Social Interaction Based Video Recommendation:

Recommending YouTube Videos to Facebook Users

Bin Nie[1] · Honggang Zhang[1] · Yong Liu[2]

[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**

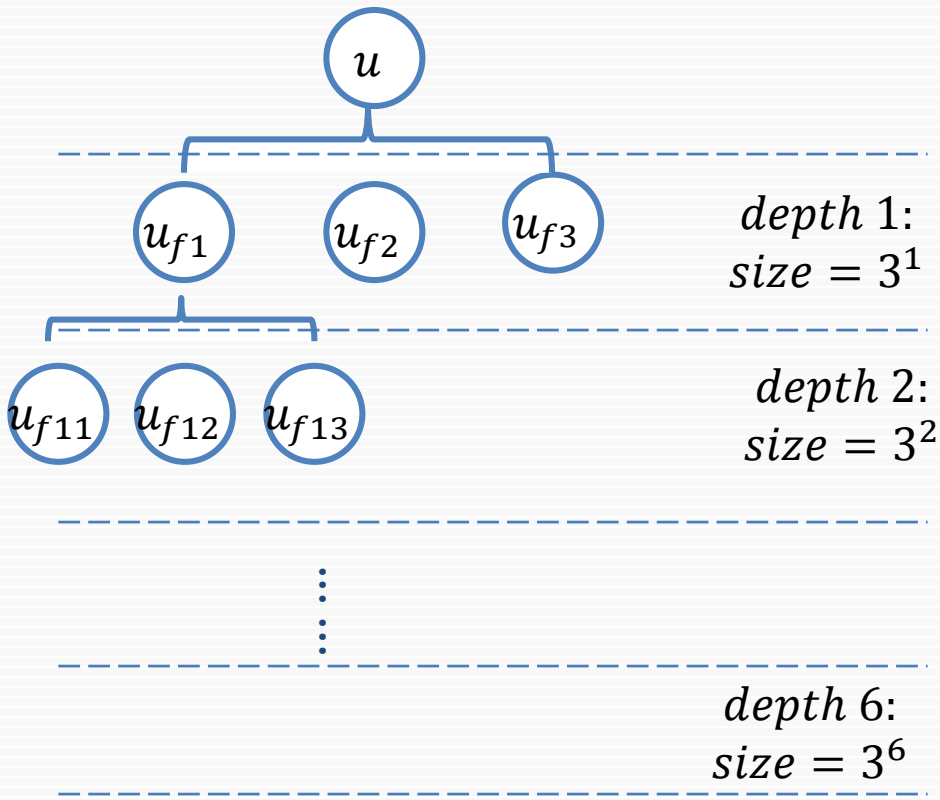
How to Recommend Videos by Leveraging Social Interactions?

