# Influence Propagation in Social Networks: a Data Mining Perspective

Francesco Bonchi
Yahoo! Research
Barcelona - Spain
bonchi@yahoo-inc.com

http://francescobonchi.com/

# Acknowledgments

Amit Goyal (University of British Columbia, Vancouver, Canada)

Laks V.S. Lakshmanan (University of British Columbia, Vancouver, Canada)

Michael Mathioudakis (University of Toronto, Canada)

Yahoo! Research Barcelona Web Mining group:

Antti Ukkonen

Ilaria Bordino

Aris Gionis

Carlos Castillo

Ingmar Weber

# Overview

## Part 1: Background

Social influence

WOMM, Viral marketing

Influence Maximization

Prior art

Missing pieces and open questions

## Part 2: Adding some pieces

Learning Influence Probability

Sparsifying Influence Networks

## Part 3: Direct Mining

Credit Distribution model

Leaders-and-Tribes model

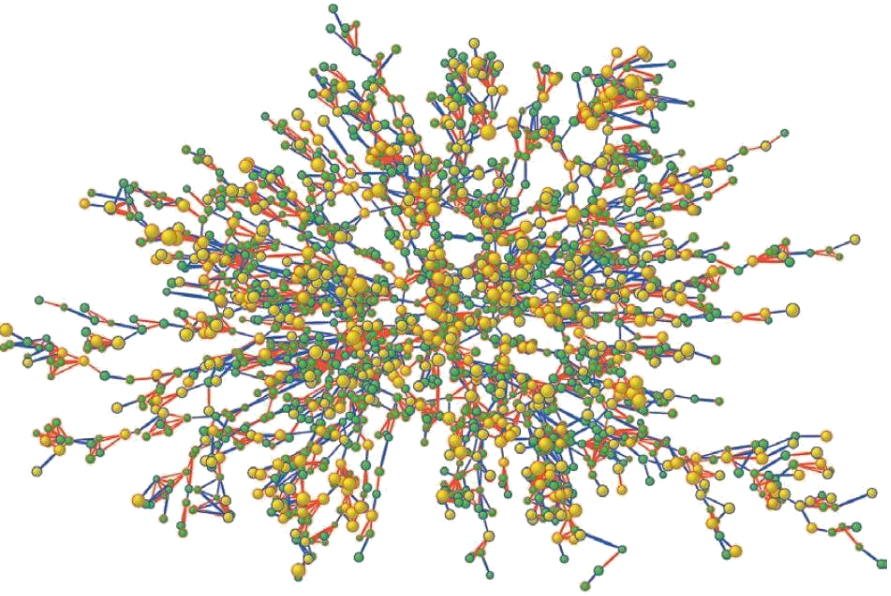
## Summary and conclusions

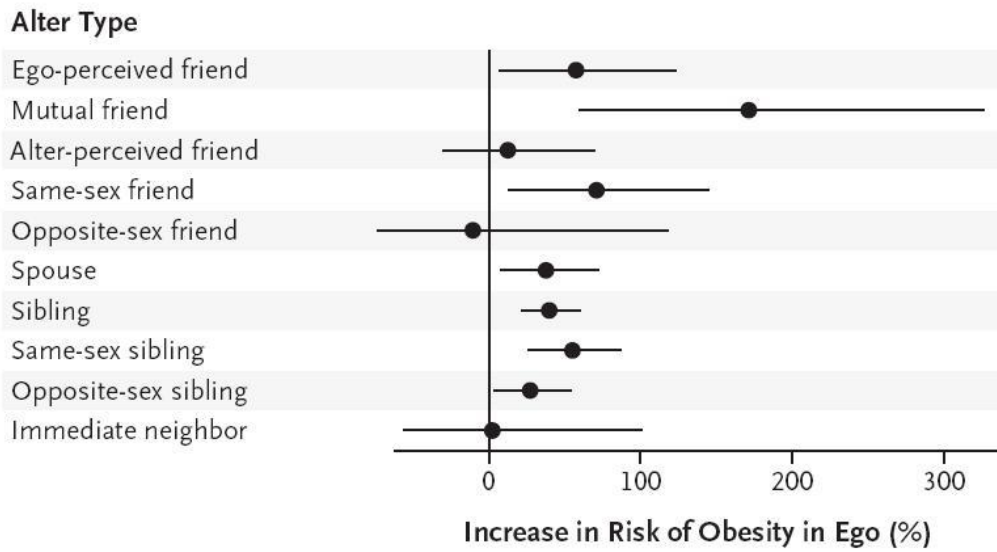
RP# denote interesting, (more or less) open "Research Problems"

# The Spread of Obesity in a Large Social Network over 32 Years

Christakis and Fowler, New England Journal of Medicine, 2007

Data set: 12,067 people from 1971 to 2003, 50K links






Obese Friend → 57% increase in chances of obesity

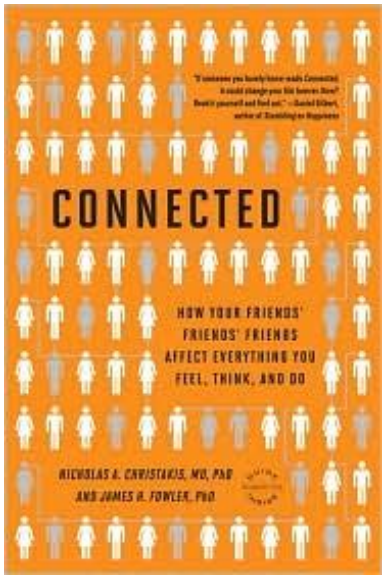
Obese Sibling → 40% increase in chances of obesity

Obese Spouse → 37% increase in chances of obesity

# Network contagion: other examples

How your friends' friends' friends' affect everything you feel, think, and do

Christakis and Fowler




back pain (spread from West Germany to East Germany after the fall of the Berlin Wall)
suicide (well known to spread throughout communities on occasion)
sex practices (such as the growing prevalence of oral sex among teenagers)
politics (the denser your network of connections, the more ideologically intense your beliefs)

# Influence or Homophily?

## Homophily

tendency to stay together with people similar to you

*"Birds of a feather flock together"*

E.g., I'm overweight → I date overweight girls

---

## Social influence

a force that person A (i.e., the influencer) exerts on person B to introduce a change of the behavior and/or opinion of B

Influence is a causal process

E.g., my girlfriend gains weight → I gain weight too

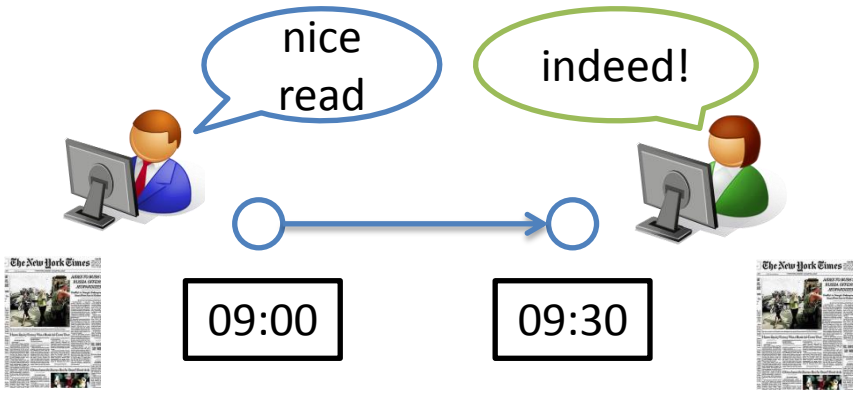
RP#1: How to distinguish social influence from homophily and other external factors

See e.g.,

Crandall et al. (KDD'08) *"Feedback Effects between Similarity and Social Influence in Online Communities"*

Anagnostopoulos et al. (KDD'08) *"Influence and correlation in social networks"*

# Influence in on-line social networks




users perform actions

post messages, pictures, video

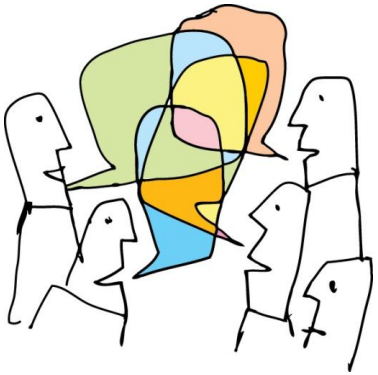
buy, comment, link, rate, share, like, retweet

users are connected with other users

interact, influence each other

actions propagate

# Social Influence Marketing
# Viral Marketing
# WOMM



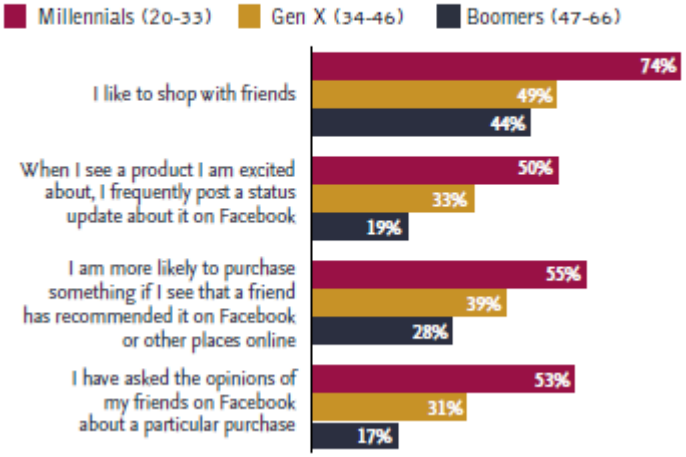
IDEA: exploit social influence for marketing

Basic assumption: word-of-mouth effect, thanks to which actions, opinions, buying behaviors, innovations and so on, propagate in a social network.

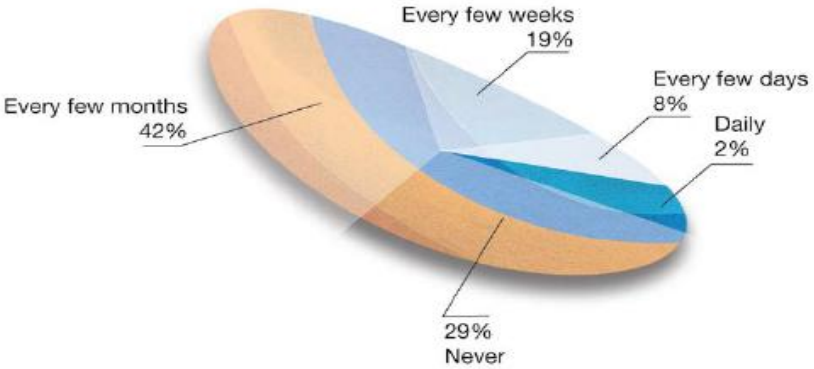
Target users who are likely to invoke word-of-mouth diffusion, thus leading to additional reach, clicks, conversions, or brand awareness

Target the influencers

Sharing and social influence




How frequently do you share recommendations online?



