Predicting and Steering Social Systems

Team members:
Jon Kleinberg    Jure Leskovec

With Lada Adamic, Ashton Anderson, Justin Cheng, Alex Dow, and Dan Huttenlocher
Increasingly we learn information through channels that are simultaneously on-line and social.

- Social sharing of information with friends.
- Collective production of knowledge.
- Question-answering by self-identified experts.

Rich source of data via digital traces.

And: these are designed social systems.

- Algorithmic filtering of content.
- Reward systems to motivate participation and effort.
Algorithmic filtering based on prediction.

- Your friends share many things over the course of a day.
- Can we predict which will be the most popular?
- We can use reshares as one measure of popularity.

Prediction as an ingredient in:

- Steering the social networking system
- Trying to understand the principles behind cascades.
Predicting Reshares
Population of cascades is highly skewed.

- Only 5% of all photos are ever reshared at all.
- But over 50% of all photo reshares occur on photos with > 500 reshares.

Predicting eventual size of a cascade is a pathological task.

- High accuracy via a trivial answer (1).
- Creating a balanced variant leads to a highly artificial distribution.

Cascade growth prediction.

- Let \( f(k) \) be median size of cascade conditional on reaching size \( k \).
- Observation on reshare cascades: \( f(k) \approx 2k \) for all \( k \).
- Given a cascade up to a certain point in time, of size \( k \), predict whether it will reach size \( f(k) \).
- Balanced task, and performance can be parametrized by \( k \).
Cascade Growth Prediction

- Given a cascade up to a certain point in time, of size $k$, predict whether it will reach size $f(k)$.

Categories of features:

- **Content**: objects in photo; caption; pos/neg emotion in photo text
- **Root**: page vs. user; degree; age/gender; facebook-age; activity level
- **Resharer**: average user properties over first $j$ resharers
- **Structural**: degrees; induced subgraph properties on first $j$ resharers; tree properties on first $j$ resharers; how many resharers escape root’s neighborhood
- **Temporal**: time until $j^{th}$ reshare; acceleration parameters
Cascade Growth Prediction

Some general observations:

- Accuracy increases with $k$.
- Temporal features very powerful, and remain important as $k$ grows.
- Features of content and original poster get less important with increasing $k$.
- High resharer depth predicts larger growth.
Further Questions

- Can fix content, vary network starting point.
- Alternately: fix root, vary content.
- Different types of content activate different sub-populations
- Clustering among resharers?
- Methodological challenge: prediction to gauge which features are important.

Next, a different design problem: motivating users via rewards.
Systems of rewards are a crucial feature of many domains:

- Government, Military
- Scientific community
- On-line participatory settings

Many can be seen as “badges.”
Badges

On-line domain has embraced the use of badges as rewards.

- Can recognize skills and achievements
- Can encourage participation and contribution

Multiple social-psychological dimensions [Antin-Churchill 2011]

- Goal-setting, instruction, reputation, status, affirmation, group identification
Focus here on incentive properties of rewards and badges.

- Part of the broad trend toward gamification [Deterding et al 2011]; see also [Easley-Ghosh 2013]

Many other approaches: e.g.

- contest/auction-based [DiPalatino-Vojnović 09, Cavallo-Jain 12, 13, Chawla-Hartline-Sivan 12]

- elicitation and evaluation of quality [Ghosh-McAfee 11, Mao et al 13, Witkowski et al 13]
I am trying to get the list of connected components in a graph with 100 million nodes. For smaller graphs, I usually use the `connected_components` function of the Networkx module in Python which does exactly that. However, loading a graph with 100 million nodes (and their edges) into memory with this module would require ca. 110GB of memory, which I don't have. An alternative would be to use a graph database which has a connected components function but I haven't found any in Python. It would seem that Dex (API: Java, .NET, C++) has this functionality but I'm not 100% sure. Ideally I'm looking for a solution in Python. Many thanks.

