

Tutorial at the World Wide Web conference
Hyderabad, India, 2011

<http://snap.stanford.edu/proj/socmedia-www>

Analytics & Predictive Models for Social Media: Part 1: Information flow

Jure Leskovec
Stanford University



About the tutorial: Social Media

- **Goal:** Introduce methods and algorithms for Social Media Analytics
- **Tutorial has two parts:**
 - **Part 1: Information Flow**
 - How do we capture and model the flow of information through networks to:
 - Predict information attention/popularity
 - Detect information big stories before they happen
 - **Part 2: Rich Interactions**
 - How do we go beyond “link”/“no-link”:
 - Predicting future links and their strengths
 - Separating friends from foes



WIKIPEDIA



Social Media: Social Content

- Media designed to be disseminated through Social interaction
- Web is no longer a static library that people passively browse
- Web is a place where people:
 - Consume and create content
 - Interact with other people



Social Media: Examples

- Places where people consume, produce and share content:
 - Internet forums
 - Blogs
 - Social networks
 - Microblogging: Twitter
 - Wikis
 - Podcasts, Slide sharing, Bookmark sharing, Product reviews, Comments, ...
- Facebook traffic tops Google (for USA)
 - March 2010: FB > 7% of US traffic
http://money.cnn.com/2010/03/16/technology/facebook_most_visited



WIKIPEDIA



Social Media: Opportunities

- Any user can share and contribute content, express opinions, link to others
- This means: Web (Social Media) captures the pulse of humanity!
 - Can directly study opinions and behaviors of millions of users to gain insights into:
 - Human behavior
 - Marketing analytics, product sentiment



WIKIPEDIA



Social Media: Challenges

- Traditionally:
 - Web is static library
 - Search engines crawl and index the Web
 - Users issue queries to find what they want
- Today:
 - On-line information reaches us in small increments from real-time sources and through social networks
- How should this change our understanding of information, and of the role of networks?



Social Media: Challenges

- Web as universal library vs. Web as current awareness medium
 - Real-time information flow in social networks
 - Real-time search:
 - “Tell me about X” vs. “Tell me what’s hot now”
 - Predictive models of human interactions
 - We need finer resolution than presence/absence of a link



WIKIPEDIA



Social Media: Rich & Big Data

- Rich and big data:
 - Billions users, billions contents
 - Textual, Multimedia (image, videos, etc.)
 - Billions of connections
 - Behaviors, preferences, trends...
- Data is open and easy to access
 - It's easy to get data from Social Media
 - Datasets
 - Developers APIs
 - Spidering the Web

For the list of datasets see tutorial website:
<http://snap.stanford.edu/proj/socmedia-www>
and also: <http://snap.stanford.edu/data>

Social Media Datasets

- **Social Tagging:**
 - CiteULike, Bibsonomy, MovieLens, Delicious, Flickr, Last.FM...
 - <http://kmi.tugraz.at/staff/markus/datasets/>
- **Yahoo! Firehose**
 - 750K ratings/day, 8K reviews/day, 150K comments/day, status updates, Flickr, Delicious...
 - http://developer.yahoo.net/blog/archives/2010/04/yahoo_updates_firehose.html
- **MySpace data** (real-time data, multimedia content, ...)
 - <http://blog.infochimps.org/2010/03/12/announcing-bulk-redistribution-ofmyspace-data/>
- **Spinn3r Blog Dataset, JDPA Sentiment Corpus**
 - <http://www.icwsm.org/data/>

Tutorial Outline

- Part 1: Information flow in networks
 - 1.1: Data collection: How to track the flow?
 - 1.2: Correcting for missing data
 - 1.3: Modeling and predicting the flow
 - 1.4: Infer networks of information flow
- Part 2: Rich interactions

Part 1 of the Tutorial: Overview

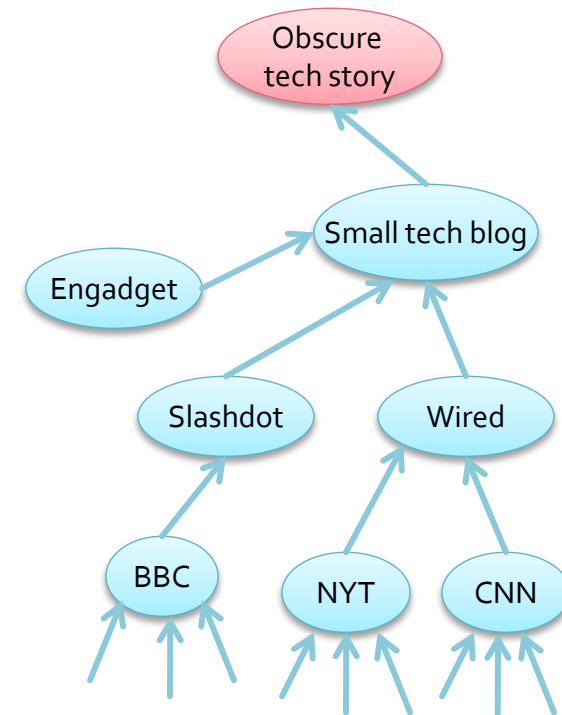
- Information flow through Social Media
 - Analyzing underlying mechanisms for the real-time spread of information through on-line networks
- Motivating questions:
 - How do messages spread through social networks?
 - How to predict the spread of information?
 - How to identify networks over which the messages spread?

Social Media Data: Spinn3r

- Spinn3r Dataset: <http://spinn3r.com>
 - 30 million articles/day (50GB of data)
 - 20,000 news sources + millions blogs and forums
 - And some Tweets and public Facebook posts
- What are basic “units” of information?
 - Pieces of information that propagate between the nodes (users, media sites, ...)
 - phrases, quotes, messages, links, tags

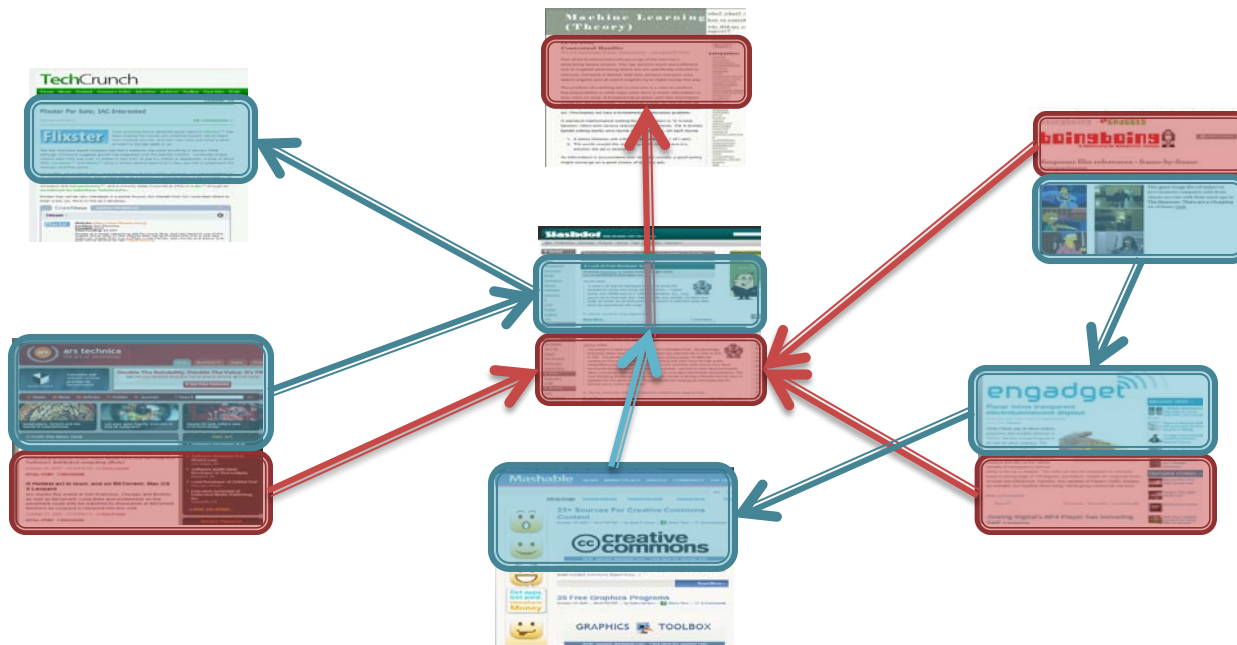
Tracing Information Flow

- Would like to track **units** of information that:
 - correspond to pieces of information:
 - events, articles, ...
 - vary over the order of days,
 - and can be handled at large scale
- Ideas:
 - (1) Cascading links to articles
 - Textual fragments that travel relatively unchanged:
 - (2) URLs and hashtags on Twitter
 - (3) Phrases inside quotes: “...”

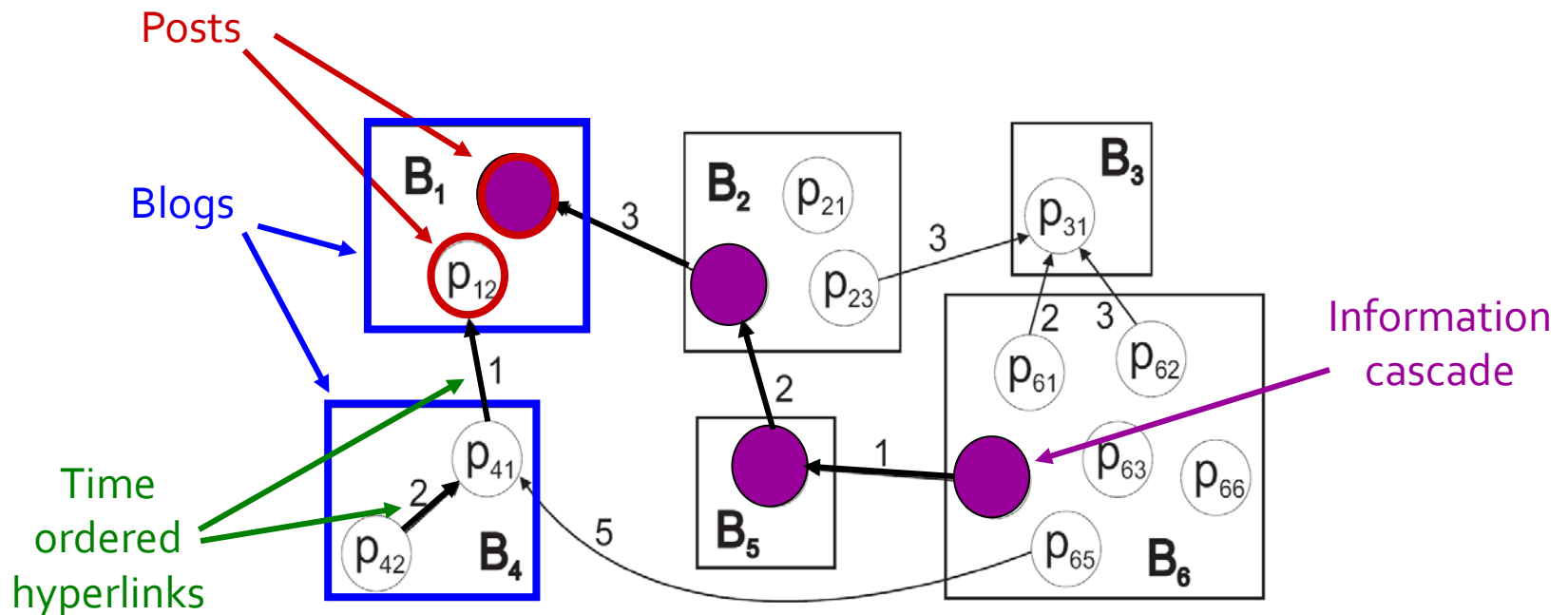


Tracing Information (1): Hyperlinks

- Bloggers write posts and refer (**link**) to other posts and the **information propagates**



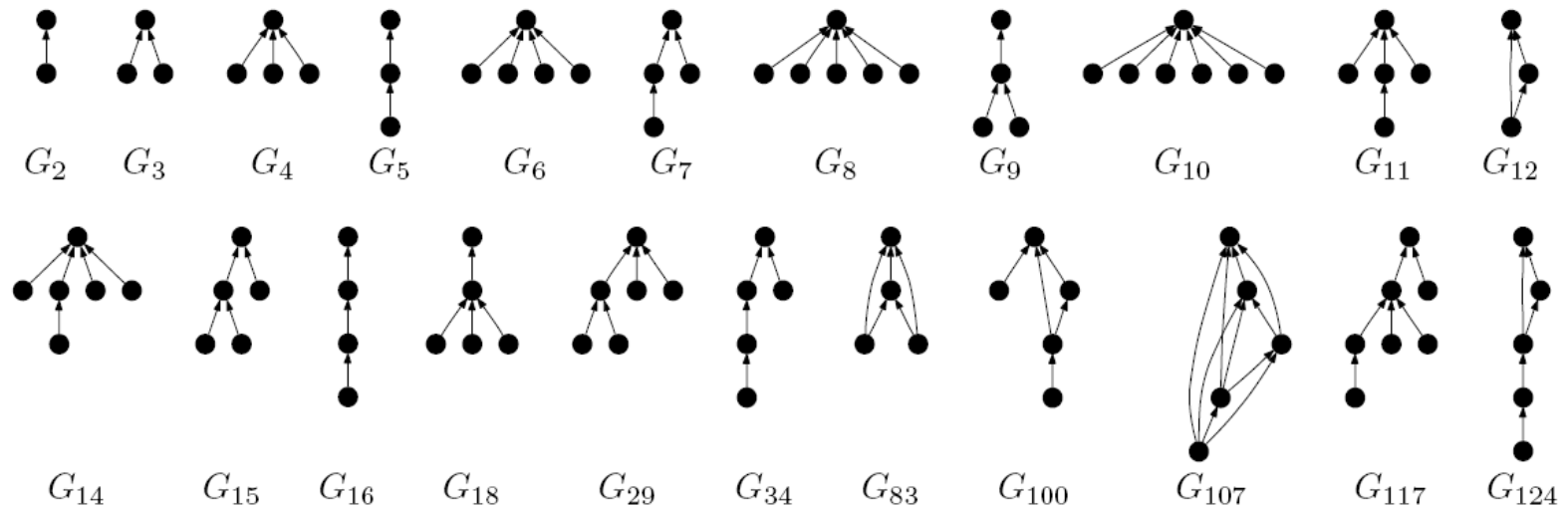
Cascading hyperlinks



- Identify **cascades – graphs** induced by a time ordered propagation of information

Cascade Shapes

- Cascade shapes (ordered by decreasing frequency)
 - 10 million posts and 350,000 cascades



- Cascades are mainly stars (trees)
- Interesting relation between the cascade frequency and structure

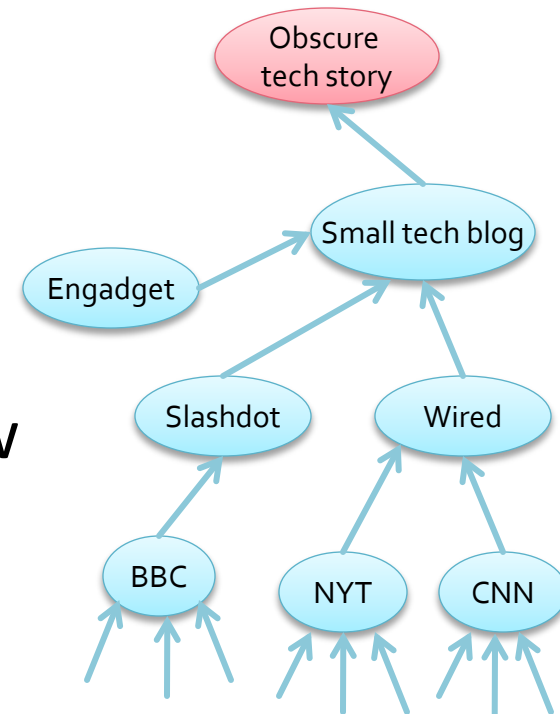
Tracing hyperlinks: Pros/Cons

■ Advantages:

- Unambiguous, precise and explicit way to trace information flow
- We obtain both the times as well as the trace (graph) of information flow

■ Caveats:

- Not all links transmit information:
 - Navigational links, templates, ads
- Many links are missing:
 - Mainstream media sites do not create links
 - Bloggers “forget” to link the source



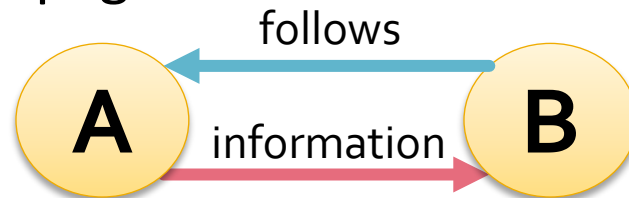
Tracing Information (2): Twitter

- Twitter information network:
 - Each user generates a stream of tweets
 - Users then subscribe to “follow” the streams of others
- 3 ways to track information flow in Twitter:
 - (1) Trace the spread a “hashtag” over the network
 - (2) Trace the spread of a particular URL
 - (3) Re-tweets

Tracing information on Twitter (1)

■ (1) Tracing hashtags:


- Users annotate tweets with short tags
- Tags naturally emerge from the community
- Given the Twitter network and time stamped posts
 - If user A used hashtag #egypt at t_1 and user B follows A and B first used the same hashtag at some later time this means A propagated information to B



Realtime results for Mubarak

 unpais Cientos de miles de egipcios piden la dimisión de Mubarak | Un Pais <http://t.co/prnRxnY>
less than 20 seconds ago via Tweet Button

 ibrahimhabib @AJEnglish Mubark fortune US70 bn <http://www.guardian.co.uk/world/2011/feb/04/hosni-mubarak-family-fortune>
less than 20 seconds ago via webfrom Hackensack, NJ

 Reika_25 RT @fluutokies #Obama [polite mode off]: #Mubarak is an old, extremely stubborn mad man, who needs a psychiatrist to be convinced to leave. #jan25 #egypt
less than 20 seconds ago via web

 itmustbecamel RT @carmelva: RT @SarahZaaimi: a twitter user: Are they any anti-mubarak apps available for the iphone? #Egypt #jan25
less than 20 seconds ago via TweetDeck

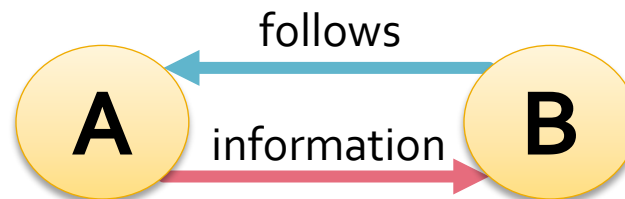
Tracing information on Twitter (2)

■ (2) Tracing URLs:

- Many tweets contain shortened (hashed) URLs
 - Short-URLs are “personalized”
 - If two users shorten the same URL it will shorten to different strings
- Given the Twitter network and time stamped posts
 - If user A used URL_1 at t_1 and B follows A and B used the same URL later then A propagated information to B

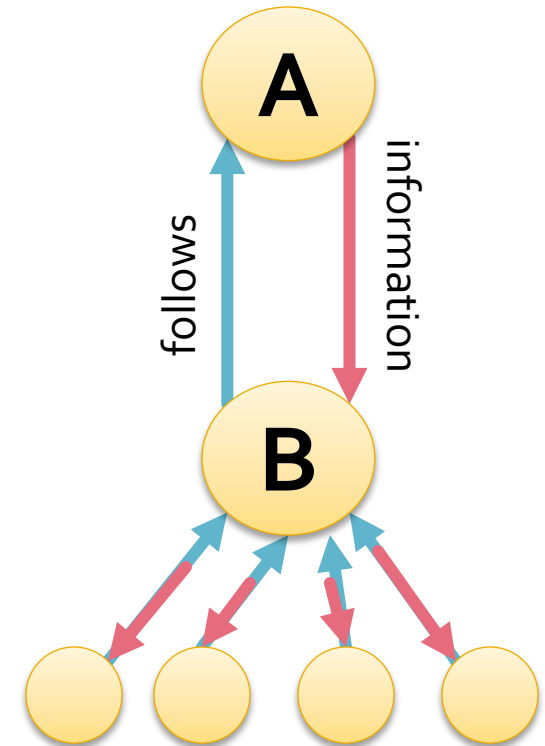
Realtime results for Mubarak

The screenshot shows four tweets from a Twitter search for 'Mubarak'. The first tweet is from 'unpais' with a shortened URL 'http://t.co/pmRxnY' highlighted in a yellow box. The second tweet is from 'ibrahimhabib @AJEnglish' with a full URL 'http://www.guardian.co.uk/world/2011/feb/04/hosni-mubarak-family-fortune'. The third tweet is a retweet from 'Refka_25' with the text '#Mubarak is an old, extremely stubborn mad man, who needs a psychiatrist to be convinced to leave. #jan25 #egypt'. The fourth tweet is a retweet from 'itmustbecamel' with the text 'Are they any anti-mubarak apps available for the iphone? #Egypt #jan25'.



