Part I → Part II → Part III → Part IV → Part V

Other Issues

Part IV: Outline

- Testing Homophily/Influence
- Learning Influence Models
- Model-based versus Memory-based Approaches
- Influence vs. Adoption/Revenue
- Handling Competition
- Participation Maximization
- Paying Attention to Budget and Time

The Influence of Big Business by Michael Messina

"We are here in to defend democracy around the world."

Testing Homophily/Influence

Sources of Correlation

- **Social influence** (induction)
  - One person performing an action **causes** people connected to her to do the same
- **Homophily** (selection)
  - Similar individuals are more likely to be connected: proverbial birds of a feather …
- Confounding factors: external influences Friends likely to live in same city and upload pix of same landmarks; a lot of users rate avatar ‘cos of its popularity; …

[Anagnostopoulos, Kumar, & Mahdian KDD 2008]

Shuffle test (1/3)

- Want to test if there is correlation in node activation, given D = (G, W).
  - G – social graph; W = $\{u_1, \ldots, u_m\}$ – nodes that acted (along with timestamps).
- Influence model: each user flips a coin at each time t, to decide to (not) act.
- Prob. depends on time, user, and their #active friends. Fit a logistic function for estimating probs:

correlation

$$p(k) = e^{\alpha \, ln(k+1)+\beta} / [1 + e^{\alpha \, ln(k+1)+\beta}]$$

[Anagnostopoulos et al. KDD 2008]

Shuffle test (2/3)

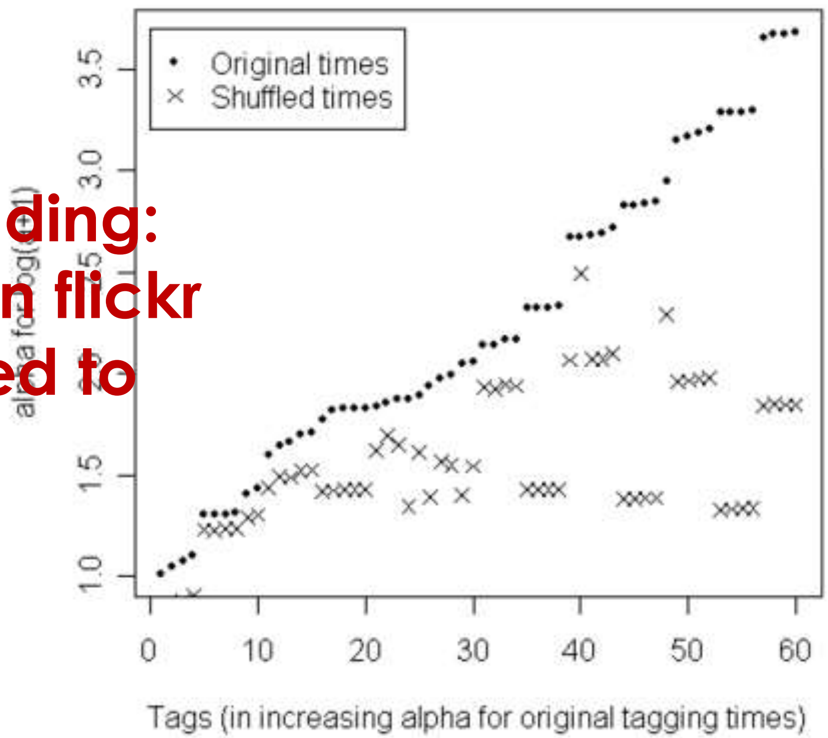
- Learn correlation on both original data D = (G,W) and on D' = (G,W') obtained by a random shuffle: randomly permute activation times of $u_1, \ldots, u_m$. → α, α'.

- If original data D came from an influence model, α' should significantly drop from α.

Shuffle test (3/3)

- Infer influence weights
- Randomize activation times in each cascade
- Infer influence weights again
  - Should be lower

A Key empirical finding: Tagging behavior in flickr cannot be attributed to influence.



[Anagnostopoulos et al. KDD 2008]

Matched sampling

- Match pairs of nodes that are “twins”
  - E.g. same age, same location, etc.
  - Match a node with no adopting friends, with a node with k adopting friends
- Verify if the node with adopting friends is more likely to adopt
- Main finding: matching random pairs reveals gross overestimates of influence by traditional methods: homophily explains >50% of perceived contagion.
  - Data: Yahoo! IM network, adoption of mobile app.

[Aral et al. PNAS 2009]

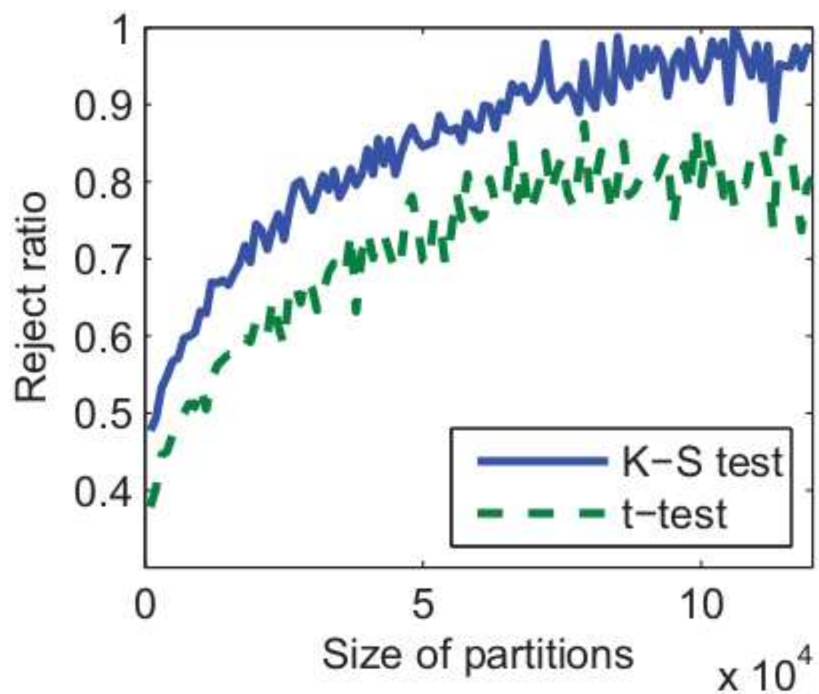
Effect of rating from friends

- Do WoM recommendations influence user ratings? If yes, how do you quantify it?
  - Focus on ***posterior evaluation***; *surprising findings.*

- For a given item i and a user u, build a triple $\langle friendRec(i,u), rating(i,u), friendRating(i,u)\rangle$ = <m',r,r'>

Group by (similarity on) $friendRating(i,u)$ and in each bucket **test if $\boldsymbol{rating(i,u)}$ is independent from $\boldsymbol{friendRec(i,u)}$**

Experimental results: **not independent** ⇒ friend adoption influences user's ratings



[Huang, Cheng,Shen, Zhou, Jin WSDM 2012]

Learning Influence Models

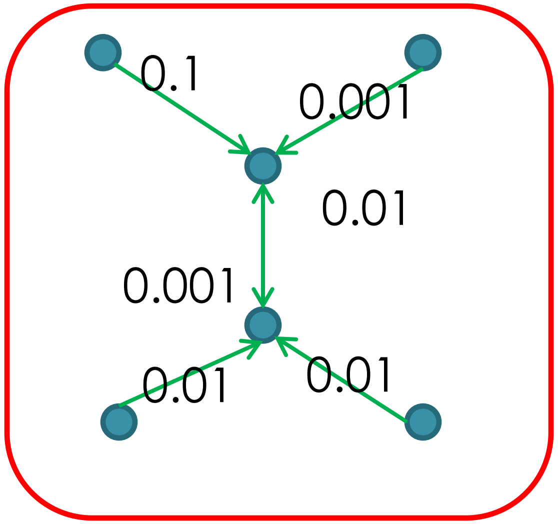
Where do the numbers come from?

Learning influence models

- Where do **influence probabilities** come from?
  - Real world social networks don't have probabilities!
  - Can we learn the probabilities from action logs?
  - Sometimes we don't even know the social network
  - Can we learn the social network, too?
- 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?

Where do the weights come from?

- Influence Maximization – Gen 0: academic collaboration networks (real) with weights assigned arbitrarily using some models:
  - Trivalency: weights chosen uniformly at random from {0.1, 0.01, 0.001}.

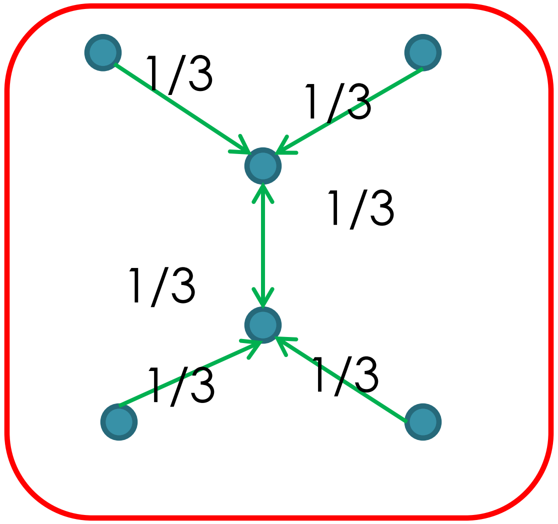


Where do the weights come from?