# Bring me the influencers!

Influencers increase brand awareness/product conversion through WOMM

Influencers advocate brand

Influencers influence a purchasing action from their peers


## Some of the many startups involved in social influence

Klout (http://klout.com)

Measure of overall influence online (mostly twitter, now FB and linkedin)

Score = function of true reach, amplification probability and network influence

Claims score to be highly correlated to clicks, comments and retweets

Peer Index (http://www.peerindex.net)

Identifies/Scores authorities on the social web by topic

SocialMatica (http://www.socialmatica.com)

Ranks 32M people by vertical/topic, claims to take into account quality of authored content

Influencer50 (http://www.influencer50.com)

Clients: IBM, Microsoft, SAP, Oracle and a long list of tech companies

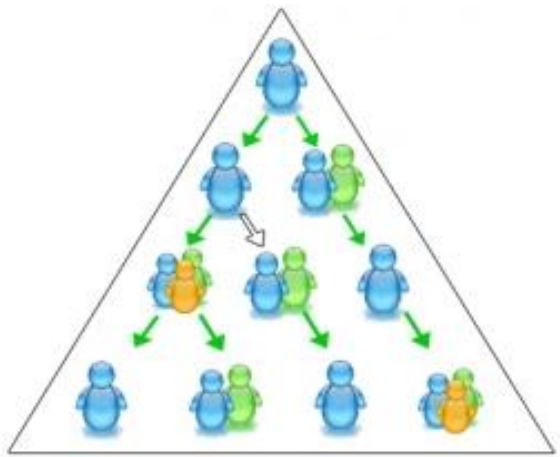
Svnetwork, Bluecalypso, CrowdBooster, Sproutsocial,TwentyFeet, EmpireAvenue, Twitaholic

(there's more … )

# Viral Marketing and Influence Maximization

Business goal (Viral Marketing): exploit the "word-of-mouth" effect in a social network to achieve marketing objectives through self-replicating viral processes

Mining problem statement (Influence Maximization): find a seed-set of influential people such that by targeting them we maximize the spread of viral propagations



Hot topic in Data Mining research since 10 years:

Domingos and Richardson *"Mining the network value of customers"* (KDD'01)

Domingos and Richardson *"Mining knowledge-sharing sites for viral marketing"* (KDD'02)

Kempe et al. *"Maximizing the spread of influence through a social network"* (KDD'03)

# Not only marketing

Information propagation

Social media analytics

Spread of rumors

Interest, trust, referral

Innovation adoption

Epidemics

Feed ranking

Expert finding

“Friends” recommendation

Social recommendation

Social search

# Influence Maximization Problem

following Kempe et al. (KDD'03) *"Maximizing the spread of influence through a social network"*

Given a propagation model *M*, define influence of node set *S*,
$\sigma_M(S)$ = expected size of propagation, if *S* is the initial set of active nodes

Problem: Given social network *G* with arcs probabilities/weights, budget k, find k-node set *S* that maximizes $\sigma_M(S)$

Two major propagation models considered:

independent cascade (IC) model

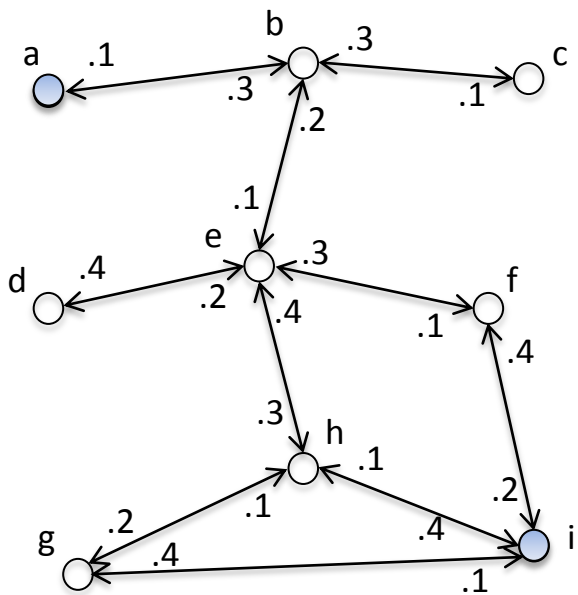
linear threshold (LT) model

# Independent Cascade Model (IC)

Every arc *(u,v)* has associated the probability *p(u,v)* of *u* influencing *v*

Time proceeds in discrete steps

At time t, nodes that became active at t-1 try to activate their inactive neighbors, and succeed according to *p(u,v)*

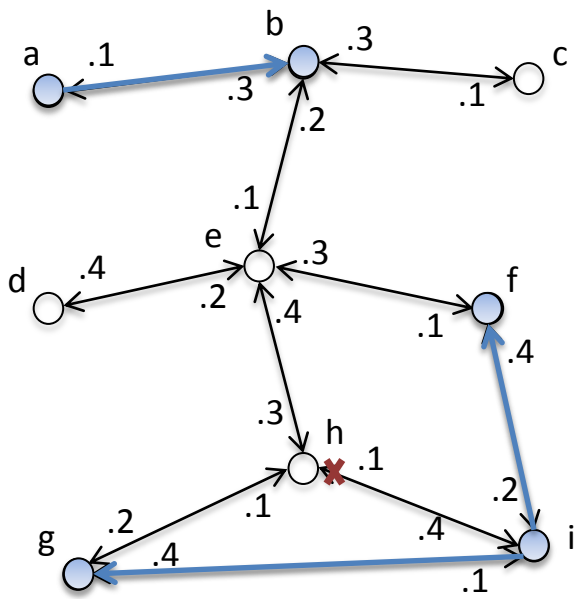


# Independent Cascade Model (IC)

Every arc *(u,v)* has associated the probability *p(u,v)* of *u* influencing *v*

Time proceeds in discrete steps

At time t, nodes that became active at t-1 try to activate their inactive neighbors, and succeed according to *p(u,v)*

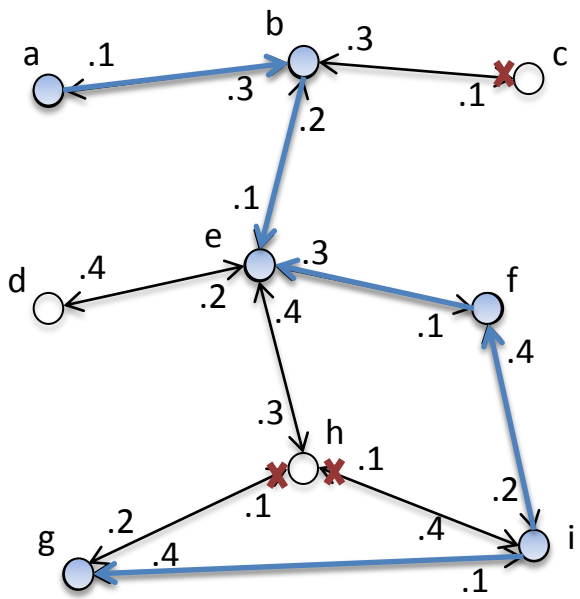



# Independent Cascade Model (IC)

Every arc *(u,v)* has associated the probability *p(u,v)* of *u* influencing *v*

Time proceeds in discrete steps

At time t, nodes that became active at t-1 try to activate their inactive neighbors, and succeed according to *p(u,v)*

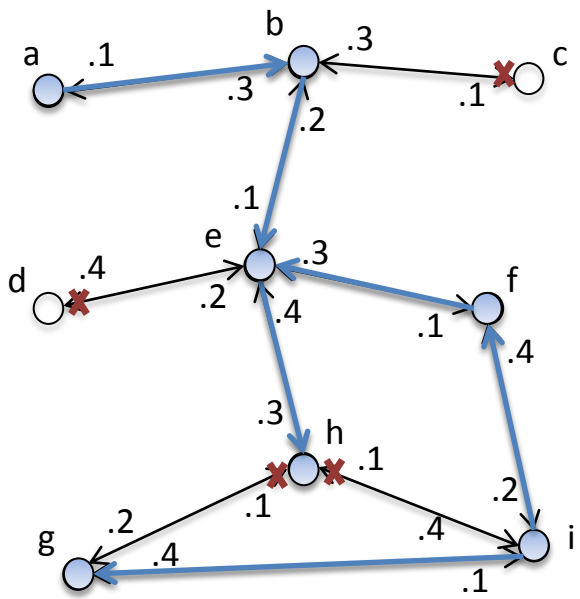



# Independent Cascade Model (IC)

Every arc *(u,v)* has associated the probability *p(u,v)* of *u* influencing *v*

Time proceeds in discrete steps

At time t, nodes that became active at t-1 try to activate their inactive neighbors, and succeed according to *p(u,v)*



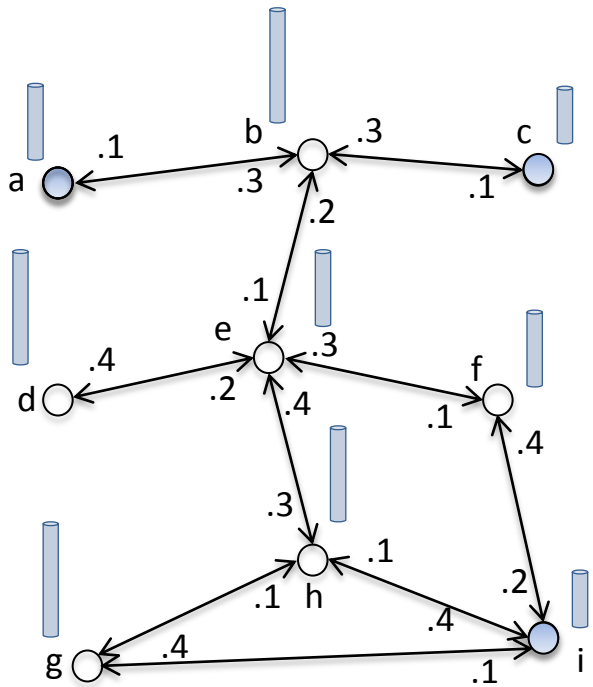

# Linear Threshold Model (LT)

Every arc *(u,v)* has associated a weight *b(u,v)* such that the sum of incoming weights in each node is $\leq 1$

