set *S* of size *k*:** largest expected cascade size ***f(S)*** if set ***S*** is activated



Network, each edge activates with prob. p_{uv}

Activate edges by coin flipping

Multiple realizations ***i***. Each realization is a "parallel universe"



- **Want to solve:**

$$\arg\max_{|S|=k} f(S) = \frac{1}{|I|}\sum_{i\in I} f_i(S)$$

Consider ***S={a,d}*** then:
$f_1(S)=5$, $f_2(S)=4$, $f_3(S)=3$
and **$f(S) = 12$**

… influence set of node *a*
… influence set of node *d*

Plan: Prove 2 things

(1) Our f(S) is submodular

<u>(2) Hill Climbing gives near-optimal solutions</u>

(for monotone submodular functions)

Proof for Hill Climbing

Claim:

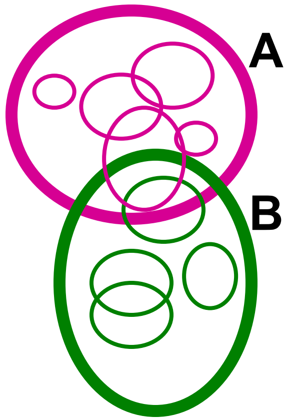
When f(S) is monotone and submodular then Hill climbing produces active set S where: $f(S) \geq \left(1 - \frac{1}{e}\right) \cdot f(OPT)$

- In other words: $f(S) \geq 0.63 \cdot f(OPT)$

- **The setting:**
  - Keep adding nodes that give the largest gain
  - Start with $S_0 = \{\}$, produce sets $S_1, S_2, \dots, S_k$
  - **Add elements one by one**
  - Let $OPT = \{t_1 \dots t_k\}$ be **the optimal set (OPT)** of size k
- **We need to show:** $f(S) \geq (1 - \frac{1}{e})\, f(OPT)$

Proof Overview

- **Define:** **Marginal gain:** $\delta_i = f(S_i) - f(S_{i-1})$
- **Proof: 3 steps:**
  - **0)** Lemma: $f(A \cup B) - f(A) \le \sum_{j=1}^{k} [f(A \cup \{b_j\}) - f(A)]$
    - **where:** $B = \{b_1, \dots, b_k\}$ and $f(\cdot)$ is submodular
  - **1)** $\boldsymbol{\delta_{i+1} \ge \frac{1}{k}[f(OPT) - f(S_i)]}$
  - **2)** $\boldsymbol{f(S_{i+1}) \ge \left(1 - \frac{1}{k}\right) f(S_i) + \frac{1}{k} f(OPT)}$
  - **3)** $\boldsymbol{f(S_k) \ge \left(1 - \frac{1}{e}\right) f(OPT)}$



Step zero: Basic Hill Climbing Fact

- $f(A \cup B) - f(A) \leq \sum_{j=1}^{k}[f(A \cup \{b_j\}) - f(A)]$
  - **where:** $B = \{b_1, \dots, b_k\}$ and $f(\cdot)$ is submodular
- **Proof:**
  - Let $\boldsymbol{B_i} = \{\boldsymbol{b_1}, \dots \boldsymbol{b_i}\}$, so we have $\boldsymbol{B_1}, \boldsymbol{B_2}, \dots, \boldsymbol{B_k} (= \boldsymbol{B})$
  - $f(A \cup \mathrm{B}) - f(A) = \sum_{i=1}^{k}[f(A \cup B_i) - f(A \cup B_{i-1})]$
  - $= \sum_{i=1}^{k}[f(A \cup B_{i-1} \cup \{b_i\}) - f(A \cup B_{i-1})]$
  - $\leq \sum_{i=1}^{k}[f(A \cup \{b_i\}) - f(A)]$

Work out the sum. Everything but 1st and last term cancel out:

$$\cancel{f(A \cup B_1)} - f(A \cup B_0)$$
$$+ \cancel{f(A \cup B_2)} - \cancel{f(A \cup B_1)}$$
$$+ \cancel{f(A \cup B_3)} - \cancel{f(A \cup B_2)} \dots$$
$$+ f(A \cup B_k) - \cancel{f(A \cup B_{k-1})}$$

By submodularity since $A \cup X \supseteq A$

Step one: What is δ_i gain at step *i*?