- Influence Maximization – Gen 0: academic collaboration networks (real) with weights assigned arbitrarily using some models:
  - Weighted Cascade: $w_{uv} = \frac{1}{d_v^{in}}$.

Other variants: uniform (constant), WC with parallel edges.

Weight assignment not backed by real data. ☹



Inference problems

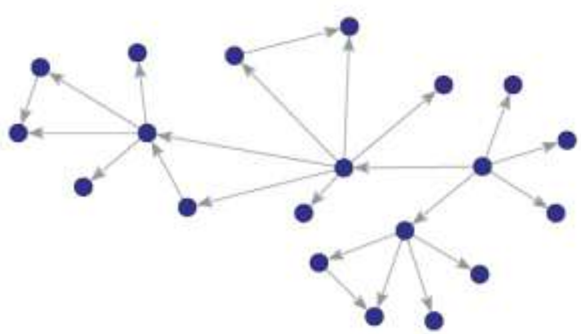
- Given a log $A = \{\langle u_1, a_1, t_1 \rangle, \dots\}$
- P1. Social network not given
  - Infer network and edge weights
- P2. Social network given
  - Infer edge weights
- P3. Social network and attribution given
  - Explicit "trackbacks" to parent user

$$A = \{\langle u_1, a_1, t_1, p_1 \rangle, \dots\}$$

  - Simple counting

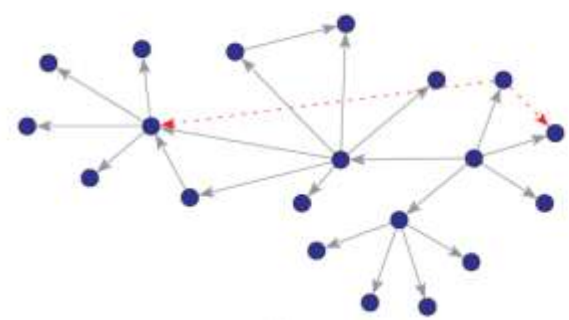
P1. Social network not given

- Observe activation times, assume probability of a successful activation decays (e.g., exponentially) with time



Actual network

$$\langle u_1, a_1, t_1 \rangle,$$
$$\langle u_2, a_2, t_2 \rangle,$$
$$\langle u_3, a_3, t_3 \rangle,$$
$$\langle u_4, a_4, t_4 \rangle,$$
$$\ldots$$

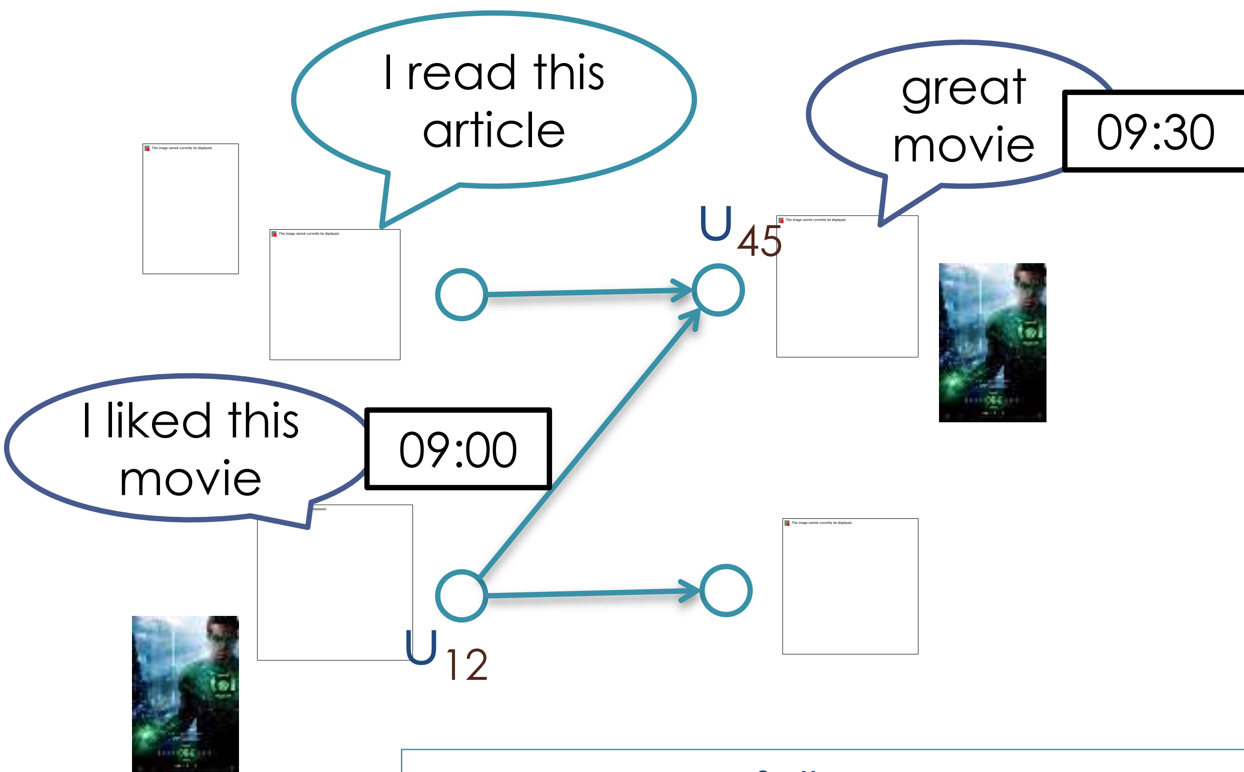


Learned network

[Gomez-Rodriguez, Leskovec, & Krause KDD 2010]

P2. Social network given

Input data: (1) social graph and (2) action 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}

P2. Social network given

- D(0), D(1), … →D(t) nodes that acted at time t.
- $C(t) = \cup_{\tau \leq t} D(\tau)$. → cumulative.
- $P_w(t+1) = 1 - \Pi_{v \in N^{in}(w) \cap D(t)} (1 - \kappa_{vw})$.
- Find $\theta = \{\kappa_{vw}\}$ that maximizes likelihood

$$L(\theta; D) = (\Pi_{t=0}^{T-1} \Pi_{w \in D(t+1)} P_w(t+1)) -$$ ← success

$$(\Pi_{t=0}^{T-1} \Pi_{v \in D(t)} \Pi_{w \in N^{out}(v) \setminus C(t+1)} (1 - \kappa_{vw}))$$ ← failure

Very expensive (not scalable)

Assumes influence weights remain constant over time

[Saito et al. KES 2008]

P2. Social network given

- Several models of influence probability
  - in the context of General Threshold model + time
  - consistent with IC and LT models
- With or without explicit attribution
- 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 → 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, and L. WSDM2010]

Influence models

Static Models: probabilities are static and do not change over time.

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

Continuous Time (CT) Models: probabilities decay exponentially in time

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

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

Discrete Time (CT) Models: Active neighbor *u* of v remains contagious in

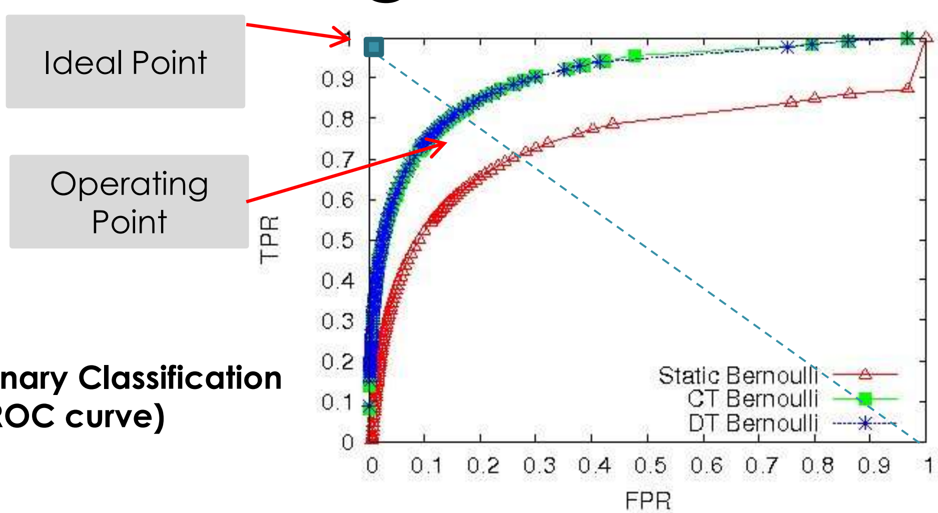
[t, t+ τ(u,v)], has constant influence prob *p(u,v)* in the interval and 0 outside.

Monotone, submodular, and incremental!

