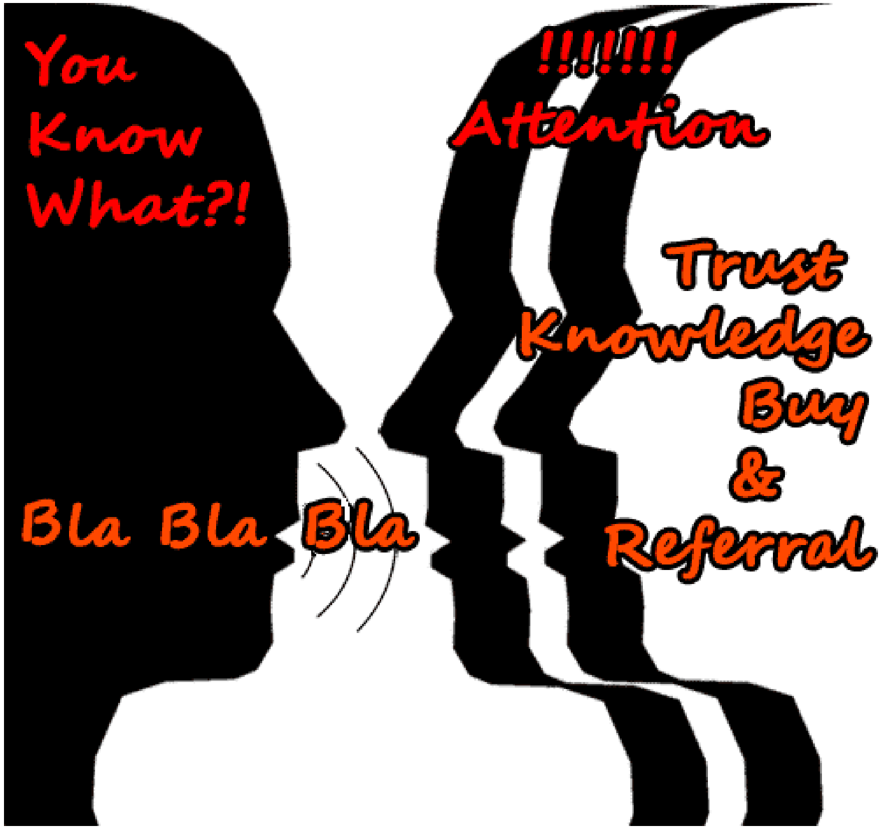
Learning Influence Probabilities in Social Networks

Amit Goyal[1]	U. of British Columbia
Francesco Bonchi[2]	Yahoo! Research
Laks V. S. Lakshmanan[1]	U. of British Columbia

1 2

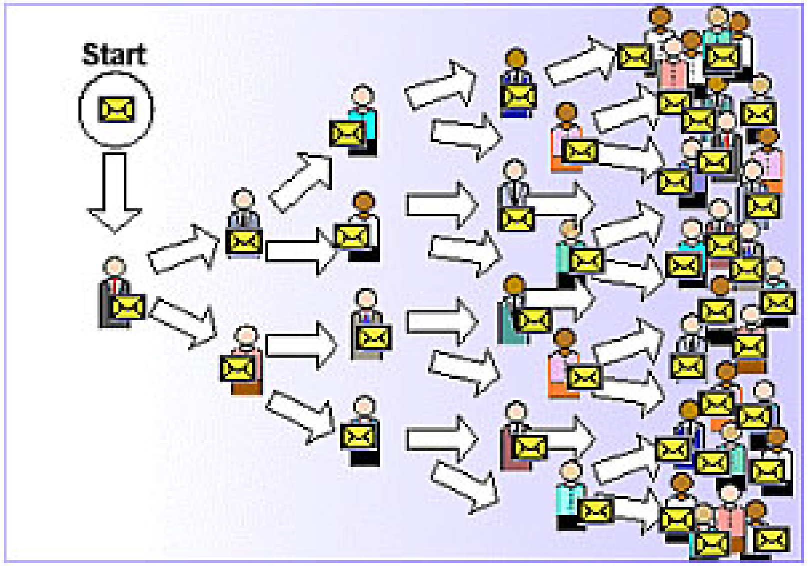
Word of Mouth and Viral Marketing

- We are more influenced by our friends than strangers
- 68% of consumers consult friends and family before purchasing home electronics (Burke 2003)



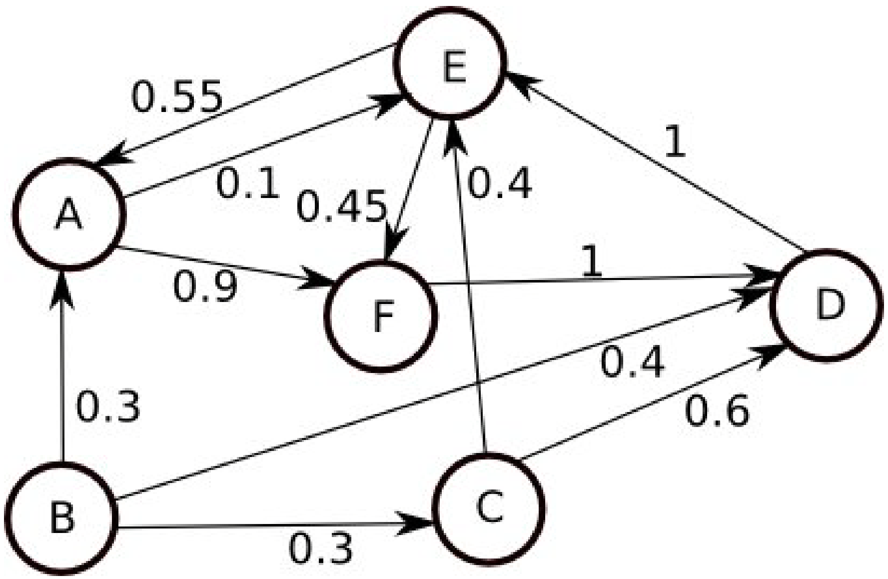
Viral Marketing

- Also known as Target Advertising
- Initiate chain reaction by Word of mouth effect
- Low investments, maximum gain



Viral Marketing as an Optimization Problem

- Given: Network with influence probabilities
- Problem: Select *top-k* users such that by targeting them, the spread of influence is maximized
- *Domingos et al 2001, Richardson et al 2002, Kempe et al 2003*



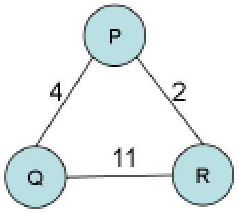
- How to calculate true influence probabilities?

