The Social Hourglass: Enabling Socially-aware Applications and Services

Adriana Iamnitchi
University of South Florida
anda@cse.usf.edu
Much Social Information Available

- Connects people through relationships
  - Object centric: use of same objects
  - Person centric: declared relationships or co-participation in events, groups, etc.

Logos of various social media platforms are shown.
Scenario 1: CallCensor

Want to go for a beer?

BUY! SELL!
Scenario 2: Sofa Surfer
Scenario 3: Data Placement
Current Approach: Vertically Integrated Socially-aware Applications

- Spam filtering
- Sybil identification
- Personalized search
- Target marketing
- Medical emergency notifications
- ...

Data Source → Application
So, What’s Wrong?

• Application-limited collection and use of social information
  – High bootstrap cost
  – Limited (potentially inaccurate) information. E.g., Information from online social networks
    • Hidden incentives to have many “friends”
    • All relationships equal
    • Symmetric relationships

• Newer proposals to merge different sources of social (and sensor) information for one app
Our Proposal: An Infrastructure for Social Computing

- Sofa Surfer
- Roommate Finder
- CallCensor

Social Computing Platforms:
- Facebook
- Brightkite
- YouTube
- Loopt
- Yelp
We want an infrastructure that:

• Can fuse information from various sources
  – Existing or future
• Allow user to control own information
  – What is collected
  – Where it is stored
  – Who can access it
• Provide social *knowledge* to a variety of applications:
  – Social inferences (may be non-trivial)

The Social Hourglass: an Infrastructure for Socially-aware Applications and Services, Iamnitchi et al., *IEEE Internet Computing*, May/June 2012
The Social Hourglass Architecture

Social signals encode social data
Social sensors analyze social signals and quantify social relations
Personal aggregators collect, fuse, and personalize sensor output
Social knowledge service provides a mechanism for storing social data. Stored data can be accessed by applications through an API
Applications make use of social data accessed via social knowledge service

Applications
Social Inference API
Social Data Management
Personal Aggregators
Social Sensors
Social Signals
Social Sensors

Consume existing social signals
• Location
• Collocation
• Schedule (e.g., Google calendar)
• Mobile phone activity (calls, sms)
• Online social network interactions
• Email
• Personal relations (family)
• Shared content
• Shared interest (e.g., CiteULike)
• …
Work in Progress: Social Sensor for Gaming Interactions

What’s difficult:
- Variability in playing habits
- Variability in playing skills
- Time patterns
Aggregators

- Act as the user’s personal assistant
- Runs on trusted device (cell phone)
- Responsible for
  - Managing passwords for various applications
  - Personalization
  - Identity management
Social signals encode social data. Social sensors analyze social signals and quantify social relations. Personal aggregators collect, fuse, and personalize sensor output. Social knowledge service provides a mechanism for storing social data. Stored data can be accessed by applications through an API. Applications make use of social data accessed via social knowledge service.
Social Graph
Distributed Social Graph

<table>
<thead>
<tr>
<th>User ID</th>
<th>Owned Peers</th>
<th>Trusted Peers</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>1,2</td>
<td></td>
</tr>
<tr>
<td>B</td>
<td>1,2</td>
<td></td>
</tr>
<tr>
<td>C</td>
<td>1,2,3</td>
<td></td>
</tr>
<tr>
<td>D</td>
<td>2,3,4,5,6</td>
<td></td>
</tr>
<tr>
<td>E</td>
<td>3,4</td>
<td></td>
</tr>
<tr>
<td>F</td>
<td>3,5,6</td>
<td></td>
</tr>
<tr>
<td>G</td>
<td>4</td>
<td></td>
</tr>
<tr>
<td>H</td>
<td>5,6,7</td>
<td></td>
</tr>
</tbody>
</table>

Peer 1

Peer 5
Prometheus: User-Controlled P2P Social Data Management for Socially-Aware Applications, Kourtellis et al., Middleware 2010
Social Inference Functions

The social graph management service exports an API that implement social inferences.
API for Applications: Social Inference Functions

• 5 basic social inference functions:
  • `relation_test (ego, alter, a, w)`
  • `top_relations (ego, a, n)`
  • `neighborhood (ego, a, w, radius)`
  • `proximity (ego, a, w, radius, distance)`
  • `social_strength (ego, alter)`

• More complex functions can be built
Social Strength

- Quantifies strength between ego and alter
- Result normalized to consider overall activity
- Search all paths of maximum 2 social hops
- One approach to quantify social strength. Others are certainly possible.

\[
NW(ego, neighbor_i) = \frac{\sum_{all-labels} w(ego, neighbor_i)}{\max_{all-neighbors} \left( \sum_{all-labels} w(ego, neighbor) \right)}
\]

\[
strength(path(N_1, N_2, \ldots, N_K)) = \max_{all-paths} \frac{\min_{s=1..K-1} \left( NW(N_s, N_{s+1}) \right)}{K}
\]
Thank you!

Work funded by US National Science Foundation grants CNS-0831785 and CNS-0952420