Evaluation

- Flickr groups dataset (action=joining)
  - ~1.3M nodes, 40M edges, 36M actions
  - 80/20 training/testing split

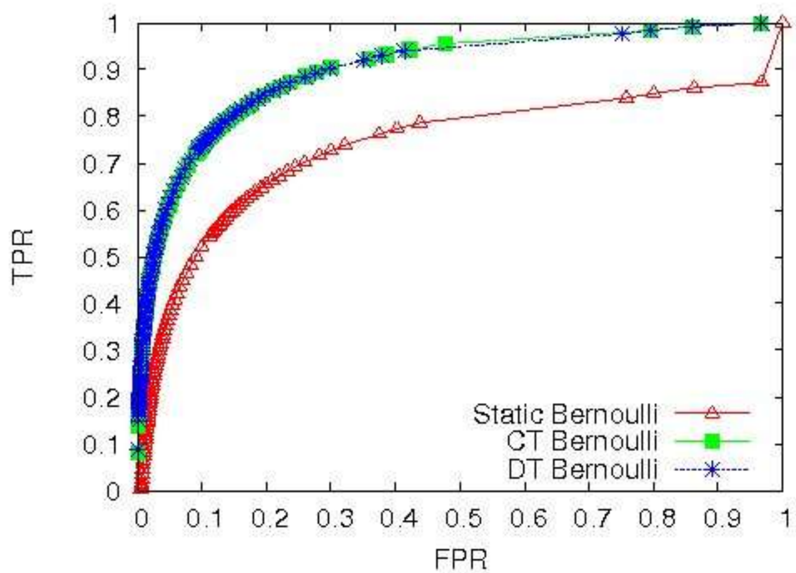
- Predict whether user will become active or not, given active neighbors

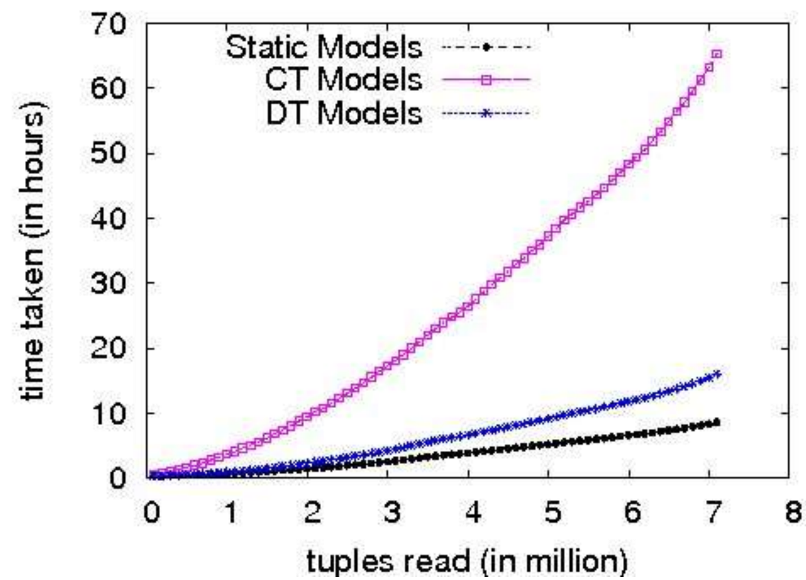


Binary Classification (ROC curve)

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

Comparison of Static, CT and DT models

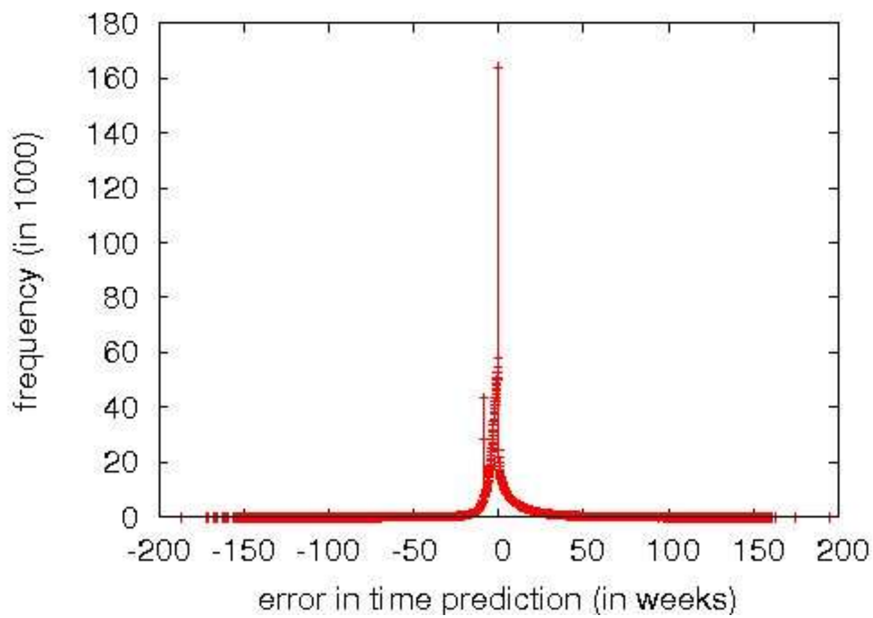




- Time-conscious models better than the static model
  - CT and DT models perform equally well

- Static and DT models are far more efficient compared to CT models because of their incremental nature

Predicting Time – Distribution of Error



- Operating Point is chosen corresponding to
  - TPR: 82.5%, FPR: 17.5%.
- Most of the time, error in the prediction is very small

Learning Influence Probabilities Takeaways

- Influence network and weights not always available

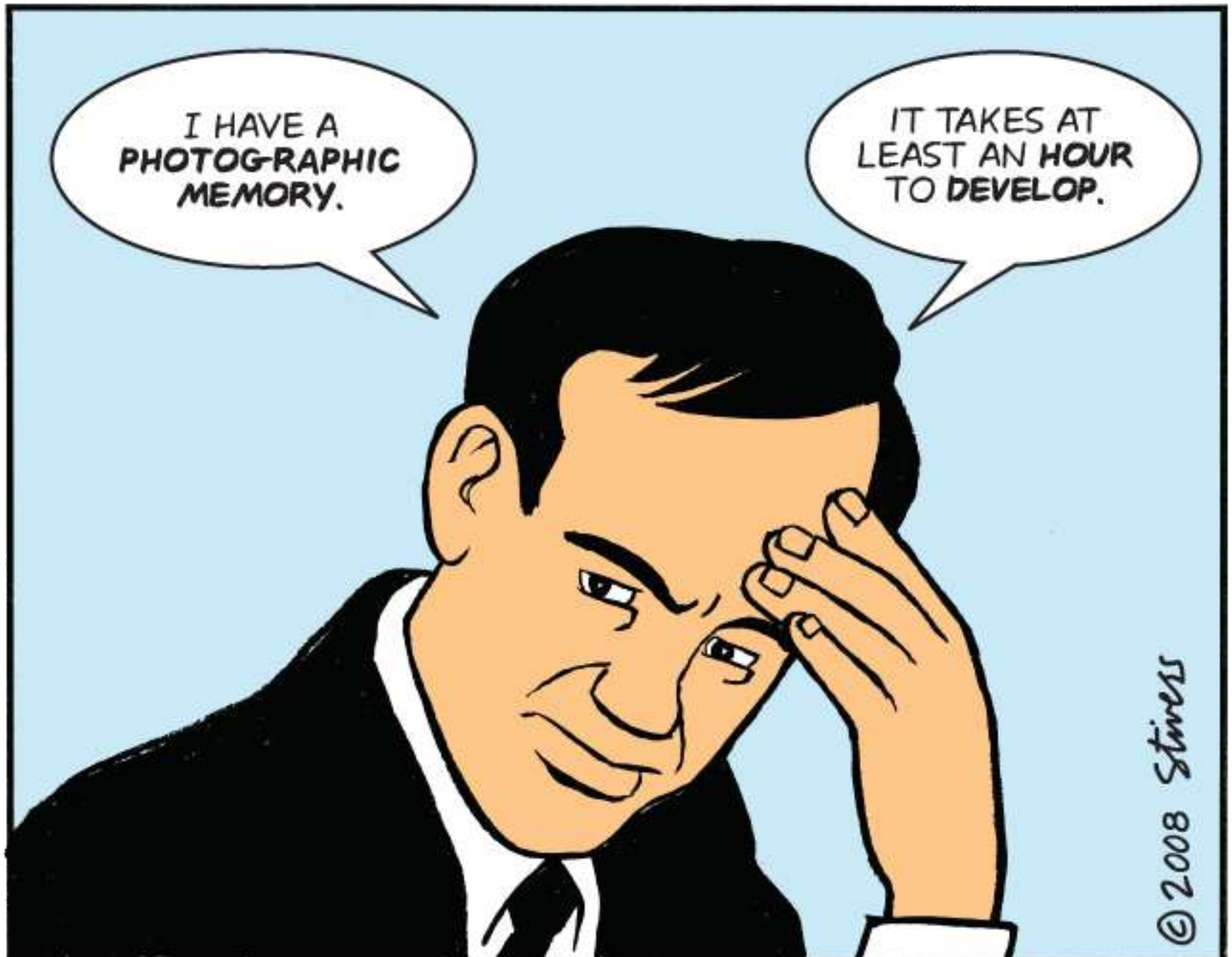
- Learn from the action log
  - [Gomez-Rodriguez et al. 2010]: Infer social network and edge weights
  - [Saito et al. 2008]: Infer edge weights using EM approach
  - [Goyal et al. 2010]: Infer both static and time-conscious models of influence

- Using CT models, it is possible to predict even the time at which a user will perform it with a good accuracy.