Some Questions

- Where do those influence probabilities come from?
  - Available real world datasets don't have prob.!
- Can we learn those probabilities from available data?
- Previous Viral Marketing studies ignore the effect of time.
  - How can we take time into account?
    - Do probabilities change over time?
  - Can we predict time at which user is most likely to perform an action.
- What users/actions are more prone to influence?

Input Data

- We focus on actions.
- Input:
  - Social Graph: P and Q become friends at time 4.
  - Action log: User P performs actions a1 at time unit 5.



User	Action	Time
P	a1	5
Q	a1	10
R	a1	15
Q	a2	12
R	a2	14
R	a3	6
P	a3	14

Our contributions (1/2)

- Propose several probabilistic influence models between users.
  - Consistent with existing propagation models.
- Develop efficient algorithms to learn the parameters of the models.
- Able to predict whether a user perform an action or not.
- Predict the time at which she will perform it.

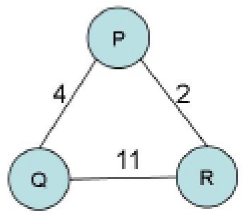
Our Contributions (2/2)

- Introduce metrics of users and actions influenceability.
  - High values => genuine influence.

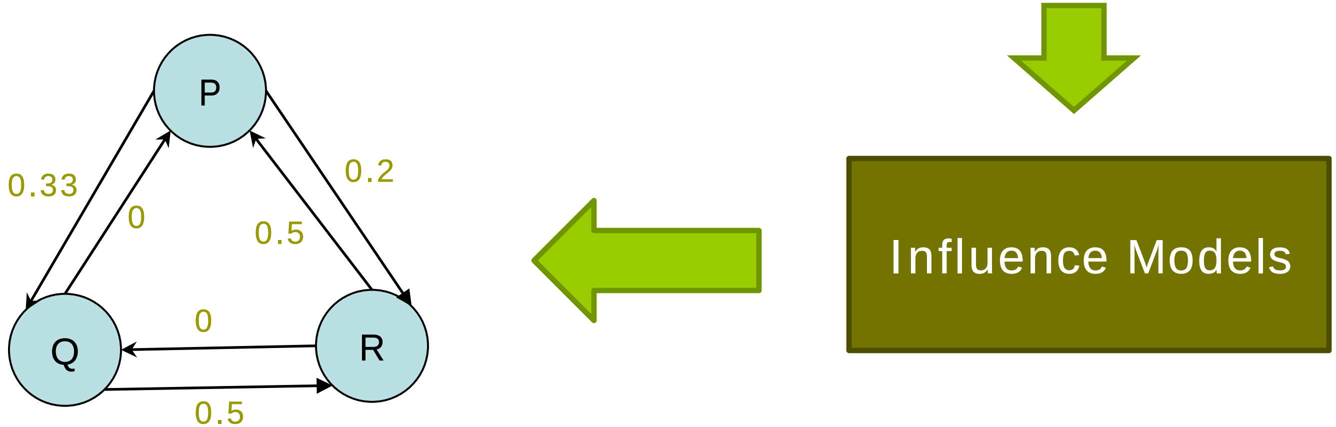
- Validated our models on Flickr.

Overview

- Input:
  - Social Graph: P and Q become friends at time 4.
  - Action log: User P performs actions a1 at time unit 5.