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**

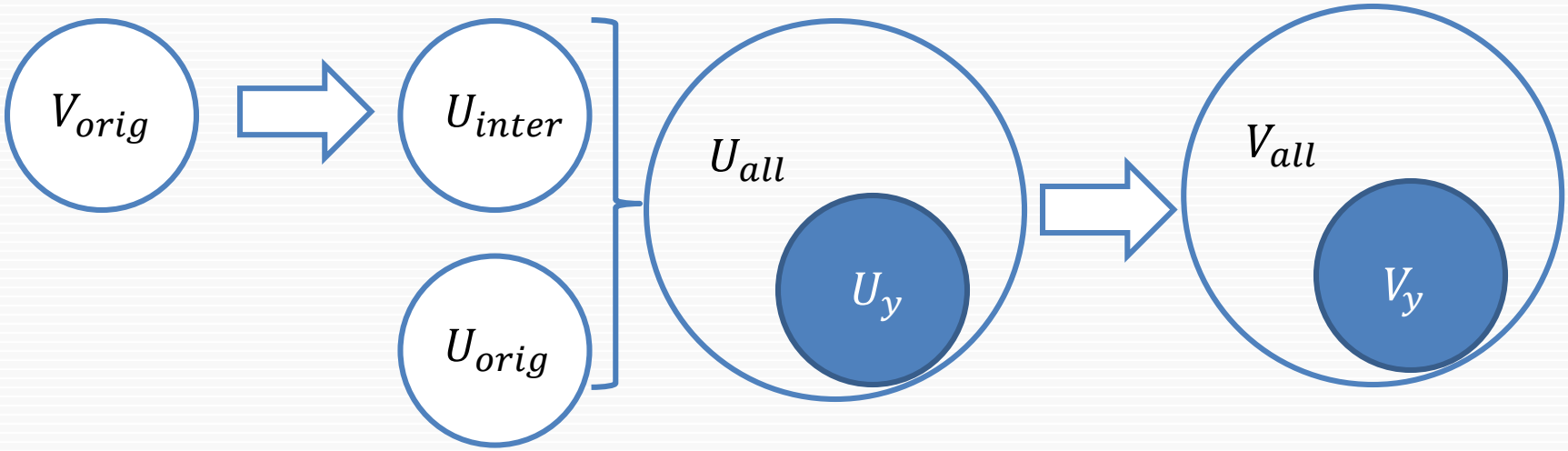


- Ideally, from each root user, we can get 1093 nodes. ($1093 \times 6 = 6{,}558$ in total.)
- Actually, from all six root users, we get $6{,}466$ nodes, denoted as U_{orig}.

- 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



	All	YouTube
# of Users	32,209	19,609
# of Direct Friends	111,486	101,176
# of Two-hops Friends	4,512,762	238,552
# of videos	294,826	134,151

It's the number of YouTube videos with **unique** links!

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

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

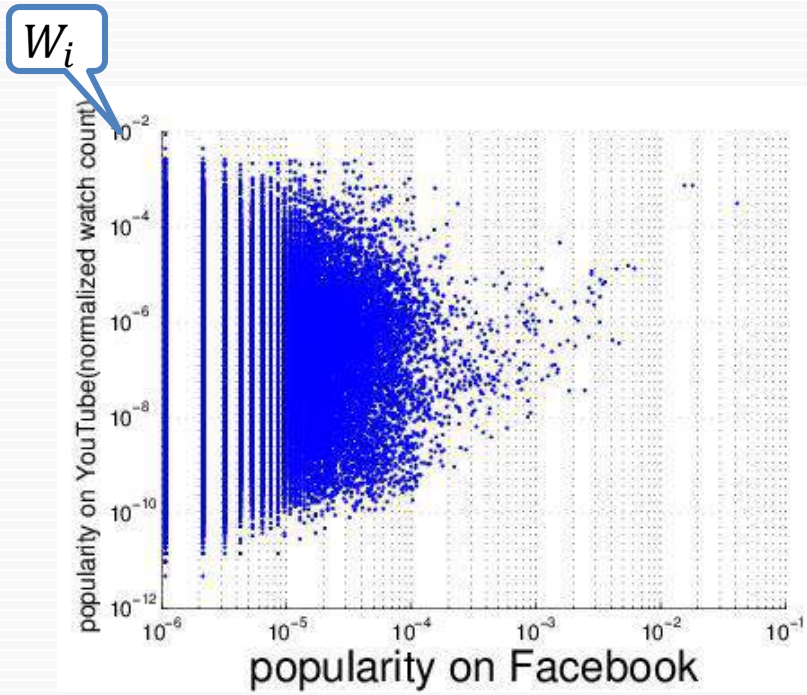
  - Popularity on YouTube

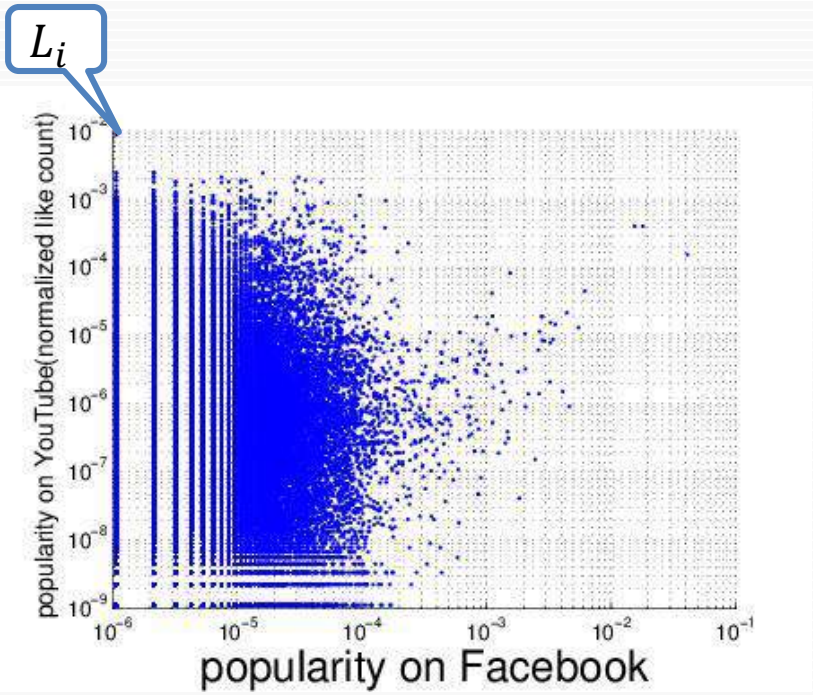
$$W_i = w_i / \sum_{j=1}^{n} w_j \qquad L_i = l_i / \sum_{j=1}^{n} l_j$$