SciPy has a `connected_components` algorithm. It expects as input the adjacency matrix of your graph in one of its `sparse matrix formats` and handles both the directed and undirected cases.

Building a sparse adjacency matrix from a sequence of \((i, j)\) pairs `adj_list` where \(i\) and \(j\) are (zero-based) indices of nodes can be done with

```
ij_indices = indices = np.array(*adj_list)
```
Our Model

A population of users and a site designer.

- Designer would like a certain frequency of activities.
- Designer creates badges, which have value to users.

- User trades off between preferred mix of activities and reaching the badge.
- We’d like to see this produce effects on both engagement and “steering” – balancing activities differently.
Our Model

- Action types $A_1, A_2, \ldots, A_n$. (ask, answer, vote, off-site, ...)
- User’s state is $n$-dimensional.
- User has preferred distribution $p$ over action types.
- User exits system with probability $\delta > 0$ each step.

- Each badge $b$ is a monotone subset of the state space; reward $V_b$ is conferred when the user enters this subset.
- User can pick distribution $x \neq p$ to get badge more quickly; comes at a cost $g(x, p)$.
- Sets up an optimization problem for the user.
What a Solution Looks Like

![Graph showing the number of $A_1$ actions vs the number of $A_2$ actions. The graph illustrates the solution space, where each point represents a combination of actions. The x-axis represents the number of $A_1$ actions, ranging from 0 to 20, and the y-axis represents the number of $A_2$ actions, ranging from 0 to 20. The graph is populated with arrows indicating possible combinations of actions.](image-url)
A One-Dimensional Version

Example: Badge at 25 actions of type 1.

- Canonical behavior: user “steers” in $A_1$ direction; then resets after receiving the badge.
Evaluating Qualitative Predictions

Consider two cumulative badges on StackOverflow.

- **Civic Duty badge**: Vote at least 300 times.
- **Electorate**: Vote on at least 600 questions (plus some other technical conditions).

5-dimensional state space: \((Q, A, Q\text{-vote}, A\text{-vote}, \text{off-site})\).

![Graphs showing Civic Duty and Electorate badges](image)
The Badge Placement Problem

General question: how should you “place” badges in action space to achieve desired effects?

- Current collaboration with Coursera on badge placement.

Concrete question: Suppose you can define a badge $b$ of value $V_b$, and you want to achieve an aggregate action distribution of $q$. Where should you place the badge?

Example: Place badge on type 1 to achieve max number of type 1 actions.

$\delta = 0.01$ in example.
An Experiment on Coursera

Top byline:

Thread byline:

Badge ladder:

**Badge Series (2 earned)**

<table>
<thead>
<tr>
<th>Badge Series</th>
<th>BRONZE</th>
<th>SILVER</th>
<th>GOLD</th>
<th>DIAMOND</th>
</tr>
</thead>
<tbody>
<tr>
<td>The Reader</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>The Supporter</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>The Contributor</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>The Conversation Starter</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Top Posts</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

*The Reader*
To earn the next badge (Silver), you must read 30 threads from your classmates.

*The Supporter*
To earn the next badge (Silver), you must vote on 15 posts that you find interesting or useful.

*The Contributor*
To earn the next badge (Bronze), you must post 3 replies that your classmates find interesting.

*The Conversation Starter*
To earn the next badge (Bronze), you must start 3 threads that your classmates find interesting.

*Top Posts*
To earn the next badge (Bronze), you must write a post that gets 5 upvotes from your classmates.
Multiple ways in which rewards can create incentives.

- Choice of threshold: absolute standard or top-$k$ [Easley-Ghosh 2013].
- Incentivizing certain rates/speeds of activity [Ghosh-Kleinberg 2013].
- Estimation of parameters from trace data, for purposes of designing badges or other incentives.
- Where does the value of rewards/badges/credit come from? Intrinsic sense of progress, social interaction, ...?