User	Action	Time
P	a1	5
Q	a1	10
R	a1	15
Q	a2	12
R	a2	14
R	a3	6
P	a3	14



Background

General Threshold (Propagation) Model

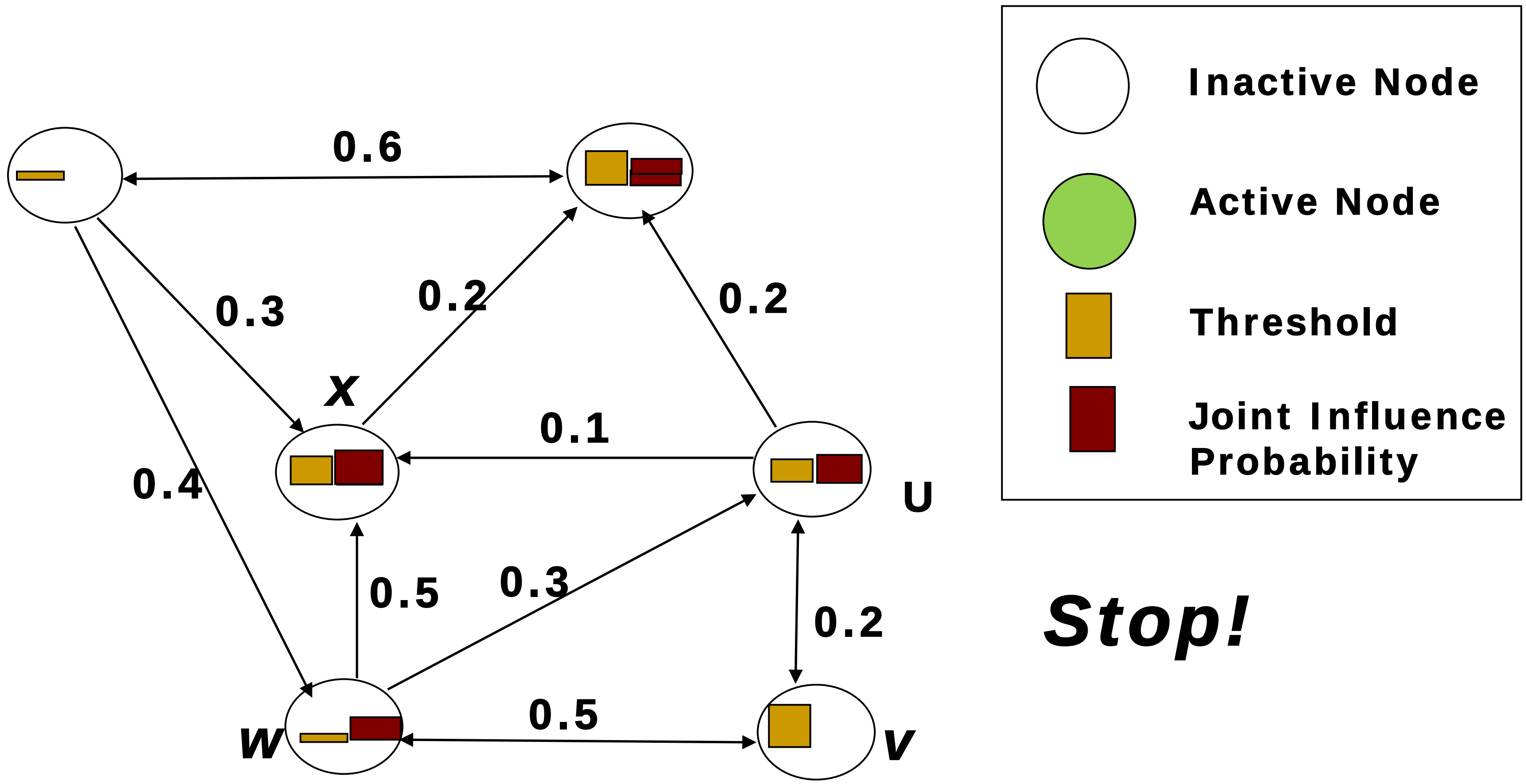
- At any point of time, each node is either active or inactive.

- More active neighbors => u more likely to get active.

- Notations:
  - S = {active neighbors of u}.
  - $p_u(S)$: Joint influence probability of S on u.
  - Θ_u: Activation threshold of user u.

- When $p_u(S) >= \Theta_u$, u becomes active.

General Threshold Model - Example



Stop!

Source: David Kempe's slides

Our Framework

Solution Framework

- Assuming independence, we define

$$p_u(S) = 1 - \prod_{v \in S}(1 - p_{v,u})$$

  - $p_{v,u}$: influence probability of user v on user u
- Consistent with the existing propagation models – monotonocity, submodularity.
- It is incremental. i.e. $p_u(S \cup \{w\})$ can be updated incrementally using $p_u(S)$ and $p_{w,u}$
- Our aim is to learn $p_{v,u}$ for all edges.

Influence Models

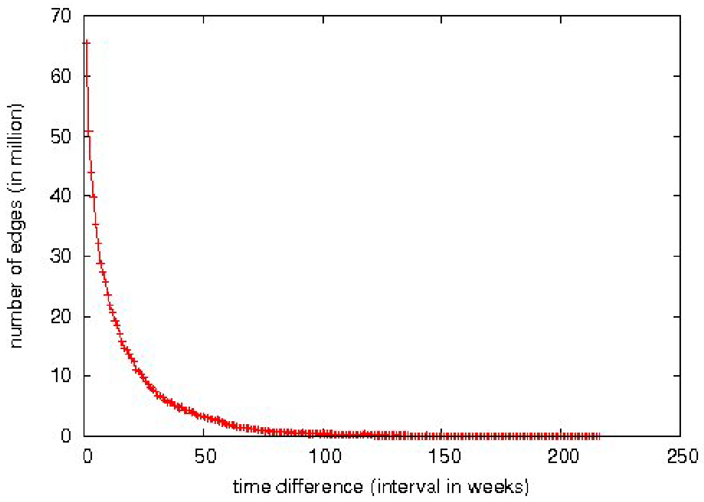
- Static Models
  - Assume that influence probabilities are static and do not change over time.
- Continuous Time (CT) Models
  - Influence probabilities are continuous functions of time.
  - Not incremental, hence very expensive to apply on large datasets.
- Discrete Time (DT) Models
  - Approximation of CT models.
  - Incremental, hence efficient.

Static Models

- 4 variants
  - Bernoulli as running example.
- Incremental hence most efficient.
- We omit details here

Time Conscious Models

- Do influence probabilities remain constant independently of time?
- We propose Continuous Time (CT) Model
  - Based on exponential decay distribution



NO

Continuous Time Models

- Best model.
- Capable of predicting time at which user is most likely to perform the action.
- Not incremental
  - Discrete Time Model
    - Based on step time functions
    - Incremental