watch count

like count

Compare Popularity of Videos on Facebook and YouTube





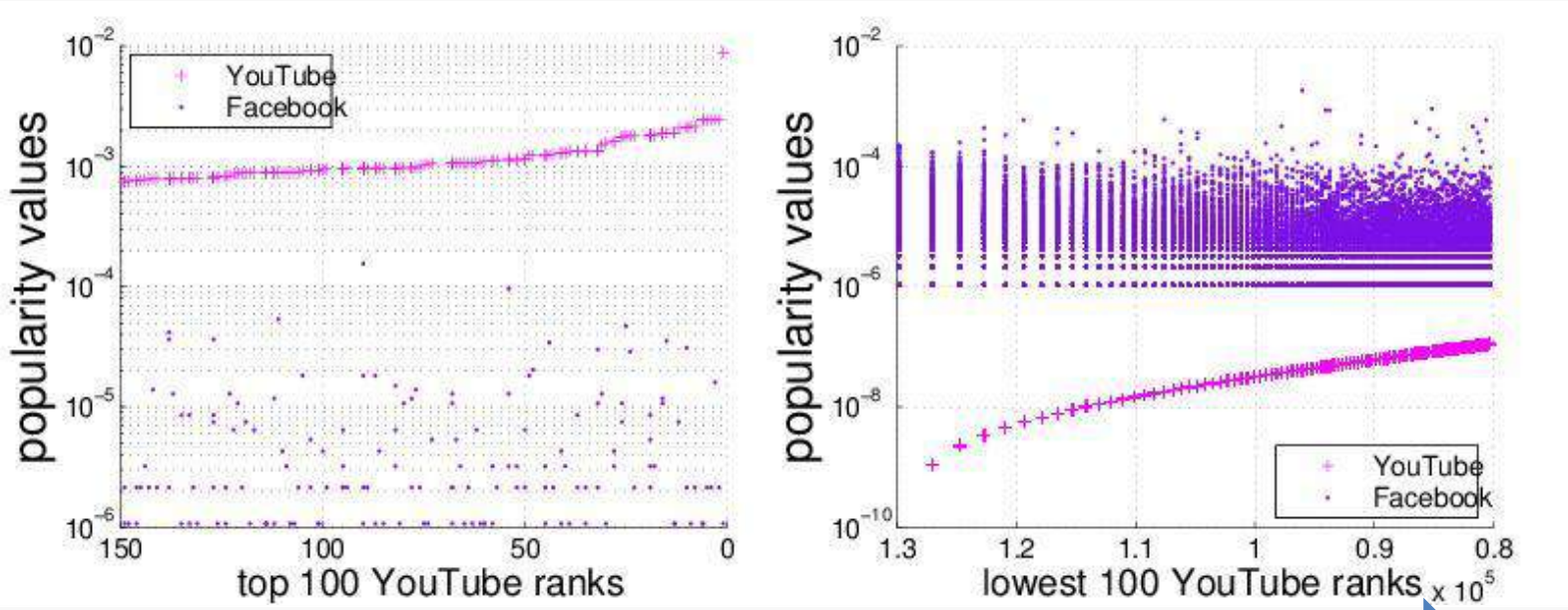
Pearson Coefficient $= 0.03$

Pearson Coefficient $= 0.02$

It **might?** be effective to exploit social interaction information to improve RS