Remember: $\boldsymbol{\delta_i = f(S_i) - f(S_{i-1})}$

- $f(OPT) \le f(S_i \cup OPT)$ (by monotonicity)
- $= f(S_i \cup OPT) - f(S_i) + f(S_i)$
- $\le \sum_{j=1}^{k}\left[f\left(S_i \cup \{t_j\}\right) - f(S_i)\right] + f(S_i)$ (by prev. slide)
- $\le \sum_{j=1}^{k}[\delta_{i+1}] + f(S_i)$
- $= f(S_i) + k\,\delta_{i+1}$
- **Thus:** $f(OPT) \le f(S_i) + k\,\delta_{i+1}$
- $\Rightarrow \boldsymbol{\delta_{i+1} \ge \frac{1}{k}[f(OPT) - f(S_i)]}$

$OPT = \{ t_1, \ldots t_k \}$
t_j is j-th element of the optimal solution.

Rather than choosing t_j let's greedily choose **the best element q_i**, which gives a gain of δ_{i+1}.
So, $\boldsymbol{f\left(S_i \cup \{t_j\}\right) \le \delta_{i+1}}$.
This is the "hill-climbing" assumption.

Step two: What is f(S_{i+1})?

- **We just showed:** $\delta_{i+1} \geq \frac{1}{k}[f(OPT) - f(S_i)]$
- **What is $f(S_{i+1})$?**
  - $f(S_{i+1}) = f(S_i) + \delta_{i+1}$
  - $\geq f(S_i) + \frac{1}{k}[f(OPT) - f(S_i)]$
  - $\geq \left(1 - \frac{1}{k}\right) f(S_i) + \frac{1}{k} f(OPT)$
- **What is $f(S_k)$?**

Step three: What is $f(S_k)$?

- **Claim:** $f(S_i) \geq \left[1 - \left(1 - \frac{1}{k}\right)^i\right] f(OPT)$

Proof by induction:

- $\boldsymbol{i = 0}$:
  - $f(S_0) = f(\{\}) = 0$
  - $\left[1 - \left(1 - \frac{1}{k}\right)^0\right] f(OPT) = 0$

Step three: What is $f(S_k)$?

- **Given that this is true for S_i:** $f(S_i) \geq \left[1-\left(1-\frac{1}{k}\right)^i\right] f(OPT)$

Proof by induction:

- **At $i+1$:**
  - $f(S_{i+1}) \geq \left(1-\frac{1}{k}\right) f(S_i) + \frac{1}{k} f(OPT)$
  - $\geq \left(1-\frac{1}{k}\right) \underbrace{\left[1-\left(1-\frac{1}{k}\right)^i\right] f(OPT)}_{\text{the claim}} + \frac{1}{k} f(OPT)$
  - $= \left[1-\left(1-\frac{1}{k}\right)^{i+1}\right] f(OPT)$

Two slides ago we showed:

$$f(S_{i+1}) \geq \left(1-\frac{1}{k}\right) f(S_i) + \frac{1}{k} f(OPT)$$

What is f(S$_k$)?

- **Thus:**

$$f(S) = f(S_k) \geq \left[1 - \underbrace{\left(1 - \frac{1}{k}\right)^k}_{\leq \frac{1}{e}}\right] f(OPT)$$

- **So:**

$$f(S_k) \geq \left(1 - \frac{1}{e}\right) f(OPT)$$

qed.

Apply inequality: $1 + x \leq e^x$ where $x = -\frac{1}{k}$

Solution Quality

We just proved:

- Hill climbing finds solution S which
  $\boldsymbol{f(S) \geq (1-1/e)*f(OPT)}$ i.e., $f(S) \geq 0.63*f(OPT)$

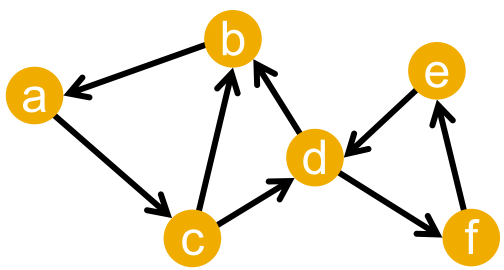
- This is a **data <u>independent</u> bound**
  - This is a worst case bound
  - No matter what is the input data,
    we know that the Hill-Climbing **will never do worse** than $0.63*f(OPT)$

Evaluating our $f(S)$?