Tracing information on Twitter (3)

- (3) Re-tweets:
 - Explicit information diffusion mechanism on Twitter
 - B sees A's tweet and "forwards" it to its follower by re-tweeting
 - By following re-tweet cascades we establish the information flow



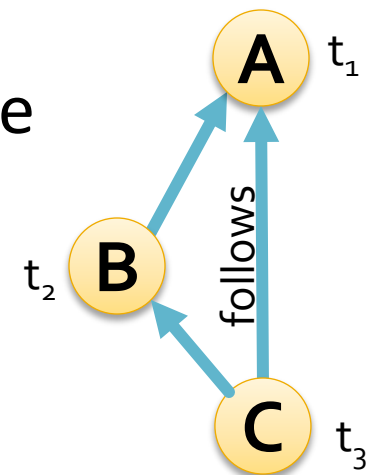
Tracing Information (2): Twitter

■ Advantages:

- Large network, relatively easy to collect data
 - **Caveat:** If data is incomplete cascades break into pieces!
- Many different diffusion mechanisms

■ Caveats:

- Not clear whether hashtags really diffuse
- Due to “personalization” easier to argue URLs diffuse
- Problem with all is that we do not know the “influencer”



Who “influenced” C?

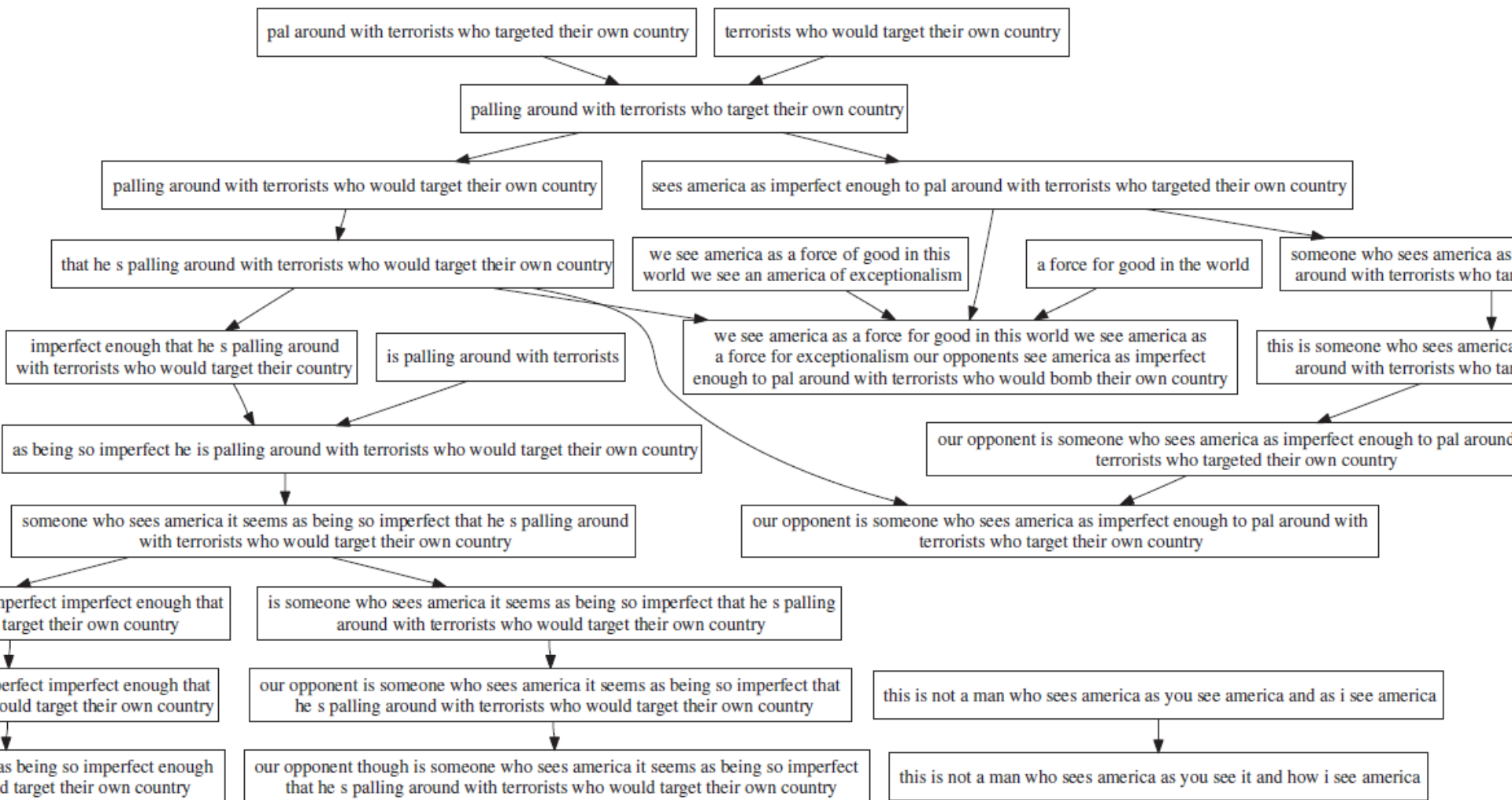
Side note: Twitter data

- Twitter network:
 - Kwak et al.: <http://an.kaist.ac.kr/traces/WWW2010.html>
- Tweets:
 - 500 million tweets over 7 months
 - Go to <http://snap.stanford.edu/data/twitter7.html> and “view source” and you will find the links to the data commented out 😊

Tracing Information (3): Memes

- Meme: A unit of cultural inheritance
- Extract textual fragments that travel relatively unchanged, through many articles:
 - Look for phrases inside quotes: “...”
 - About 1.25 quotes per document in Spinn3r data
 - Why it works?
 - Quotes...
 - are integral parts of journalistic practices
 - tend to follow iterations of a story as it evolves
 - are attributed to individuals and have time and location

Challenge: Quotes Mutate

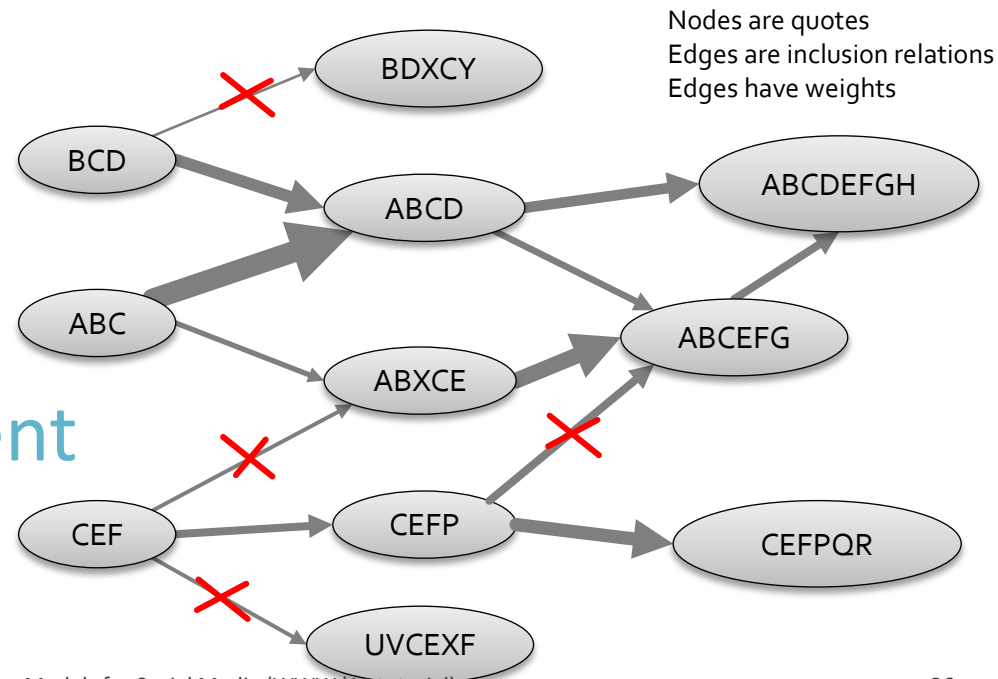


Quote: Our opponent is someone who sees America, it seems, as being so imperfect, imperfect enough that he's palling around with terrorists who would target their own country.

Finding Mutational Variants

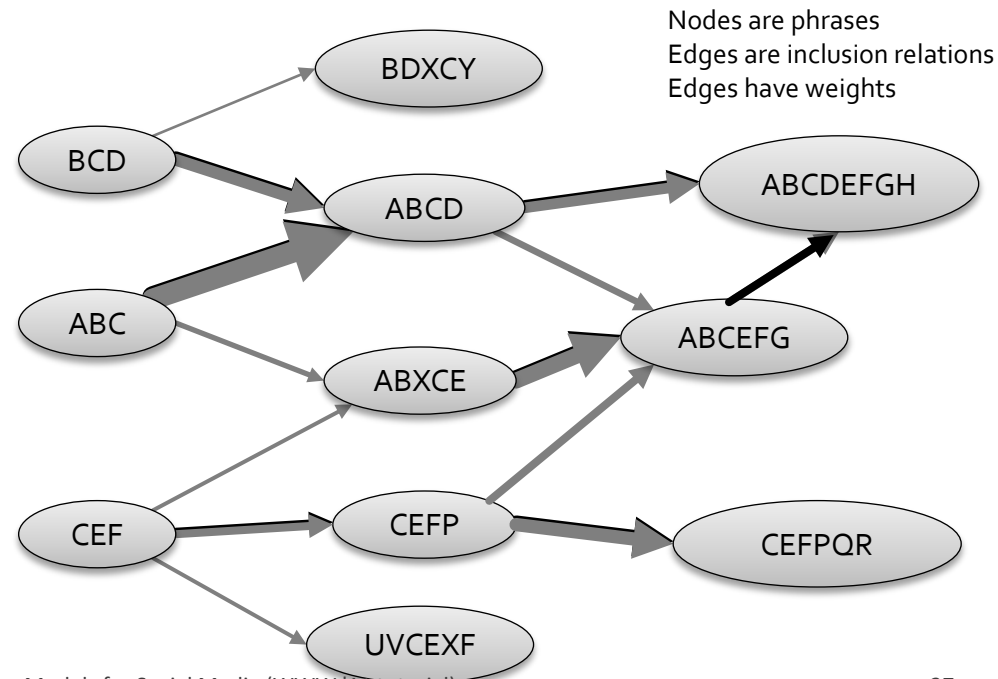
- **Goal:** Find mutational variants of a quote
- Form approximate quote inclusion graph
 - Shorter quote is approximate substring of a longer one

- **Objective:** In DAG (approx. quote inclusion), **delete min total edge weight s.t. each connected component has a single "sink"**

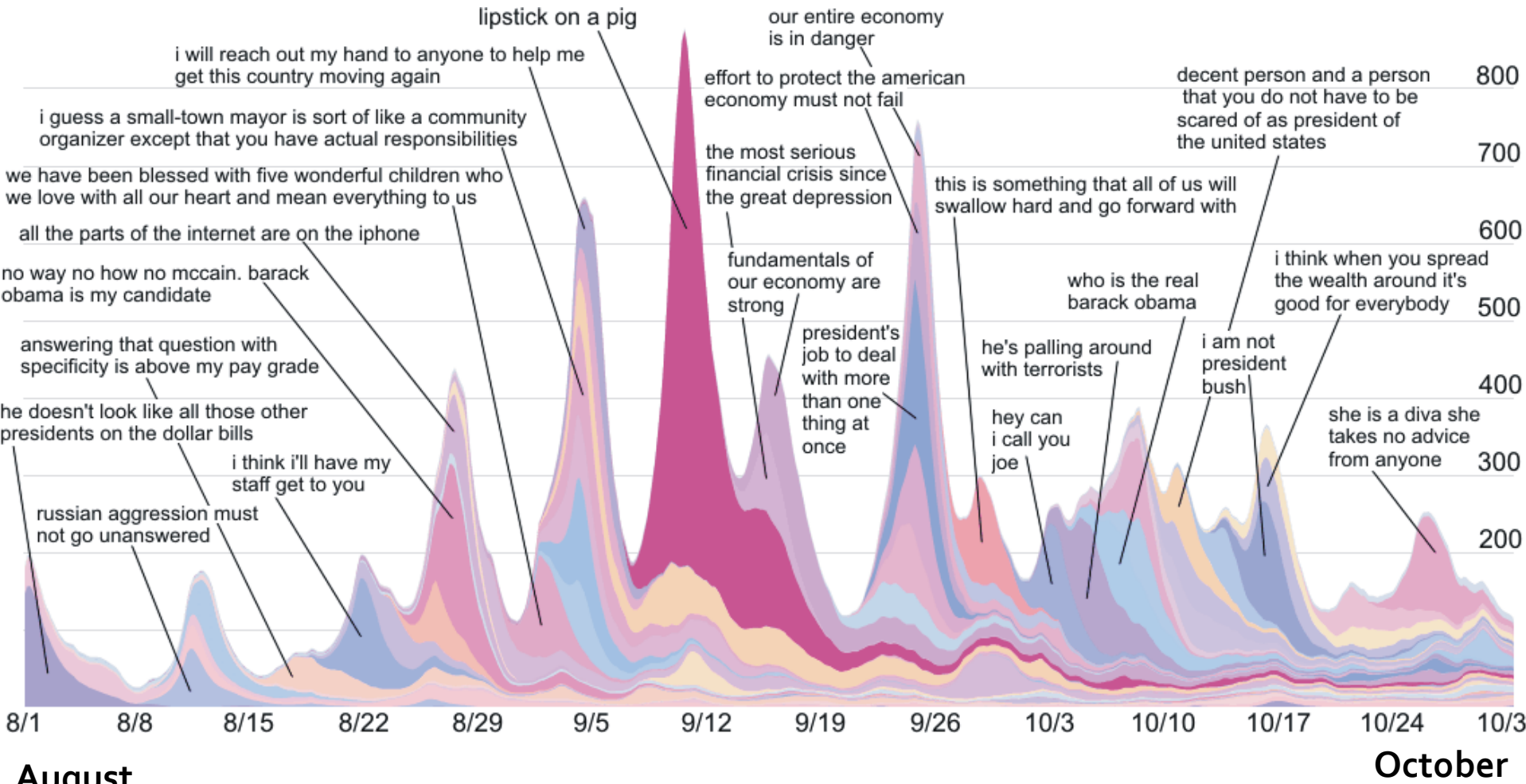


Finding Mutational Variants

- DAG-partitioning is NP-hard but heuristics are effective:
 - **Observation:** enough to know node's parent to reconstruct optimal solution
 - **Heuristic:** Proceed top down and assign a node (keep a single edge) to the strongest cluster



Insights: Quotes reveal pulse of media



Volume over time of top 50 largest total volume quote clusters

<http://memetracker.org>

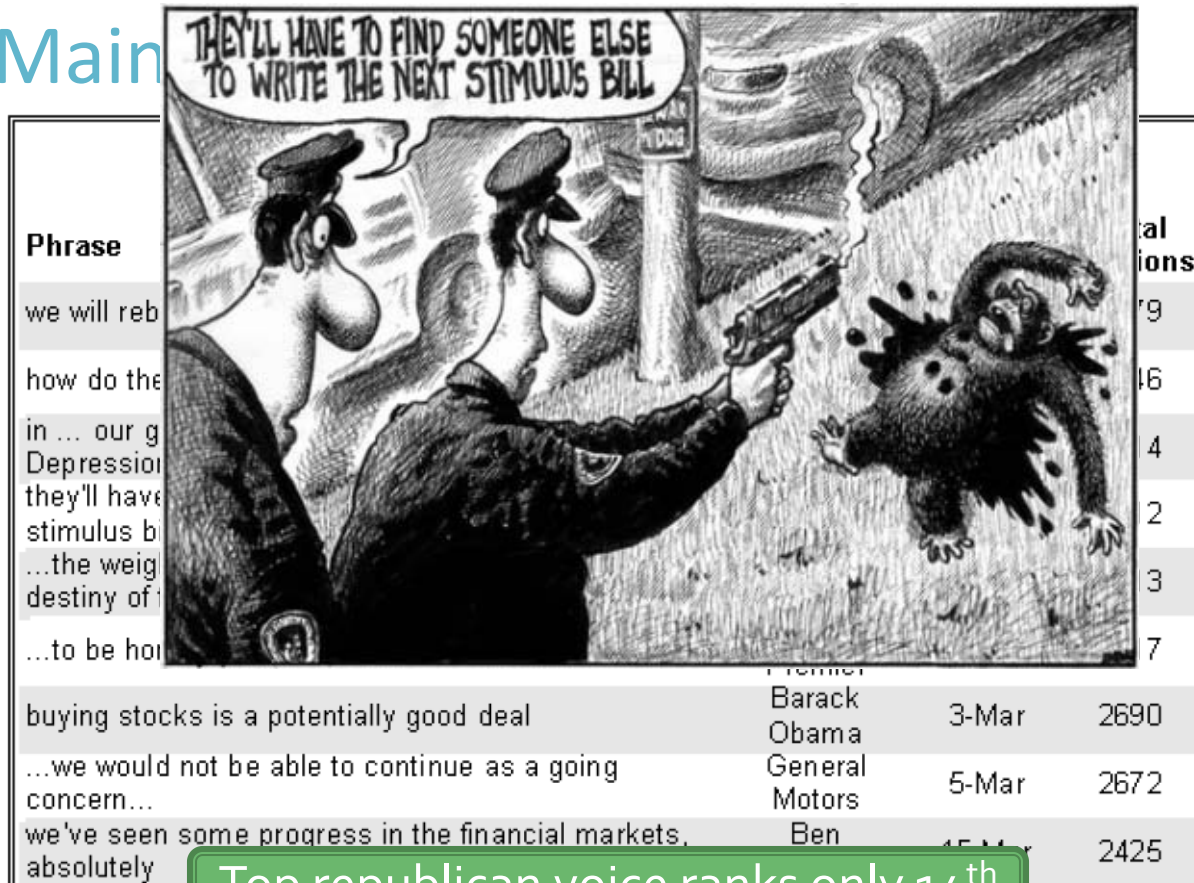
Insights: When sites mention quotes?