Time proceeds in discrete steps

Each node *v* picks a random threshold $\vartheta_v \sim U[0,1]$

A node *v* becomes active when the sum of incoming weights from active neighbors reaches $\vartheta_v$



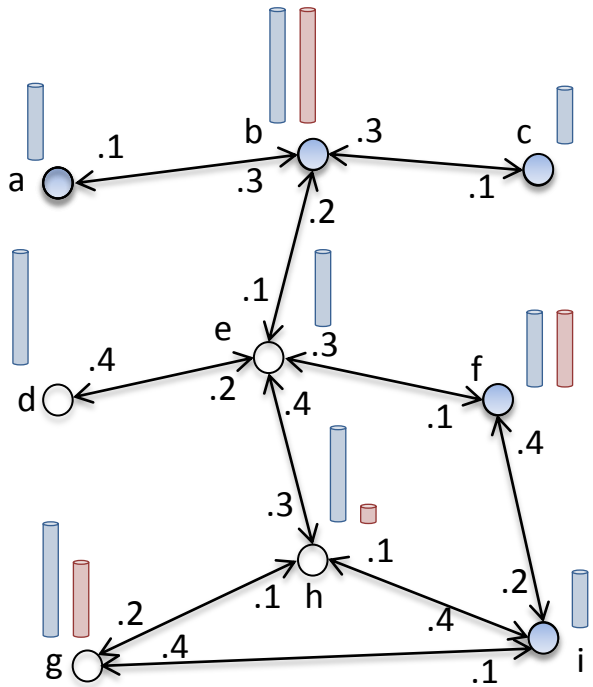

# Linear Threshold Model (LT)

Every arc *(u,v)* has associated a weight *b(u,v)* such that the sum of incoming weights in each node is $\leq 1$

Time proceeds in discrete steps

Each node *v* picks a random threshold $\vartheta_v \sim U[0,1]$

A node *v* becomes active when the sum of incoming weights from active neighbors reaches $\vartheta_v$



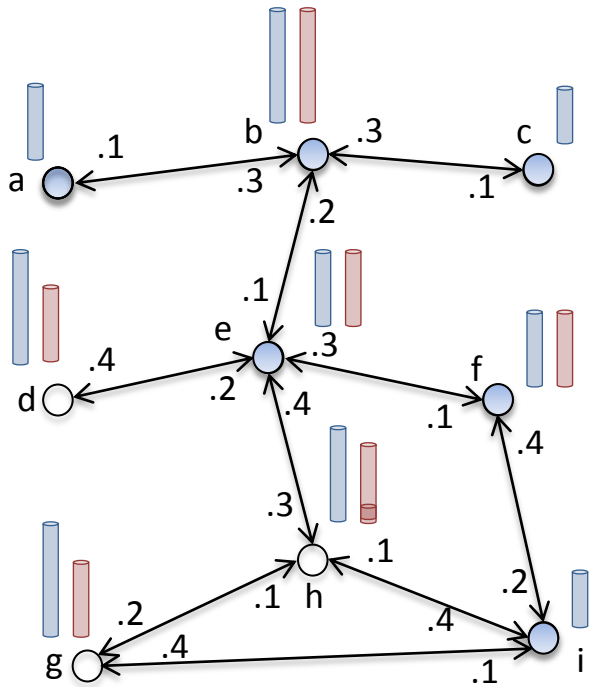
# Linear Threshold Model (LT)

Every arc *(u,v)* has associated a weight *b(u,v)* such that the sum of incoming weights in each node is $\leq 1$

Time proceeds in discrete steps

Each node *v* picks a random threshold $\vartheta_v \sim U[0,1]$

A node *v* becomes active when the sum of incoming weights from active neighbors reaches $\vartheta_v$



# Known Results

Bad news: NP-hard optimization problem for both IC and LT models

Good news: we can use Greedy algorithm

**Algorithm 1** Greedy

**Input:** $G, k, \sigma_m$

**Output:** seed set $S$

1: $S \leftarrow \emptyset$

2: **while** $|S| < k$ **do**

3: select $u = \arg\max_{w \in V \setminus S} (\sigma_m(S \cup \{w\}) - \sigma_m(S))$

4: $S \leftarrow S \cup \{u\}$

$\sigma_M(S)$ is monotone and submodular

Theorem*: The resulting set $S$ activates at least (1- 1/e) > 63% of the number of nodes that any size-k set could activate

Bad news: computing $\sigma_M(S)$ is #P-hard under both IC and LT models

step 3 of the Greedy Algorithm above can only be approximated by MC simulations

*Nemhauser et al. "An analysis of approximations for maximizing submodular set functions – (i)" (1978)

# Influence Maximization: prior art

Much work has been done following Kempe et al. mostly devoted to heuristichs to improve the efficiency of the Greedy algorithm:

E.g.,

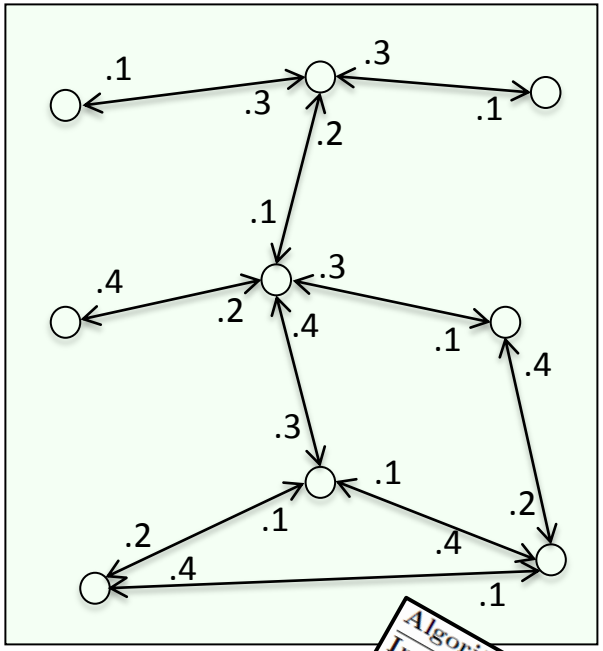
Kimura and Saito (PKDD'06) *"Tractable models for information diffusion in social networks"*

Leskovec et al. (KDD'07) *"Cost-effective outbreak detection in networks"*

Chen et al. (KDD'09) *"Efficient influence maximization in social networks"*

Chen et al. (KDD'10) *"Scalable influence maximization for prevalent viral marketing in large-scale social networks"*

Chen et al. (ICDM'10) *"Scalable influence maximization in social networks under the linear threshold model"*




**Algorithm 1** Greedy

**Input:** $G, k, \sigma_m$

**Output:** seed set $S$

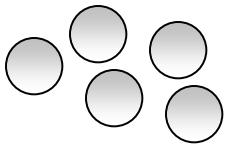
$S \leftarrow \emptyset$

**while** $|S| < k$ **do**

select $u = \arg\max_{w \in V \setminus S}$

$S \leftarrow S \cup \{u\}$

Seed set



RP#2: scalability of the Influence Maximization framework

RP#3: how likely is Viral Marketing to be successful in the real-world?

# Missing pieces and open questions
(tackled in Part 2)

## Where do influence probabilities come from?

Real world social networks don't have probabilities!

How can we learn those probabilities from available propagations data?

## How important is to accurately learn the probabilities?

What is the relative importance of the graph structure and the edge probabilities in the influence maximization problem?

## Does influence probability change over time?

Yes!

How can we take time into account?

Can we predict the time at which user is most likely to perform an action?

## Do we really have to use the whole social graph?

# More missing pieces
## (tackled in Part 3)

Influence maximization based on MC simulations + learning the influence probabilities is computationally expensive.

Can we avoid the costly learning + simulation approach?

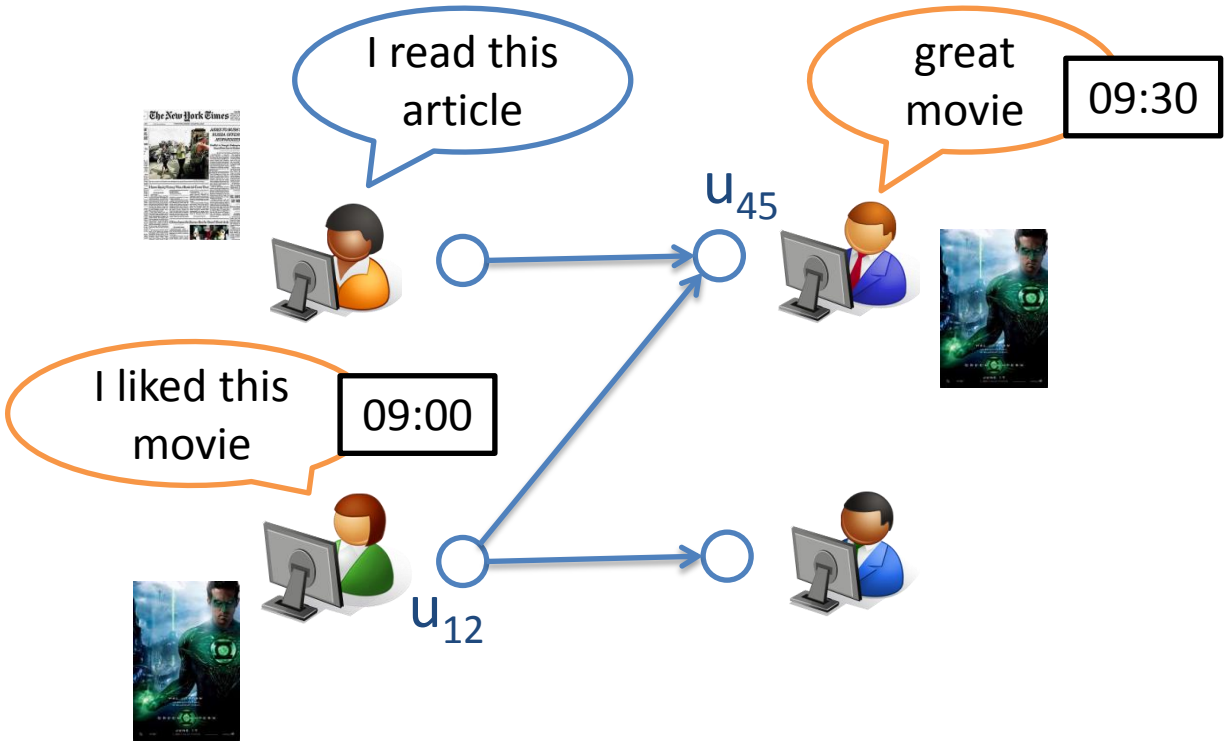
instead directly mine the available propagation traces to build a model of influence spread for any given seed set?