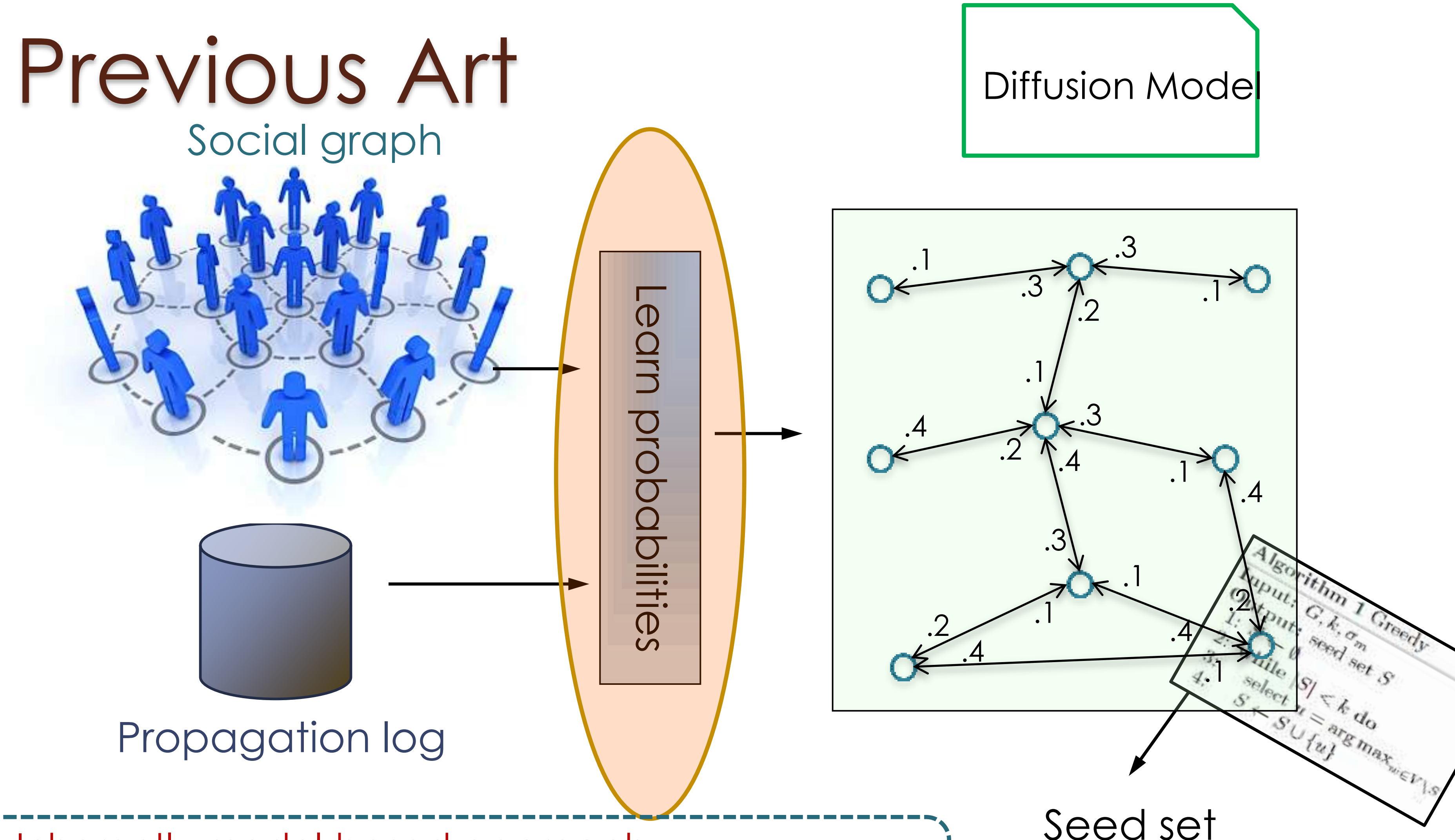
- Introduce metrics of users and actions influenceability.
  - High values => easier prediction of influence.
  - Can be utilized in Viral Marketing decisions.

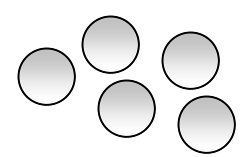
Memory-based and Model-based Approaches for Influence Maximization

And a little memory always helps!



Previous Art





Inherently model-based approach
Can we instead use memory (of what happened in the past) directly?

Why learning from data matters

- Methods compared (IC model):
  - WC, TV, UN (no learning)
  - EM (learned from real data – Expectation Maximization method)
  - PT (learned then perturbed ± 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, & L. VLDB 2012]

Why learning from data matters – experiments*

1.

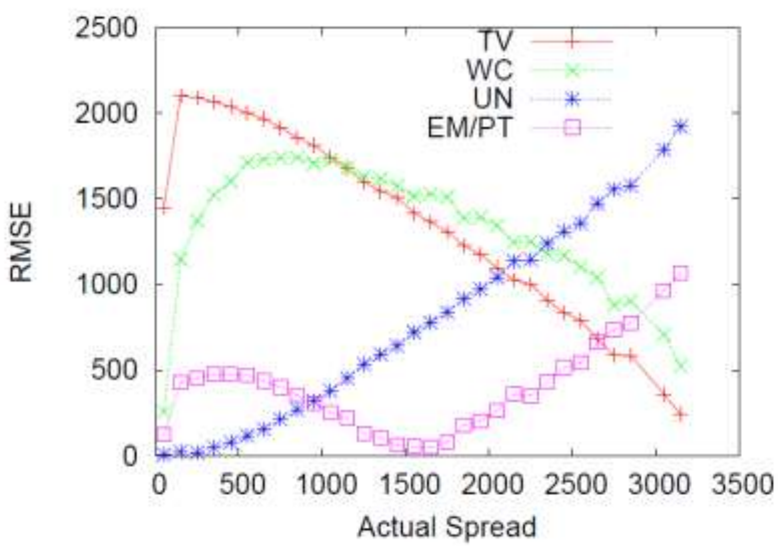
Flixster_Small

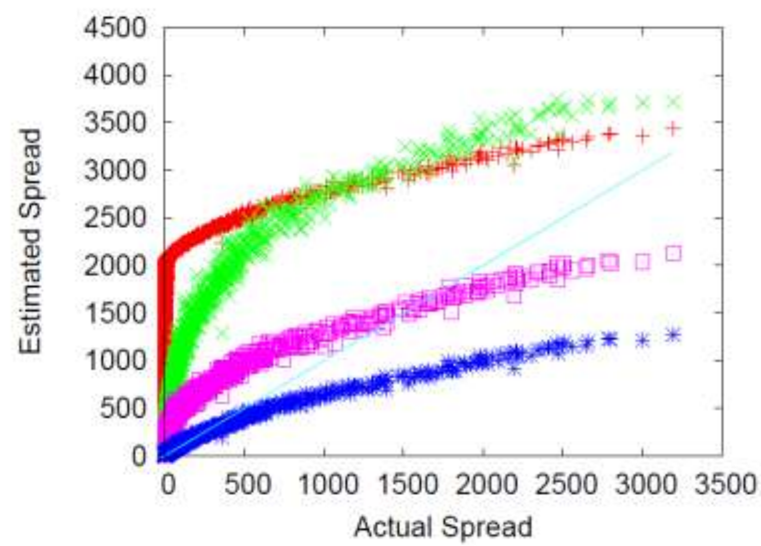
	UN	WC	TV	EM	PT
UN	50	25	5	6	6
WC		50	9	3	2
TV			50	3	2
EM				50	44
PT					50

Flickr_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				

2.





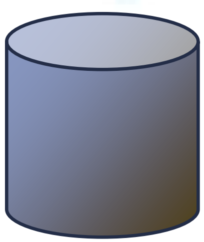
Memory-based Approach

- Instead of learning probabilities from available propagation traces
  (sampling possible worlds from model, using simulation to estimate expected spread)
- **Use the actual/real worlds corresponding to the propagations that actually happened to estimate spread!**

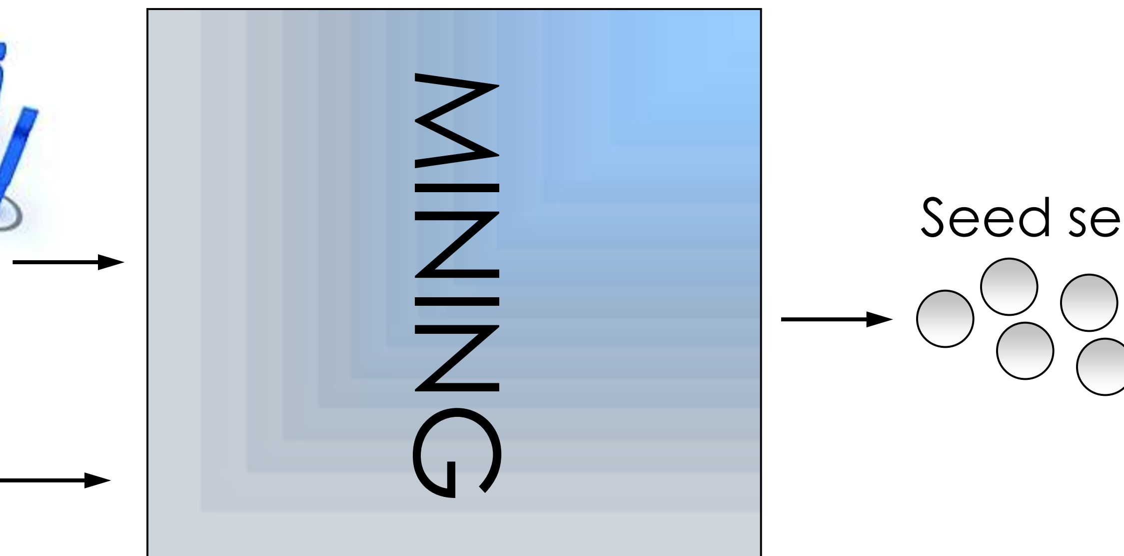
[Goyal et al. VLDB 2012]

Direct mining

Social graph



Propagation log



Seed set

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!