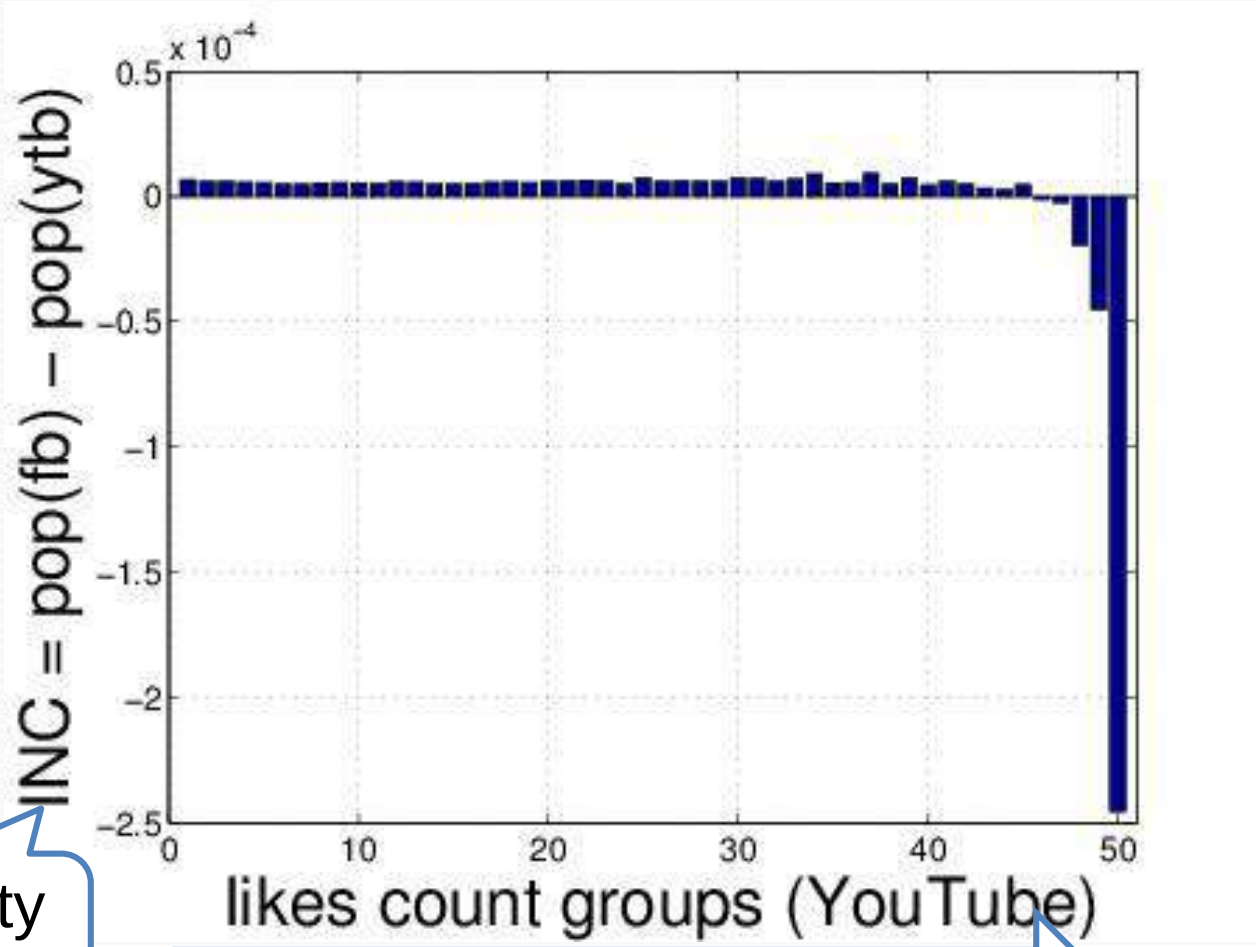
Video Popularity Comparison on Facebook and YouTube