- **How to evaluate influence maximization $f(S)$?**
  - Still an open question of how to compute it efficiently
- **But: Very good estimates by simulation**
  - Repeating the diffusion process often enough (polynomial in ***n***; 1/ε)
  - Achieve ***(1± ε)*-approximation** to ***f(S)***
  - Generalization of Nemhauser-Wolsey proof: Greedy algorithm is now a ***(1-1/e- ε)*-approximation**

RECAP: Influence Maximization

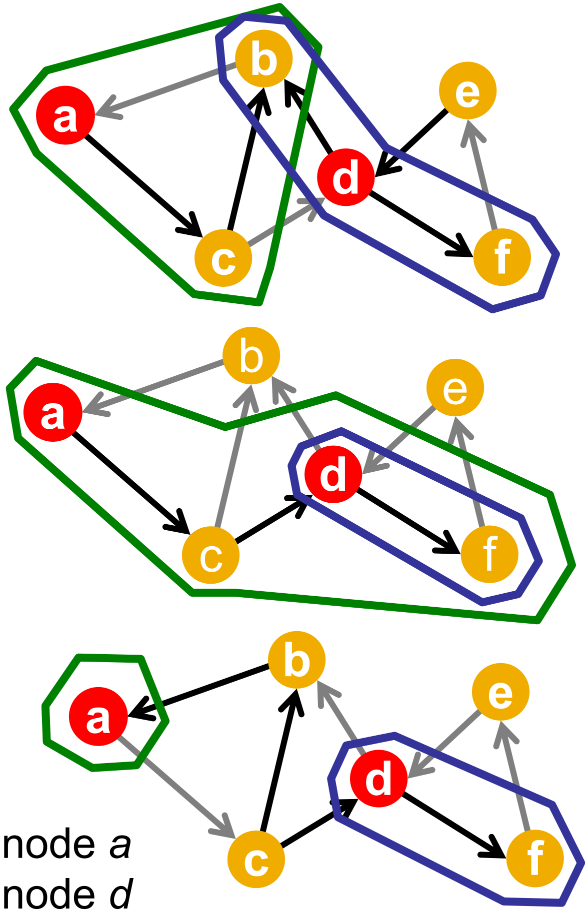
- **Find most influential set *S* of size *k*:** largest expected cascade size ***f(S)*** if set ***S*** is activated



Network, each edge activates with prob. p_{uv}

Activate edges by coin flipping

Multiple realizations ***i***. Each realization is a "parallel universe"



- **Want to solve:**

$$\arg\max_{|S|=k} f(S) = \frac{1}{|I|}\sum_{i\in I} f_i(S)$$

Consider ***S={a,d}*** then:
$f_1(S)=5$, $f_2(S)=4$, $f_3(S)=3$
and **$f(S) = 1/3*(5+4+3)=4$**