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

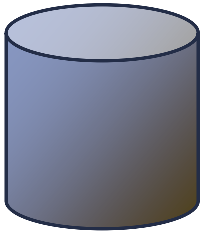
sampling "possible worlds" (MC simulations)

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

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

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

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



[Goyal et al. VLDB 2012]

The sparsity issue

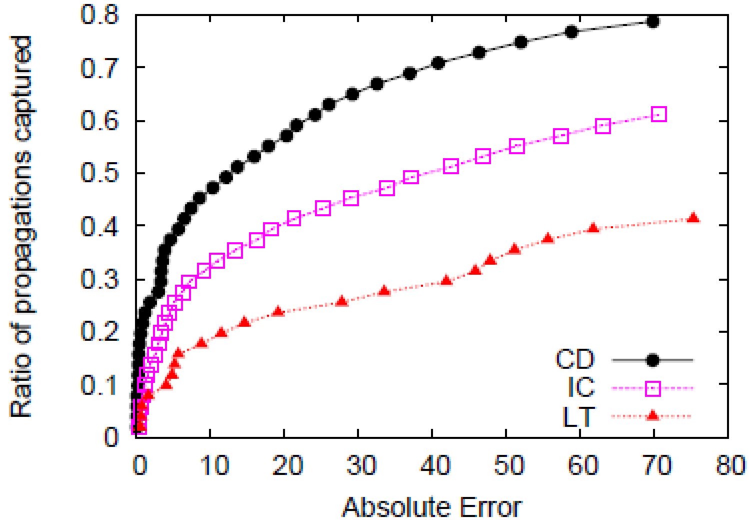
We can not estimate directly $Pr[path(S,u)=1]$ as:

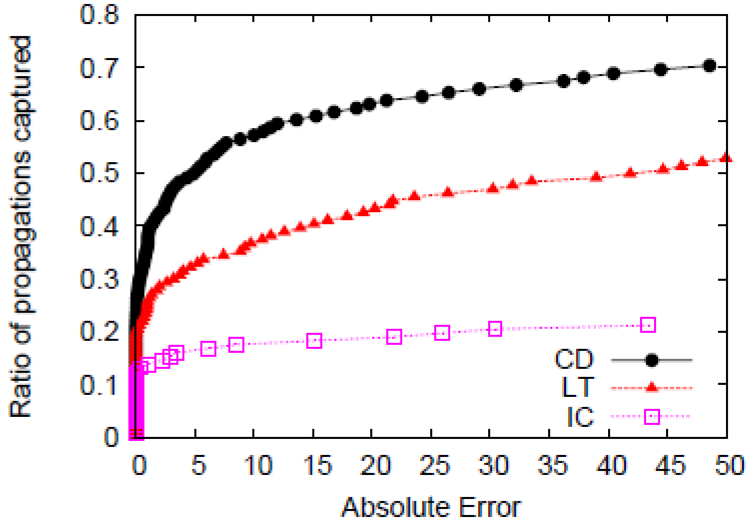
$$\frac{\text{(\# actions in which } S \text{ is the seed-set and } u \text{ participates)}}{\text{(\# actions in which } S \text{ is the seed-set)}}$$

- None or too few actions where *S* is effectively the seed set i.e., initiators).
- Take a ***u-centric*** perspective instead:
- Each time *u* performs an action we distribute influence credit for this action, back to her ancestors
- learns different level of user-influenceability
- Time-aware

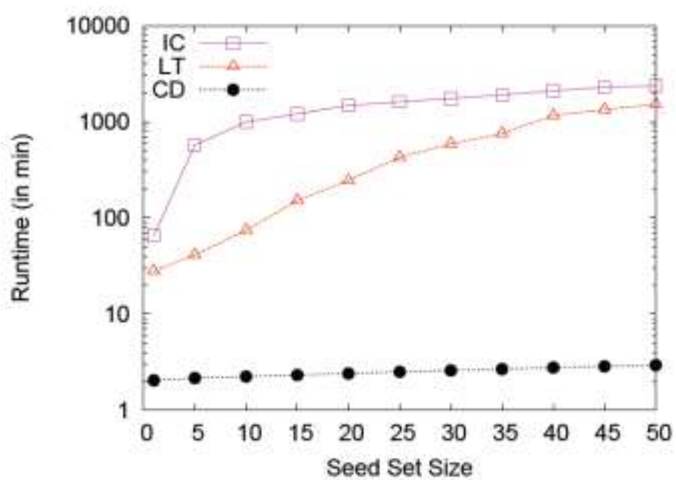
Experiments

Datasets: Flixster Flickr





Dataset: Flixster small



Influence vs. Adoption/Revenue

I want to buy it but …

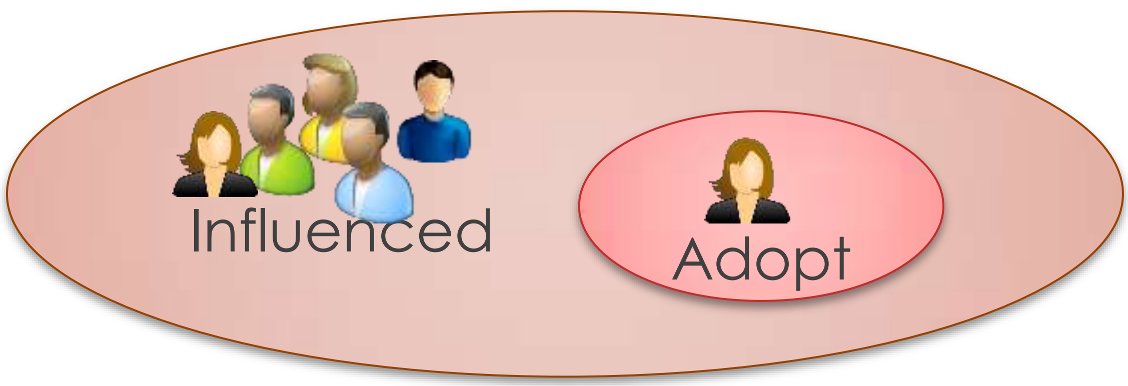
Influence vs. Adoption vs. Profit

- If a user gets influenced, it doesn't necessarily imply she'll adopt the product.
- Classical models:
  - influenced ➔ adopt.
  - Profit captured by proxy: expected spread!
- Need models and algorithms for VM taking these distinctions into account.

[Bhagat, Goyal, & L. WSDM 2012]

Influence $\nRightarrow$ Adoption

- Observation: Only a subset of influenced users actually adopt the marketed product



- Awareness/information spreads in an epidemic-like manner while adoption depends on factors such as product quality and price

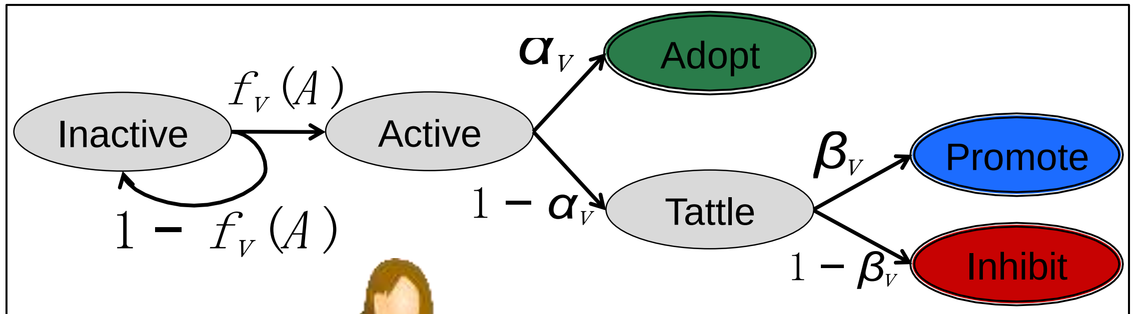
[Kalish MS, 95]

Influence $\nRightarrow$ Adoption

- Moreover, there exist users who help in information propagation without actually adopting the product – tattlers.