- Can classify individual sources by their typical timing relative to the peak aggregate intensity

	Rank	Lag [h]	Reported	Site
Professional blogs	1	-26.5	42	hotair.com
	2	-23	33	talkingpointsmemo.com
	4	-19.5	56	politicalticker.blogs.cnn.com
	5	-18	73	huffingtonpost.com
	6	-17	49	digg.com
	7	-16	89	breitbart.com
	8	-15	31	thepoliticalcarnival.blogspot.com
	9	-15	32	talkleft.com
	10	-14.5	34	dailykos.com
	News media	30	-11	32
34		-11	72	cnn.com
40		-10.5	78	washingtonpost.com
48		-10	53	online.wsj.com
49		-10	54	ap.org

Insights: Quotes on Great depression

- Pew's project for Excellence in journalism
- Media coverage of the current economic crisis
- Main



Speech in congress

Dept. of Labor release



60-minutes interview

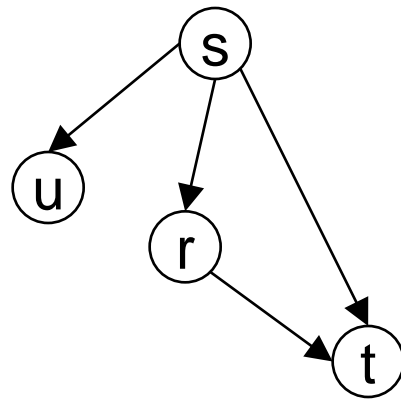
Top republican voice ranks only 14th

Tutorial Outline

- **Part 1: Information flow in networks**
 - 1.1: Data collection: How to track the flow?
 - 1.2: Correcting for missing data
 - 1.3: Modeling and predicting the flow
 - 1.4: Detecting/maximizing influence
- **Part 2: Rich interactions**

Information Diffusion Cascades

Network



(s, r, t) 

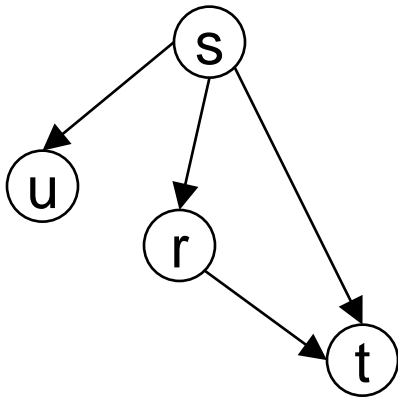
s  r

r  t 

- Users are nodes in a social network.
- We know the network
- We focus on some action users have performed (e.g., tweeted about “mubarak”)
- We may or may not know which node influenced which other node

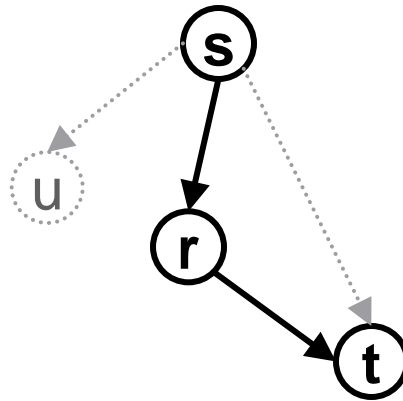
Information Diffusion Cascades

Network



Influence Cascade

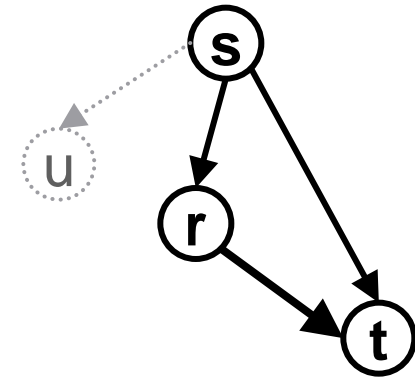
(e.g., Twitter re-tweets)



(s, r, t)
s → r
r → t

Network Cascade

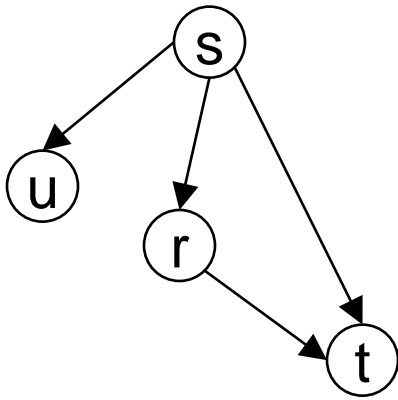
(e.g., Twitter hashtags)



(s, r, t)

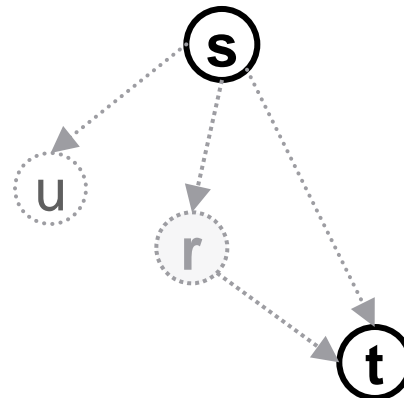
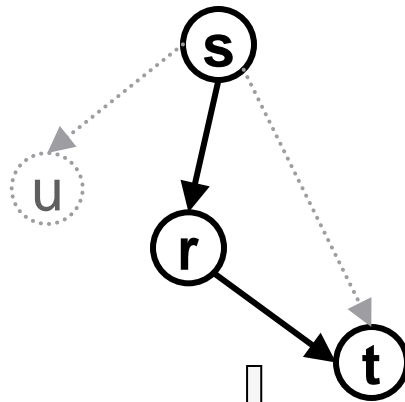
What happens with missing data?

Network



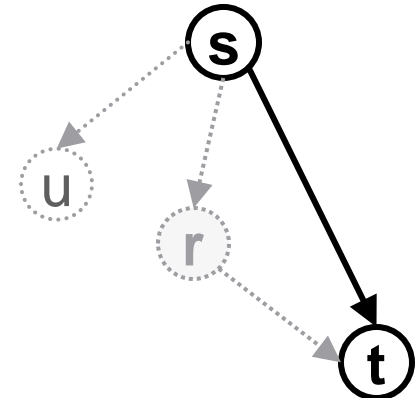
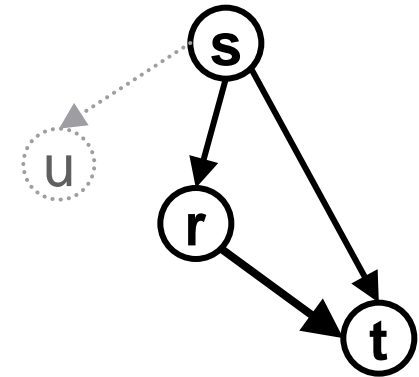
Influence Cascade

(e.g., Twitter re-tweets)



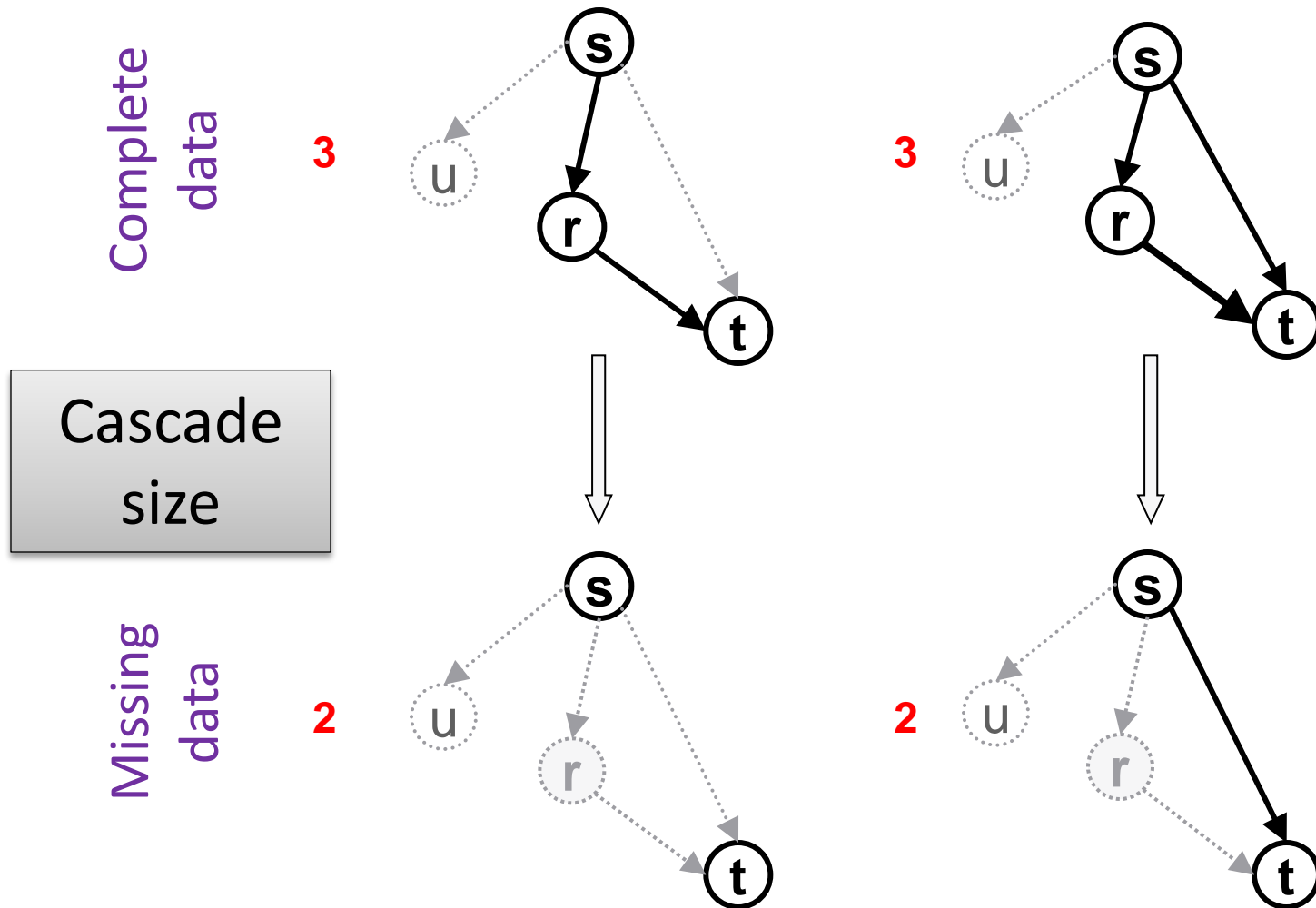
Network Cascade

(e.g., Twitter hashtags)

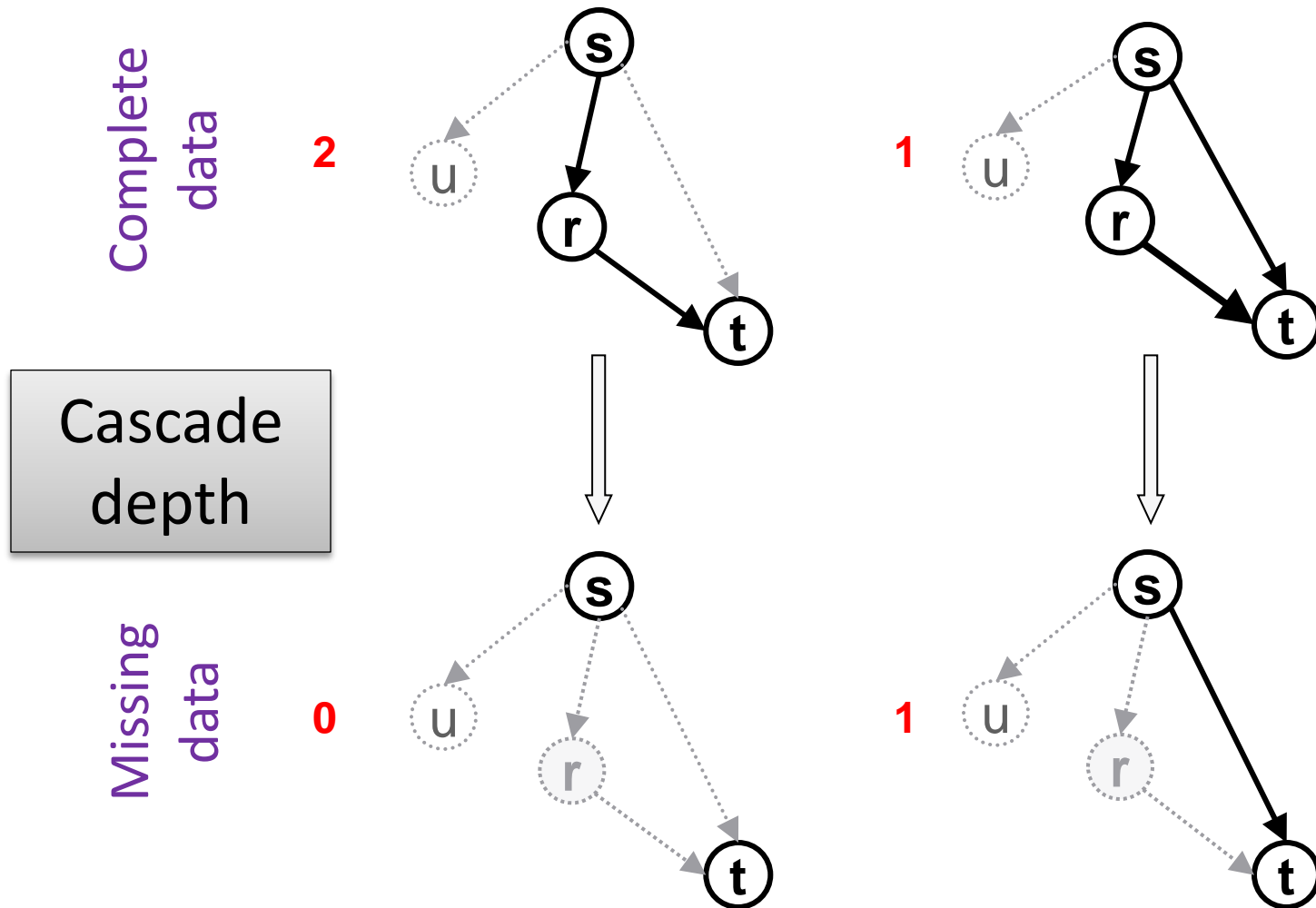


Data about
node *r* is
missing

Missing Data Distorts Cascades (1)



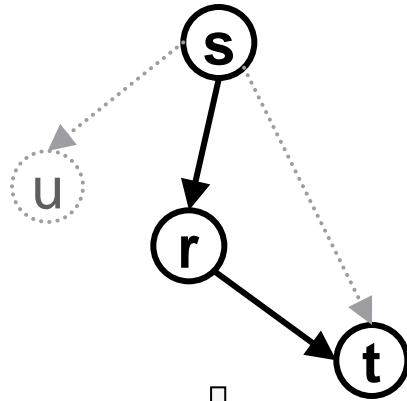
Missing Data Distorts Cascades (2)



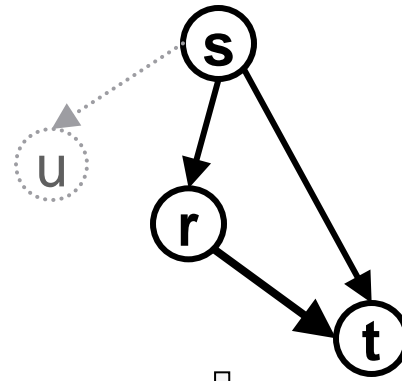
Missing Data Distorts Cascades (3)

