Social Interaction Based Video Recommendation: Recommending YouTube Videos to Facebook Users

Bin Nie¹ · Honggang Zhang¹ · Yong Liu²

¹ Fordham University, Bronx, NY. Email: {bnie, hzhang44}@fordham.edu
² NYU Poly, Brooklyn, NY. Email: yongliu@poly.edu
Background

- Online videos are popular topics for OSNs
- Recent researches: Recommender Systems can benefit from OSNs
  - But, they mainly use…
    - Synthetic Network
    - Pure content-sharing websites
- Our Work, based on Real-World Data
How to Recommend Videos by Leveraging Social Interactions?

Our research:

- **Step1**: Collect Data from Facebook and YouTube
- **Step2**: Analyze Video Popularity and Social Interactions
- **Step3**: Develop Social-Interaction Based Algorithms to Recommend Videos
Data Collection: Sampling Facebook Network

- Pick a user as a root, and start a random-walk, neighbor-limited breadth-first search process

\[ u \]

\[ u \quad u_f1 \quad u_f2 \quad u_f3 \]

\[ u_f1 \quad u_f11 \quad u_f12 \quad u_f13 \]

depth 1: size = 3^1

depth 2: size = 3^2

depth 6: size = 3^6

- Ideally, from each root user, we can get 1093 nodes. (1093 × 6 = 6,558 in total.)
- Actually, from all six root users, we get 6,466 nodes, denoted as \( U_{orig} \).
- Collect videos uploaded or shared by users in \( U_{orig} \) → video set \( V_{orig} \)
- Sampling time interval: April 18, 2013 ~ May 30, 2013

* other types: status, photo, link, swf, checkin, offer.
Data Collection: Finalize Data Set

\[ V_{orig} \rightarrow U_{inter} \rightarrow U_{all} \rightarrow V_{all} \]

<table>
<thead>
<tr>
<th></th>
<th>All</th>
<th>YouTube</th>
</tr>
</thead>
<tbody>
<tr>
<td># of Users</td>
<td>32,209</td>
<td>19,609</td>
</tr>
<tr>
<td># of Direct Friends</td>
<td>111,486</td>
<td>101,176</td>
</tr>
<tr>
<td># of Two-hops Friends</td>
<td>4,512,762</td>
<td>238,552</td>
</tr>
<tr>
<td># of videos</td>
<td>294,826</td>
<td>134,151</td>
</tr>
</tbody>
</table>

It’s the number of YouTube videos with **unique** links!
How to Recommend Videos by Leveraging Social Interactions?

Our research:

- Step 1: Collect Data from Facebook and YouTube
- Step 2: Analyze Video Popularity and Social Interactions
- Step 3: Develop Social-Interaction Based Algorithms to Recommend Videos
Video Popularity

- Study YouTube videos set $V_y$ and users set $U_y$

- For a video $v_i \in V_y$,
  - Popularity on Facebook
    $$F_i = c_i / \sum_{j=1}^{n} c_j$$
    # of sharing + # of interactions
  - Popularity on YouTube
    $$W_i = w_i / \sum_{j=1}^{n} w_j$$
    Watch count
    $$L_i = l_i / \sum_{j=1}^{n} l_j$$
    Like count
Compare Popularity of Videos on Facebook and YouTube

$W_i$  

$p_i$  

$L_i$  

Pearson Coefficient $= 0.03$  

Pearson Coefficient $= 0.02$

It **might** be effective to exploit social interaction information to improve RS
Video Popularity Comparison on Facebook and YouTube

- Look at videos ranked among top 100 and lowest 100 on YouTube

like count in the increasing order
Video Popularity Difference
A Global View

* Each group has at least 100 videos
Video Popularity Difference
A Global View

\[ \text{INC} = \text{pop(\text{fb})} - \text{pop(\text{ytb})} \]

\( G_1 \) \hspace{1cm} \text{pop increase} \hspace{1cm} \text{INC} = \text{pop(\text{fb})} - \text{pop(\text{ytb})} \hspace{1cm} G_{45} \hspace{1cm} G_{46} \hspace{1cm} \text{pop decrease} \hspace{1cm} G_{50} \n
Mining social interactions might potentially help to improve recommendation of videos, especially Niche Videos!

