Collaborative Filtering for Orkut Communities: Discovery of User Latent Behavior

Wen-Yen Chen
Computer Science
University of California, Santa Barbara

Joint work with
Jon Chu (MIT)
Junyi Luan (PKU)
Hongjie Bai (Google)
Edward Chang (Google)
description: hey everyone...this room is to commemorate an addition to the already popular nintendo franchise...the Nintendo Wii. Points of view, opinions, and comments will be greatly welcomed concerning the definite success of the Wii that is to come (well i'm buying one anyway). "Together, Wii will take over the world"

"Famitsu asked survey takers which hardware they expect to win the next generation race. The Wii won this vote at 73.0%, with PS3 coming in second at 22.6% and Xbox 360 getting 4.4%.

language: English (UK)
category: Games
owner: Yasin «Fatine»
moderators: Adnaan
type: public
content privacy: open to non-members
forum: anonymous
location: United Kingdom
created: May 19, 2005
members: 1,546

forum

<table>
<thead>
<tr>
<th>topic</th>
<th>posts</th>
<th>last post</th>
</tr>
</thead>
<tbody>
<tr>
<td>Need help, PAL vs NTSC</td>
<td>5</td>
<td>4/5/09</td>
</tr>
<tr>
<td>Want any Wii Games???</td>
<td>3</td>
<td>3/21/09</td>
</tr>
<tr>
<td>ANY</td>
<td>1</td>
<td>3/12/09</td>
</tr>
<tr>
<td>[off] meu novo video no YouTube</td>
<td>1</td>
<td>2/23/09</td>
</tr>
<tr>
<td>Wii Cricket, How good is it?</td>
<td>3</td>
<td>2/12/09</td>
</tr>
</tbody>
</table>
Motivation

Social-network sites are popular and attract millions of users a day
• Facebook, Orkut, Myspace, Twitter…
• Orkut has more than 130M users, 30M communities, 10K communities created daily

Rapid growth of user-generated data available
• Communities, images, videos, posts, friendships…
• Information overload problem

We focus on personalized community recommendation task
• Collaborative filtering (CF) approach
Collaborative Filtering (CF)

The operative assumption underlying collaborative filtering

- Users who were similar in the past are likely to be similar in the future
- Use similar users’ behaviors to make recommendations

Algorithms of three different types

- Memory-based
- Model-based
- Association rules
Collaborative Filtering for Orkut Communities

Investigate two algorithms from very different domains

- Association rules mining (ARM)
  - Discover associations between communities (explicit relations)
  - Users joining “NYY” usually join “MLB”, rule: NYY → MLB
  - Target user joins “NYY”, being recommended “MLB”
  - Fewer common users between “New York Mets” and “MLB”, no rules
Collaborative Filtering for Orkut Communities

Investigate two algorithms from very different domains

• Association rules mining (ARM)
  – Discover associations between communities (explicit relations)
  – Users joining “NYY” usually join “MLB”, rule: NYY → MLB
  – Target user joins “NYY”, being recommended “MLB”
  – Fewer common users between “New York Mets” and “MLB”, no rules

• Latent Dirichlet Allocation (LDA)
  – Model user-community using latent aspects (implicit relations)
  – Implicit relation exists between “NYM” and “MLB” via latent structure
Formulate ARM to Community Recommendation

View user as a transaction and his joined communities as items

<table>
<thead>
<tr>
<th>User</th>
<th>Communities</th>
</tr>
</thead>
<tbody>
<tr>
<td>$u_1$</td>
<td>${c_1, c_3, c_7}$</td>
</tr>
<tr>
<td>$u_2$</td>
<td>${c_3, c_7, c_8, c_9}$</td>
</tr>
<tr>
<td>$u_3$</td>
<td>${c_2, c_3, c_8}$</td>
</tr>
<tr>
<td>$u_4$</td>
<td>${c_1, c_8, c_9}$</td>
</tr>
</tbody>
</table>

Frequent Itemsets | Support
--- | ---
$\{c_1\}$ & 2
$\{c_3\}$ & 3
$\{c_7\}$ & 2
$\{c_8\}$ & 3
$\{c_9\}$ & 2
$\{c_3, c_7\}$ & 2
$\{c_3, c_8\}$ & 2
$\{c_8, c_9\}$ & 2

FP-growth

Supp\text{\_threshold} = 2

Association Rules | Support | Confidence
--- | --- | ---
$c_3 \Rightarrow c_7$ & 2 | 66.7%
$c_3 \Rightarrow c_8$ & 2 | 66.7%
$c_7 \Rightarrow c_3$ & 2 | 100%
$c_8 \Rightarrow c_3$ & 2 | 66.7%
$c_8 \Rightarrow c_9$ & 2 | 66.7%
$c_9 \Rightarrow c_8$ & 2 | 100%