How can we do this efficiently?

# Part 2:
# Learning Influence Probability
# Sparsifying Influence Networks

# Data! Data! Data!

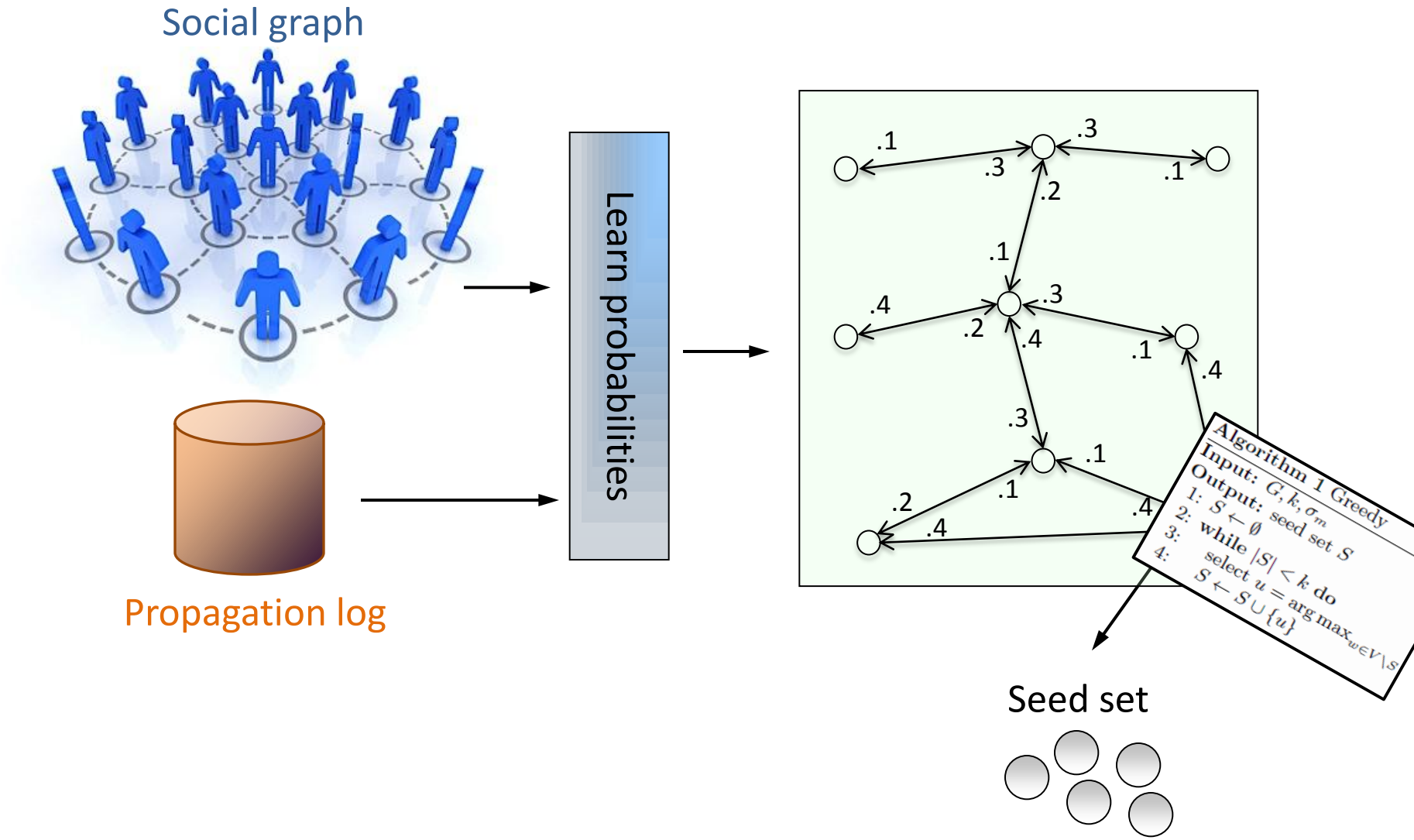
We have 2 pieces of input data:
(1) social graph and (2) a log of past propagations



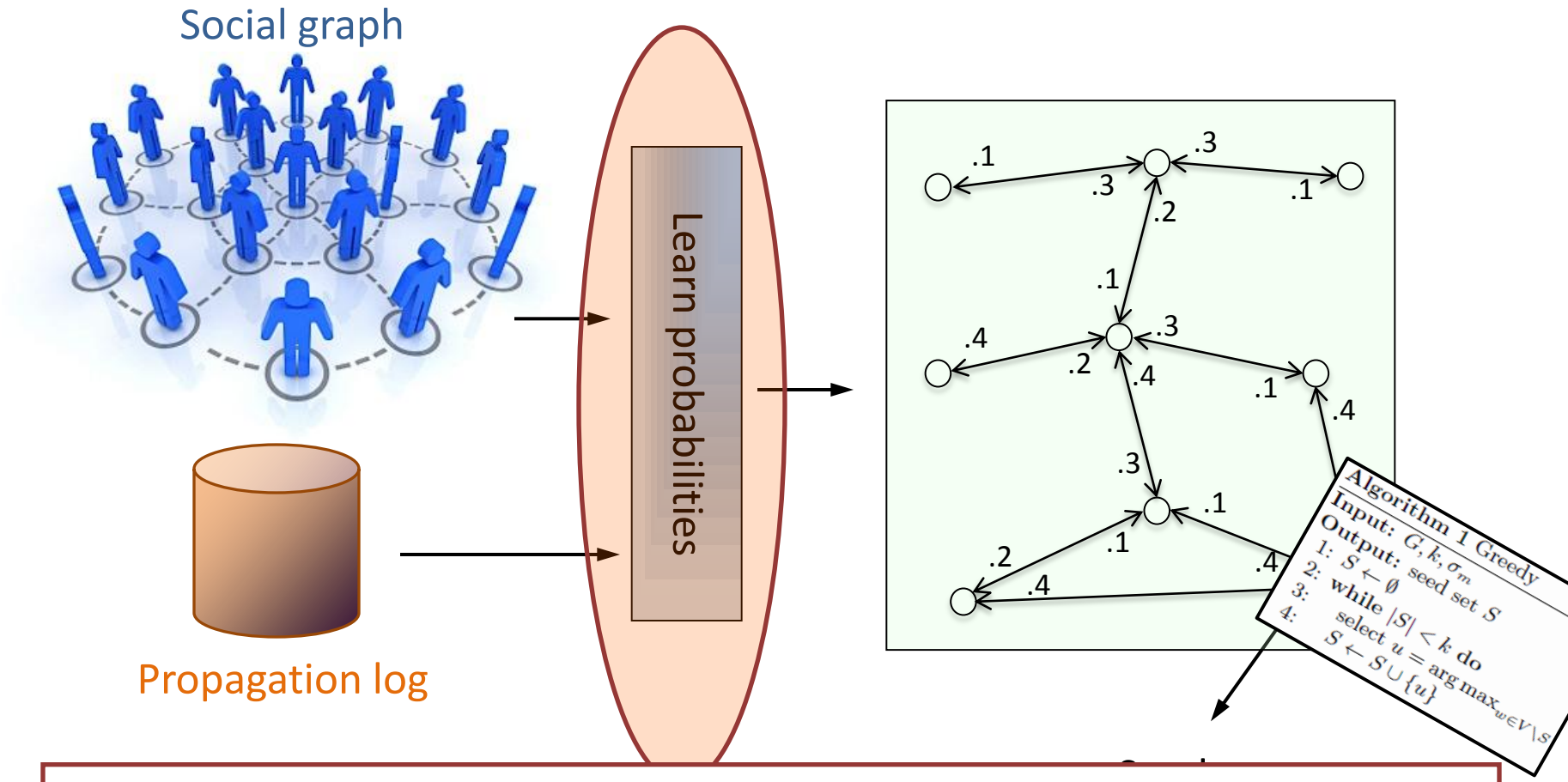

| Action | Node | Time |
|---|---|---|
| a | $u_{12}$ | 1 |
| a | $u_{45}$ | 2 |
| a | $u_{32}$ | 3 |
| a | $u_{76}$ | 8 |
| b | $u_{32}$ | 1 |
| b | $u_{45}$ | 3 |
| b | $u_{98}$ | 7 |

$u_{45}$ follows $u_{12}$ – arc $u_{12} \rightarrow u_{45}$

# The general picture



# The general picture



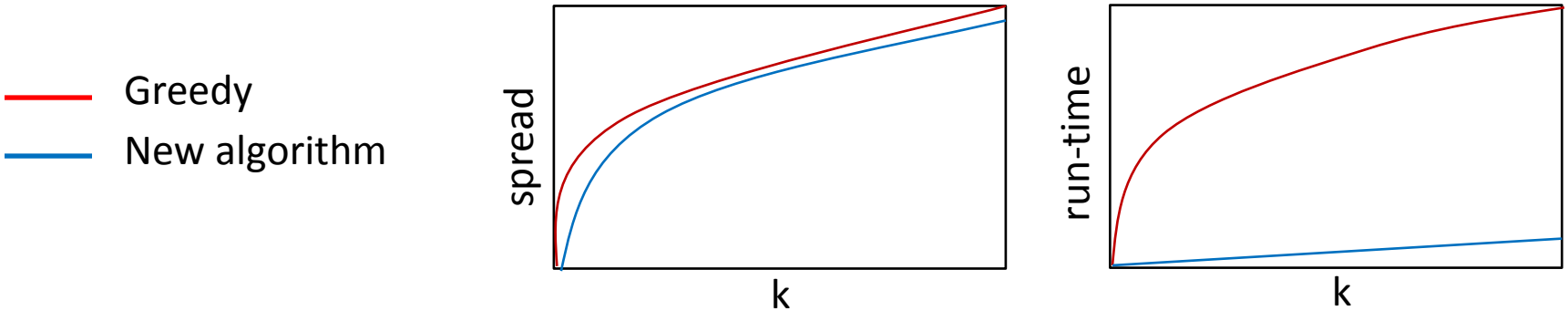

- Saito, Nakano, and Kimura (KES'08) *"Prediction of information diffusion probabilities for independent cascade model"* → IC model
- Goyal, Bonchi, Lakshmanan (WSDM'10) *"Learning influence probabilities in social networks"* → General threshold model + **Time**

# Prior art typical experimental assessment

Assuming IC (or LT) model,

compare the influence spread achieved by seed sets selected by different algorithms

Spread computed by means of IC (or LT) propagation simulations (lack of ground truth!)