Complete
data

2



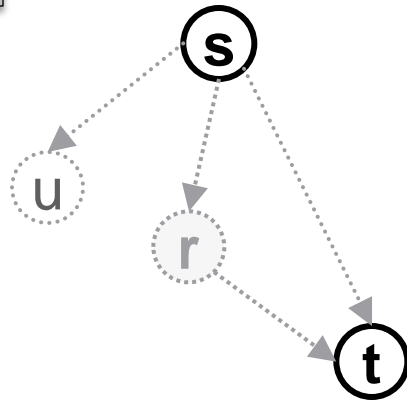
2



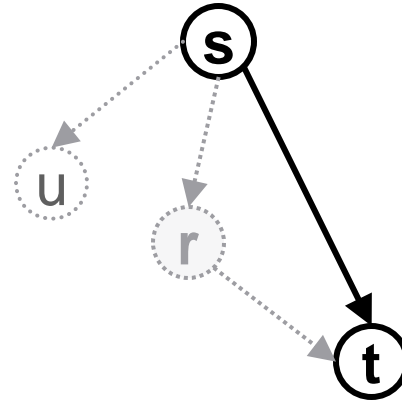
Participation:
#non-leaves

Missing
data

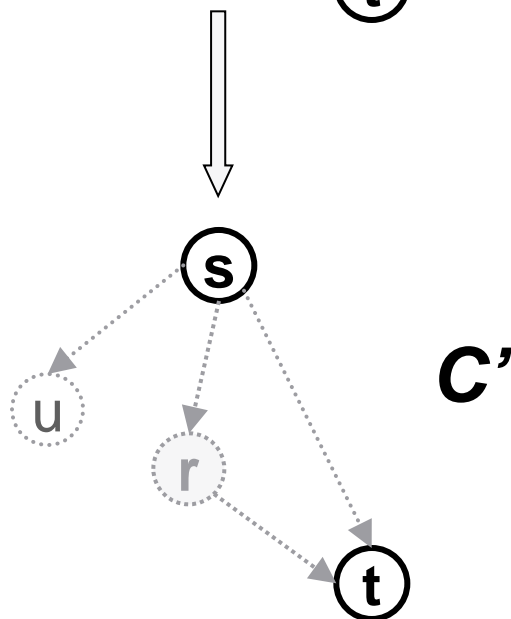
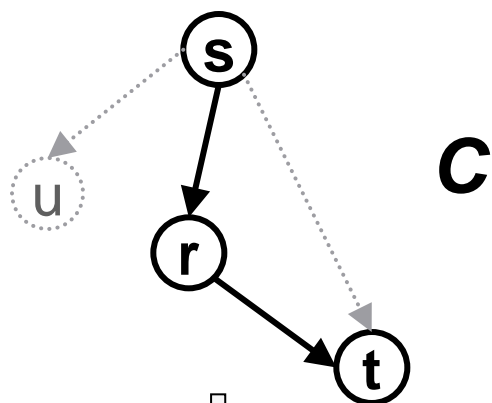
0



1

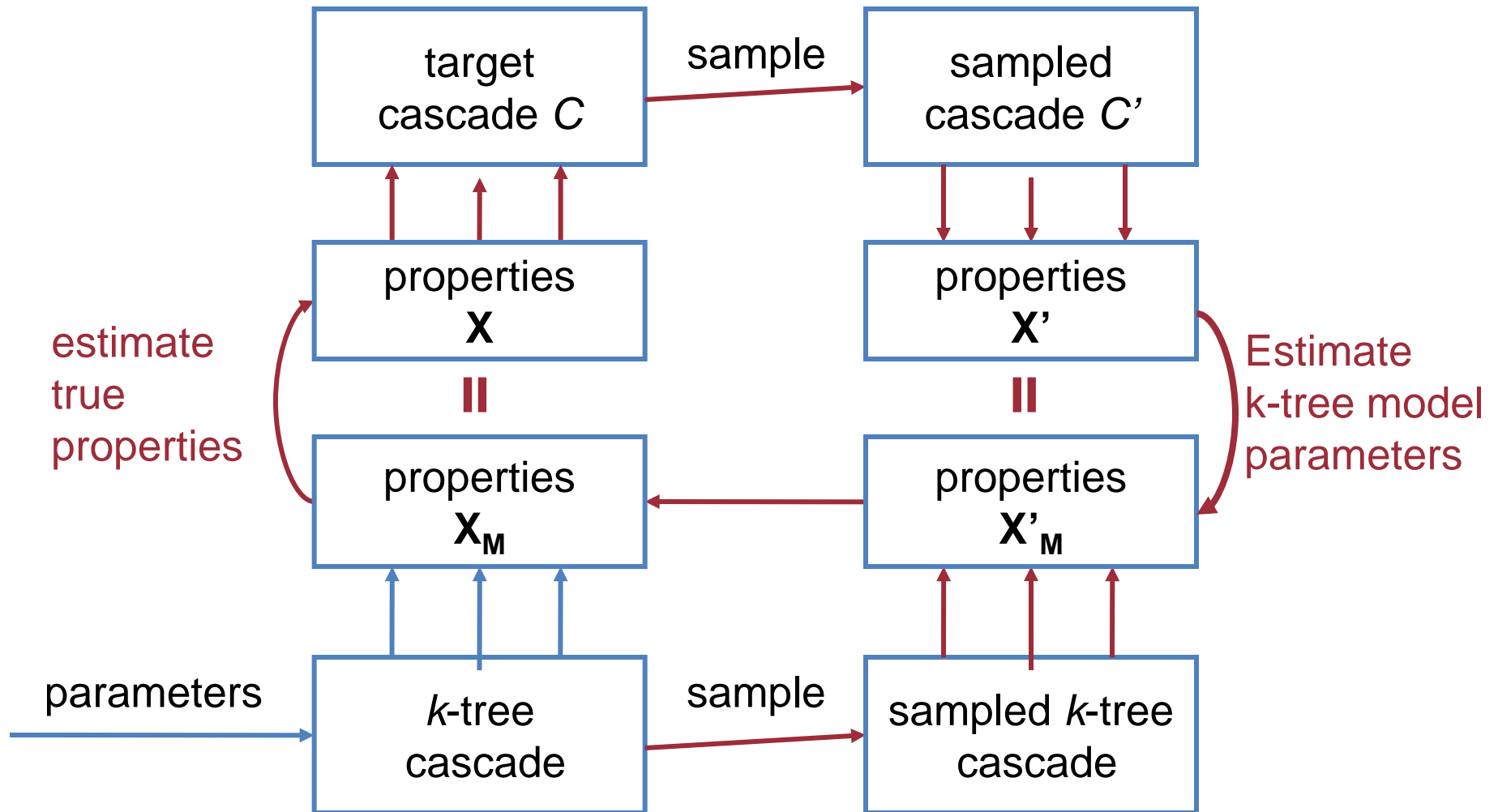


Problem Statement



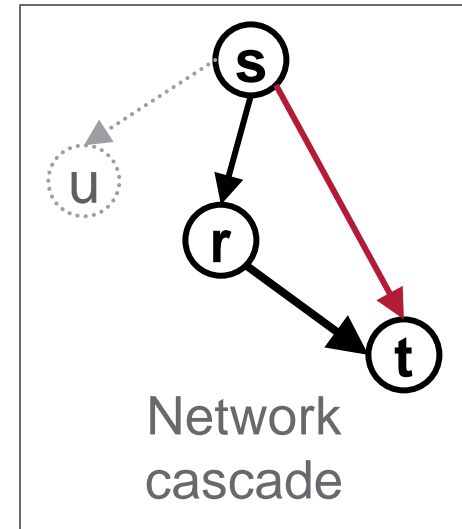
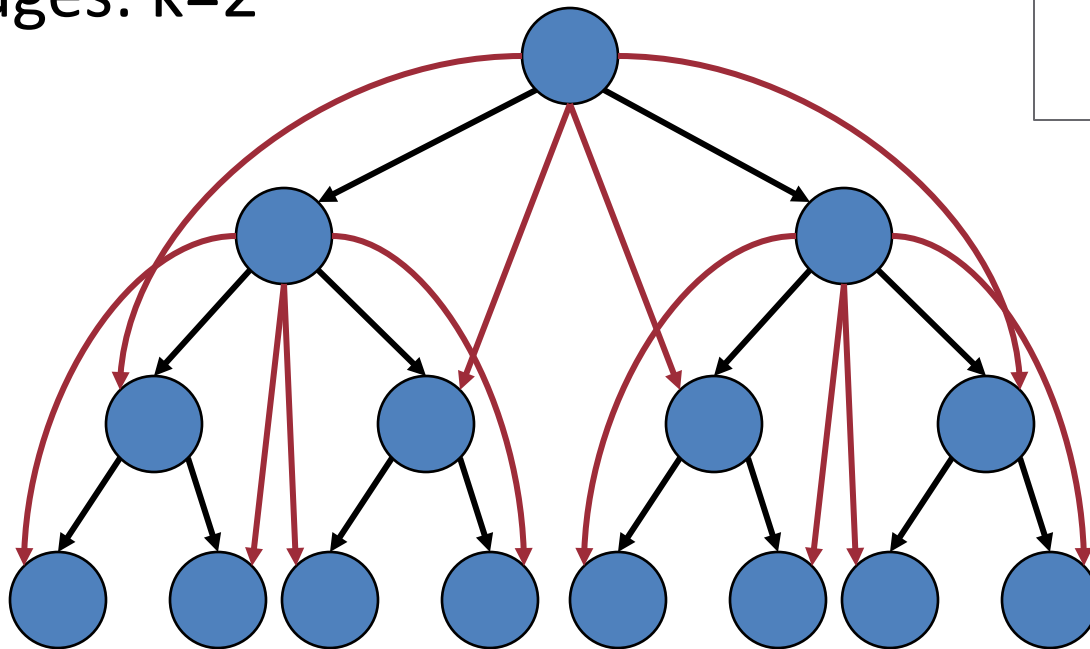
- **Goal:** Find properties X of the complete cascade C
- We only have access to cascade C' that is C with missing data
 - Missing data regime: Each node of C is missing independently with probability p
- Properties X' of C' are inaccurate ($X \neq X'$)

Methodology

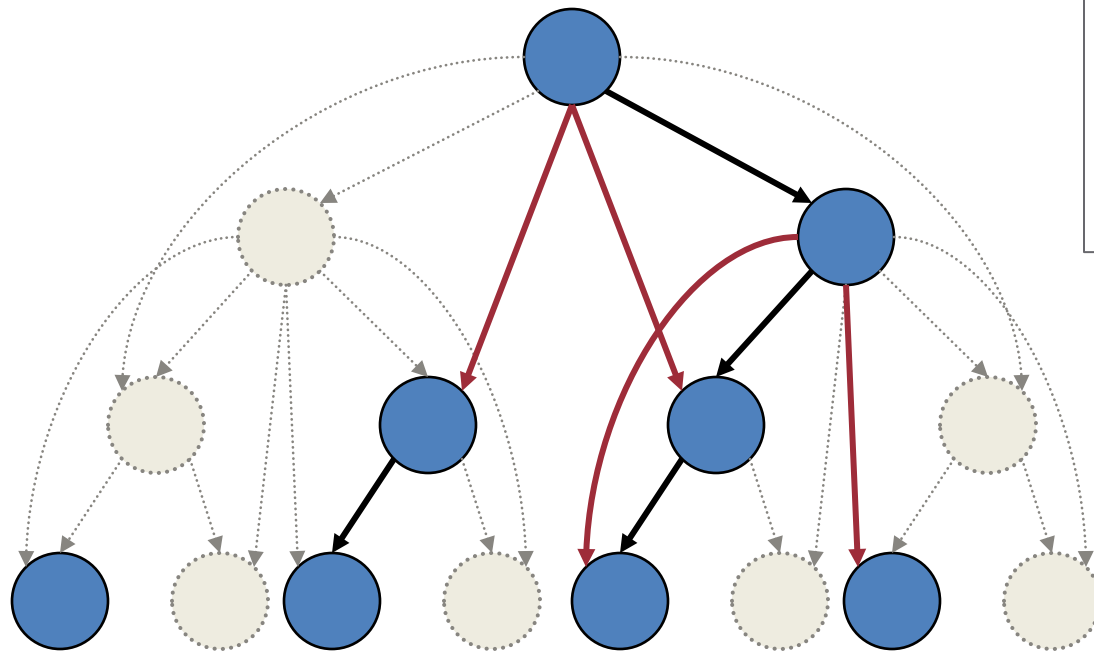


k-Tree Model of Cascades

- 3 Parameters of the model:
 - Branching: $b=2$
 - Depth: $h=3$
 - In-edges: $k=2$



Properties of k-trees with missing data

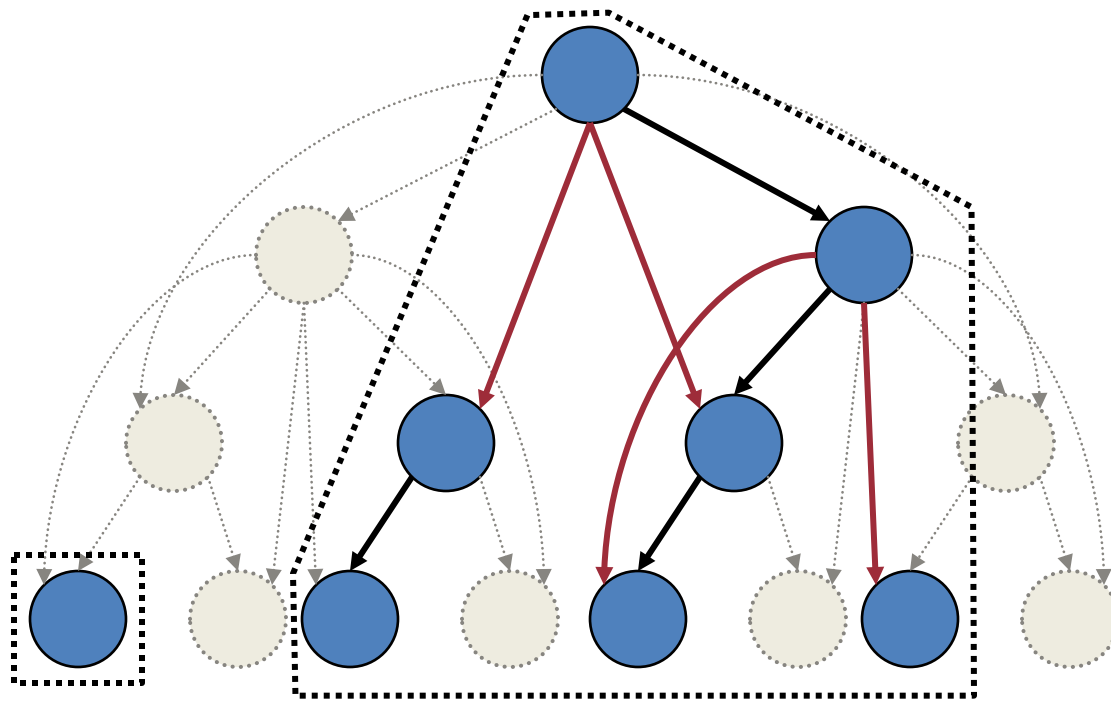


p = fraction of nodes
observed in the sample

Interested in k-tree
properties as a function of p

- Relatively simple to reason about
- Can calculate an exact expressions for many different properties of the k-tree

Example: # of Connected Components



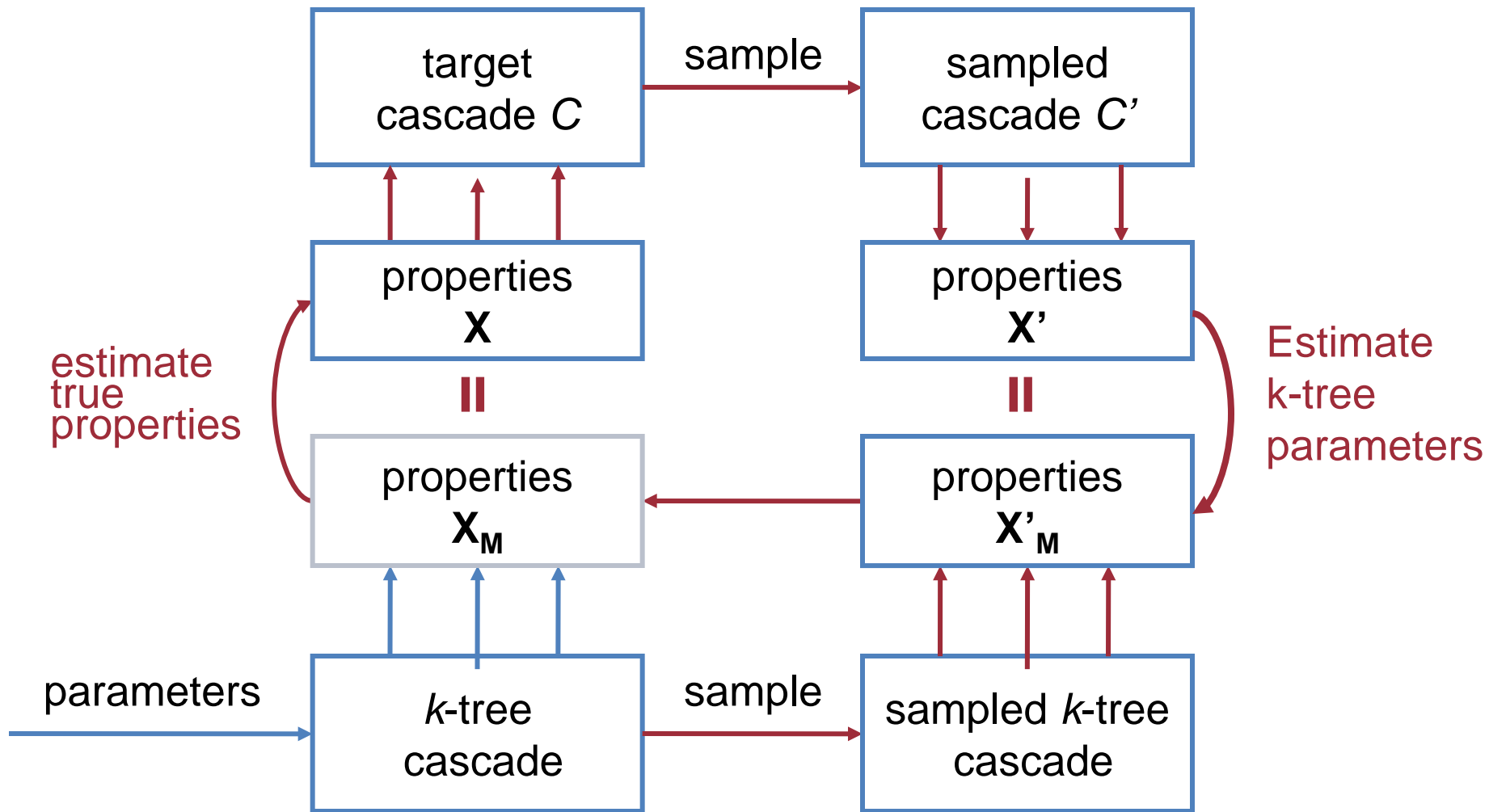
Number of connected components as a function of k-tree model parameters (b, h, k) and missing data rate p :

$$\#CC = \frac{p^2}{b-1} \left(\frac{b(1-b^k)}{b-1} + kb^{h+1} \right)$$

Properties of k-trees

- Derive equations linking k-tree parameters with a cascade properties:
 - # of nodes
 - # of edges
 - # of connected components = $\frac{p^2}{b-1} \left(\frac{b(1-b^k)}{b-1} + kb^{h+1} \right)$
 - # of isolated nodes
 - # of leaves
 - Average node degree
 - Out-degree of non-leaves
 - Size of the largest connected component
- For each property find an “equation” that links model parameters (p,b,h,k) to the value

