An Incentive-Compatible Distributed Recommendation Model

José M. Vidal
Computer Science and Engineering
University of South Carolina
Columbia, SC, USA.
http://jmvidal.cse.sc.edu

14 July 2003
Recommender Systems

- All participants submit their preferences to a central server.
- Server provides recommendations.

Problems:

- Server becomes a central point of failure and attack target.
- Assumes users are willing to provide information to server.
Paying for Recommendations

- A common idea is to have agents pay to read other’s recommendations. But this has some problems.

  1. Processing payments adds overhead and third party.
  2. Users don’t generally trust agents with their money.

- Trade recommendations.
The Model

<table>
<thead>
<tr>
<th>$D$</th>
<th>set of documents, $d \in D$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$A$</td>
<td>set of agents, $i \in A$</td>
</tr>
<tr>
<td>$L_i(d)$</td>
<td>a proposition that is true if $i$ likes $d$</td>
</tr>
<tr>
<td>$R_i(d)$</td>
<td>a proposition that is true if $i$ has read $d$</td>
</tr>
<tr>
<td>$P_r$</td>
<td>the payoff for reading a liked document.</td>
</tr>
<tr>
<td>$C_r$</td>
<td>the cost of reading a document.</td>
</tr>
<tr>
<td>$C_m$</td>
<td>the cost of sending a message.</td>
</tr>
</tbody>
</table>
Utilities

• Each agent gets a utility

\[ U_i(d) = \begin{cases} 
  P_r - C_r & \text{if } L_i(d) \\
  -C_r & \text{otherwise.}
\end{cases} \]

• When two agents meet each must decide whether to tell the other about a document it likes:

<table>
<thead>
<tr>
<th></th>
<th>Nothing</th>
<th>Send</th>
</tr>
</thead>
<tbody>
<tr>
<td>Nothing</td>
<td>0,0</td>
<td>( x_i(j), -C_m )</td>
</tr>
<tr>
<td>Send</td>
<td>(-C_m, x_j(i))</td>
<td>( x_i(j) - C_m, x_j(i) - C_m )</td>
</tr>
</tbody>
</table>
Utilities

- where

\[ x_i(j) = r_i(j)(Pr[L_i(d) \mid L_j(d)] \cdot (P_r - C_r) + (1 - Pr[L_i(d) \mid L_j(d)]) \cdot (-C_r)) \]

\( r_j(i) \) probability that \( i \) receives a document from \( j \) that it has not read.

- Calculating the value of \( x_i(j) \) is usually impossible for \( i \).

- If we assume that \( i \) knows which documents \( j \) has read then we can approximate it using observations:

\[ x_i(j) \approx \frac{|L_i^i \cap L_j^j|}{|L_j^j|} \cdot (P_r - C_r) + \left(1 - \frac{|L_i^i \cap L_j^j|}{|L_i^i|}\right) \cdot (-C_r). \]
Model Analysis

• The payoff matrix is often a Prisoner’s Dilemma, so we expect Tit-for-Tat to be the evolutionary stable strategy (so, they will Send).

• The actual values of $x_i$ can be approximated with observations.

• If two agents have completely uncorrelated preferences then they will rather choose documents randomly.
An Agent’s Choice

• An agent must also decide whether to ask an agent for a recommendation or pick a document at random—the classic explore vs. exploit problem.

• We can calculate the expected utility to be gained from a recommendation: \( x_i(j) \) as well as from reading a random document.

• Agent makes stochastic decision.

• When the agent receives a request it only replies if it expects more utility from the recommendation \( x_i(j) \) than it costs to send the message \( C_m \).
Experimental Model

- We represent each document with an n-dimensional binary vector $d$.
- An agent’s preference is also an n-dimensional vector $p_i$.
- We say that $L_i(d)$ is true if and only if

$$\frac{d \cdot p_i}{n} \leq r$$

for some constant $r$. 
Implementation
• $c$ is clustering probability. 100 documents.
• New documents appear constantly.
• Gain when $c = 1$ and new documents continue to appear.
• Total utility gain for every agent when $c=0.9$. 
Greedy Agents

- agent-explore = .05. Better for $c = .9$ but worse for other $c$ values.
Greedy Agents—Individual Gains

- Gain of each agent for $c = 1$ with agent-explore set to .05
For $c = .9$ the utility is higher but some are left behind.
Conclusions and Future Work

- Using information as currency is a viable approach but several obstacles need to be overcome.

- Agents that have most common interests perform better. Need better ways to support unique agents (by finding friends quickly?)

- The value of a new recommendation evaporates very quickly (unlike money). Need to re-think sharing strategy: share immediately?

- Need to consider hostile agents.