* Each group has at least 100 videos
Interest Similarity vs. Social Distance

- **Cosine Similarity**

  \[
  \text{cosineSim}(a, b) = \frac{2}{\sqrt{4 \times 3}} = 0.58
  \]

<table>
<thead>
<tr>
<th></th>
<th>(v_1)</th>
<th>(v_2)</th>
<th>(v_3)</th>
<th>(v_4)</th>
<th>(v_5)</th>
</tr>
</thead>
</table>
  Alice | 2 | 0 | 4 | 1 | 3 |
  Bob   | 0 | 3 | 2 | 1 | 0 |

  Alice interacted with \(v_3\) 4 times

- **Social Distance**
  - One-hop friends
  - Two-hops friends
  - More-than-two-hops friends
Interest Similarity vs. Social Distance

NO significant correlation between users’ interest similarity and social distance!
How to Recommend Videos by Leveraging Social Interactions?

Our research:

- **Step 1:** Collect Data from Facebook and YouTube
- **Step 2:** Analyze Video Popularity and Social Interactions
- **Step 3:** Develop Social-Interaction Based Algorithms to Recommend Videos
## Example Scenario

- **5 users and 10 videos** $\Rightarrow 5 \times 10$ matrix $A$

<table>
<thead>
<tr>
<th></th>
<th>$v_0$</th>
<th>$v_1$</th>
<th>$v_2$</th>
<th>$v_3$</th>
<th>$v_4$</th>
<th>$v_5$</th>
<th>$v_6$</th>
<th>$v_7$</th>
<th>$v_8$</th>
<th>$v_9$</th>
</tr>
</thead>
<tbody>
<tr>
<td>Alice</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>Bob</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>Cindy</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>David</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Eddie</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
</tr>
</tbody>
</table>

* interested: at least interacted with this video once!
Example Scenario

- 5 users and 10 videos $\rightarrow$ $5 \times 10$ matrix $A$

<table>
<thead>
<tr>
<th>Alice</th>
<th>0</th>
<th>1</th>
<th>1</th>
<th>0</th>
<th>1</th>
<th>0</th>
<th>0</th>
<th>1</th>
<th>1</th>
<th>1</th>
</tr>
</thead>
</table>

$V_{a,1}$

$V_{a,0}$

grouping
Example Scenario

- **5 users and 10 videos** → **$5 \times 10$ matrix $A$**

| Alice | 0 | 1 | 1 | 0 | 1 | 0 | 0 | 1 | 1 | 1 | 1 |

We consider users who have interaction with at least 50 videos in $V_y$.

$V_{a,test} = 20\%$ of $V_{a,1}$

$V_{a,1}$

$V_{a,train}$

$V_{a,0}$

Alice’s candidate video set $V_{a,cad}$
Suppose recommend videos to Alice,

- Rank videos in $V_{a,cad}$ according to the like count, or watch count from YouTube in non-increasing order
- Recommend Top-k videos in $V_{a,cad}$ to Alice
CollabRecommend Algorithm

Suppose recommend videos to Alice,

- \( \forall l \in U_y \), calculate \( \cosineSim(a, l) \) using \( V_{a,train} \)
- \( S_a = \text{Top-50 similar} \)

Selecting Users

Scoring

- Score videos in \( V_{a,cad} \) according to \( S_a \) users’ interest vectors

Recommending

- Recommend Top-k videos, denoted as \( V_{a,rec} \)
Example: How to Score $V_{a,rec}$?

- Alice’s $V_{a,cad} = \{v_1, v_2, v_3, v_4, v_5\}$
- Top 5 users in Alice’s $S_a = \{u_1, u_2, u_3, u_4, u_5\}$

### Alice’s Score Vector

<table>
<thead>
<tr>
<th></th>
<th>$v_1$</th>
<th>$v_2$</th>
<th>$v_3$</th>
<th>$v_4$</th>
<th>$v_5$</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>

### $u_1$’s Interest Vector

|     | 1     | 1     | 0     | 0     | 1     |

### $u_2$’s Interest Vector

|     | 0     | 1     | 1     | 0     | 0     |

### $u_3$’s Interest Vector

|     | 0     | 0     | 1     | 1     | 1     |

### $u_4$’s Interest Vector

|     | 0     | 0     | 1     | 0     | 0     |

### $u_5$’s Interest Vector

|     | 0     | 0     | 1     | 0     | 1     |
Example: How to Score $V_{a,rec}$?