… influence set of node *a*
… influence set of node *d*

Experiments and Concluding Thoughts

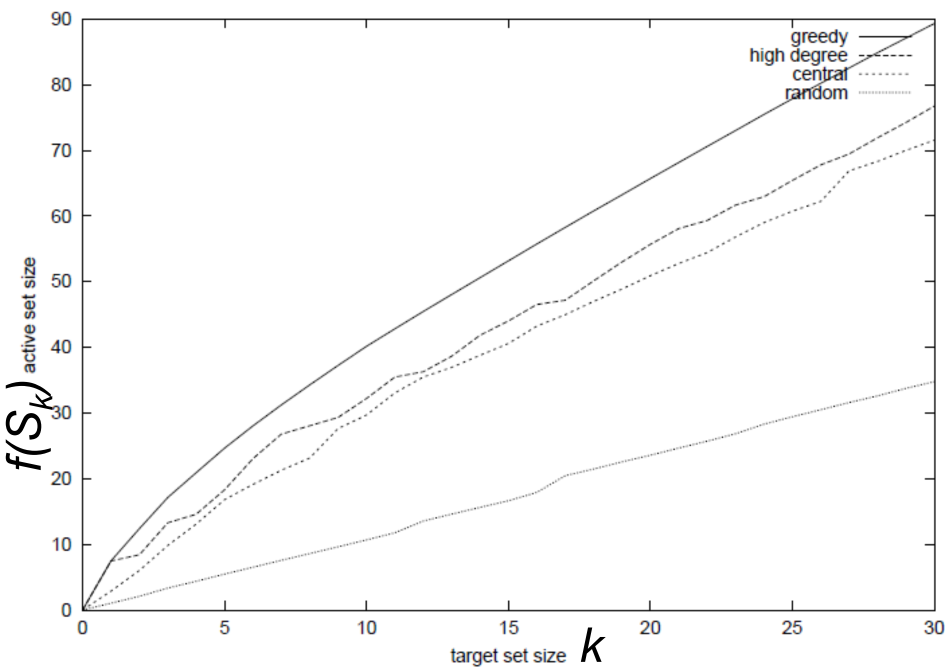
Experiment Data

- **A collaboration network**: co-authorships in papers of the arXiv high-energy physics theory:
  - 10,748 nodes, 53,000 edges
  - **Example cascade process:** Spread of new scientific terminology/method or new research area
- **Independent Cascade Model:**
  - **Case 1:** Uniform probability ***p*** on each edge
  - **Case 2:** Edge from **v** to **w** has probability **1/deg(w)** of activating **w**.

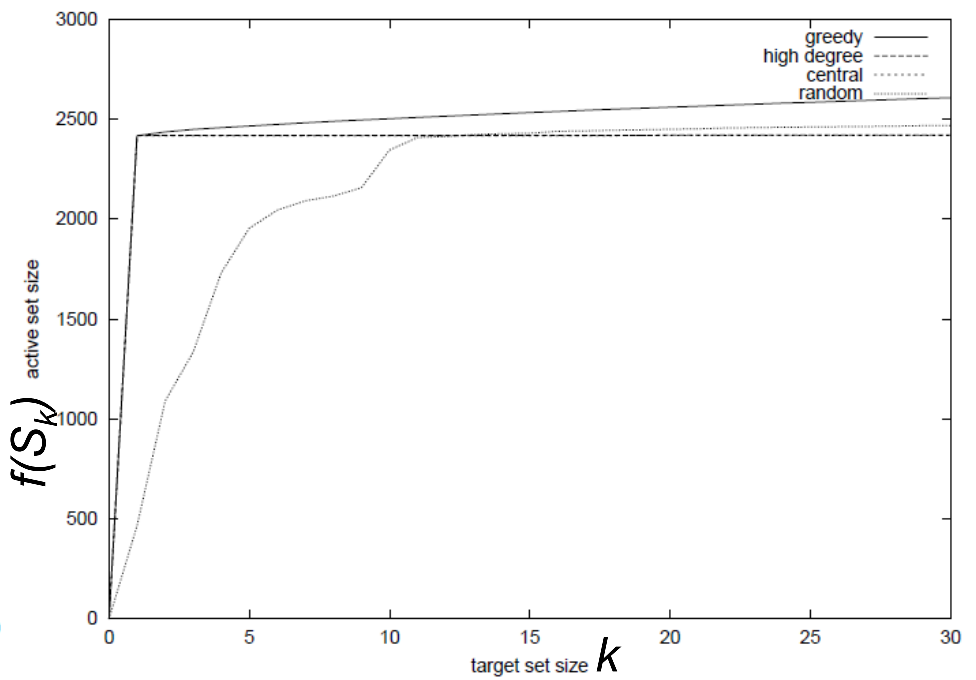
Experiment Settings

- **Simulate the process 10,000 times for each targeted set**
  - Every time re-choosing edge outcomes randomly
- **Compare with other 3 common heuristics**
  - **Degree centrality:** Pick nodes with highest degree
  - **Closeness centrality:** Pick nodes in the "center" of the network
  - **Random nodes:** Pick a random set of nodes