Recommendation based on rules

- If joining $(c_7, c_8)$, being recommended $c_3$ (1.667) and $c_9$ (0.667)
- $\text{supp}(A) = \# \text{ of transactions containing } A$
- $\text{supp}(A \Rightarrow B) = \text{supp}(A,B)$
- $\text{conf}(A \Rightarrow B) = \text{supp}(A,B) / \text{supp}(A)$
Formulate LDA to Community Recommendation

View users as docs, communities as words and membership counts as co-occurrence counts

- α, β: symmetric Dirichlet priors
- θ: per-user topic distribution
- φ: per-topic community distribution

Gibbs sampling

\[
P(z_i = j | w_i = c, z_{-i}, w_{-i}) \propto \frac{C_{c_j}^{CZ} + \beta}{\sum_{c'} C_{c_{j'}}^{CZ} + M\beta} \cdot \frac{C_{u_j}^{UZ} + \alpha}{\sum_{u'} C_{u_{j'}}^{UZ} + K\alpha}
\]

Recommendations based on learned model parameters

- \( \xi_{cu} = \sum_z \phi_{cz} \theta_{zu} \)
Parallelization

We parallelized both ARM and LDA

• Parallel ARM effort [RecSys’08]
• Focus more on parallel LDA

We have two parallel frameworks

• MapReduce
• Message Passing Interface (MPI)
MapReduce

- User specified Map and Reduce functions
- Map: generates a set of intermediate key/value pairs
- Reduce: reduce the intermediate values with the same key
- Read/Write data using disk I/O
- Intensive I/O cost but provide fault-tolerance mechanism

Message Passing Interface (MPI)

- Send/receive data to/from machine’s memory
- Machines can communicate via MPI library routines
- Lazy checkpoints for fault-tolerance
- Suitable for algorithms with iterative procedures
Parallelization

We have $P$ machines and distribute the computation by rows

Each machine $i$

- Computes **local variables** $C_{c_j}^{C_Z}(i)$ and $C_{u_j}^{U_Z}(i)$
- Gets **global variable** $C_{c_j}^{C_Z} = \sum_i C_{c_j}^{C_Z}(i)$
  - AllReduce operation

**Community-topic count**

**User-topic count**

**Computation cost**

- Before: $O(NLK) \times (\# \text{ of iterations})$
- After: $O\left(\frac{NLK}{P}\right) \times (\# \text{ of iterations})$

- $N$: # of users
- $L$: avg # of communities per user
- $K$: # of topics
We have $P$ machines and distribute the computation by rows.

Each machine $i$

- Computes local variables $C_{cj}^{CZ}(i)$ and $C_{uj}^{UZ}(i)$
- Gets global variable $C_{cj}^{CZ} = \sum_i C_{cj}^{CZ}(i)$

Communication cost

- Communication: $O\left(\alpha \cdot \log P + \beta \cdot \frac{P - 1}{P} KM + \gamma \cdot \frac{P - 1}{P} KM\right)$

- startup time of a transfer
- transfer time per unit
- computational time for reduction
Empirical Study

Orkut data
• Community membership data
• 492,104 users and 118,002 communities
• User/community data are anonymized to preserve privacy

Evaluations
• Recommendation quality using top-$k$ ranking metric
• Rank difference between ARM and LDA
• Latent information learned from LDA
• Speedup
Evaluation metric

- Output values of two algorithms cannot be compared directly
- Ranking metric: top-\(k\) recommendation [Y. Koren KDD’08]

Evaluation protocol

- Randomly withhold one community from user’s joined communities
  - Training set for algorithms
- Select \(k-1\) additional random communities not in user’s joined communities
- Evaluate set: the withheld community together with \(k-1\) other communities
  - Order the communities by predicted scores
  - Obtain the corresponding rank of the withheld community (0, …, \(k-1\))
- The lower the rank, the more successful the recommendation
Top-$k$ recommendation performance

Macro-view (0% - 100%), where $k = 1001$

**ARM:** higher the support, worse the performance

**LDA:** consistent performance with varying # of topics
Top-\(k\) recommendation performance (cont.)

Micro-view (0\% - 2\%), where \(k = 1001\)

ARM is better when recommending list up to 3 communities

LDA is consistently better when recommending a list of 4 or more
Rank Differences

Rank differences under different parameters

- ARM-50: best-performing ARM
- LDA-30: worst-performing LDA, LDA-150: best-performing LDA
- Rank difference = LHS – RHS

More withhelod communities have positive rank differences

- LDA generally ranks better than ARM
- LDA is better → much better, ARM is better → a little better
Rank differences under different parameters

- ARM-2000: worst-performing ARM
- LDA-30: worst-performing LDA, LDA-150: best-performing LDA

Similar patterns but fewer rank difference 0

- Increase in the positive rank difference
- Higher support value causes fewer rules for ARM → narrow coverage
Analysis of Latent Information from LDA (cont.)