- Alice’s $V_{a,cad} = \{v_1, v_2, v_3, v_4, v_5\}$
- Top 5 users in Alice’s $S_a = \{u_1, u_2, u_3, u_4, u_5\}$
- Alice’s Score Vector

<table>
<thead>
<tr>
<th></th>
<th>$v_1$</th>
<th>$v_2$</th>
<th>$v_3$</th>
<th>$v_4$</th>
<th>$v_5$</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>1</td>
<td>2</td>
<td>4</td>
<td>1</td>
<td>3</td>
</tr>
</tbody>
</table>

- $u_1$’s Interest Vector

<table>
<thead>
<tr>
<th></th>
<th>$v_1$</th>
<th>$v_2$</th>
<th>$v_3$</th>
<th>$v_4$</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>1</td>
<td>1</td>
<td></td>
<td>0</td>
</tr>
</tbody>
</table>

- $u_2$’s Interest Vector

<table>
<thead>
<tr>
<th></th>
<th>$v_1$</th>
<th>$v_2$</th>
<th>$v_3$</th>
<th>$v_4$</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>0</td>
<td>1</td>
<td></td>
<td>0</td>
</tr>
</tbody>
</table>

- $u_3$’s Interest Vector

<table>
<thead>
<tr>
<th></th>
<th>$v_1$</th>
<th>$v_2$</th>
<th>$v_3$</th>
<th>$v_4$</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>0</td>
<td>0</td>
<td></td>
<td>1</td>
</tr>
</tbody>
</table>

- $u_4$’s Interest Vector

<table>
<thead>
<tr>
<th></th>
<th>$v_1$</th>
<th>$v_2$</th>
<th>$v_3$</th>
<th>$v_4$</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>0</td>
<td>0</td>
<td></td>
<td>0</td>
</tr>
</tbody>
</table>

- $u_5$’s Interest Vector

<table>
<thead>
<tr>
<th></th>
<th>$v_1$</th>
<th>$v_2$</th>
<th>$v_3$</th>
<th>$v_4$</th>
<th>$v_5$</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
</tr>
</tbody>
</table>

Alice’s Final Score Vector
Example: How to Score $V_{a,rec}$?

- Alice’s $V_{a,cad} = \{v_1, v_2, v_3, v_4, v_5\}$
- Top 5 users in Alice’s $S_a = \{u_1, u_2, u_3, u_4, u_5\}$
- Alice’s Score Vector

<table>
<thead>
<tr>
<th></th>
<th>$v_1$</th>
<th>$v_2$</th>
<th>$v_3$</th>
<th>$v_4$</th>
<th>$v_5$</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>1</td>
<td>2</td>
<td>4</td>
<td>1</td>
<td>3</td>
</tr>
</tbody>
</table>

Alice’s Final Score Vector

Sorting

Recommend Top-2 to Alice, denoted as $V^2_{a,rec}$
Video-related Interactions

- Bob interacted with Alice’s video twice.
Video-related Interactions

- Bob interacted with Alice’s video twice.
- Alice interacted with Bob’s video once.
- Total video-related interactions between Alice and Bob = 2 + 1 = 3
SocialRecommend Algorithm

Suppose recommend videos to Alice,

- **Similarity**
  \[ S_a = \text{Top-50 similar} \]
- **Social Interaction**
  \[ S_a = S_a \cup \text{Top-50 users with most interactions} \]

\[ |S_a| \leq 100! \]
Evaluate Recommendation Accuracy

- **Top-k Hit-Ratio**

\[
H(a, k) = \frac{|V_{a,rec}^k \cap V_{a,test}|}{|V_{a,test}|}
\]

- **NaiveYouTube** (based on like count)
- **CollabRecommend**
- **SocialRecommend**

Most significant improvement
Evaluate Recommendation Algorithm

- Recall

\[
\text{recall} = \frac{\left| U_y \right|}{\sum_{i=1}^{\left| U_y \right|} \left| V_{i,\text{rec}} \cap V_{i,\text{test}} \right|} / \left( \sum_{i=1}^{\left| U_y \right|} \left| V_{i,\text{test}} \right| \right)
\]

About 20% ~ 25% relative accuracy improvement
Recall

\[
\text{recall} = \frac{\sum_{i=1}^{k} \left| V_{i,\text{rec}} \right|}{\left| U_{y} \right|}
\]

Recommendation Algorithm

- Social-Recommend

Vary number of candidate users
Conclusion

- Video popularity on Facebook is not always well-aligned with video popularity on YouTube.
- With rich social interactions information, even simple algorithm could achieve significantly better recommendation accuracy.
Future Work

- Expand Dataset
  - More diverse

- Improve SocialRecommend Performance
  - More robust
  - Better hit-ratio and recall
Thank you!