Independent Cascade Model



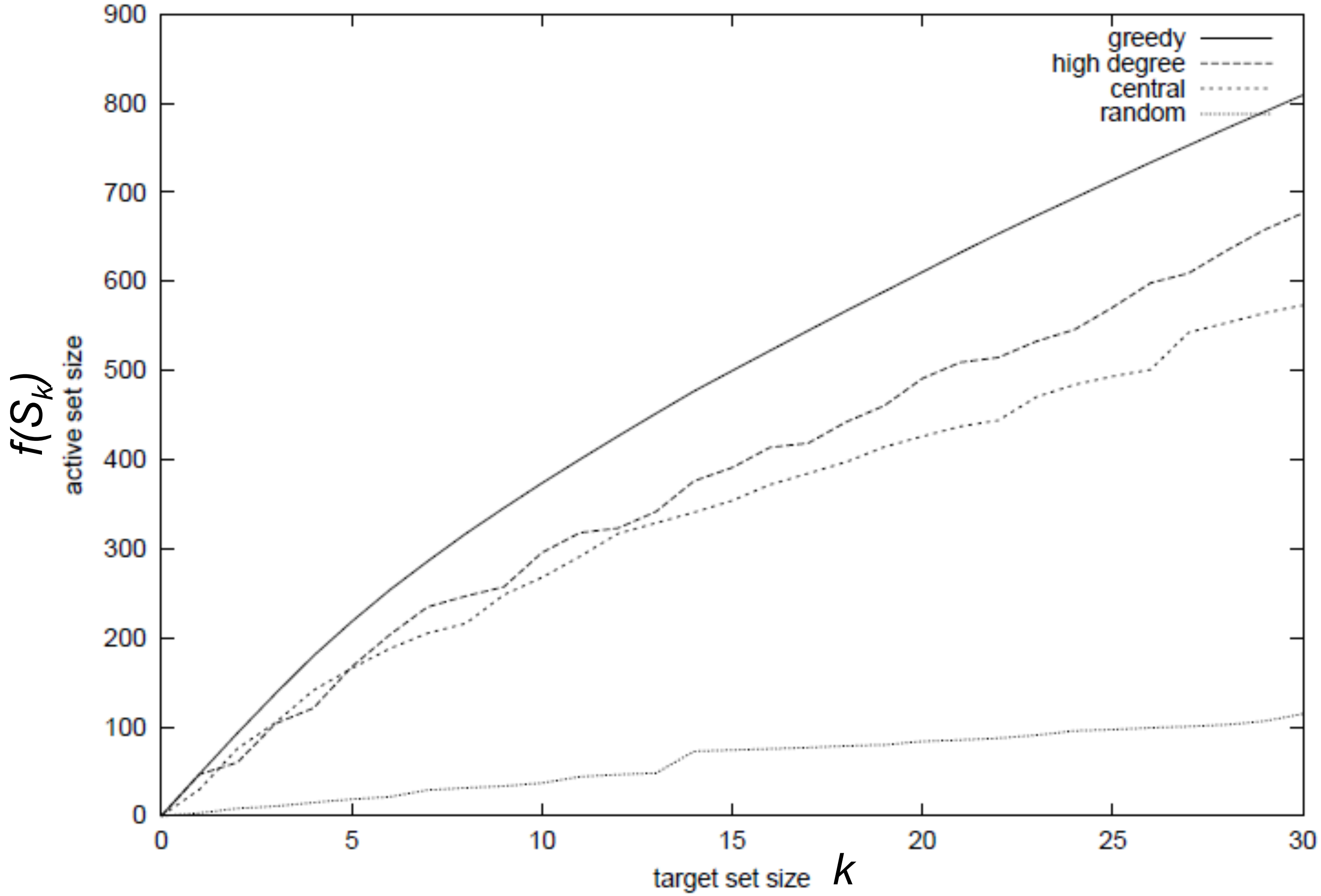
$p_{uv} = 0.01$



$p_{uv} = 0.10$

Uniform edge firing probability p_{uv}

Independent Cascade Model



$p_{uv}=1/\deg(v)$

Non-uniform edge firing probability p_{uv}

Discussion

- **Notice:** Greedy approach is slow!
  - For a given network ***G***, repeat 10,000s of times:
    - Flip coin for each edge and determine influence sets under coin-flip realization *i*
    - Each node *u* is associated with 10,000s influence sets X_u^i
  - **Greedy's complexity is $\boldsymbol{O(k \cdot n \cdot R \cdot M)}$**
    - n … number of nodes in *G*
    - k … number of nodes to be selected/influenced
    - R … number of simulation rounds
    - m … number of edges in *G*

Cottage Industry of Heuristics

- Many researchers have since proposed heuristics that work well in practice and run faster than the greedy algorithm

[Chen, Wang, Yang]