<u>Using simple methods of assigning probabilities:</u>

**WC** (weighted cascade) p(u,v) = 1/in_degree(v)

**TV** (trivalency) selected uniformly at random from the set *{0.1, 0.01, 0.001}*

**UN** (uniform) all edges have same probability (e.g. *p = 0.01)*

# Why learning from data matters – experiments*

- Methods compared (IC model):
  - WC, TV, UN (no learning)
  - EM (learned from real data – Expectation Maximization method)
  - PT (learned than perturbed $\pm$ 20%)

- Data:
  - 2 real-world datasets (with social graph + propagation log): Flixster and Flickr
  - On Flixster, we consider "rating a movie" as an action
  - On Flickr, we consider "joining a group" as an action
  - Split the data in training and test sets – 80:20

- Compare the different ways of assigning probabilities:
  1. Seed sets intersection
  2. Given a seed set, we ask to the model to predict its spread (ground truth on the test set)

* Goyal, Bonchi, Lakshmanan (VLDB'12) *"A Data-Based Approach to Social Influence Maximization"*

# Why learning from data matters – experiments*

1. <u>Seed sets intersection (*k = 50*)</u>

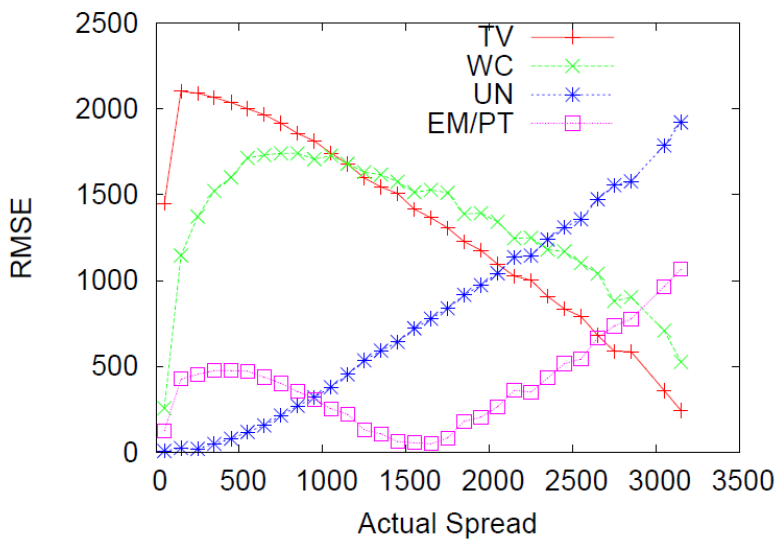
| | UN | WC | TV | EM | PT |
|---|---|---|---|---|---|
| UN | 50 | 25 | 5 | 6 | 6 |
| WC | | 50 | 9 | 3 | 2 |
| TV | | | 50 | 3 | 2 |
| EM | | | | 50 | 44 |
| PT | | | | | 50 |

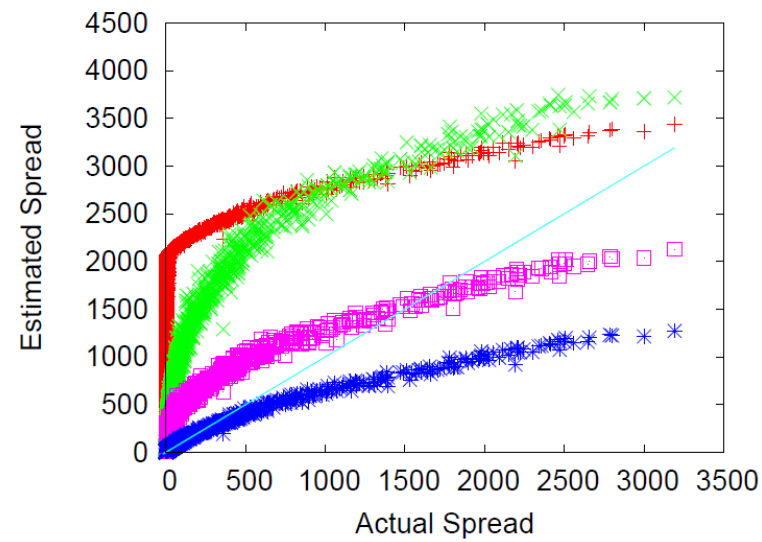
FLIXSTER_SMALL

| | PT | EM | TV | WC | UN |
|---|---|---|---|---|---|
| UN | 0 | 0 | 44 | 19 | 50 |
| WC | 0 | 0 | 17 | 50 | |
| TV | 0 | 0 | 50 | | |
| EM | 44 | 50 | | | |
| PT | 50 | | | | |

FLICKR_SMALL

2. <u>Given a seed set, we ask to the IC model to predict its spread (on the test set)</u>




* Goyal, Bonchi, Lakshmanan (VLDB'12) *"A Data-Based Approach to Social Influence Maximization"*

# Learning influence probabilities*

Propose several models of influence probability

in the context of General Threshold model + time

consistent with IC and LT models

Models able to predict whether a user will perform an action or not

predict the time at which she will perform it

Introduce metrics of user and action influenceability

high values $\rightarrow$ genuine influence

Develop efficient algorithms to learn the parameters of the models

minimize the number of scans over the propagation log

Incrementality property

* Goyal, Bonchi, Lakshmanan (WSDM'10) *"Learning Influence Probabilities In Social Networks"*

# Influence models

<u>Static Models:</u> probabilities are static and do not change over time.

Bernoulli: $p_{vu} = \frac{A_{v2u}}{A_v}$ Jaccard: $p_{vu} = \frac{A_{v2u}}{A_{v|u}}$

<u>Continuous Time (CT) Models:</u> probabilities decay exponentially in time

$$p_{uv}^{t} = p_{uv}^{0} exp\left(-\frac{t - t_v}{\tau_{uv}}\right)$$

Not incremental, hence very expensive to apply on large datasets.

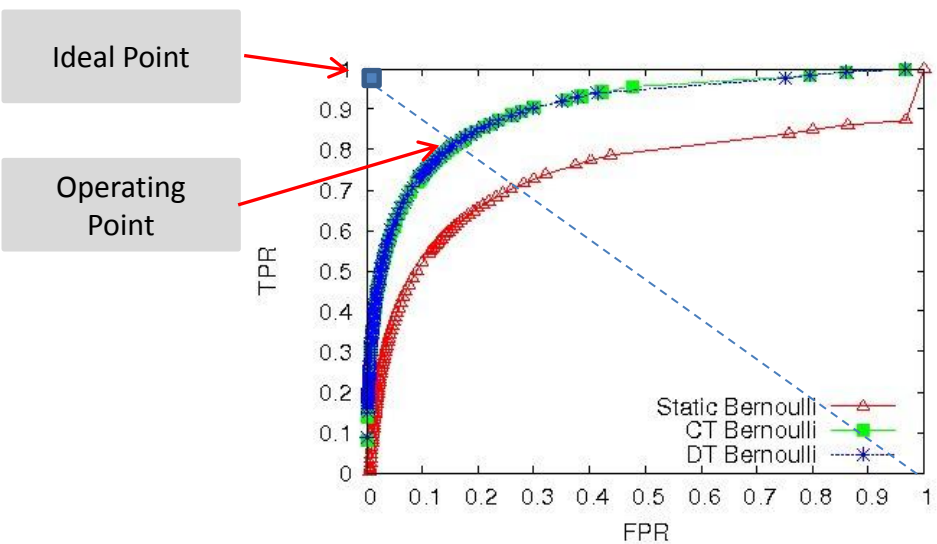
<u>Discrete Time (CT) Models:</u> Active neighbor *u* of *v* remains contagious in [t, t+ $\tau$(u,v)], has constant influence prob *p(u,v)* in the interval and 0 outside.

Monotone, submodular, and incremental!

# Evaluation

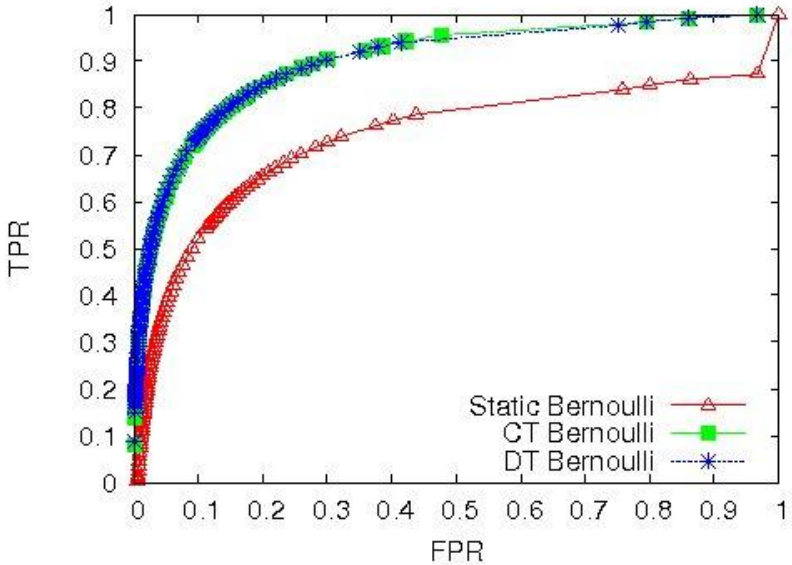
- Sample of a Flickr dataset (#users ~ 1.3 million #edges ~ 40.4 million)
- "Joining a group" is considered as action
- #tuples in action log ~ 35.8 million
- split the action data into training (80%) and testing (20%)

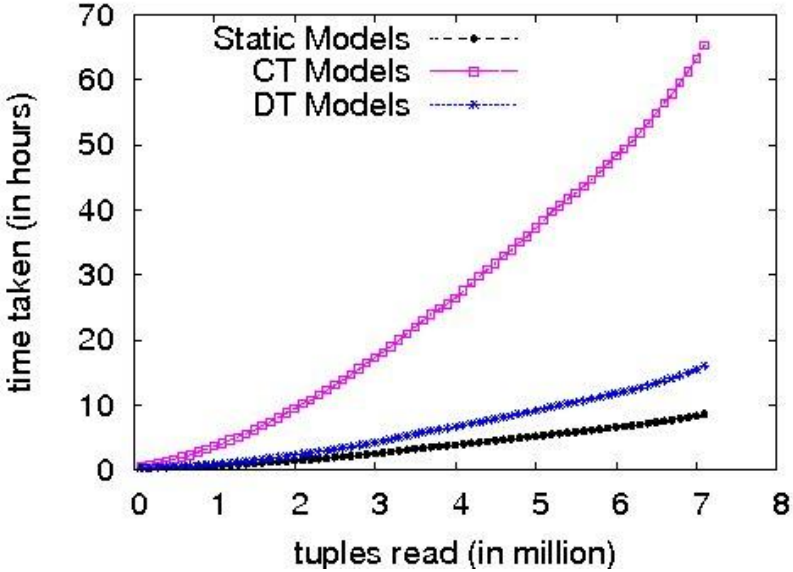
- We ask the model to predict whether user will become active or not, given all the neighbors who are active
  - Binary Classification (ROC curve)




| | | Reality | |
|---|---|---|---|
| | | Active | Inactive |
| Prediction | Active | TP | FP |
| | Inactive | FN | TN |
| | Total | P | N |