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



like count in the increasing order

Video Popularity Difference
A Global View

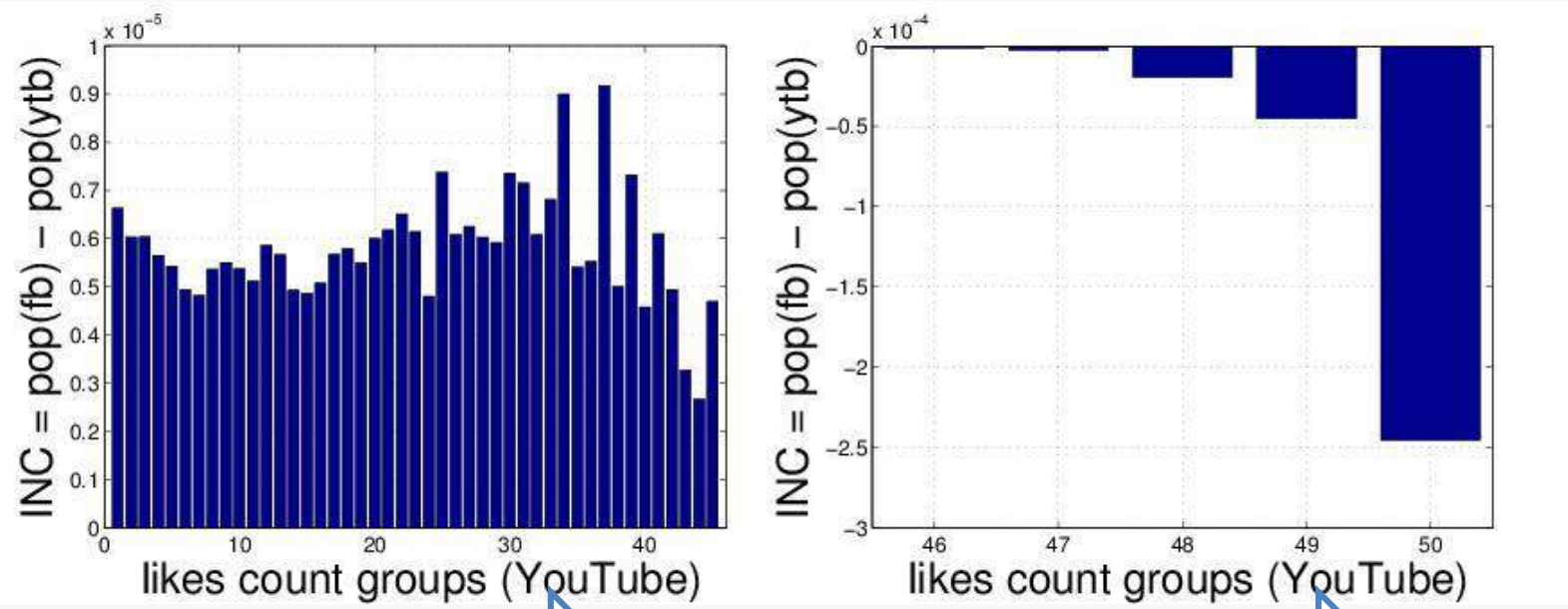


popularity increase

like count in the increasing order

* Each group has at least 100 videos

Video Popularity Difference
A Global View



G_1 pop increase G_{45} G_{46} pop decrease G_{50}

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

Alice interacted with v_3 4 times

Binary Values

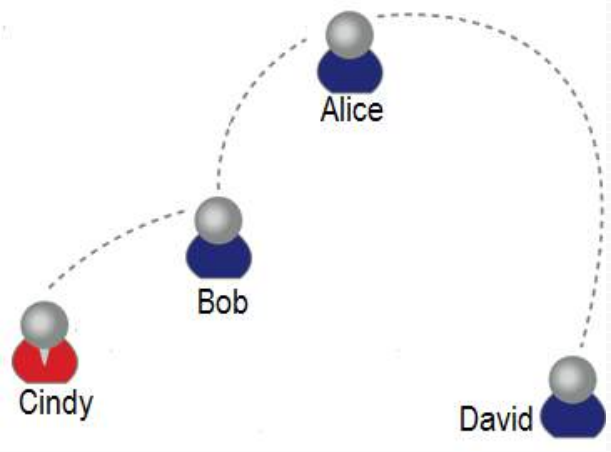
- **Cosine Similarity**

	v_1	v_2	v_3	v_4	v_5
Alice	2	0	4	1	3
Bob	0	3	2	1	0

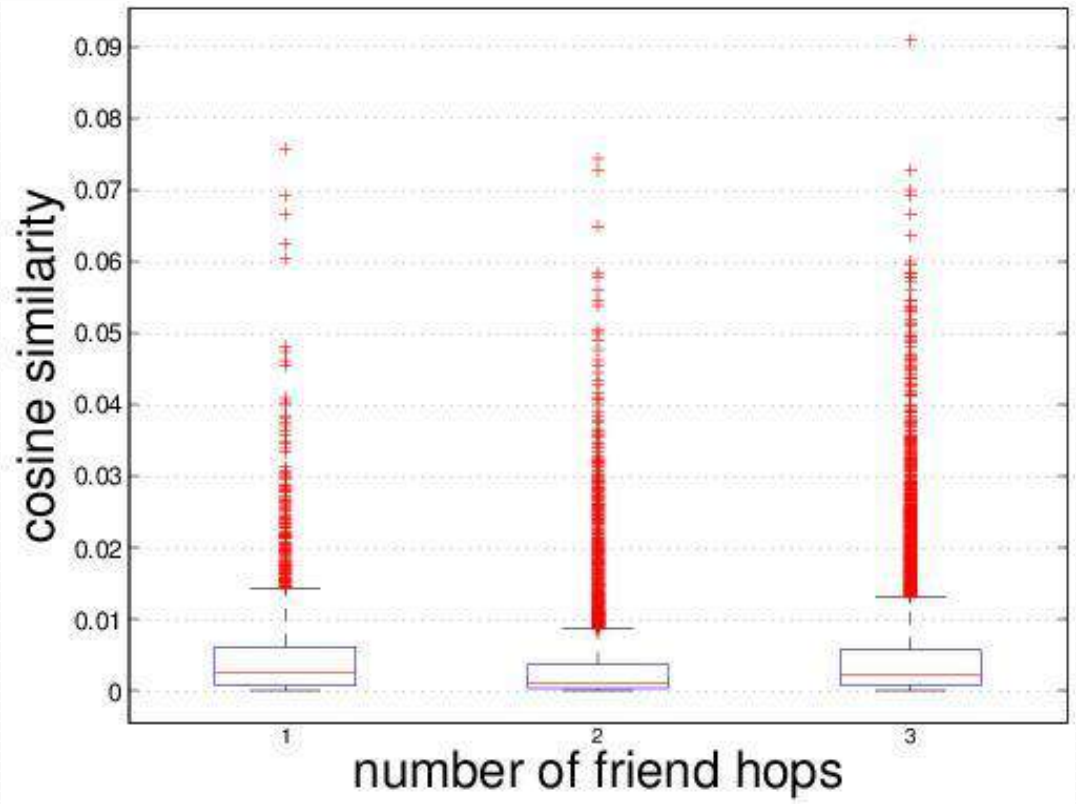
Range: [0,1]

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

- **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:

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

Example Scenario

- 5 **users and** 10 **videos → 5 × 10 matrix** A

Alice's Interest Vector

	v_0	v_1	v_2	v_3	v_4	v_5	v_6	v_7	v_8	v_9
Alice	0	1	1	0	1	0	0	1	1	0
Bob	1	0	0	0	1	1	0	0	0	1
Cindy	0			1	0			0	1	1
David	0			0	0			1	0	0
Eddie	1	0	1	1	0	1	0	0	1	1

0→ Alice is not interested in v_3

1→ Alice is interested in v_4

* interested: at least interacted with this video once!

Example Scenario

- 5 **users and** 10 **videos → 5×10 matrix** A