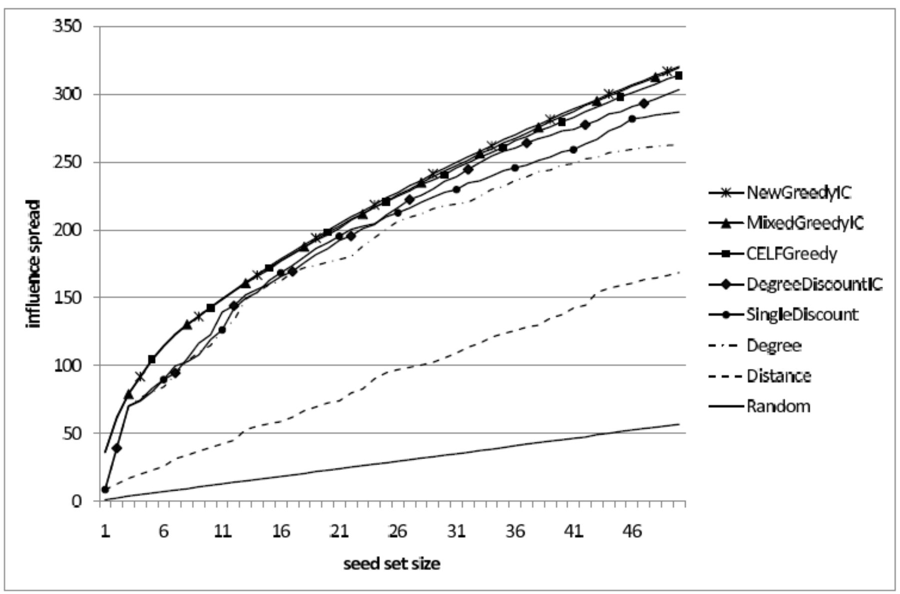


Figure 2: Influence spreads of different algorithms on the collaboration graph NetPHY under the independent cascade model ($n = 37,154$, $m = 231,584$, and $p = 0.01$).

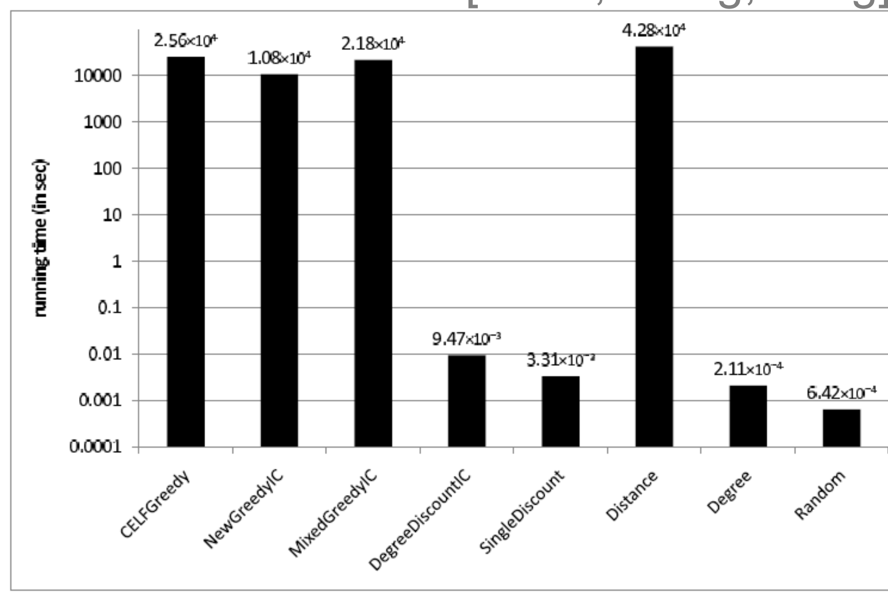


Figure 4: Running times of different algorithms on the collaboration graph NetPHY under the independent cascade model ($n = 37,154$, $m = 231,584$, $p = 0.01$, and $k = 50$).

Open Questions

- **More realistic viral marketing:**
  - Different marketing actions increase **likelihood** of initial activation, for **several** nodes at once
- **Study more general influence models:**
  - Find trade-offs between generality and feasibility
- **Deal with negative influences:**
  - Model competing ideas
- Obtain more data (better models) about how activations occur in real social networks