Our Model (LT-C)

Active
Friends



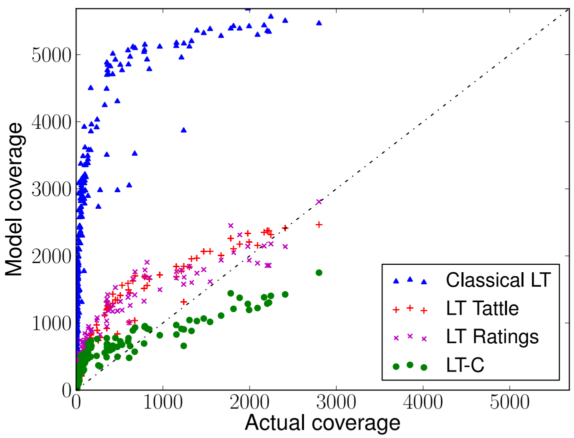
User v

- Model Parameters
  - *A* is the set of active friends
  - $f_V(A)$ is the activation function $$f_v(A) = \frac{\sum_{u \in A} w_{u,v}(r_{u,i} - r_{\min})}{r_{\max} - r_{\min}}$$
  - $r_{U,i}$ is the (predicted) rating for product i given by user u
  - α_V is the probability of user v adopting the product
  - β_V is the probability of user v promoting the product

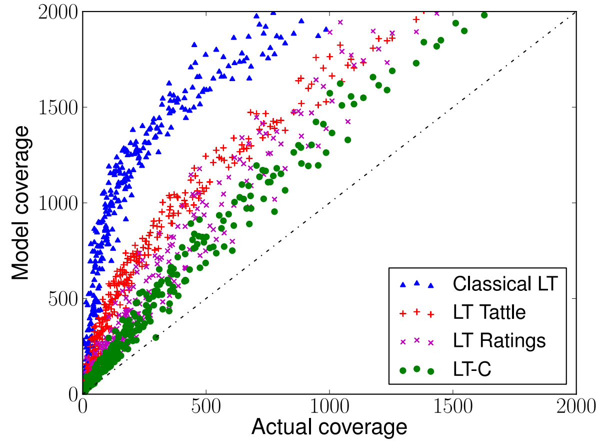
Maximizing Product Adoption

- Problem: Given a social network and product ratings, find *k* users such that by targeting them the expected spread (expected number of adopters) under the LT-C model is maximized
- Problem is NP-hard
- The spread function is monotone and submodular yielding a (1-1/e)-approximation to the optimal using a greedy approach

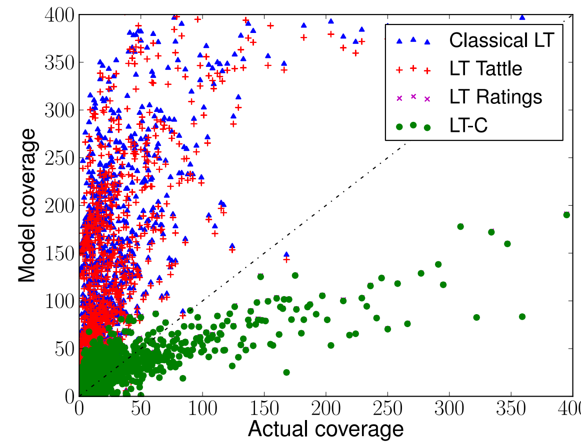
Evaluation: Spread Estimates



Flixster



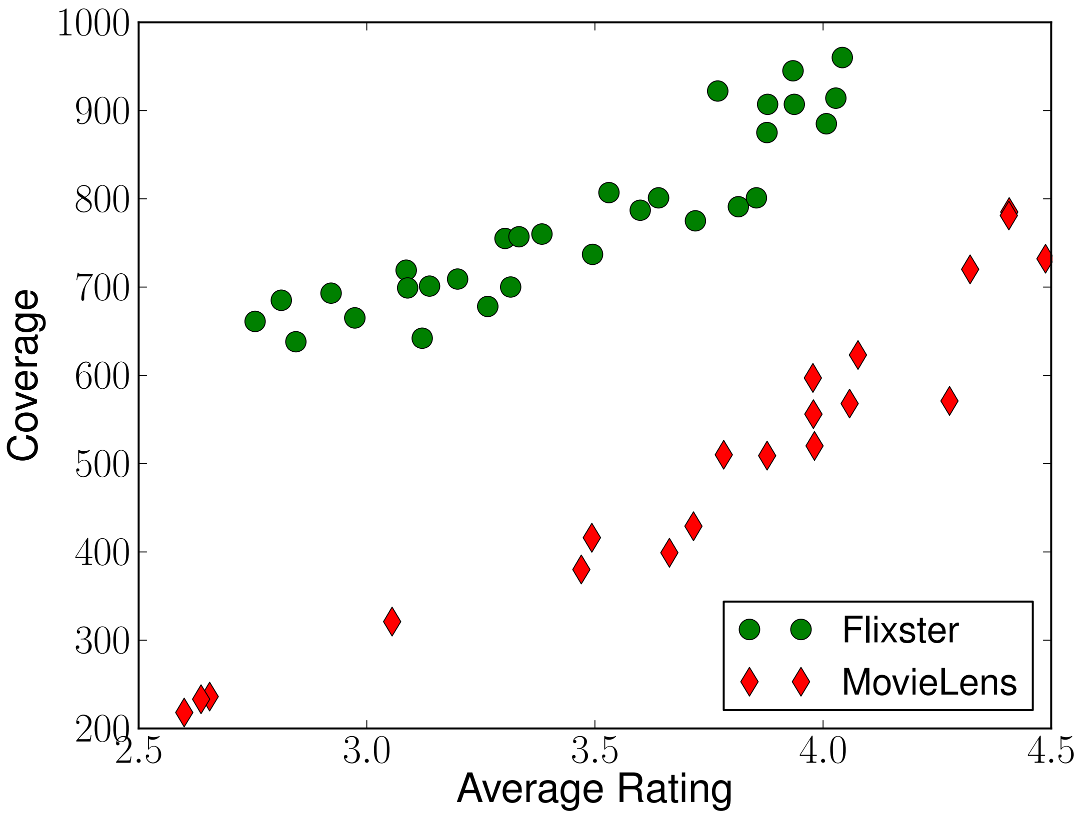
MovieLens



Last.fm

- Our model (LT-C) better predicts spread for all datasets

Spread depends on product quality

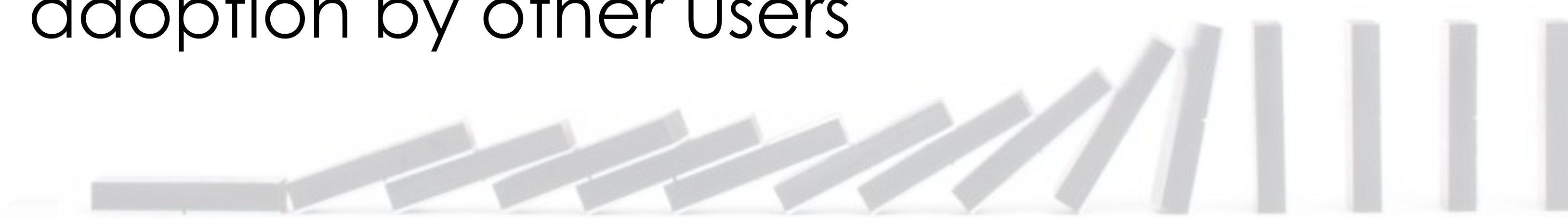


Better quality products have better coverage

Classical LT model on the other hand predicts equal coverage for all products

Key Takeaways

- Only a fraction of users who are influenced do adopt the product
- The influence of an adopter on her friends is a function of the adopter's experience with the product, in addition to propagation probability
- Non-adopters can play a role of "information bridges" helping in spreading the influence/information, and thus adoption by other users



Handling Competitions


Sacre bleu!
25% OFF
HA!
50% OFF
© Grimmy & Clark

Competitive influence diffusion

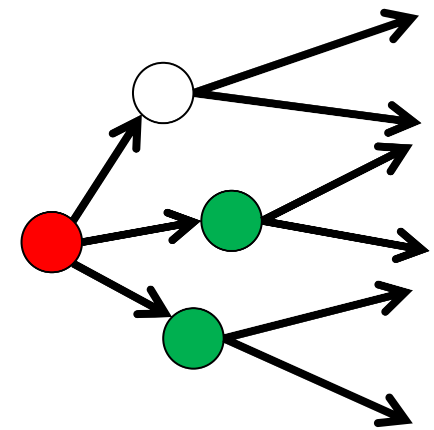
- Influence maximization vs. influence blocking maximization
- Modeling competitive diffusion
- Endogenous competition: emergence and propagation of negative opinions

Influence blocking maximization

- Problem:
  - Given the negative activation status,
  - find k positive seeds
  - minimize the further negative influence, or maximize the expected number of "saved" or "blocked" nodes from negative influence --- *negative influence reduction*
- Extension of the IC model [Budak et al. WWW 2011]
- Extension of the LT model [He et al. SDM 2012]