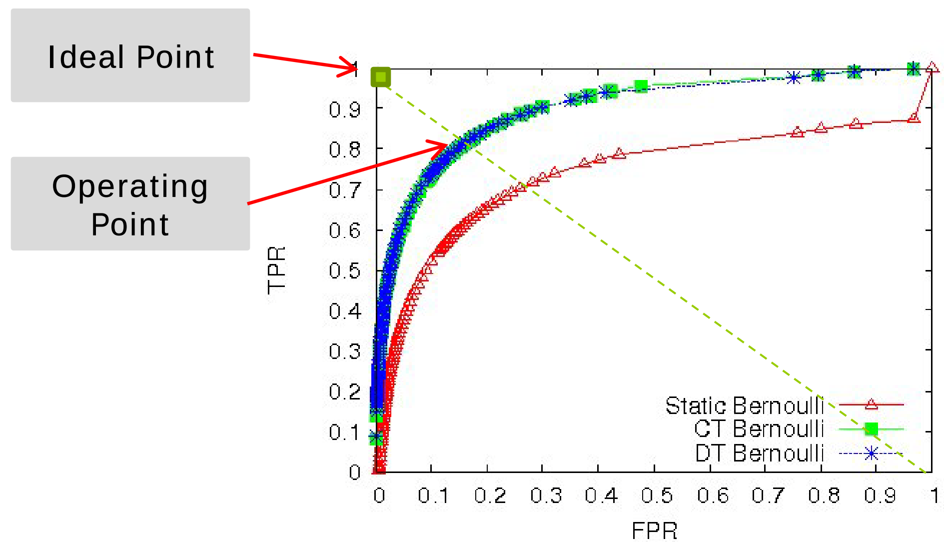
Evaluation Strategy (1/2)

- Split the action log data into training (80%) and testing (20%).
  - User "James" have joined "Whistler Mountain" community at time 5.
- In testing phase, we ask the model to predict whether user will become active or not
  - Given all the neighbors who are active
  - Binary Classification

Evaluation Strategy (2/2)

- We ignore all the cases when none of the user's friends is active
  - As then the model is inapplicable.
- We use ROC (Receiver Operating Characteristics) curves
  - True Positive Rate (TPR) vs False Positive Rate (FPR).
  - TPR = TP/P
  - FPR = FP/N

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



Algorithms

- Special emphasis on efficiency of applying/testing the models.
  - Incremental Property

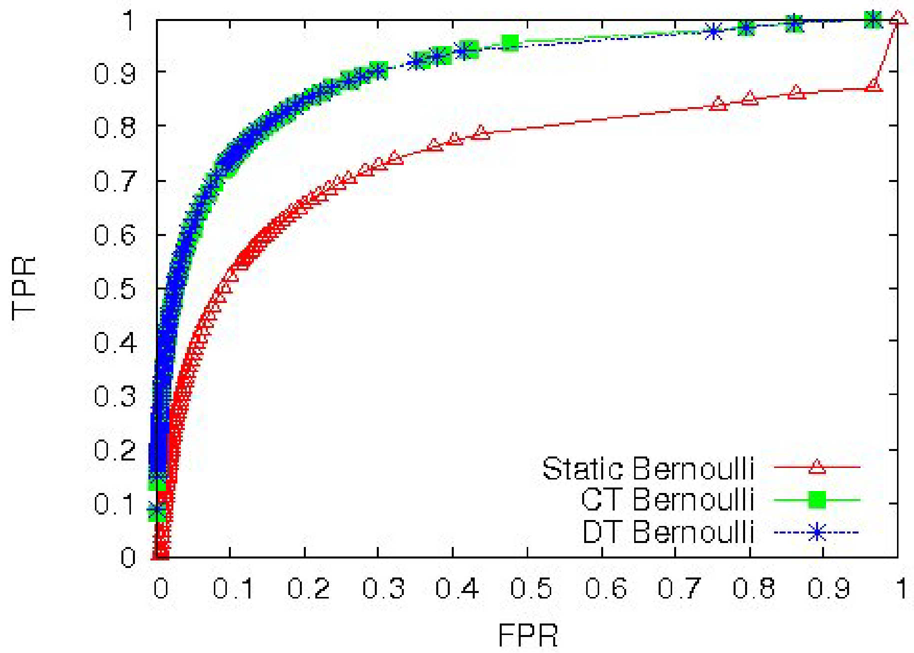
- In practice, action logs tend to be huge, so we optimize our algorithms to minimize the number of scans over the action log.
  - Training: 2 scans to learn all models simultaneously.
  - Testing: 1 scan to test one model at a time.

Experimental Evaluation

Dataset

- Yahoo! Flickr dataset
- “Joining a group” is considered as action
  - User “James” joined “Whistler Mountains” at time 5.
- #users ~ 1.3 million
- #edges ~ 40.4 million
- Degree: 61.31
- #groups/actions ~ 300K
- #tuples in action log ~ 35.8 million

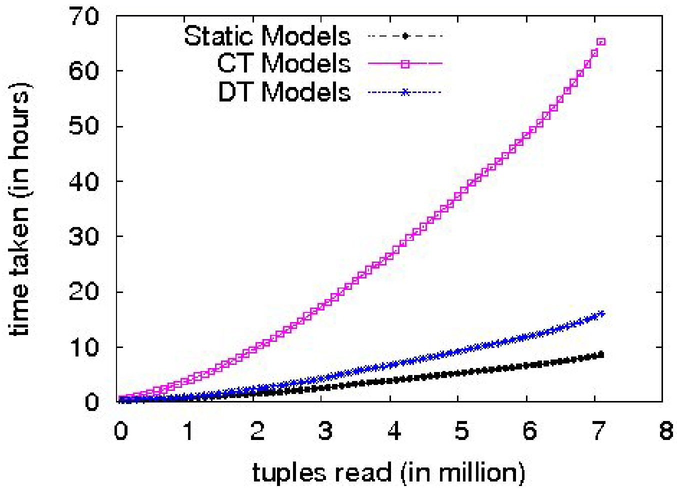
Comparison of Static, CT and DT models



- Time conscious Models are better than Static Models.
- CT and DT models perform equally well.