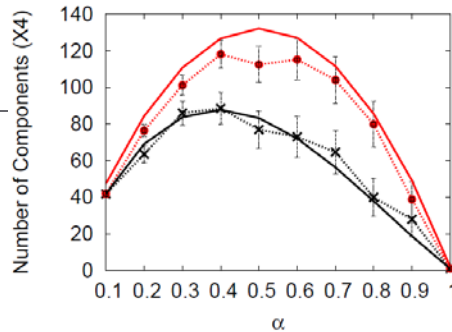
Methodology Overview



Estimating k-tree Parameters

$$X'(\alpha p) = \frac{(\alpha p)^2}{b-1} \left(\frac{b(1-b^k)}{b-1} + kb^{h+1} \right)$$

where X' is the # of connected components in the αp sample

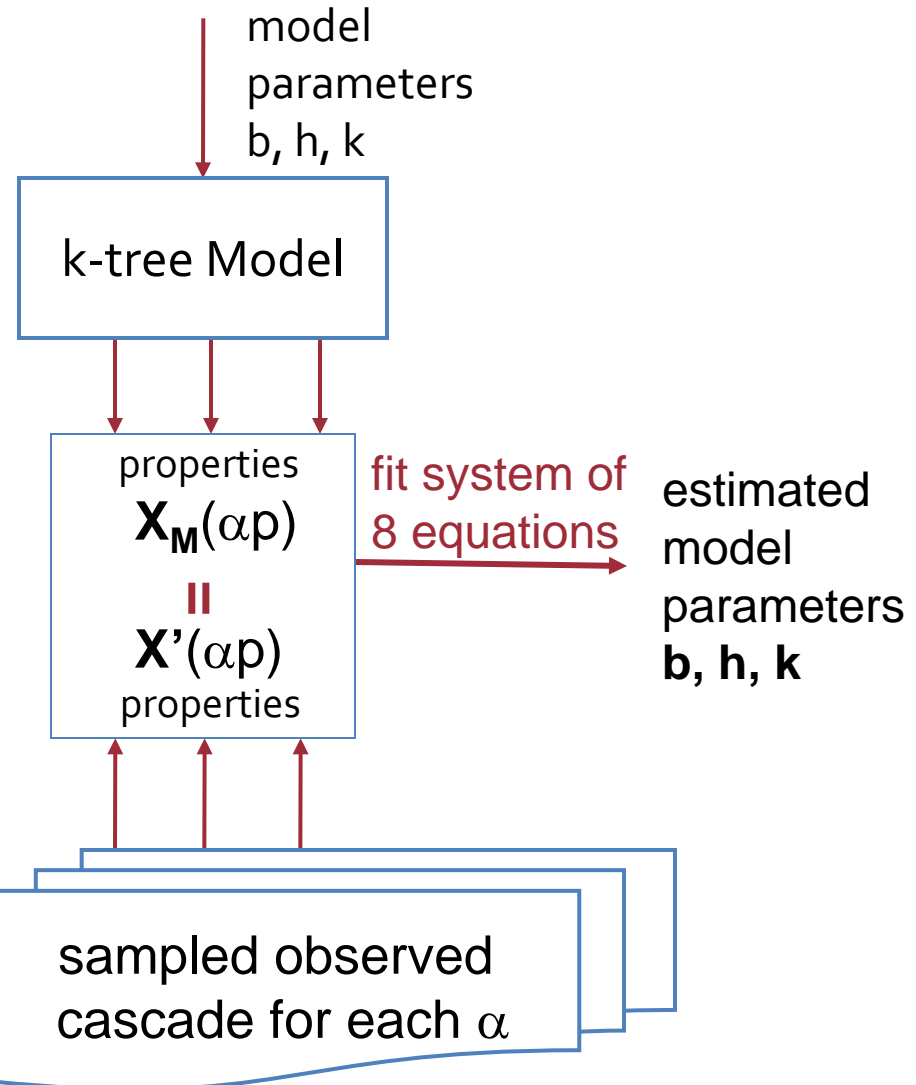


Sample p fraction

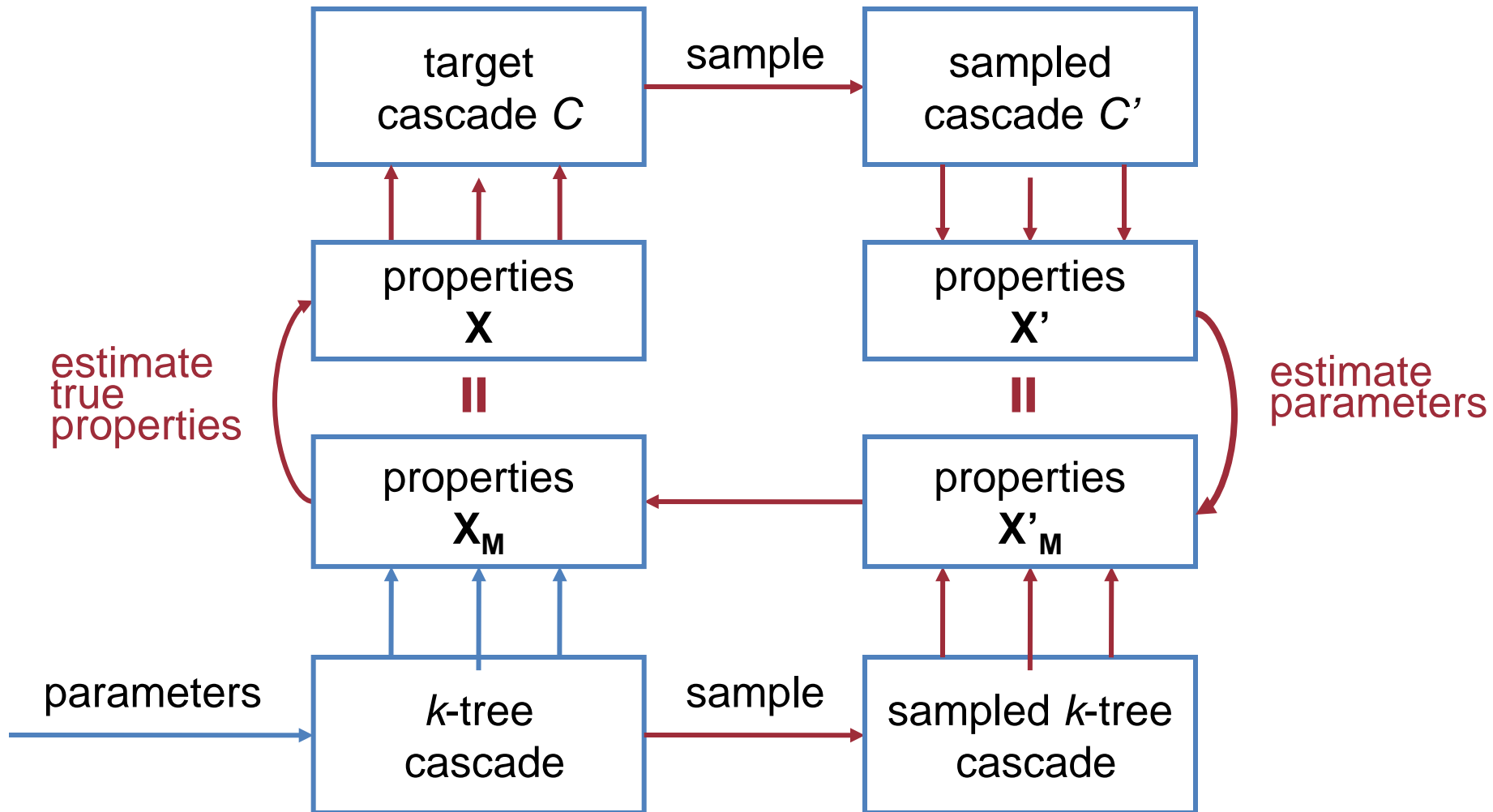
observed cascade C'

subsample α fraction

sampled observed cascade for each α



Methodology Overview



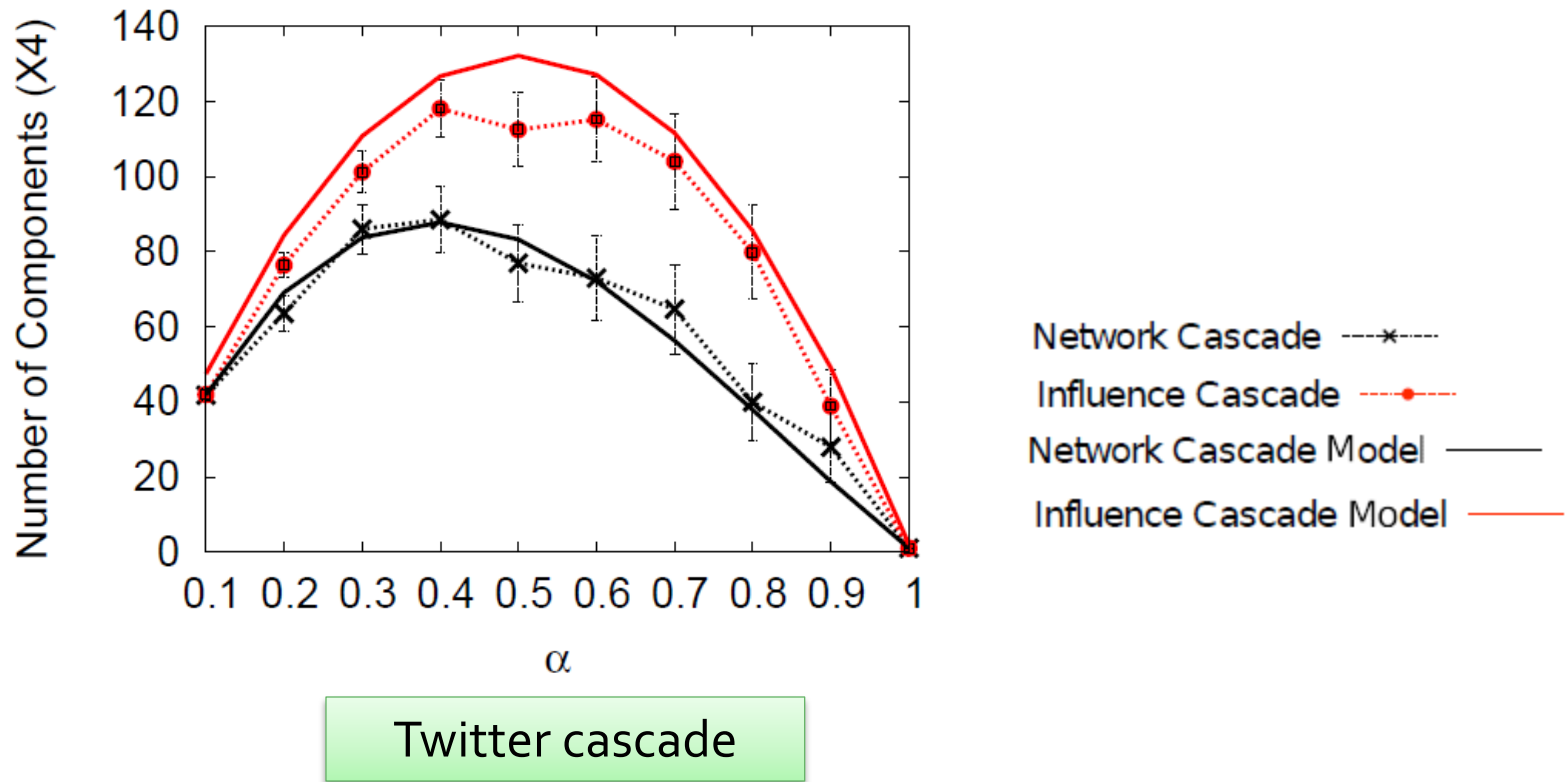
Experiments

- Are k-trees a good model for cascades?
- How well can we correct for missing data?
- Do parameters of the model correspond to real cascade properties?

Data Sets

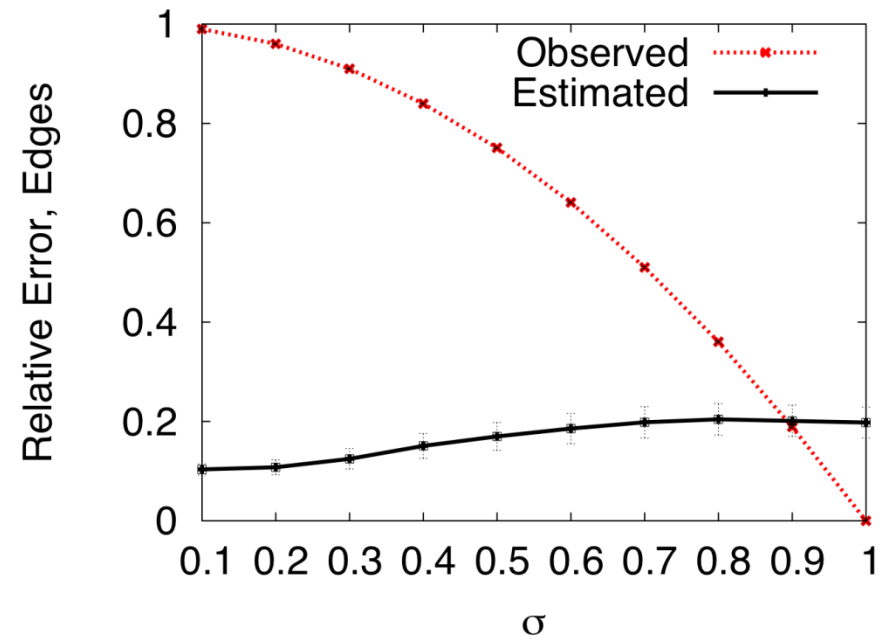
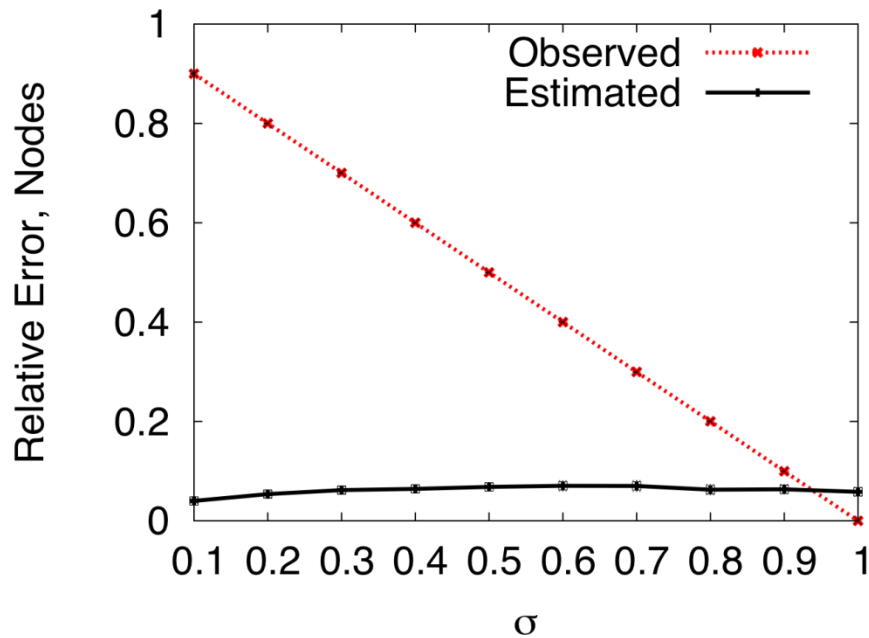
- **Twitter:**
 - Network of 72 million nodes and 2 billion edges
 - Complete set of 350 million tweets over 6 months
 - **Influence cascade:** URL retweet cascades
 - **Network cascade:** Pretend we do not know URL of which friend did you forward
- **Sprinn3r blogs:**
 - English blogosphere posts over 2 months
 - Influence cascades: Cascades formed by links between the blog posts

Is k-tree a good cascade model?



- Yes! Curve of the model matches the empirically measured values of a real cascade.