Multiple Campaign IC model

- Two campaigns, positive vs. negative
- General case:
  - each campaign has an independent set of IC parameters
  - negative influence reduction is not submodular
- Special cases:
  - high effectiveness property: positive campaign has propagation probability of one
  - campaign oblivious IC : positive and negative campaigns have the same parameters
  - tie-breaking rule: positive campaign dominance
  - negative influence reduction is submodular



[Budak, Agrawal and Abbadi, WWW 2011]

Competitive linear threshold model

- two campaigns, each has a different set of LT parameters (influence weights)
- each nodes has two thresholds, negative and positive thresholds, drawn uniformly at random from [0,1]
- positive and negative campaigns use their own LT parameters to diffuse
- negative campaign dominates (could be changed to an arbitrary dominance probability)

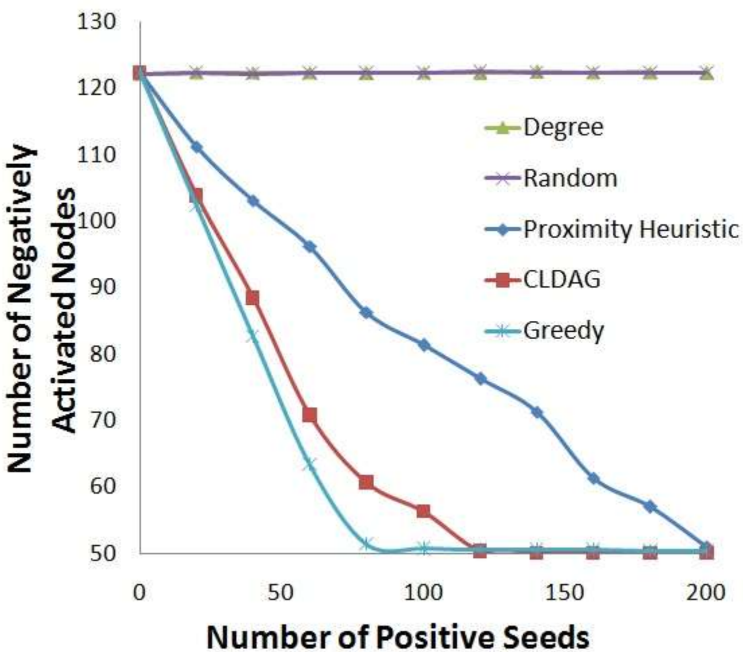
[He et al. SDM 2012]

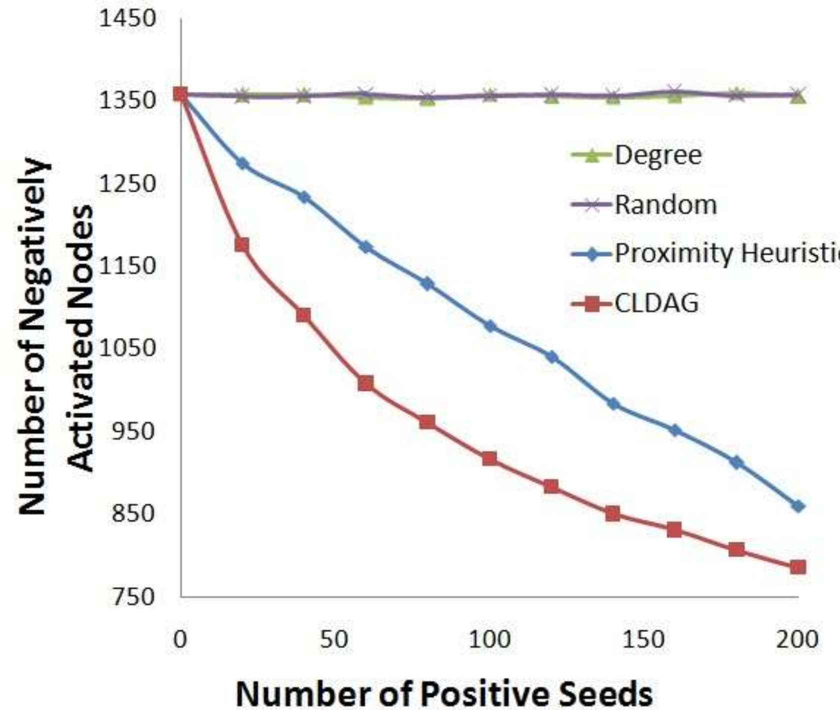
Influence blocking maximization under CLT

- negative influence reduction is submodular
- allows greedy approximation algorithm
- fast heuristic CLDAG:
  - reduce influence computation on local DAGs
  - use dynamic programming for LDAG computations

[He et al. SDM 2012]

Performance of the CLDAG



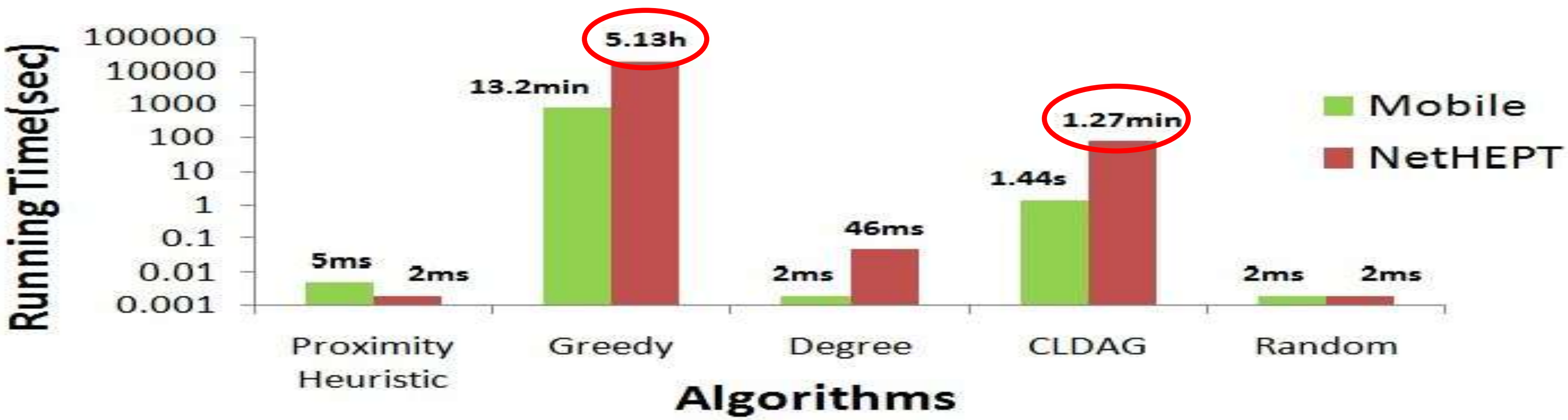


- with Greedy algorithm
- 1000 node sampled from a mobile network dataset
- 50 negative seeds with max degrees

- without Greedy algorithm
- 15K node NetHEPT, collaboration network in arxiv
- 50 negative seeds with max degrees

[He et al. SDM 2012]

Scalability—Real dataset



- Scalability Result for subgraph with greedy algorithm

[He et al. SDM 2012]

Attacker/defender game for competitive influence diffusion

- a zero-sum game
  - attacker selects negative seeds to maximize its influence
  - defender selects positive seeds to minimize attacker's influence
- Maximin strategy
  - compute mixed Nash equilibrium for both simultaneous-move and leader-follower Stackelberg games
  - inefficient, need full payoff matrix
- Double oracle algorithm
  - attacker uses any influence maximization algo. as attacker oracle
  - defender uses any influence blocking maximization algo. as defender oracle
  - iteration: use oracles to enlarge strategy space, use Maximin to compute mixed equilibrium on the current strategy space

[Tsai, Nguyen and Tambe, AAAI 2012]

Endogenous Competition: Effect of Negative Opinions

"PERSONALLY, I THOUGHT IT STUNK!"