Runtime

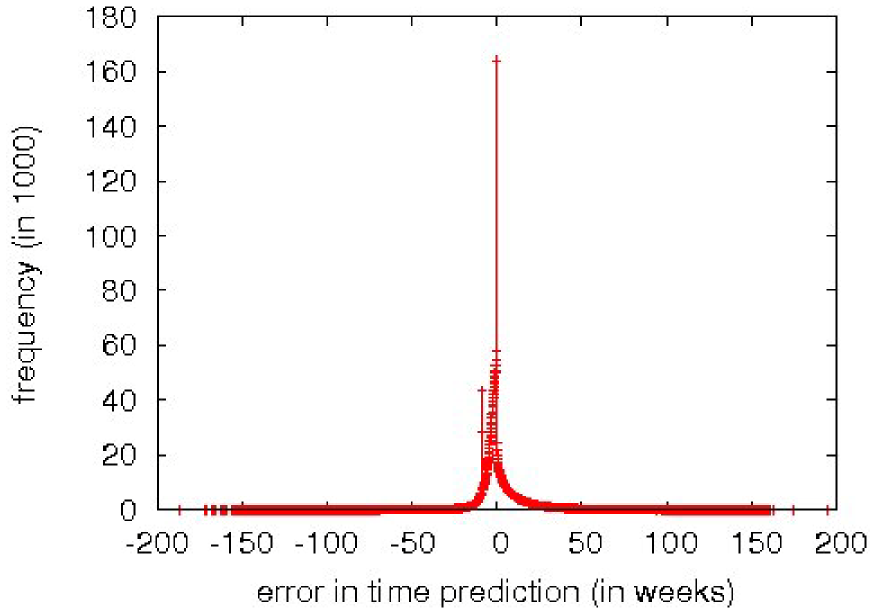
Testing



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

Predicting Time – Distribution of Error

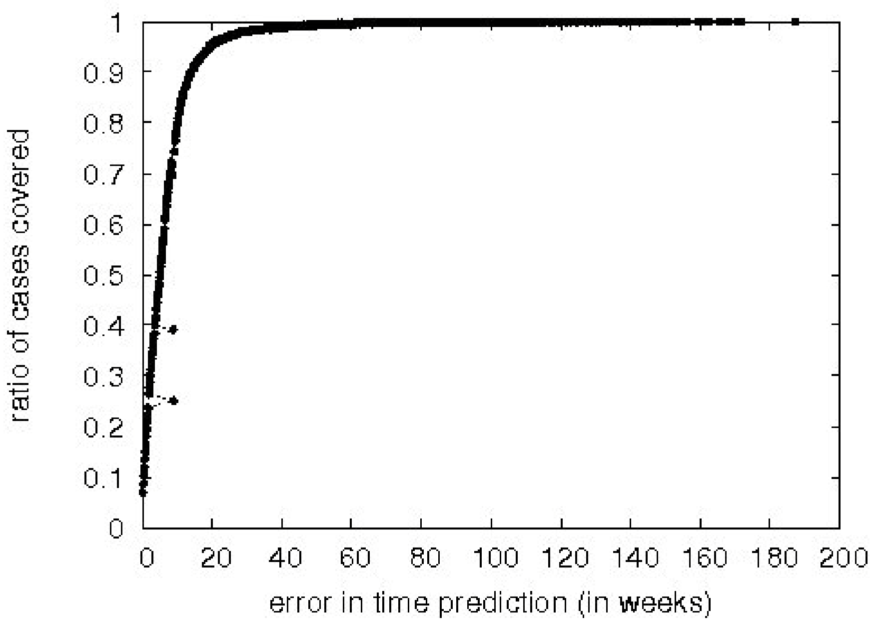
- Operating Point is chosen corresponding to
  - TPR: 82.5%, FPR: 17.5%.



- 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

Predicting Time – Coverage vs Error

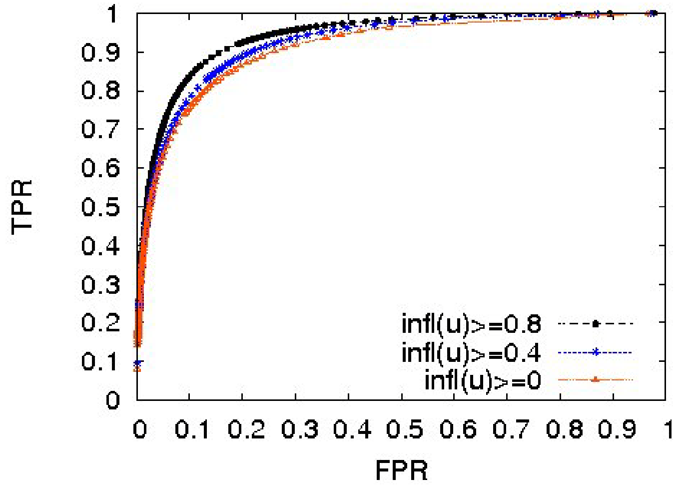
- Operating Point is chosen corresponding to
  - TPR: 82.5%, FPR: 17.5%.



- A point (x,y) here means for y% of cases, the error is within $\pm x$
- In particular, for 95% of the cases, the error is within 20 weeks.

User Influenceability

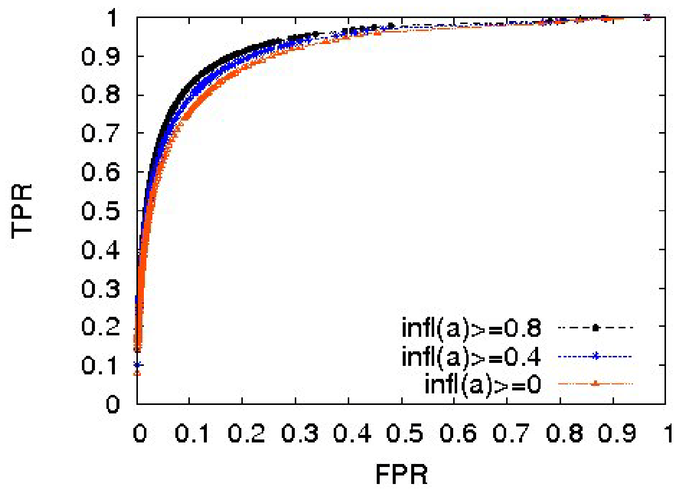
- Some users are more prone to influence propagation than others.
- Learn from Training data



- Users with high influenceability => easier prediction of influence => more prone to viral marketing campaigns.