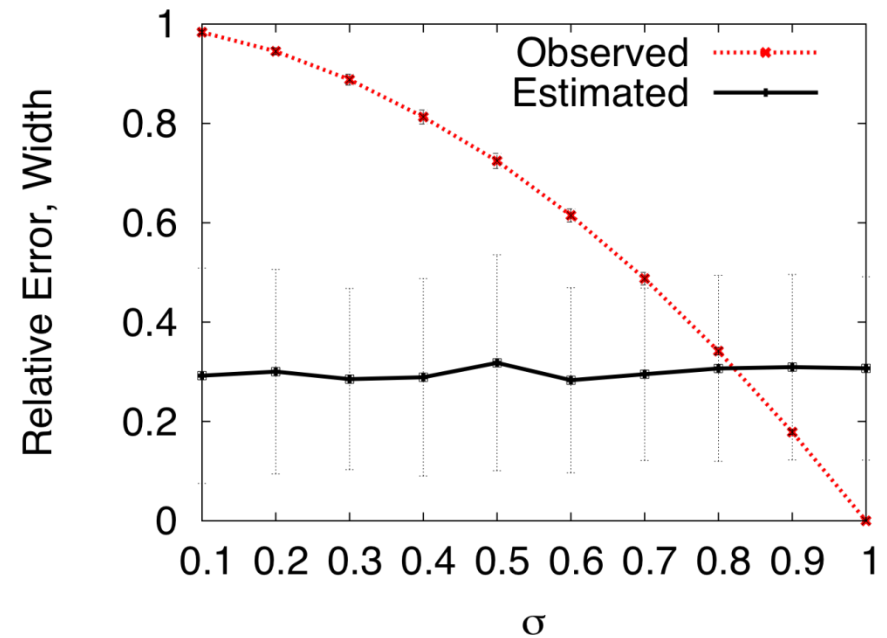
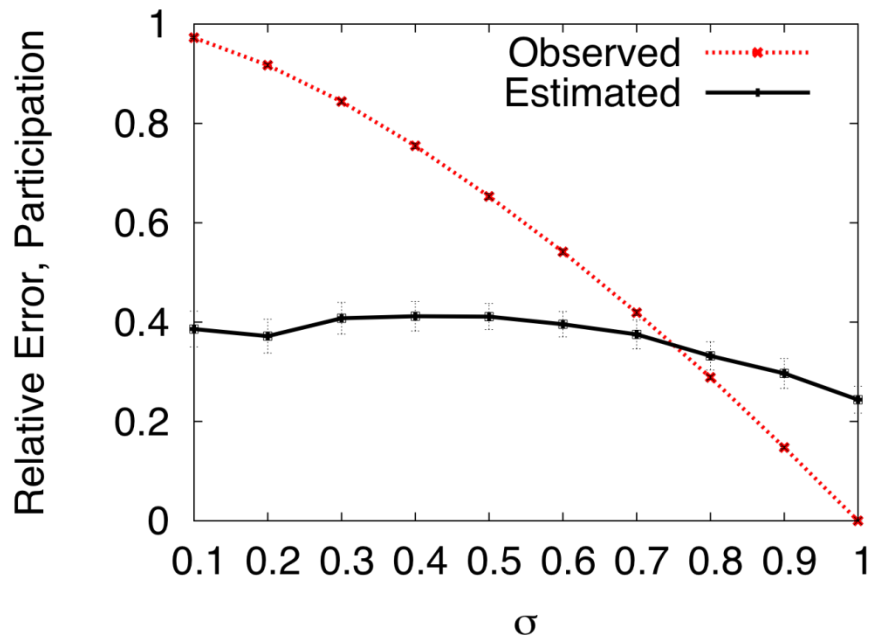
Can Correct for Missing Data?



Twitter cascades

- Do significantly better than the observed uncorrected values!

Can Correct for Missing Data?



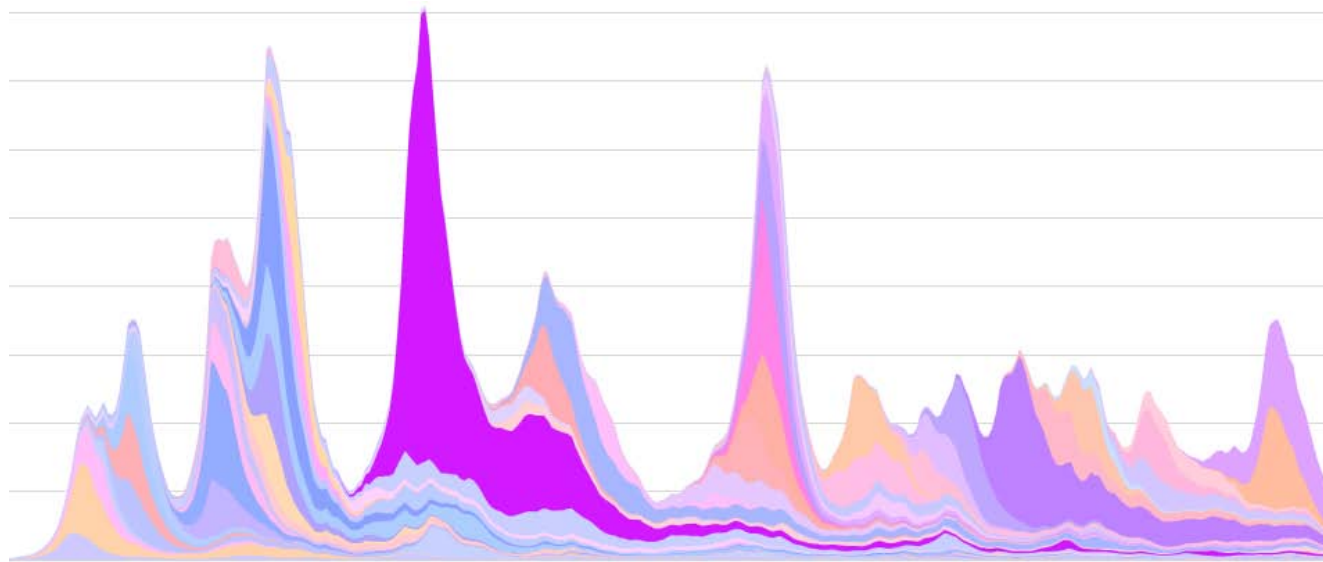
Twitter cascades

- Our method is most effective when more than 20% of the data is missing
- Works well even with 90% of the data missing

Tutorial Outline

- **Part 1: Information flow in networks**
 - 1.1: Data collection: How to track the flow?
 - 1.2: Correcting for missing data
 - 1.3: Modeling and predicting the flow
 - 1.4: Infer networks of information flow
- **Part 2: Rich interactions**

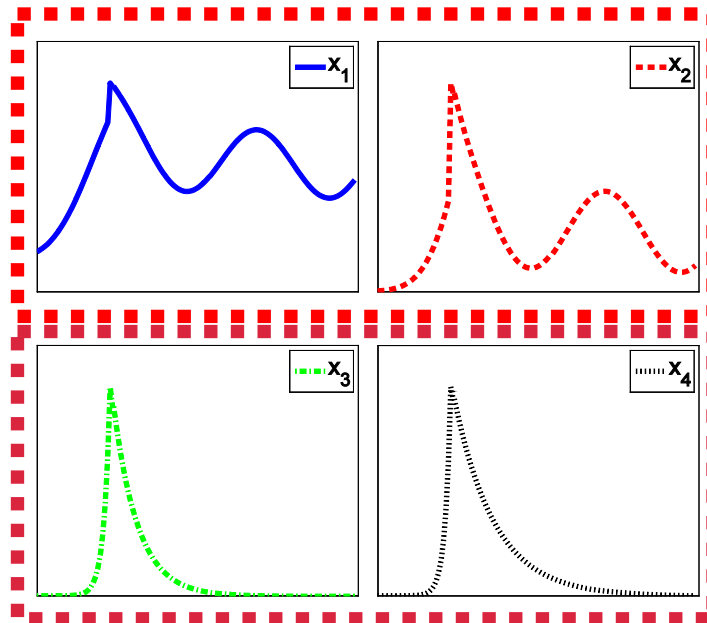
Patterns of Information Attention



- **Q: What are temporal patterns of information attention?**
 - **Item i :** Piece of information (e.g., quote, url, hashtag)
 - **Volume $x_i(t)$:** # of times i was mentioned at time t
 - Volume = number of mentions = attention = popularity
 - **Q: Typical classes of shapes of $x_i(t)$**

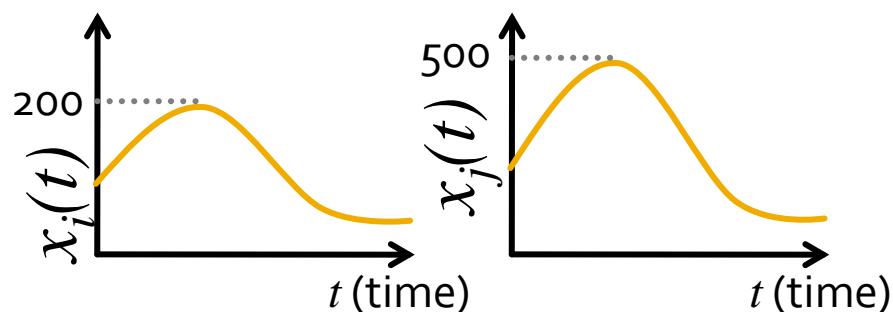
Discovering Attention Patterns

- **Given:** Volume of an item over time
- **Goal:** Want to discover types of shapes of volume time series

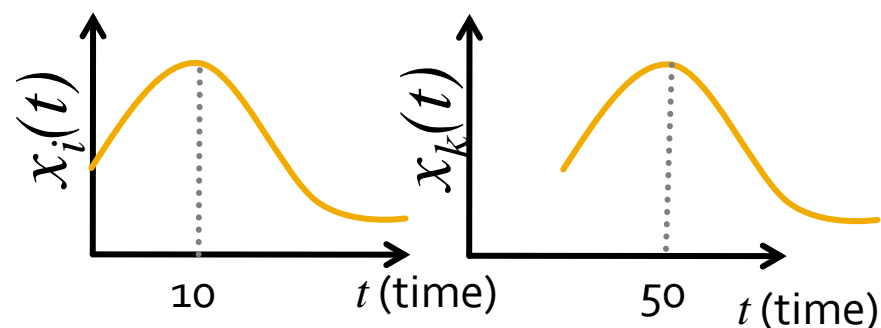


Clustering Temporal Signatures

- Goal: Cluster time series & find cluster centers
- Time series distance function needs to be:



Invariance to scaling

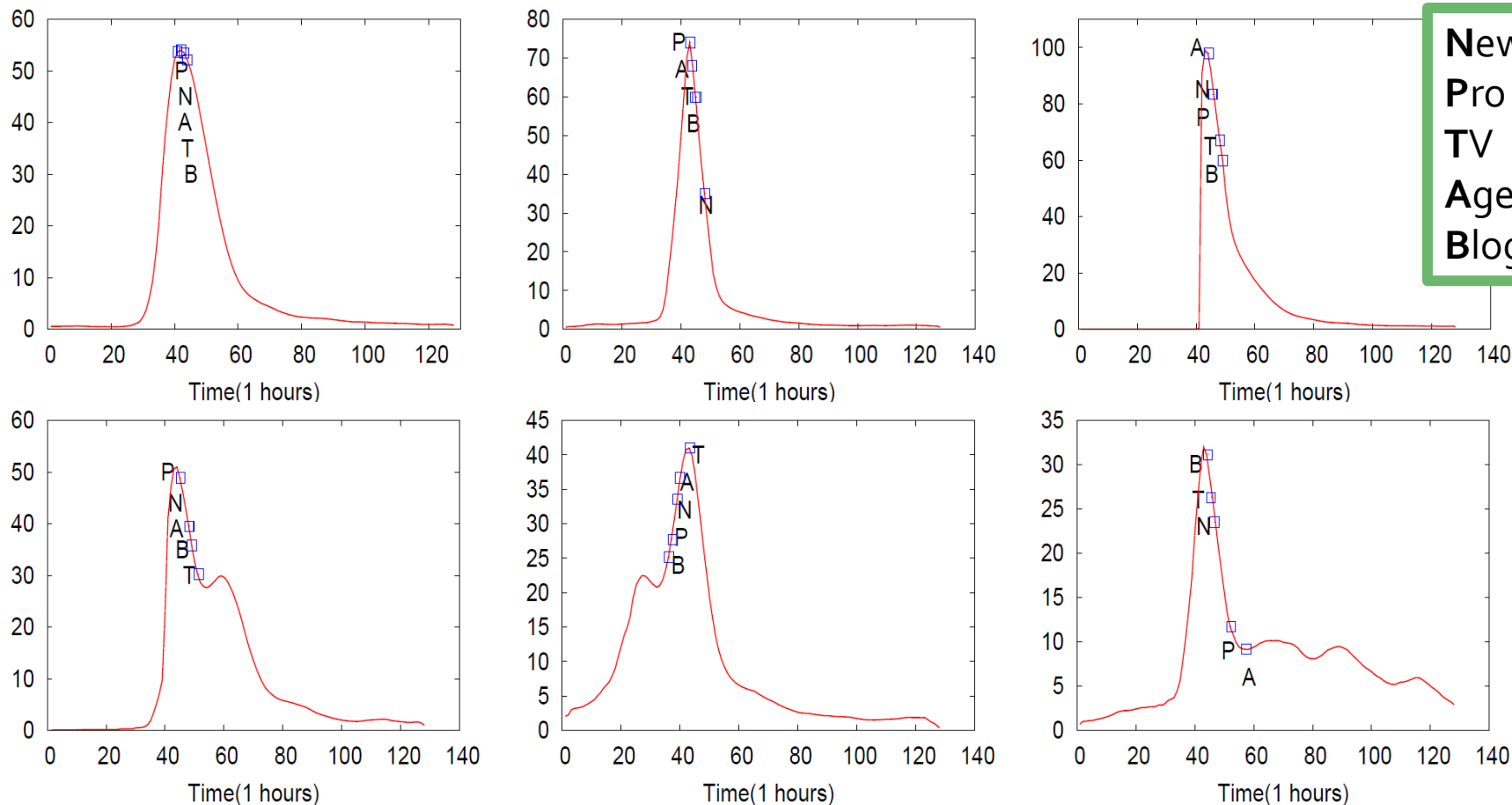


Invariance to translation

$$d(x, y) = \min_{a, q} \sum_t (x(t) - a \cdot y(t - q))^2$$

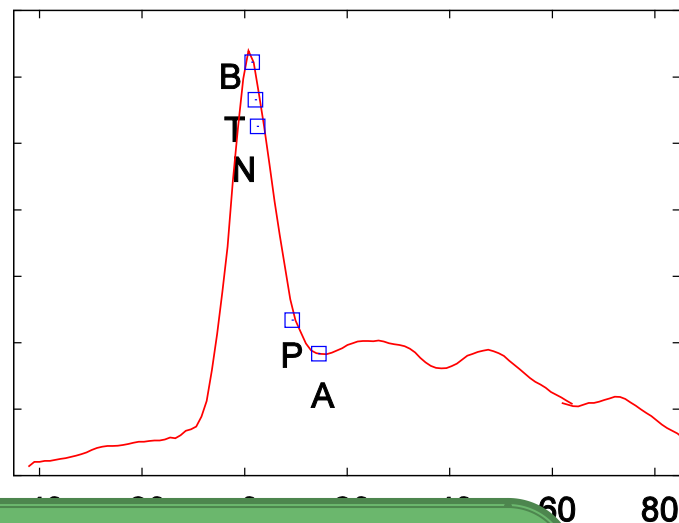
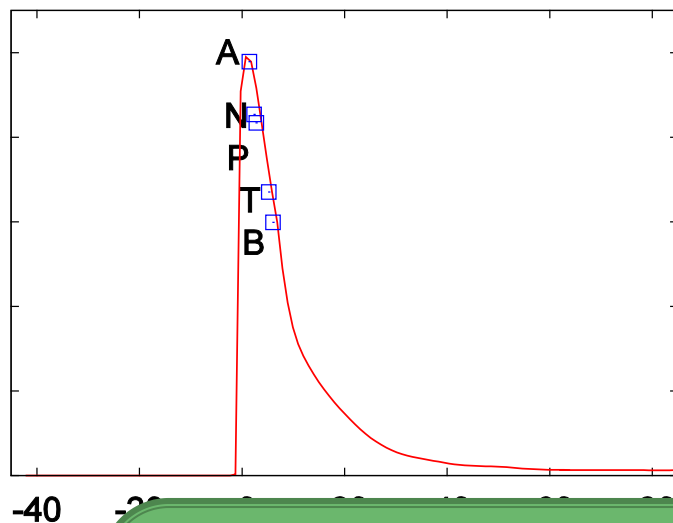
- K-Spectral Centroid clustering [WSDM '11]

Patterns of Attention



- **Quotes:** 1 year, 172M docs, 343M quotes
- **Same 6 shapes for Twitter:** 580M tweets, 8M #tags

Analysis of Attention Patterns



Different media give rise
to different patterns

- Spike

Agency

- Slow &

- Blogs

the mainstream media

- Blog volume = 29.1%

mainstream media

- Blog volume = 53.1%

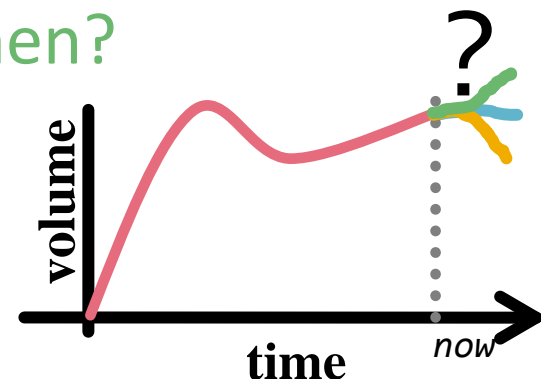
Predicting Attention

■ How much attention will information get?

■ Who reports the information and when?

- 1h: Gizmodo, Engadget, Wired
- 2h: Reuters, Associated Press
- 3h: New York Times, CNN

■ How many will mention the info at time 4, 5, ...?

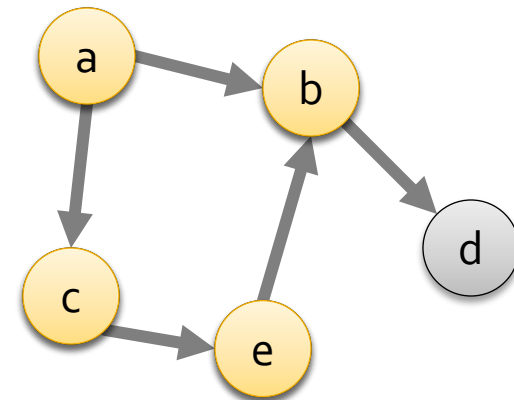


■ Motivating question:

- If NYT mentions info at time t
- How many subsequent mentions of the info will this generate at time $t+1, t+2, \dots$?

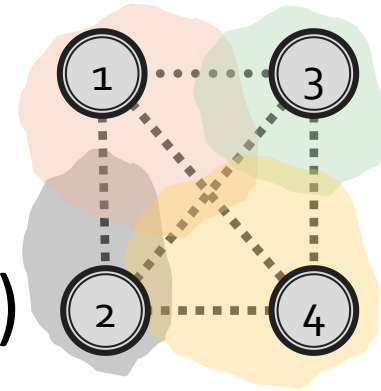
Predicting Information Diffusion

- **Goal:**
 - Predict future attention (number of mentions)
- **Traditional view:**
 - In a network “infected” nodes spread info to their neighbors
- **Problem:**
 - The network may be unknown
- **Idea:** Predict the future attention based on which nodes got “infected” in the past



Linear Influence Model

- **Idea:** Predict the volume based on who got infected in the past
- **Solution:** Linear Influence Model (LIM)
 - Assume no network
 - Model the global influence of each node
 - Predict future volume from node influences
- **Advantages:**
 - No knowledge of network needed
 - Contagion can “jump” between the nodes



LIM: Strategy

t	M(t)	V(t)
1	U, W	2
2	V, X, Y	3
3		?