User1 whom LDA ranks better

User2 whom ARM ranks better

joined communities

<table>
<thead>
<tr>
<th>Community #</th>
<th>Community Name</th>
<th>Category</th>
<th>Size</th>
</tr>
</thead>
<tbody>
<tr>
<td>60640</td>
<td>Java certification</td>
<td>Computers/Internet</td>
<td>731</td>
</tr>
<tr>
<td>58544</td>
<td>professor Ayaz Isazadeh</td>
<td>Alumni/Schools</td>
<td>19</td>
</tr>
<tr>
<td>25422</td>
<td>persiancomputing</td>
<td>Computers/Internet</td>
<td>39</td>
</tr>
<tr>
<td>100953</td>
<td>Iranian J2EE developers</td>
<td>Computers/Internet</td>
<td>297</td>
</tr>
<tr>
<td>53474</td>
<td>web design</td>
<td>Schools/Education</td>
<td>4588</td>
</tr>
<tr>
<td>27999</td>
<td>Yazd sampaad</td>
<td>Alumni/Schools</td>
<td>17</td>
</tr>
<tr>
<td>43431</td>
<td>Tabriz university CS students</td>
<td>Alumni/Schools</td>
<td>13</td>
</tr>
<tr>
<td>80441</td>
<td>C#</td>
<td>Computers/Internet</td>
<td>2247</td>
</tr>
<tr>
<td>66948</td>
<td>Delphi</td>
<td>Computers/Internet</td>
<td>142</td>
</tr>
</tbody>
</table>

joined communities

<table>
<thead>
<tr>
<th>Community #</th>
<th>Community Name</th>
<th>Category</th>
<th>Size</th>
</tr>
</thead>
<tbody>
<tr>
<td>50279</td>
<td>Shahruk Khan fan club</td>
<td>Individuals</td>
<td>50857</td>
</tr>
<tr>
<td>44363</td>
<td>girl power</td>
<td>Religion/Beliefs</td>
<td>1647</td>
</tr>
<tr>
<td>109245</td>
<td>love never dies</td>
<td>Romance/Relationships</td>
<td>22600</td>
</tr>
<tr>
<td>111271</td>
<td>why friends break our heart</td>
<td>Romance/Relationships</td>
<td>10301</td>
</tr>
<tr>
<td>38320</td>
<td>holy angels school</td>
<td>Alumni/Schools</td>
<td>95</td>
</tr>
<tr>
<td>15760</td>
<td>why life is so unpredictable</td>
<td>Other</td>
<td>3878</td>
</tr>
<tr>
<td>8886</td>
<td>T20 WC champs</td>
<td>Recreation/Sports</td>
<td>43662</td>
</tr>
<tr>
<td>77269</td>
<td>star-one fame serial-remix</td>
<td>Other</td>
<td>403</td>
</tr>
<tr>
<td>51302</td>
<td>left right left</td>
<td>Arts/Entertainment</td>
<td>13744</td>
</tr>
<tr>
<td>65215</td>
<td>life is too short to live</td>
<td>Other</td>
<td>8197</td>
</tr>
</tbody>
</table>

Concentrated topic dist.

Scattered topic dist.
Analysis of Latent Information from LDA (cont.)

User1 whom LDA ranks better

User2 whom ARM ranks better

Joined communities

Larger communities

Overlapped at peak

User and withheld community

User and withheld community
Runtime Speedup of parallel LDA

Runtime for LDA using different number of machines

- Use up to 32 machines
- 150 topics, 500 iterations
- Reduce time from 8 hrs to 45 mins

<table>
<thead>
<tr>
<th>Machines</th>
<th>Comp</th>
<th>Comm</th>
<th>Sync</th>
<th>Total</th>
<th>Speedup</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>28911s</td>
<td>0s</td>
<td>0s</td>
<td>28911s</td>
<td>1</td>
</tr>
<tr>
<td>2</td>
<td>14543s</td>
<td>417s</td>
<td>1s</td>
<td>14961s</td>
<td>1.93</td>
</tr>
<tr>
<td>4</td>
<td>7755s</td>
<td>686s</td>
<td>1s</td>
<td>8442s</td>
<td>3.42</td>
</tr>
<tr>
<td>8</td>
<td>4560s</td>
<td>949s</td>
<td>2s</td>
<td>5511s</td>
<td>5.25</td>
</tr>
<tr>
<td>16</td>
<td>2840s</td>
<td>1040s</td>
<td>1s</td>
<td>3881s</td>
<td>7.45</td>
</tr>
<tr>
<td>32</td>
<td>1553s</td>
<td>1158s</td>
<td>2s</td>
<td>2713s</td>
<td>10.66</td>
</tr>
</tbody>
</table>

- When increasing the # of machines
  - Computation time was halved
  - Communication time increased
  - Communication has larger impact on speedup
Discovery of user latent behavior on Orkut

• Compared ARM and LDA for community recommendation task
  – Used top-k ranking metric
• Analyzed latent information learned from LDA
• Parallelized LDA to deal with large data

Future work

• Extend LDA method to consider the strength of relationship between a user and a community
• Extend ARM method to take multi-order rules into consideration

Parallel LDA code release