Action Influenceability

- Some actions are more prone to influence propagation than others.



- Actions with high user influenceability => easier prediction of influence => more suitable to viral marketing campaigns.

Related Work

- Independently, Saito et al (KES 2008) have studied the same problem
  - Focus on Independent Cascade Model of propagation.
  - Apply Expectation Maximization (EM) algorithm.
  - Not scalable to huge datasets like the one we are dealing in this work.

Other applications of Influence Propagations

- Personalized Recommender Systems
  - Song et al 2006, 2007
- Feed Ranking
  - Samper et al 2006
- Trust Propagation
  - Guha et al 2004, Ziegler et al 2005, Golbeck et al 2006, Taherian et al 2008

Conclusions (1/2)

- Previous works typically assume influence probabilities are given as input.
- Studied the problem of learning such probabilities from a log of past propagations.
- Proposed both static and time-conscious models of influence.
- We also proposed efficient algorithms to learn and apply the models.

Conclusions (2/2)

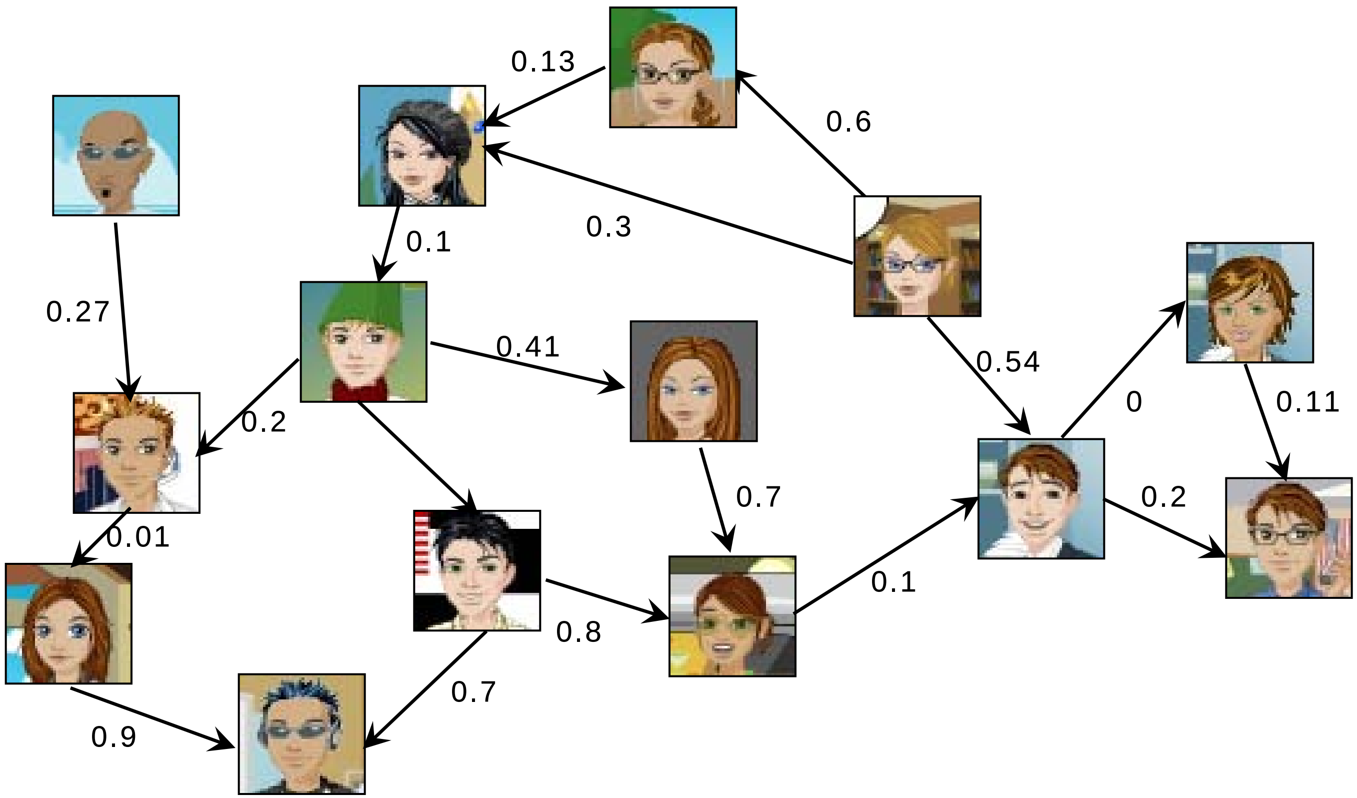
- Using CT models, it is possible to predict even the time at which a user will perform it with a good accuracy.