- **K=1 contagion:**
 - $V(t)$...number of new infections at time t
 - $M(t)$...set of newly infected nodes at time t
- How does **LIM** predict the future number of infections $V(t+1)$?
 - Each node u has an **influence function**:
 - After node u gets infected, how many other nodes tend to get infected
 - Estimate the influence function from past data
 - Predict future volume using the influence functions of nodes infected in the past

The Linear Influence Model

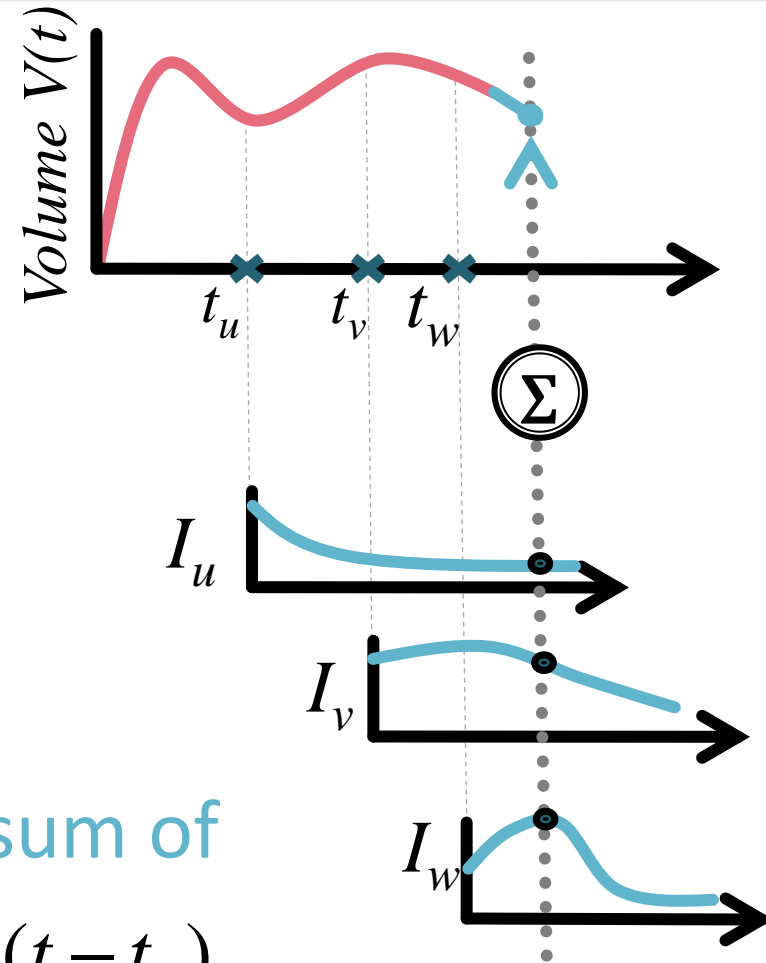
How to predict future volume $x_i(t+1)$ of info i ?

- Node u has an **influence function** $I_u(q)$:
 - $I_u(q)$: After node u gets “infected”, how many other nodes tend to get infected q hours later
 - E.g.: Influence function of CNN:
How many sites say the info after they see it on CNN?
 - Estimate the influence function from past data
- Predict future volume using the influence functions of nodes infected in the past

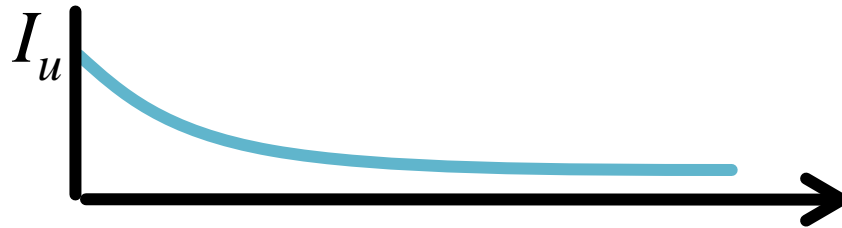
The Linear Influence Model

LIM model:

- Volume $x_i(t)$ of i at time t
- $A_i(t)$... a set of nodes that mentioned i before time t
- And let:
 - $I_u(q)$: influence function of u
 - t_u : time when u mentioned i
- Predict future volume as a sum of influences:
$$x_i(t+1) = \sum_{u \in A_i(t)} I_u(t - t_u)$$



Estimating Influence Functions



- After node u mentions the info, $I_u(q)$ other mentions tend to occur q hours later
 - $I_u(q)$ is not observable, need to estimate it
 - Make no assumption about its shape
 - Model $I_u(q)$ as a vector: $I_u(q) = [I_u(1), I_u(2), I_u(3), \dots, I_u(L)]$
 - Find $I_u(q)$ by solving a **least-squares-like** problem:

$$\min_{I_u, \forall u} \sum_i \sum_t \left(x_i(t+1) - \sum_{u \in A_i(t)} I_u(t - t_u) \right)^2$$

LIM: Influence Functions

- Discrete non-parametric influence functions:

- Discrete time units

- $I_u(t)$... non-negative vector of length L

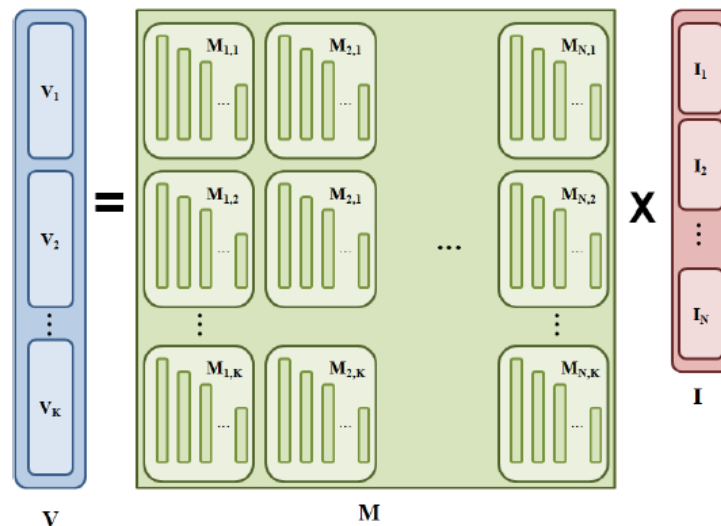
$$I_u(t) = [I_u(1), I_u(2), I_u(3), \dots, I_u(L)]$$

How do we estimate the influence functions $I_u(t)$?

- **Note:** This makes no assumption about the shape of $I_u(t)$

LIM as matrix equation

- Input data: K contagions, N nodes
- Write LIM as a matrix equation:



- Volume vector:
 $V_k(t)$... volume of contagion k at time t
- Infection indicator matrix:
 $M_{u,k}(t) = 1$ if node u gets infected by contagion k at time t
- Influence functions:
 $I_u(t)$... influence of node u on diffusion

Estimating influence functions

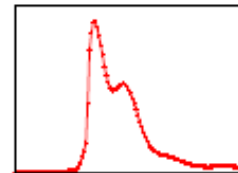
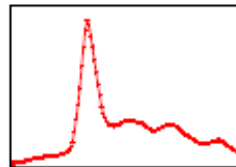
- **LIM** as a matrix equation: $V = M * I$
- Estimate influence functions:

$$\hat{\mathbf{I}} = \arg \min_{\mathbf{I} \geq 0} \|\mathbf{V} - \mathbf{M} \cdot \mathbf{I}\|_2^2$$

- Solve using Non-Negative Least Squares
 - Well known, we use Reflective Newton Method
 - Time ~ 1 sec when M is 200,000 x 4,000 matrix
- Predicting future volume: **Simple!**
 - Given M and I , then
 - $V = M * I$

LIM: Performance

- Take top 1,000 quotes by the total volume:
 - Total 372,000 mentions on 16,000 websites
- Build LIM on 100 highest-volume websites
 - $x_i(t)$... number of mentions across 16,000 websites
 - $A_i(t)$... which of 100 sites mentioned quote i and when
- Improvement in L2-norm over 1-time lag predictor

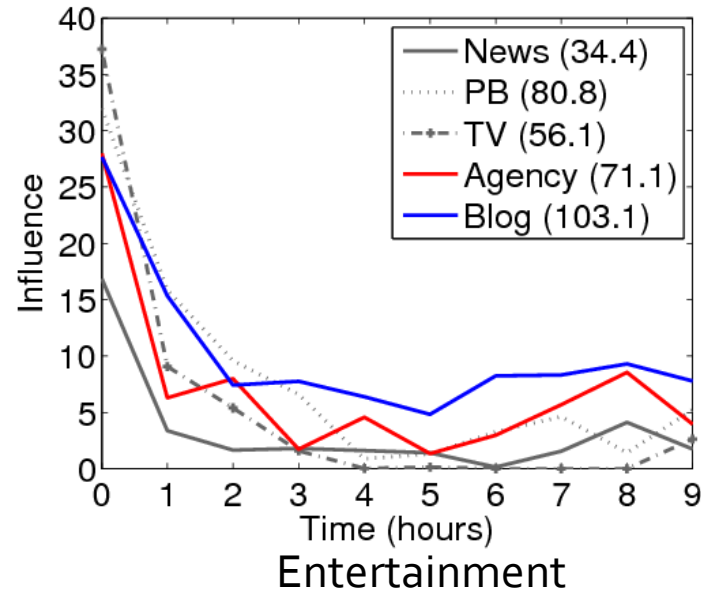
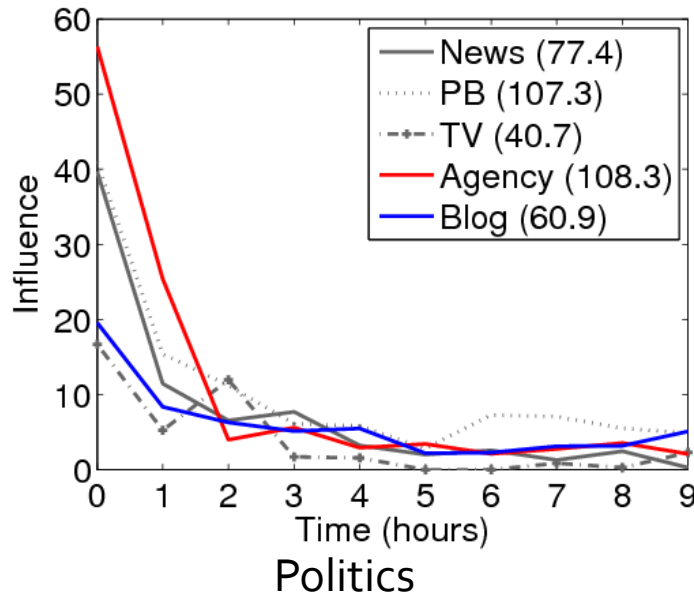


	Bursty phrases	Steady phrases	Overall
AR	7.21%	8.30%	7.41%
ARMA	6.85%	8.71%	7.75%
LIM (N=100)	20.06%	6.24%	14.31%

Analysis of Influence Functions

- Influence functions give insights:
 - **Q:** NYT writes a post on politics, how many people tend to mention it next day?
 - **A:** Influence function of NYT for political phrases!
- Experimental setup:
 - 5 media types:
 - Newspapers, Pro Blogs, TVs, News agencies, Blogs
 - 6 topics:
 - Politics, nation, entertainment, business, technology, sports
 - For all phrases in the topic, estimate average influence function by media type

Analysis of Influence



News Agencies, Personal Blogs (Blog), Newspapers, Professional Blogs, TV

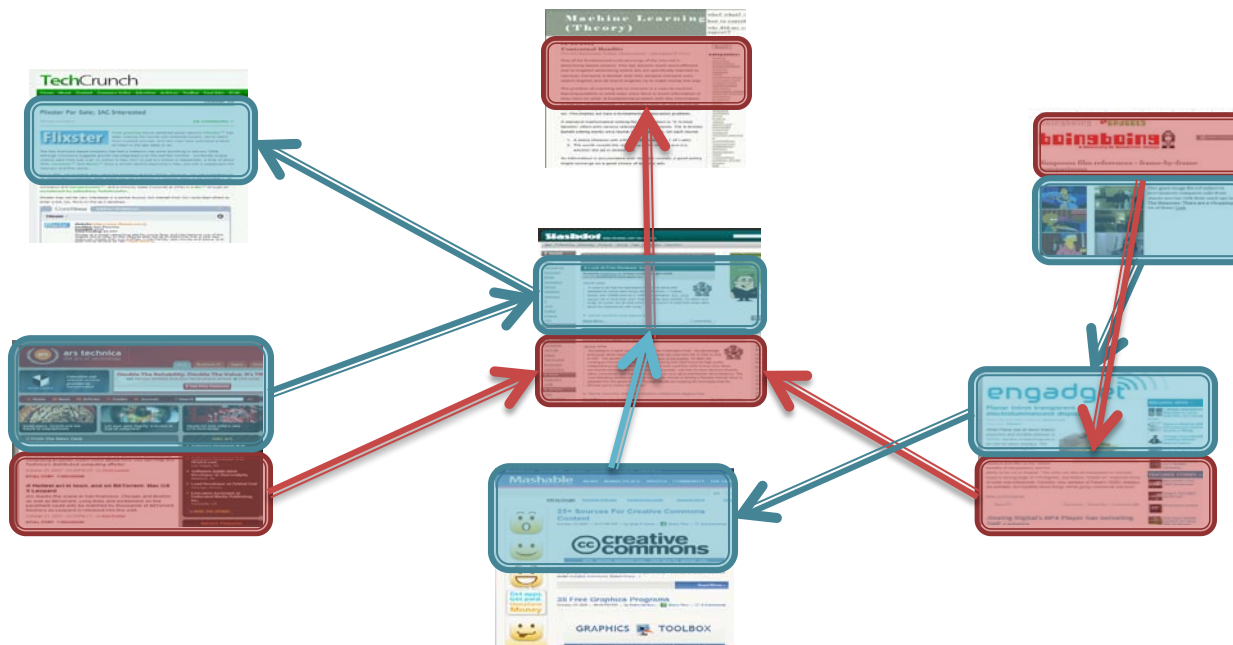
- Politics is dominated by traditional media
- Blogs:
 - Influential for Entertainment phrases
 - Influence lasts longer than for other media types

Tutorial Outline

- **Part 1: Information flow in networks**
 - 1.1: Data collection: How to track the flow?
 - 1.2: Correcting for missing data
 - 1.3: Modeling and predicting the flow
 - 1.4: Infer networks of information flow
- **Part 2: Rich interactions**

Inferring the Diffusion Network

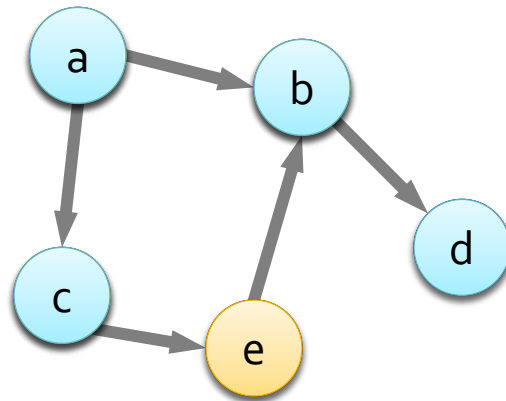
- But how does information **really** spread?



- We only see time of mention but not the edges
- Can we reconstruct (hidden) **diffusion network**?

Inferring the Diffusion Networks

- There is a **hidden** diffusion network:



- We only see **times** when nodes get “infected”:
 - c_1 : (a,1), (c,2), (b,3), (e,4)
 - c_2 : (c,1), (a,4), (b,5), (d,6)
- **Want to infer who-infects-whom network!**

Examples and Applications

	Virus propagation	Word of mouth & Viral marketing
Process	Viruses propagate through the network	Recommendations and influence propagate
We observe	We only observe when people get sick	We only observe when people buy products
It's hidden	But NOT who infected whom	But NOT who influenced whom

Can we infer the underlying network?

Plan for the NETINF Algorithm

- The plan for NETINF:
 - Define a continuous time model of propagation
 - Define the likelihood of the observed data given a graph
 - Show how to efficiently **compute** the likelihood
 - Show how to efficiently **optimize** the likelihood
 - Find a graph G that maximizes the likelihood
- Note:
 - There are super-exponentially many graphs, $O(N^{N*N})$
 - NETINF finds a near-optimal graph in $O(N^2)$!