Endogenous competition

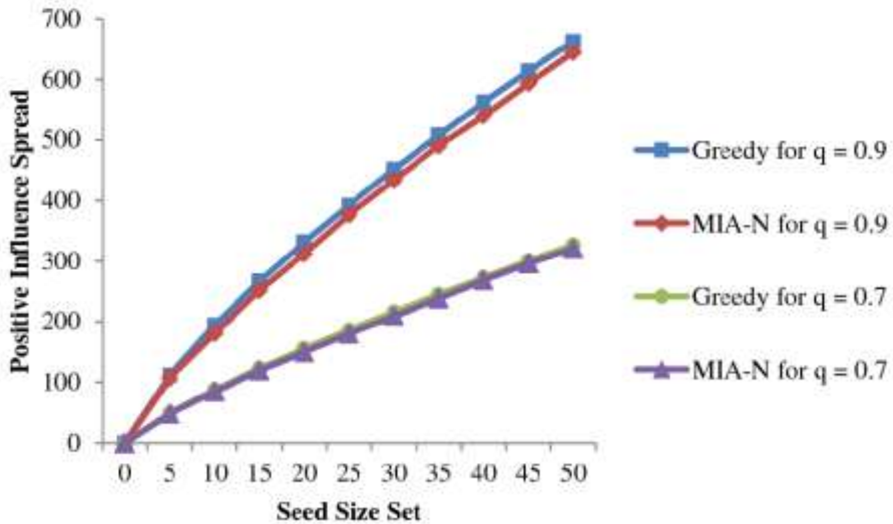
- Negative opinion generated from product defects
- Negative opinion propagates, competing with positive opinion
- Positive opinions may turn negative, but negative opinions will not turn back --- negativity bias

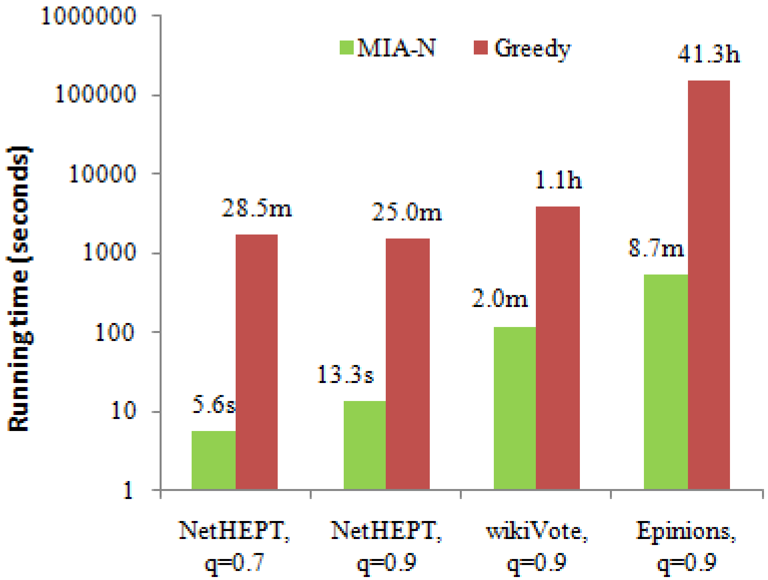
Influence maximization with negative opinions

- IC-N: extend the IC model with quality factor q
  - each positive activation has probability of $1 - q$ to turn negative
  - negative opinion propagates as positive opinion, but negative activations do not turn positive
- Maximize the positive influence
- Submodularity still holds
- MIA-N: fast heuristics using dynamic programming for efficient tree based influence spread computation

[Chen et al. SDM 2011]

Performance of MIA-N heuristic





- 15K node NetHEPT, collaboration network in arxiv
- influence spread of MIA-N matches Greedy algorithm

- MIA-N achieves 2 orders of manitude speedup

[Chen et al. SDM 2011]

Key takeaways for handling competitions

- Standard models (IC/LT) may be generalized for exogenous/endogenous competition
  - be careful, may violate submodularity

- Activation timing becomes important, due to competitions between positive and negative diffusions
  - Greedy algorithm becomes slower
  - Heuristics need dynamic programming

Other topics

- Participation maximization
  - from platform provider's point of view
  - many cascades, maximize overall spread
  - each user can be seeds for a small number of cascades
  - see [Ienco, Bonchi and Castillo, ICDM Workshops 2010; Sun et al. ICWSM 2011]
- Budget and time
  - Time-critical IM [Chen, Lu, Zhang, AAAI 2012]
  - minimize seed size, or diffusion time [Goyal, et al. SNAM 2012]

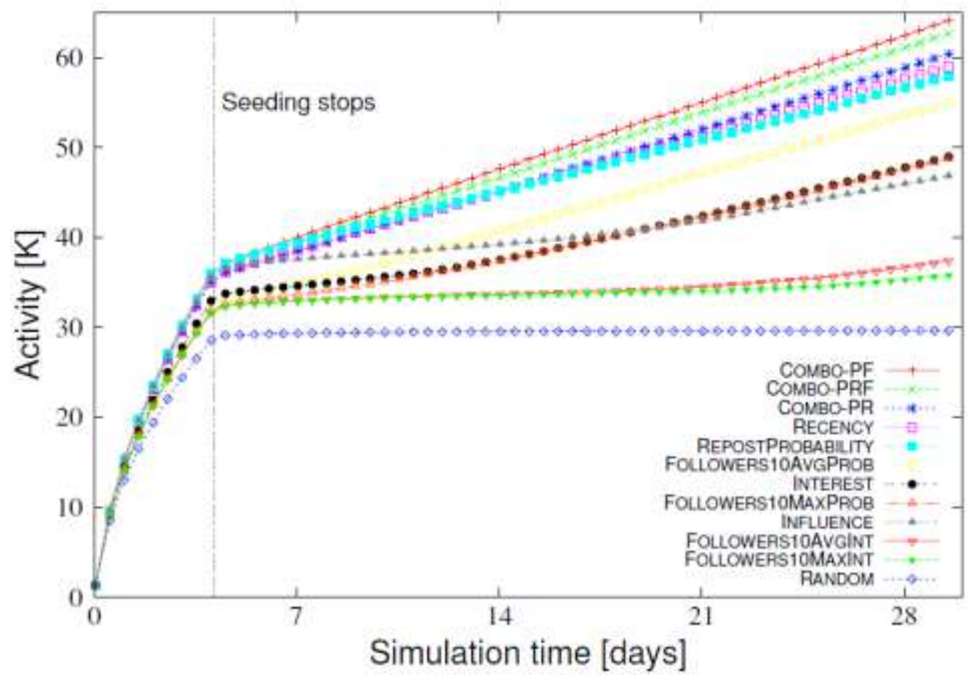
Participation Maximization

Seed allocation and participation maximization

- Multiple independent cascades from seed sets
- Each user can only act as seed for a fixed number of cascades
- Problem: find allocation of seeds to users, to maximize the total size of all cascades
  - online version: allocation has to be done when user logs in
- Applications:
  - Meme ranking
  - Topic recommendation in online discussion forums
  - Online advertising

Application: Meme Ranking

- Users see a selection of k postings by people they follow
    - **Which postings?**
- **Heuristic**: observe what each user and a small sample of her followers have re-posted

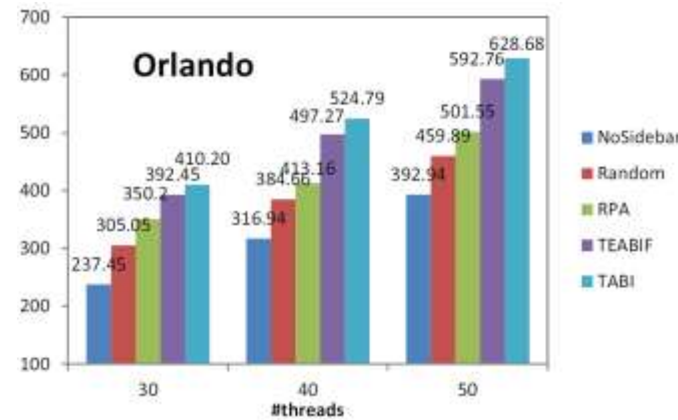


Goal:
maximize total re-posting activity by all users

[Ienco, Bonchi and Castillo, ICDM Workshops 2010]

Application: topic thread rec.

- recommend a small set of topic (or thread) to users on their sidebars
- maximize total participations of all discussion threads
    - diff. from recommender systems: not only increase participation of recommended users, but increase participation of others via social influence



- Theory: social welfare maximization with submodular functions
- RPA (randomized proportional allocation): greedy-based, very slow
- TABI: heuristic considering both self and other participation via influence

[Sun et al. ICWSM 2011]

Paying Attention to Budget and Time

Time critical influence maximization

- achieve influence maximization within a short deadline
- need to model delay in influence diffusion
  - add meeting probabilities of pair of nodes; influence occur only after individuals meet
  - extend IC and LT models, still satisfy submodularity
- fast heuristics (for the IC model extension)
  - MIA-M: need dynamic programming
  - MIA-C: conversion to standard IC model and MIA algorithm

[Chen, Lu and Zhang, AAAI 2012]

Minimizing Expenses

- **MINTSS**: Given a target spread you want to reach, how to pick the fewest seeds that realize the outcome?

  - **Problem.** Given $G = (V, E)$, a threshold η on expected spread, pick the smallest set of seeds S: $\sigma(S) \geq \eta$.
  - For hardness, approximability results and algorithms, see paper!

[Goyal, Bonchi, L., & Venkatasubramanian SNAM 2012]

Minimizing Propagation Time

- **MINTIME**: Given a seed budget and a target spread, pick seeds under budget so the target is realized as quickly as possible.
  - **Problem.** Given $G = (V, E)$, a seed budget k and a threshold η on expected spread, choose k seeds S: $\sigma(S) \geq \eta$ and the time horizon in which this happens is min.
  - For hardness, approximability results and algorithms, see paper!

[Goyal, Bonchi, L., & Venkatasubramanian SNAM 2012]

Part IV Key Takeaways

- Tests exist for homophily/influence
- Influence weights can be learned from data!
- Bypassing model and direct seed selection is possible
- Better models for Adoption/Revenue vs Influence
- Exogenous and endogenous competition can be modeled with care
- Participation maximization considers maximizing multiple influence spreads across an entire platform
- Time and budget can be considered in the objective function