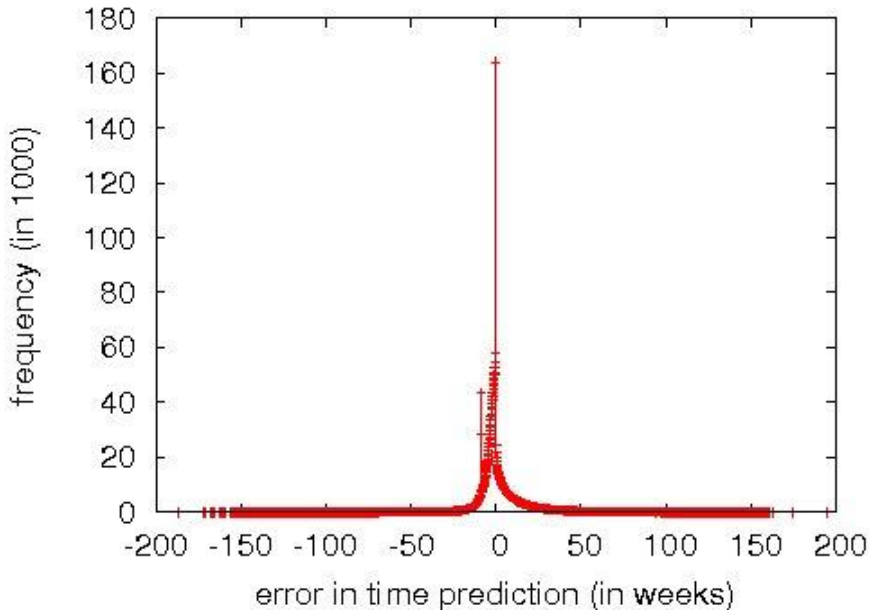
# Comparison of Static, CT and DT models




- Time conscious models are better than the static model
- CT and DT models perform equally well
- Static and DT models are far more efficient compared to CT models because of their incremental nature

# Predicting Time – Distribution of Error




For TP cases

X-axis: error in predicting time (in weeks)

Y-axis: frequency of that error

Most of the time, error in the prediction is very small

# Sparsification of Influence Networks*

which connections are most important
for the propagation of actions?

keep only important connections

data reduction
visualization
clustering
efficient graph analysis
find the backbone of influence networks

*Mathioudakis, Bonchi, Castillo, Gionis, Ukkonen (KDD'11) "Sparsification of Influence Networks"

# Sparsification

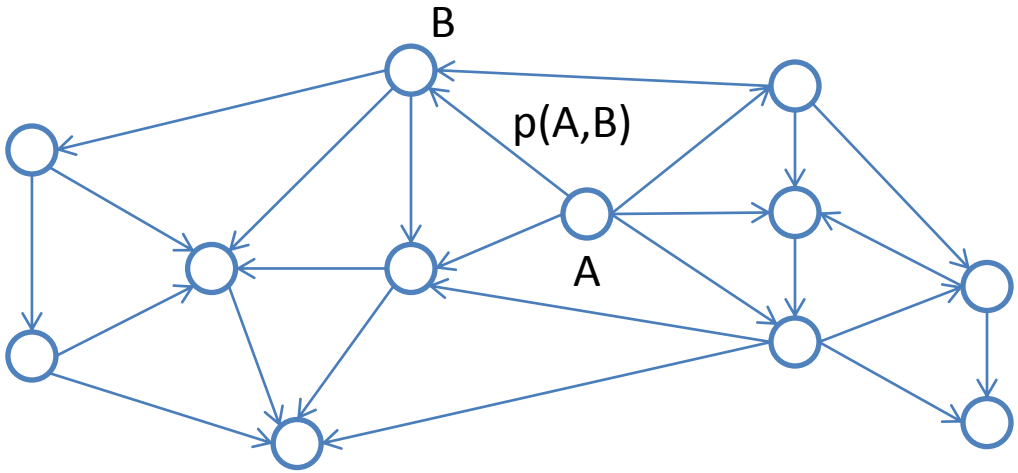
social network

set of propagations

p(A,B)

k arcs

most likely to explain propagations



# Sparsification

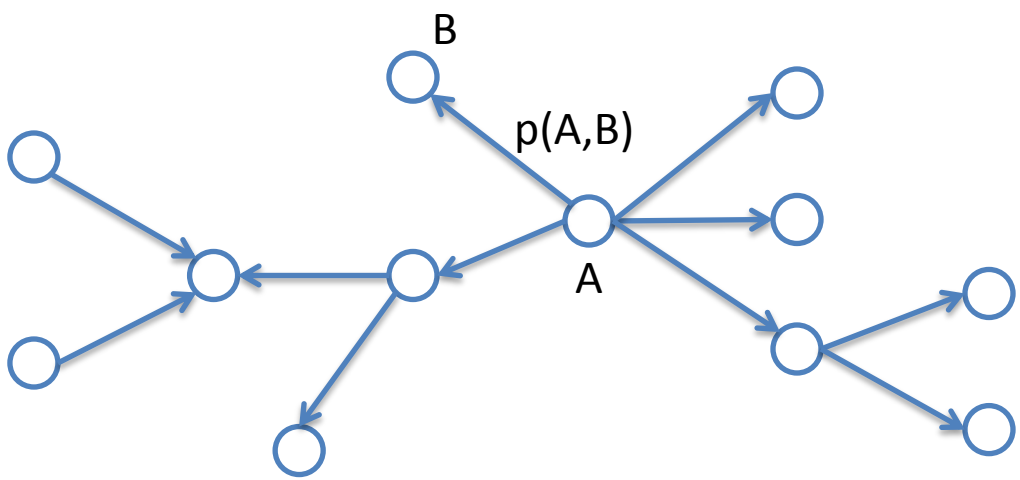
social network

set of propagations

p(A,B)

k arcs

most likely to explain propagations



# Solution

not the k arcs with largest probabilities!

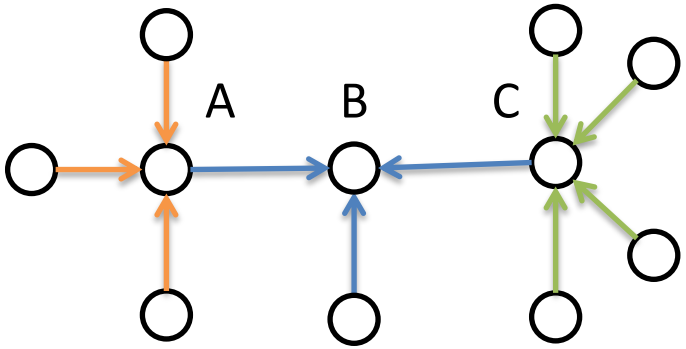
problem is NP-hard and inapproximable

sparsify separately incoming arcs of individual nodes

optimize corresponding likelihood

dynamic programming

optimal solution




$$k_A + k_B + k_C = k$$

# Spine - **sp**arsification of **i**nfluence **ne**tworks

http://www.cs.toronto.edu/~mathiou/spine/

greedy algorithm

two phases

phase 1

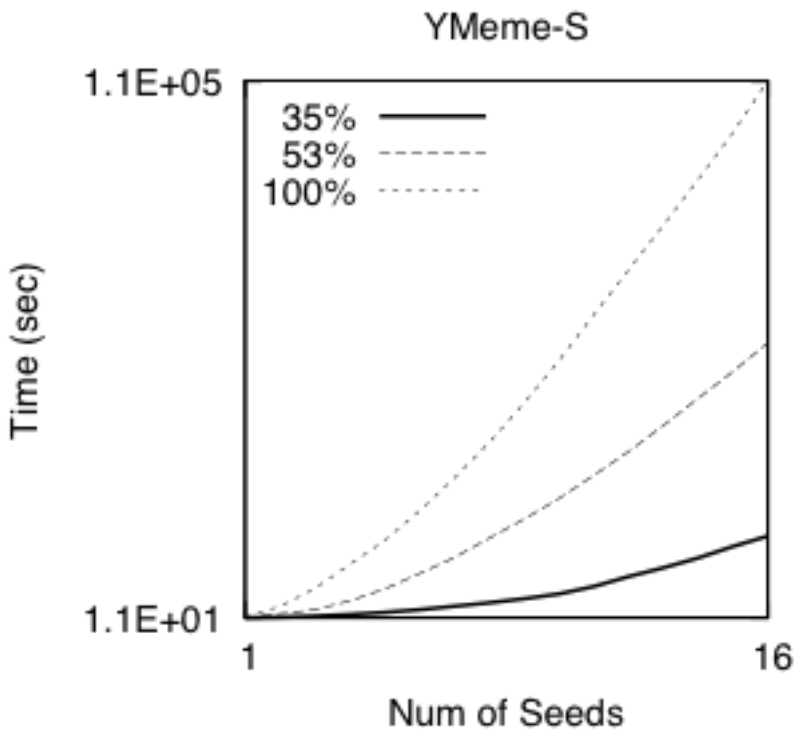
obtain a non-zero-likelihood solution

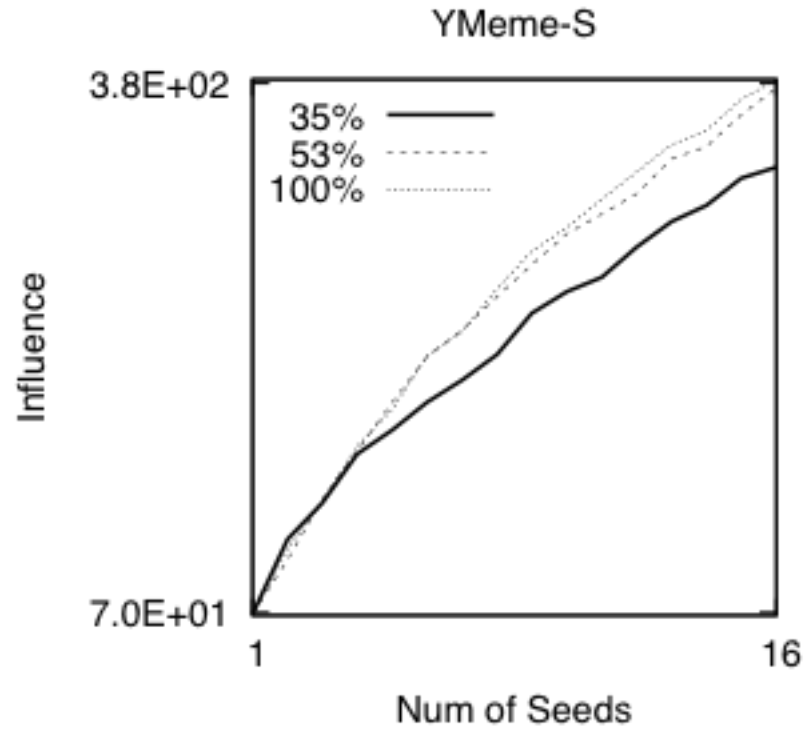
phase 2

add one arc at a time, the one that offers largest increase in likelihood

(approximation guarantee for phase 2 thanks to submodularity)

# Application to Influence Maximization



# Part 2: takeaways (1/2)

Both the social graph structure and the influence probabilities are important in the influence maximization problem

Mining real past propagation matters!