Information Diffusion Model

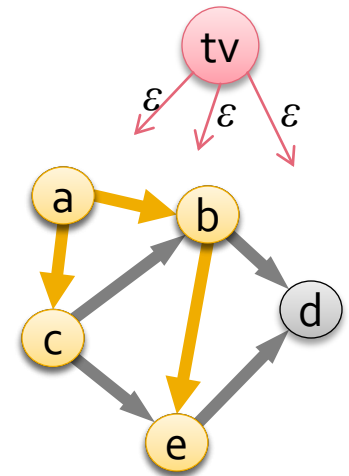
- Information Diffusion Model:
 - Cascade reaches node i at time t_i , and spreads to i 's neighbors j :

With prob. β cascade propagates along edge (i,j) and $t_j = t_i + \Delta$

- Transmission probability:

$$P_c(i,j) \propto P(t_j - t_i) \text{ if } t_j > t_i \text{ else } \varepsilon$$

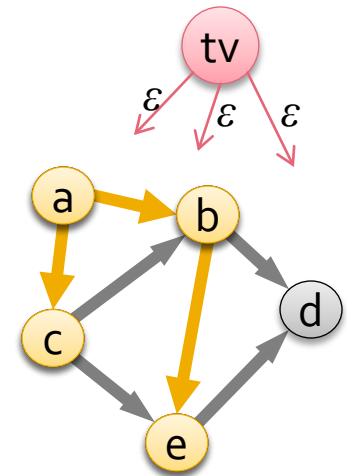
- ε captures influence external to the network
 - At any time a node can get infected from outside with small probability ε



Probability of a cascade tree

- Information Diffusion Model:
 - Cascade reaches node i at time t_i , and spreads to i 's neighbors j :

With prob. β cascade propagates along edge (i,j) and $t_j = t_i + \Delta$



- Transmission probability:

$$P_c(i,j) \propto P(t_j - t_i) \text{ if } t_j > t_i \text{ else } \varepsilon$$

- Prob. that cascade c propagates in a tree T

$$P(c|T) = \prod_{(u,v) \in E_T} \beta P_c(u,v) \prod_{u \in V_T, (u,x) \in E \setminus E_T} (1 - \beta)$$

Edges that "propagated"

Edges that failed to "propagate"

Probability of a cascade tree

- Cascade $c = \{(u, t_u), u \in V_T\}$ is defined by node infection times:

- $c = \{ (a,1), (c,2), (b,3), (e,4) \}$

- Prob. that cascade c propagates in a tree T

$$P(c|T) = \prod_{(u,v) \in E_T} \beta P_c(u,v) \prod_{u \in V_T, (u,x) \in E \setminus E_T} (1 - \beta)$$

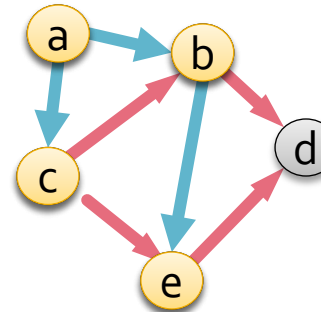
Edges that “propagated”

Edges that failed to “propagate”

- Note that 2nd term only depends on vertex set V_T of tree T (and not the edge set E_T):

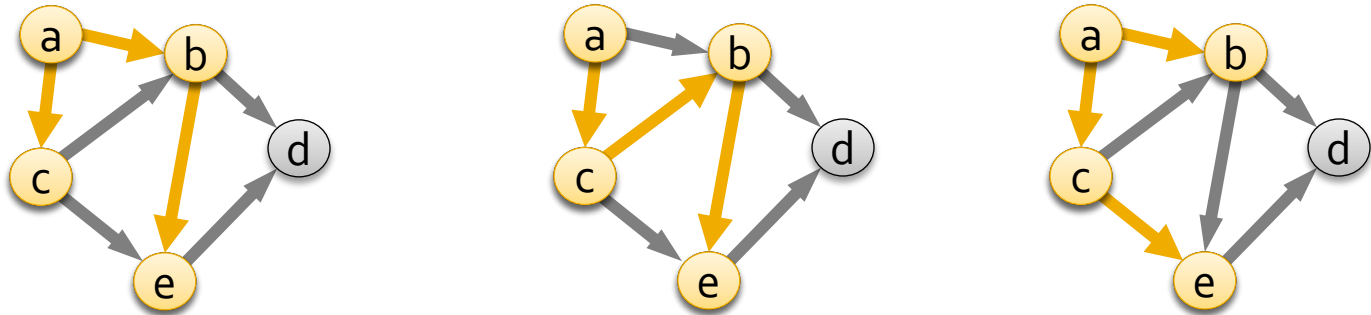
- Thus we approximate:

$$P(c|T) \approx \prod_{(u,v) \in E_T} P_c(v,u)$$



Complication: Too many trees

- There are many possible transmission trees:
 - $c = \{(a,1), (c,2), (b,3), (e,4)\}$



- Need to consider all possible directed spanning trees T supported by G :

$$P(c|G) = \sum_{T \in \mathcal{T}(G)} P(c|T)P(T|G) \propto \sum_{T \in \mathcal{T}(G)} \prod_{(u,v) \in E_T} P_c(v, u)$$

Probability of a Propagation Tree

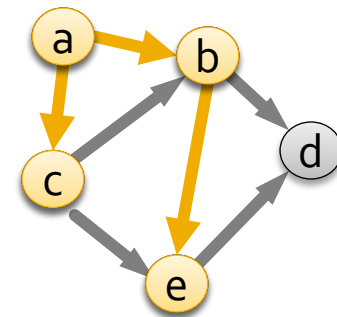
- Consider the **most likely** propagation tree
- Log-likelihood of item c in graph G :

$$F_c(G) = \max_T \log P(c|T)$$

- Log-likelihood of item c in graph G is C :

The problem is **NP-hard**:
MAX-k-COVER [KDD '10]
Our algorithm solve it
near-optimally in $O(N^2)$

$$G^* = \operatorname{argmax}_{|G| \leq k} F_C(G)$$



Submodularity to the Rescue

- Theorem: Log-likelihood $F_c(G)$ of item c is **monotonic**, and **submodular in edges of G :**
 - Let A, B be two graphs: same nodes, different edges: $A \subseteq B \subseteq V \times V$:

$$\underbrace{F_c(A \cup \{e\}) - F_c(A)}_{\text{Gain of adding an edge to a "small" graph}} \geq \underbrace{F_c(B \cup \{e\}) - F_c(B)}_{\text{Gain of adding an edge to a "large" graph}}$$

Gain of adding an edge to a "small" graph Gain of adding an edge to a "large" graph

- **Benefits:**
 - 1. Efficient (and simple) optimization algorithm
 - 2. Approximation guarantee (≈ 0.63 of OPT)
 - 3. Tight on-line bounds on the solution quality

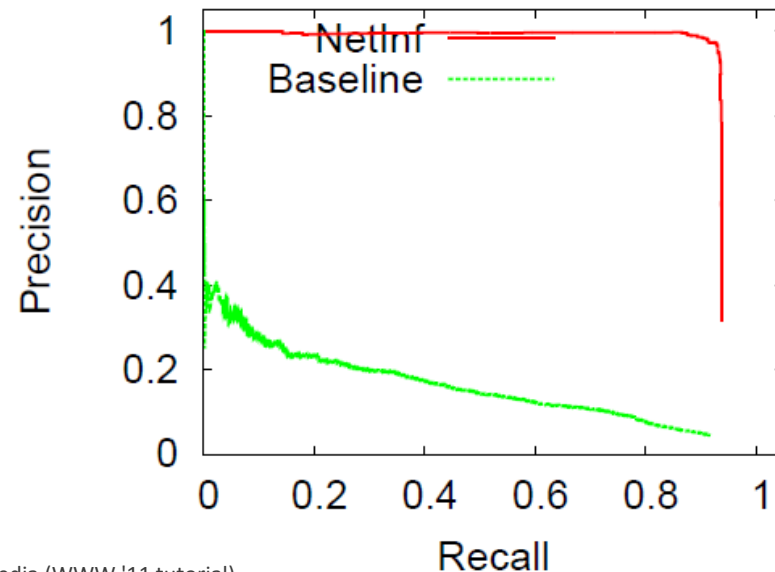
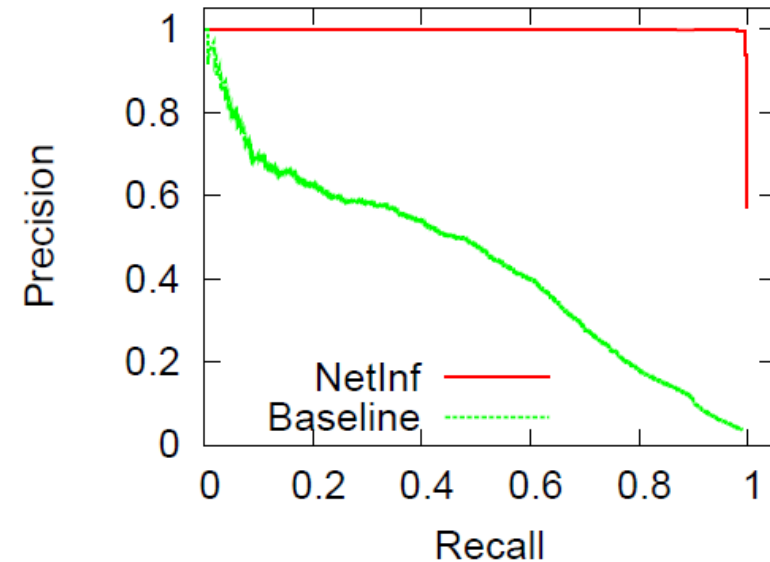
Submodularity to the Rescue

- **NetInf algorithm:** use **greedy hill-climbing** to maximize $F_C(G)$:
 - Start with empty G_0 (G with no edges)
 - Add k edges (k is parameter)
 - At every step add an **edge** to G_i that **maximizes the marginal improvement**

$$e_i = \operatorname{argmax}_{e \in G \setminus G_{i-1}} F_C(G_{i-1} \cup \{e\}) - F_C(G_{i-1})$$

Experimental Setup

- Synthetic data:
 - Take a graph G on k edges
 - Simulate info. diffusion
 - Record node infection times
 - Reconstruct G
- Evaluation:
 - How many edges of G can we find?
 - Break-even point: 0.95
 - Performance is independent of the structure of G !

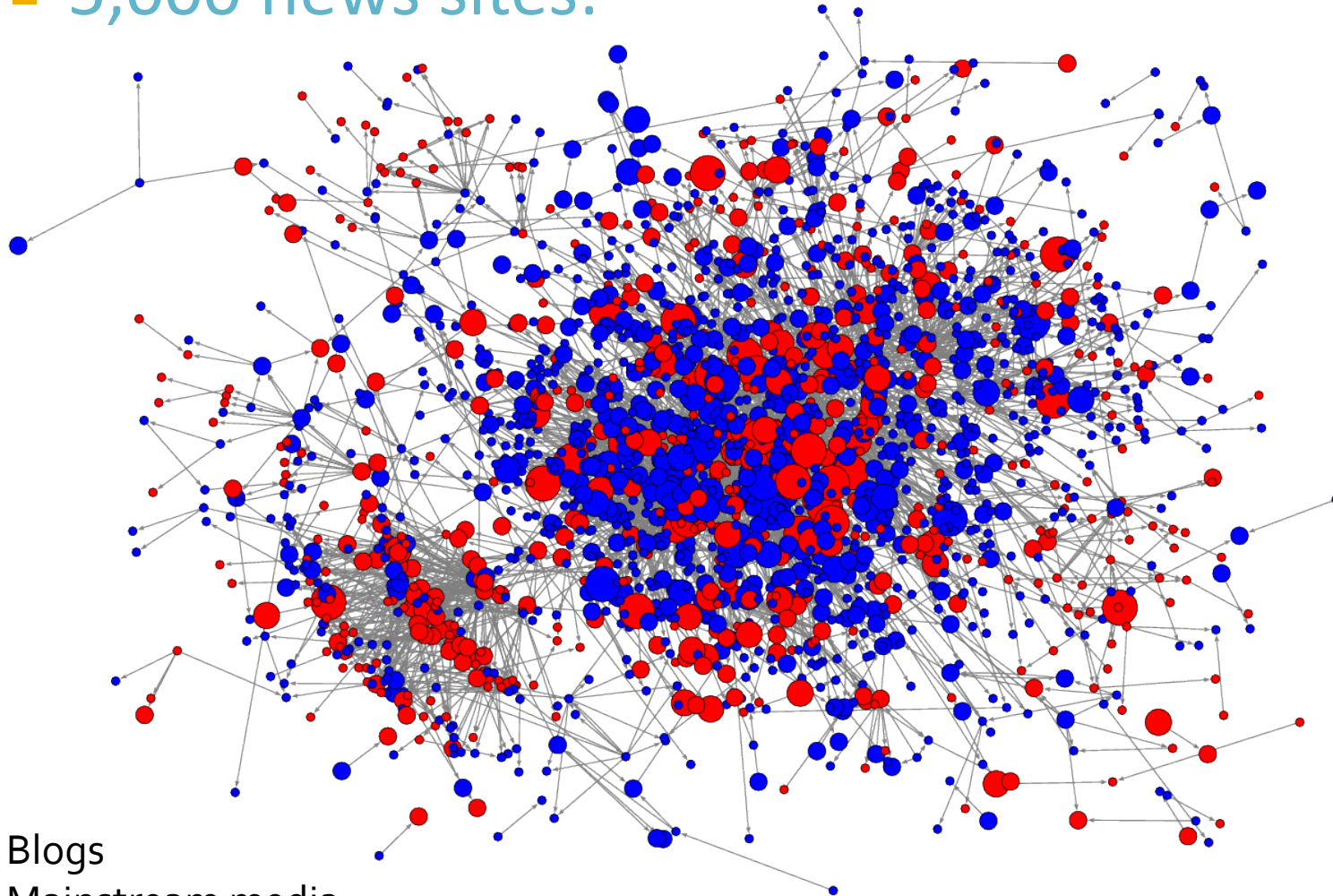


Experiments: Real Data

- Memetracker quotes:
 - 172 million news and blog articles
 - Aug '08 – Sept '09
 - Extract 343 million phrases
 - Record times $t_i(w)$ when site w mentions quote i
- Given times when sites mention quotes
- Infer the network of information diffusion:
 - Who tends to copy (repeat after) whom

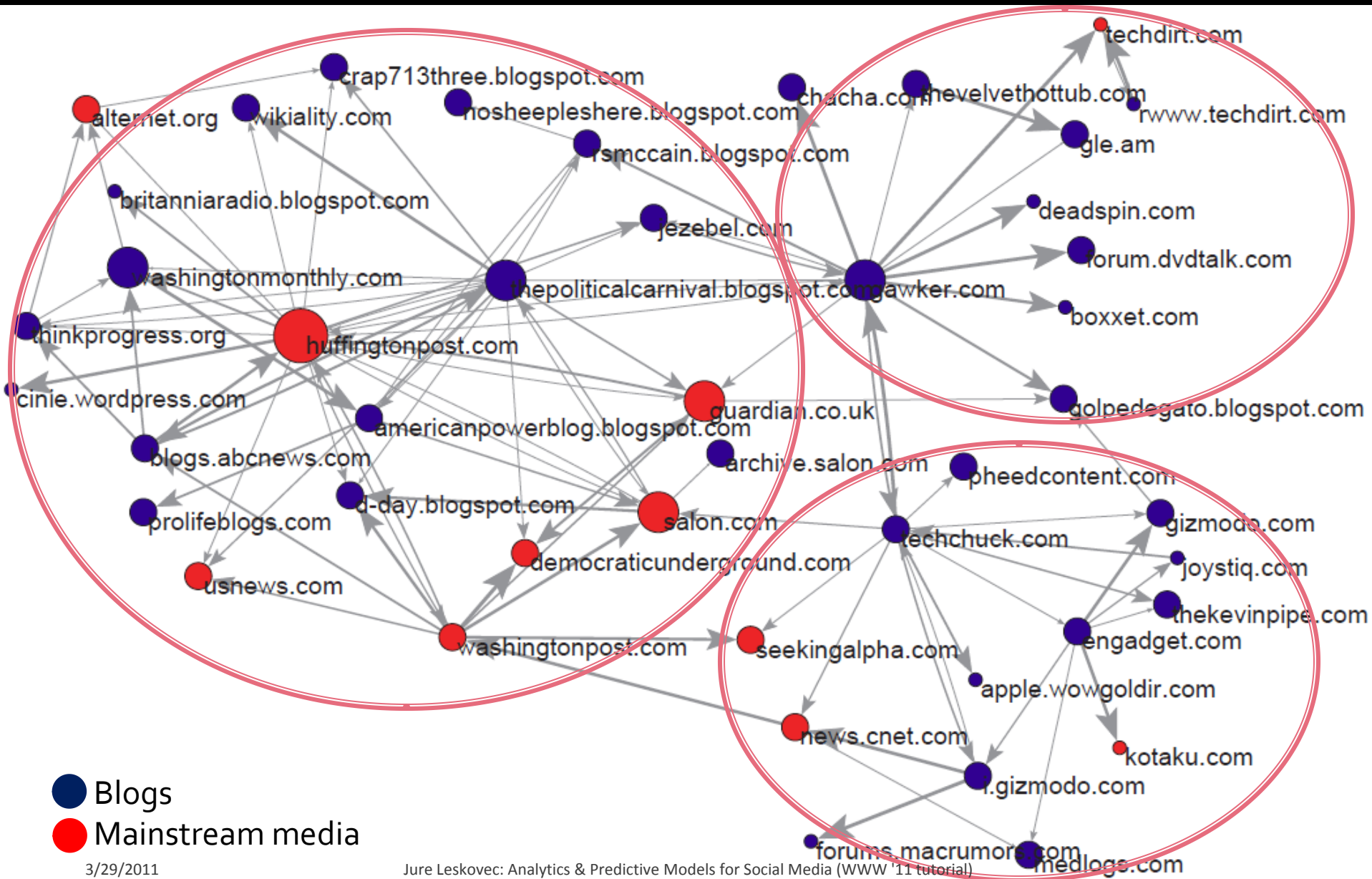
Diffusion Network

- 5,000 news sites:



- Blogs
- Mainstream media

Diffusion Network (small part)



Conclusions and Connections

- Messages arriving through networks from real-time sources requires new ways of thinking about information dynamics and consumption:
 - Tracking information through (implicit) networks
 - Quantify the dynamics of online media
 - Predict the diffusion of information
 - And infer networks of information diffusion

Further Qs: Opinion dynamics

- Can this analysis help identify dynamics of polarization [Adamic-Glance '05]?
- Connections to mutation of information:
 - How does attitude and sentiment change in different parts of the network?
 - How does information change in different parts of the network?

References

- *Meme-tracking and the Dynamics of the News Cycle*, by J. Leskovec, L. Backstrom, J. Kleinberg. KDD, 2009. <http://cs.stanford.edu/~jure/pubs/quotes-kdd09.pdf>
- *Patterns of Temporal Variation in Online Media* by J. Yang, J. Leskovec. ACM International Conference on Web Search and Data Mining (WSDM), 2011. <http://cs.stanford.edu/people/jure/pubs/memeshapes-wsdm11.pdf>
- *Modeling Information Diffusion in Implicit Networks* by J. Yang, J. Leskovec. IEEE International Conference On Data Mining (ICDM), 2010. <http://cs.stanford.edu/people/jure/pubs/lim-icdm10.pdf>
- *Inferring Networks of Diffusion and Influence* by M. Gomez-Rodriguez, J. Leskovec, A. Krause. KDD, 2010. <http://cs.stanford.edu/~jure/pubs/netinf-kdd2010.pdf>
- *On the Convexity of Latent Social Network Inference* by S. A. Myers, J. Leskovec. Neural Information Processing Systems (NIPS), 2010. <http://cs.stanford.edu/people/jure/pubs/connie-nips10.pdf>
- *Cost-effective Outbreak Detection in Networks* by J. Leskovec, A. Krause, C. Guestrin, C. Faloutsos, J. VanBriesen, N. Glance. KDD 2007. <http://cs.stanford.edu/~jure/pubs/detect-kdd07.pdf>
- *Cascading Behavior in Large Blog Graphs* by J. Leskovec, M. McGlohon, C. Faloutsos, N. Glance, M. Hurst. SDM, 2007. <http://cs.stanford.edu/~jure/pubs/blogs-sdm07.pdf>