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

Future Work

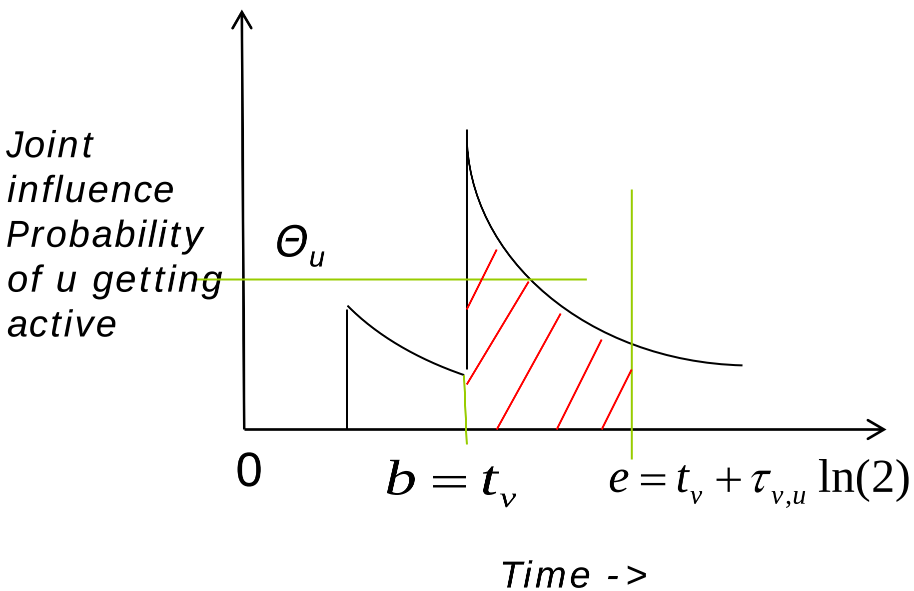
- Learning optimal user activation thresholds.
- Considering users and actions influenceability in the theory of Viral Marketing.
- Role of time in Viral Marketing.

Thanks!!



Predicting Time

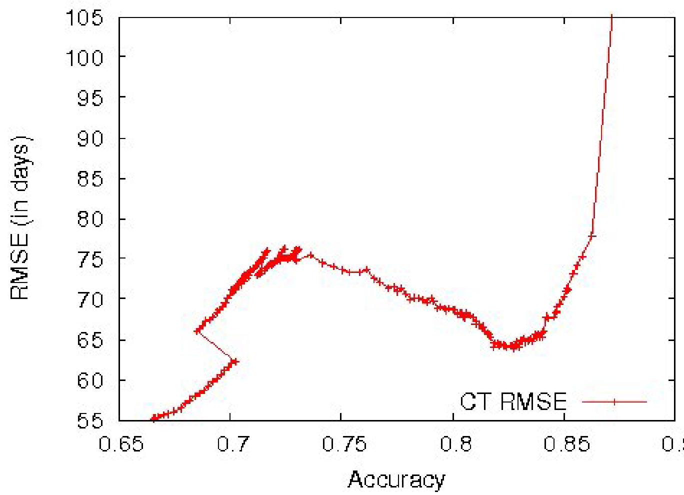
- CT models can predict the time interval [b,e] in which she is most likely to perform the action.



- $\tau_{v,u} \ln(2)$ is half life period
- Tightness of lower bounds not critical in Viral Marketing Applications.
- Experiments on the upper bound e.

Predicting Time - RMSE vs Accuracy

- CT models can predict the time interval [b,e] in which user is most likely to perform the action.
  - Experiments only on upper bound e.
- Accuracy = $\dfrac{\#\,\text{cases when the prediction of upper bound is correct}}{\#\,\text{total cases}}$
- RMSE = root mean square error
- RMSE ~ 70-80 days



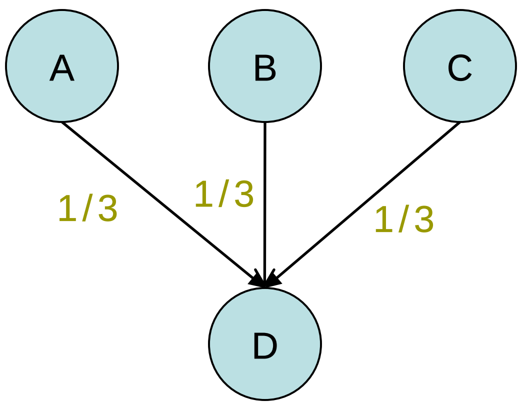
Static Models – Jaccard Index

- Jaccard Index is often used to measure similarity b/w sample sets.

- We adapt it to estimate $p_{v,u}$

$$p_{v,u} = \frac{\#\text{of actions propagated from v to u}}{\#\text{of actions performed by v or u}}$$

Partial Credits (PC)



- Let, for an action, D is influenced by 3 of its neighbors.
- Then, 1/3 credit is given to each one of these neighbors.

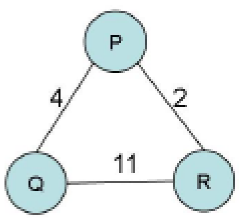
PC Bernoulli ⇒ $$p_{v,u} = \frac{\text{total credits accumulated}}{\text{total number of actions v performed}}$$

PC Jaccard ⇒ $$p_{v,u} = \frac{\text{total credits accumulated}}{\text{total number of actions v or u performed}}$$

Learning the Models

- Parameters to learn:
  - #actions performed by each user – A_u
  - #actions propagated via each edge – A_{v2u}
  - Mean life time – $\tau_{v,u}$

