Preserving Privacy and Social Influence

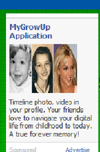
Isabelle Stanton

Social Network Privacy

- Bakstrom, Dwork and Kleinberg show removing names isn't enough
- Present passive, semi-active and active attacks against social networking graphs

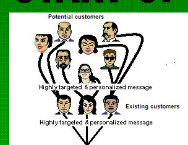
Economic Motivation

facebook



=$

START UP

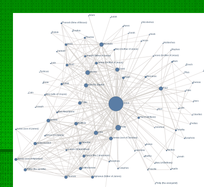


- Wants to sell ads
- Viral Marketing needs social graph

Economic Motivation

facebook

START UP



- Uses their viral marketing algorithm to find influential people
- Executes Bakstrom attack to identify them

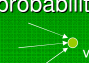
~~$~~

Goal of the Project

- Develop a perturbation scheme that preserves privacy of individuals while also approximately preserving their influence

Influence

- Modeled as a weighted graph *G=(V,E),* where $p_{u,v}$ is the probability that *u* influences *v.*
- $p_{u,v} \geq 0$
- For each v, sum of incoming probabilities at most 1,
  - For each v, $\Sigma_u \; p_{u,v} \leq 1$



- Influence of a node: Expected number of active nodes

Obtaining the (Indirect) Influence Graph

- Ask each user to rate how their friends influence them.
- Put into a matrix A
- A^2 is how a node indirectly influences its' friends' friends.
- Corresponds to a Markov Process
- $I = \Sigma A^k$

Privacy Definitions

- Def 1: If an attacker knows all the values in the original I except u then:

$$1-\varepsilon < \frac{\Pr(w(u)\in[x,y] \mid I)}{\Pr(w(u)\in[x,y])} < 1+\varepsilon$$

- Def 2: Given a perturbed version of I, I', and an edge u, the weight of u shouldn't affect I' much $1-\varepsilon < \frac{\Pr(I \mid w(u)\in[x,y])}{\Pr(I \mid w(u)\in[s,t])} < 1+\varepsilon$

Perturbation ideas

- Randomly select a value within [0,1] for each edge weight, then normalize
  - Preserves privacy but is obviously useless for preserving influence
- Randomly select a value in [1-ε, 1+ε] for each edge and multiply.
  - Influence for each node is within $(1+\varepsilon)^n$ but privacy is not preserved by any definition

My Idea

- The Influence graph is calculated as a Markov process
- A small change initially will result in a large change in the end
- Perturb the original graph instead of the end product

Original Graph Perturbation

- Nodes in clusters have approximately equal influence
- Cluster the graph
- For each inter-cluster edge, select new nodes in the cluster to assign the edge to
- Add and remove some small fraction of inter-cluster edges

- No proofs today
- Any suggestions?