

Large-Scale Online Recommendation on Graphs

Siddhartha Banerjee

In collaboration with Praneeth Netrapalli, Sujay Sanghavi and Sanjay Shakkottai
The University of Texas at Austin

BIRS Workshop, February 26, 2013

Example: News-Feeds in Social Networks

The screenshot shows a Facebook News Feed interface. At the top, there's a navigation bar with 'facebook' on the left and 'Home Profile Account' on the right. Below the navigation bar is a search bar and a 'News Feed' header. The main content area displays a list of posts from various users, including Jim Naysium, Johnny, Phil, Martha, Jen, Doghouse Diaries, Vanessa, James, Victor, and Jasmine. Each post includes a profile picture, name, and a short text snippet. On the right side, there are sections for 'Events' and 'You Don't Have Enough Friends'. The 'Events' section shows a list of events with RSVP options. The 'You Don't Have Enough Friends' section shows a list of people and a 'Find Friends' button.

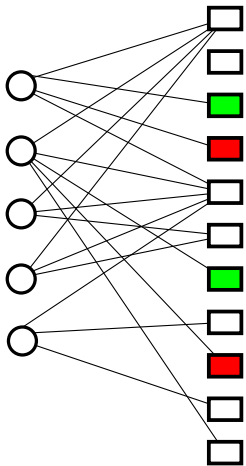
- Supply >> Demand
- Arbitrary rewards
- User feedback
- Access graph

Source: thedoghousediaries.com

- Other examples: targeted ads, personalized content curation

Online Recommendation on Graphs

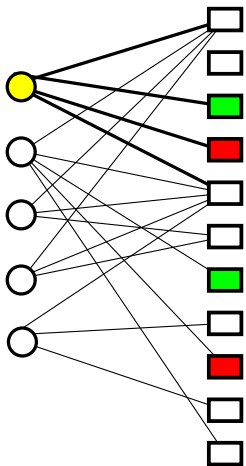
A first-cut, static model



- Known user-item access graph
- Unknown reward-function
- Users arrive randomly
- Want algorithms that are *competitive* with respect to optimum reward (i.e., with full knowledge)

Online Recommendation on Graphs

A first-cut, static model



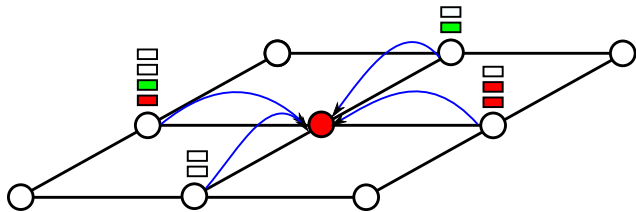
- Known user-item access graph
- Unknown reward-function
- Users arrive randomly
- Want algorithms that are *competitive* with respect to optimum reward (i.e., with full knowledge)
 - Random $\Rightarrow \gamma = \frac{1}{d_{max}}$ (for any reward-function)
 - Can we do better?

Summary of our Results

- Can do much better
 - Using some pre-processing: $\gamma = O(1/\text{makespan})$
 - This is **orderwise optimal**
- Alternate **distributed algorithm**
 - 'Samples' item i for exploration w.p. $1/d_i$
 - γ better than random, near optimal under regularity conditions
- Results hold for **large class of functions** (structural condition)
- **Scaling behavior** for showing r items
 - Linear scaling in r (optimal)
- **Dynamic settings**

Online Recommendations on Graphs

Dynamic models with node arrivals/departures



- Users arrive repeatedly (independent Poisson(1) processes)
- Items arrive to item-classes, depart after some lifetime

Thanks!