Probability of influence decay exponentially with time

it is important to develop time conscious models

Discrete time models: good compromise between accuracy and speed

Influence probability models can be use to predict if and when a user will perform an action on the basis of influence

It is important to devise algorithms that minimize the number of scans of the propagations log

RP#4: models and algorithms to learn influence probabilities from propagation data

RP#5: considering different levels of influenceability in the theory of Viral Marketing

RP#6: role of time in Viral Marketing

# Part 2: takeaways (2/2)

## Sparsification helps

reducing networks to the important parts

## highlighting the backbone of networks

RP#7: directly estimate a “sparse” set of influence probabilities

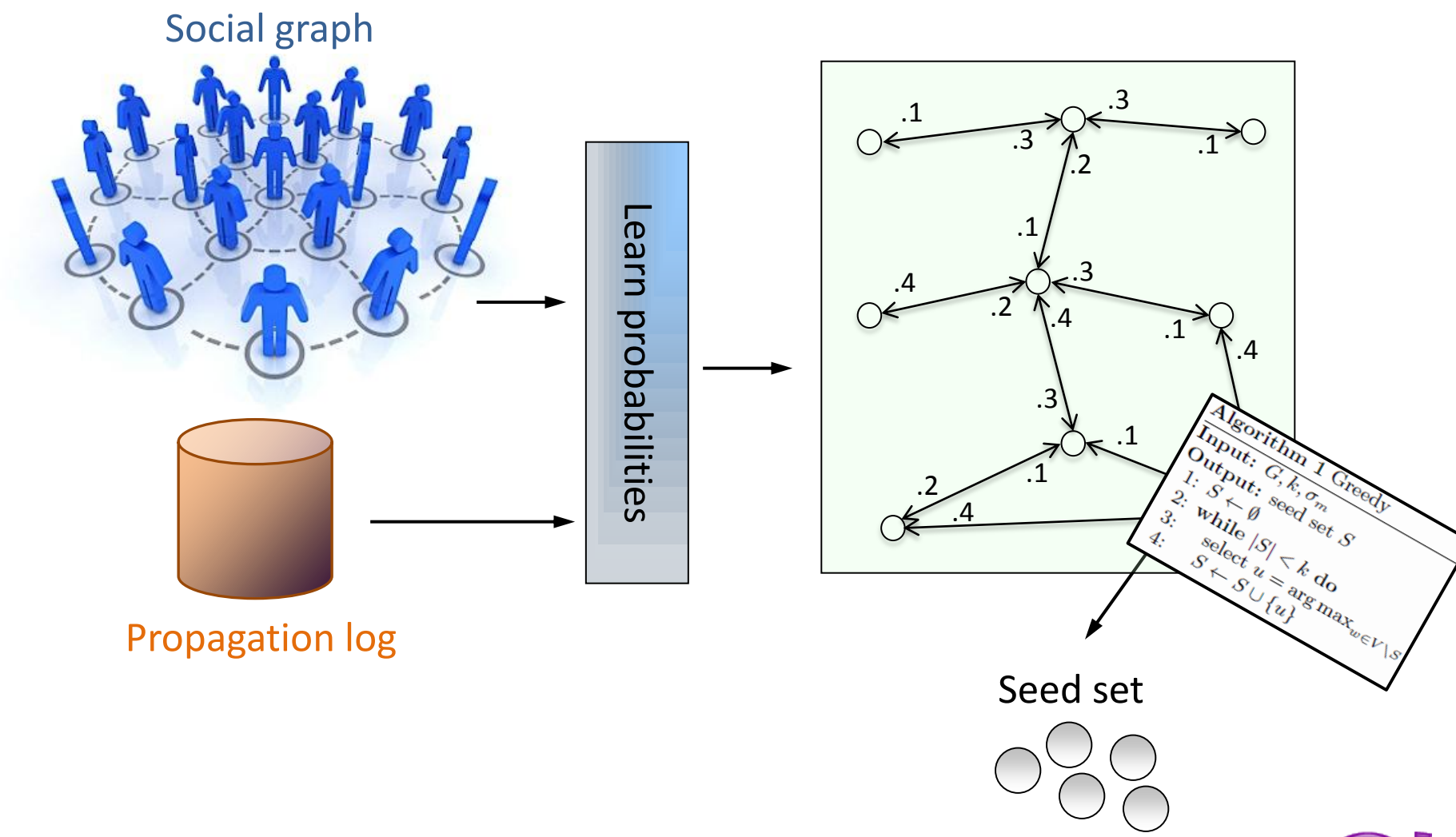
RP#8: SPINE on big data (on Hadoop)

RP#9: compare backbones of different networks

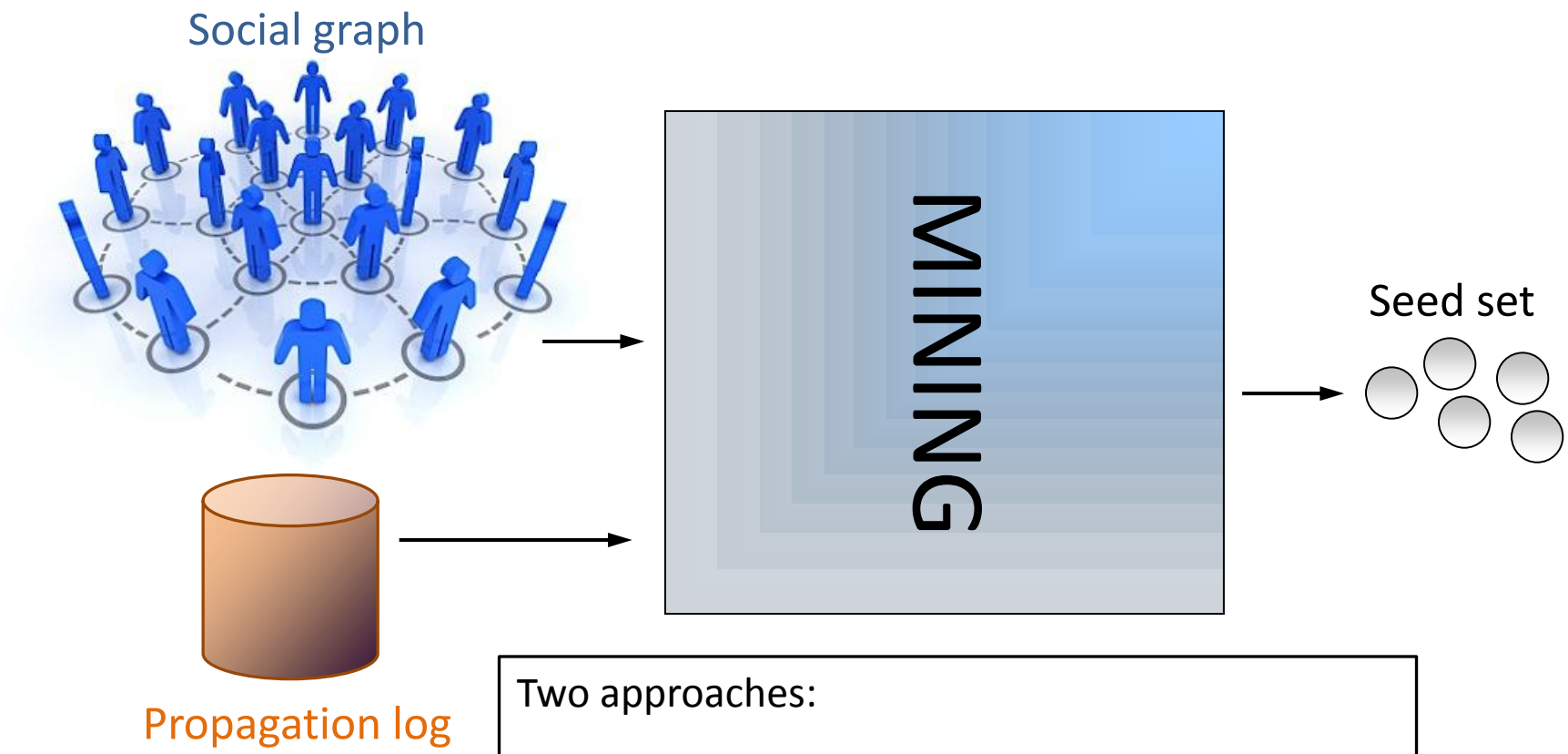
RP#10: compare properties of the original network and the sparsified

# Part 3:
# Direct Mining

# The general picture



# Direct mining




Two approaches:

1. Credit Distribution (CD) approach
2. A frequent-pattern based approach

# Expected spread: a different perspective*

Instead of simulating propagations, use available propagations!

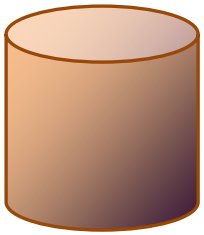
$$\sigma_m(S) = \sum_{X \in \mathbb{G}} Pr[X] \cdot \sigma_m^X(S)$$

sampling "possible worlds"
(MC simulations)

$$\sigma_m^X(S) = \sum_{u \in V} path_X(S, u)$$

$$\sigma_m(S) = \sum_{u \in V} \sum_{X \in \mathbb{G}} Pr[X]\, path_X(S, u)$$

$$\sigma_m(S) = \sum_{u \in V} E[path(S, u)] = \sum_{u \in V} \boxed{Pr[path(S, u) = 1]}$$



Estimate it in "available worlds"
(i.e., our propagation traces)

* Goyal, Bonchi, Lakshmanan (VLDB'12) *"A Data-Based Approach to Social Influence Maximization"*

# The sparsity issue

We can not estimate directly $Pr[path(S,u)=1]$ as:

$$\frac{\text{\# actions in which } S \text{ is the seed-set and } u \text{ participates}}{\text{\# actions in which } S \text{ is the seed-set}}$$

None or too few actions where *S* is effectively the seed set.