u	A_u
P	0 / 1 / 2
Q	0 / 1 / 2
R	0 / 1 / 2 / 3



P	a1	5
Q	a1	10
R	a1	15
Q	a2	12
R	a2	14
R	a3	6
P	a3	14

Input

	P	Q	R
P	X	0,0 / 1,5	0,0 / 1,10
Q	0,0	X	0,0 / 1,2 / 2,2
R	0,0 / 1,8	0,0	X

$\mathrm{A}_{\mathrm{v,u}}, \tau_{v,u}$

Propagation Models

- Threshold Models
  - Linear Threshold Model
  - General Threshold Model

- Cascade Models
  - Independent Cascade Model
  - Decreasing Cascade Model

Properties of Diffusion Models

- Monotonocity

$$p_u(S) \leq p_u(T) \text{ whenever } S \subseteq T$$

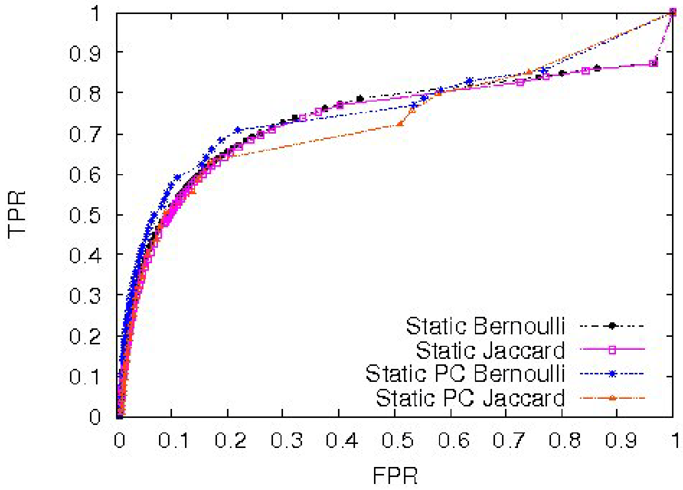
- Submodularity – Law of marginal Gain

$$p_u(S \cup \{w\}) - p_u(S) \geq p_u(T \cup \{w\}) - p_u(T)$$
$$\text{whenever } S \subseteq T$$

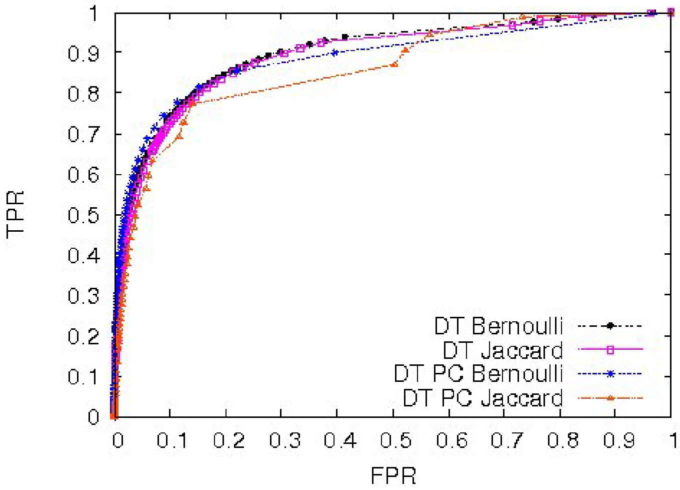
- Incrementality (Optional)

  $p_u(S \cup \{w\})$ can be updated incrementally using $p_u(S)$ and $p_{w,u}$

Comparison of 4 variants



ROC comparison of 4 variants of Static Models

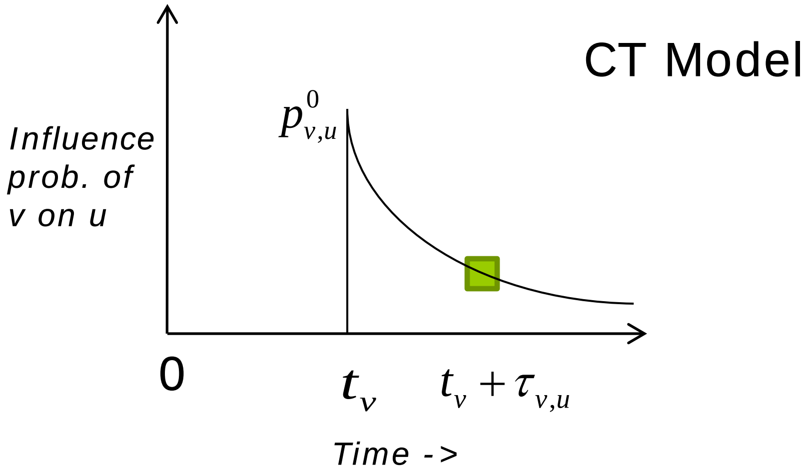


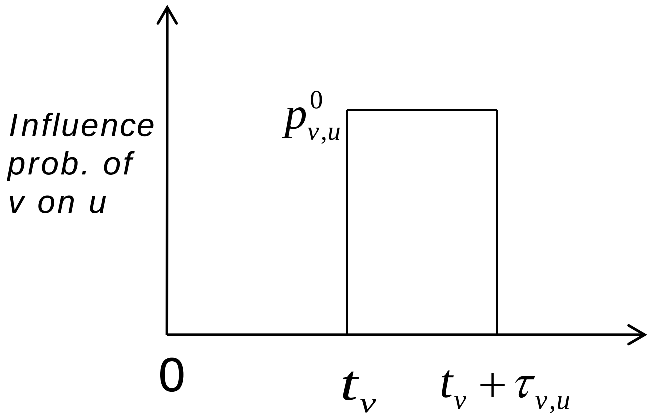
ROC comparison of 4 variants of Discrete Time (DT) Models

- Bernoulli is slightly better than Jaccard
- Among two Bernoulli variants, Partial Credits (PC) wins by a small margin.

Discrete Time Models

- Approximation of CT Models
- Incremental, hence efficient
- 4-variants corresponding to 4 Static Models





Overview

- Context and Motivation
- Background
- Our Framework
- Algorithms
- Experiments
- Related Work
- Conclusions

Continuous Time Models

- Joint influence probability

$$p_u^t(S) = 1 - \prod_{v \in S}(1 - p_{v,u}^t)$$

- Individual probabilities – exponential decay

$$p_{v,u}^t = p_{v,u}^0 e^{-(t - t_u)/\tau_{v,u}}$$

  - $p_{v,u}^0$: maximum influence probability of v on u
  - $\tau_{v,u}$: the mean life time.

Algorithms

- Training – All models simultaneously in no more than 2 scans of training sub-set (80% of total) of action log table.
- Testing – One model requires only one scan of testing sub-set (20% of total) of action log table.
- Due to the lack of time, we omit the details of the algorithms.