## Take a u-centric perspective instead:

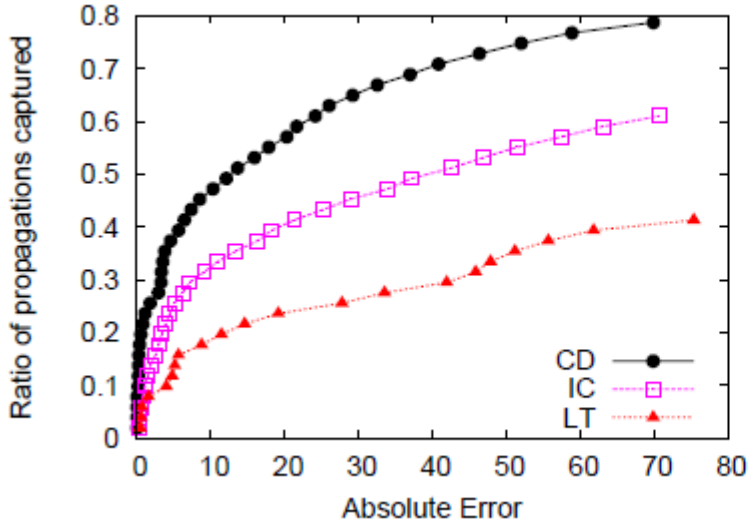
Each time *u* performs an action we distribute influence credit for this action, back to her anchestors

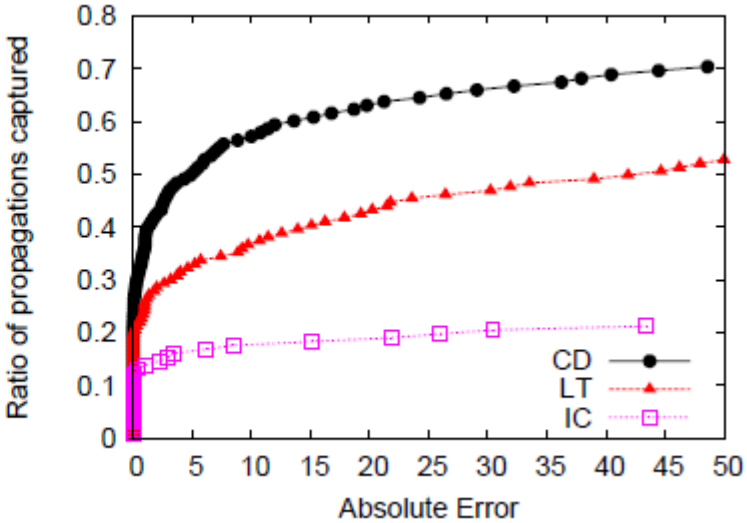
learns different level of user influenceability

Time-aware

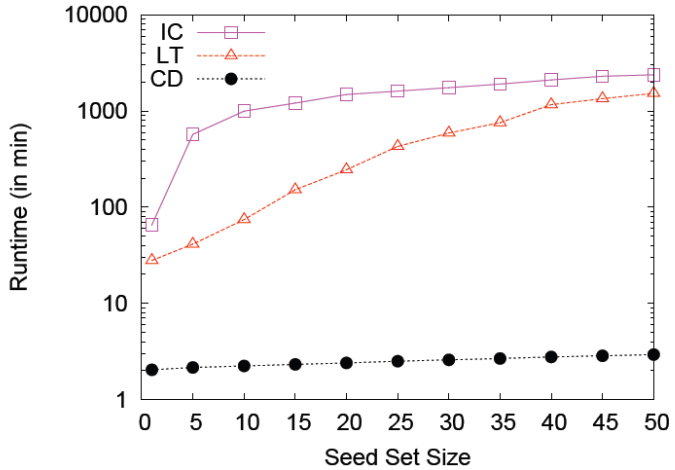
# Experiments

Datasets: Flixster Flickr

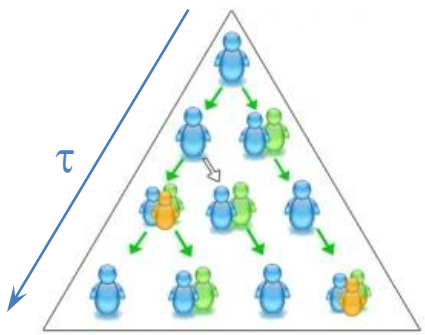



Dataset: Flixster small



# Leaders and Tribes: a pattern mining approach*




Given a time threshold $\tau$, in a given propagation, define the followers of a user $u$, those ones in the "subtree" of $u$, that activate within $\tau$ from $u$.

A user is a leader w.r.t. a given action when the number of his followers exceeds a given threshold.

A user to be identified as a leader must act as such sufficiently often, i.e., for a number of actions larger than a given threshold.

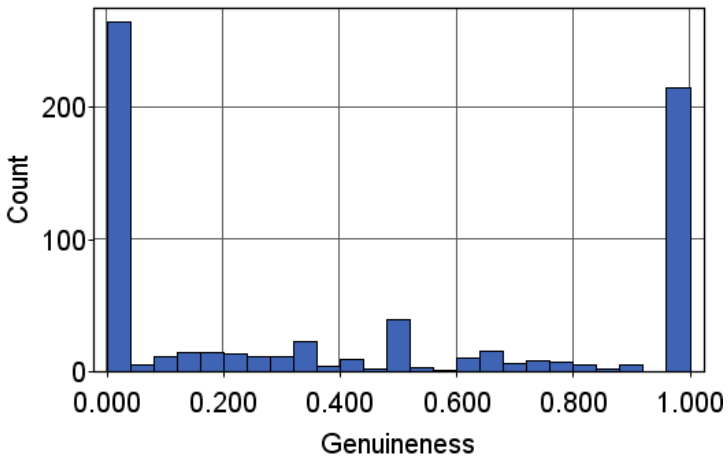
Tribe Leaders:
Previous definition does not force the set of followers for different actions to be the same. If we add this constraint we obtain tribe leaders.

Additional constraints:
- Confidence
- Genuineness

Develop efficient algorithms that make only a pass over the actions log.

*Goyal, Bonchi, Lakshmanan (CIKM'08) *"Discovering Leaders from Community Actions"*
(ICDE'09, demo ) *"GuruMine: a Pattern Mining System for Discovering Leaders and Tribes"*

# Experiments




$\pi = 9$ weeks, $\sigma = 5$

| rank | user_id | tribesize | #actions | conf. | genu. |
|---|---|---|---|---|---|
| 1 | 98711 | 25 | 11 | 0.85 | 0.73 |
| 2 | 31018 | 25 | 9 | 1 | 1 |
| 3 | 170467 | 23 | 8 | 1 | 0 |
| 4 | 20045 | 22 | 11 | 0.61 | 0.55 |
| 5 | 66331 | 21 | 13 | 0.81 | 0.92 |
| 6 | 27381 | 21 | 16 | 0.57 | 0.63 |
| 7 | 85363 | 20 | 5 | 0.63 | 0.8 |
| 8 | 144314 | 20 | 19 | 0.86 | 0.1 |
| 9 | 153181 | 19 | 22 | 0.76 | 0.82 |
| 10 | 206280 | 19 | 12 | 1 | 0.67 |

Genuineness: an almost binary concept!

Tribe leaders exhibit high confidence.

Tribe leaders with low genuineness were found dominated by other tribe leaders present in the same table

# Part 3: takeaways

Methods based on directly mining the propagations are promising

avoid the costly learning of the probabilities + simulation approach

Two models studied

"Credit Distribution" and "Leaders-and-Tribes"

Both time-conscious

Emphasis on efficient and scalable algorithms

Scan the propagations log only once

RP#11: characterization of tribes in terms of communities

# Summary

we have seen Influence Maximization prior art
missing pieces and open problems

we have filled some of the missing pieces

focussing mainly on mining the available log of past propagations together with the social graph

putting emphasis on

1) the need for clever algorithms that scan the propagations log as few times as possible

2) the temporal dimension of propagations

# Some more (almost) open problems

RP#12: community detection based on the propagations

RP#13: the competitive Viral Marketing case

RP#14: privacy of the SN users?

RP#15: more information might be available
(e.g., demographics, behavioral)

some people are more likely to buy a product than others

e.g. teenagers are more likely to buy videogames than seniors

can we compute this likelihood?

can we exploit it?

# The main open problem

Influence Maximization is still an ideal problem: how to make it actionable in the real-world?

Propagation models make many assumptions:

which are more realistic and which are less?

Which propagation model does better describe real-world?

We need techniques and benchmarks for comparing different propagation models and the associated influence maximization methods on the basis of some ground-truth

# THANKS!

http://francescobonchi.com/

# REFERENCES


Domingos and Richardson *"Mining the network value of customers"* (KDD'01)

*"Mining knowledge-sharing sites for viral marketing"* (KDD'02)

Kempe et al. *"Maximizing the spread of influence through a social network"* (KDD'03)

Kimura and Saito *"Tractable models for information diffusion in social networks"* (PKDD'06)

Saito et al. *"Prediction of information diffusion probabilities for independent cascade model"* (KES'08)

Leskovec et al. *"Cost-effective outbreak detection in networks"* (KDD'07)

Crandall et al. *"Feedback Effects between Similarity and Social Influence in Online Communities"* (KDD'08)

Anagnostopoulos et al. *"Influence and correlation in social networks"* (KDD'08)

Chen et al. *"Efficient influence maximization in social networks"* (KDD'09)

*"Scalable influence maximization for prevalent viral marketing in large-scale social networks"* (KDD'10)

*"Scalable influence maximization in social networks under the linear threshold model"* (ICDM'10)

Our work:

*"Discovering Leaders from Community Actions"* (CIKM'08)

*"GuruMine: a Pattern Mining System for Discovering Leaders and Tribes"* (ICDE'09)

*"Learning influence probabilities in social networks"* (WSDM'10)

"Sparsification of Influence Networks" (KDD'11)

*"A Data-Based Approach to Social Influence Maximization"* (VLDB'12)