Alice	0	1	1	0	1	0	0	1	1	1

grouping

Alice	1	1	1	1	1	1	0	0	0	0

$V_{a,1}$ $V_{a,0}$

Example Scenario

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

- 5 **users and** 10 **videos → 5 × 10 matrix** A

Alice	0	1	1	0	1	0	0	1	1	1

grouping

Alice	1	1	1	1	1	1	0	0	0	0

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

$V_{a,1}$

$V_{a,train}$

$V_{a,0}$

Alice's **candidate** video set $V_{a,cad}$

NaïveYouTube Algorithm

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

Selecting Videos

Recommending

- Recommend Top-k videos in $V_{a,cad}$ to Alice

CollabRecommend Algorithm

Suppose recommend videos to Alice,

Selecting Users

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

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
- u_1's Interest Vector
- u_2's Interest Vector
- u_3's Interest Vector
- u_4's Interest Vector
- u_5's Interest Vector

v_1	v_2	v_3	v_4	v_5
0	0	0	0	0
1	1	0	0	1
0	1	1	0	0
0	0	1	1	1
0	0	1	0	0
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

Alice's **Final** Score Vector

	v_1	v_2	v_3	v_4	v_5
Alice's Score Vector	1	2	4	1	3
u_1's Interest Vector	1	1		0	
u_2's Interest Vector	0	1		0	
u_3's Interest Vector	0	0		1	
u_4's Interest Vector	0	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

Alice's **Final** Score Vector

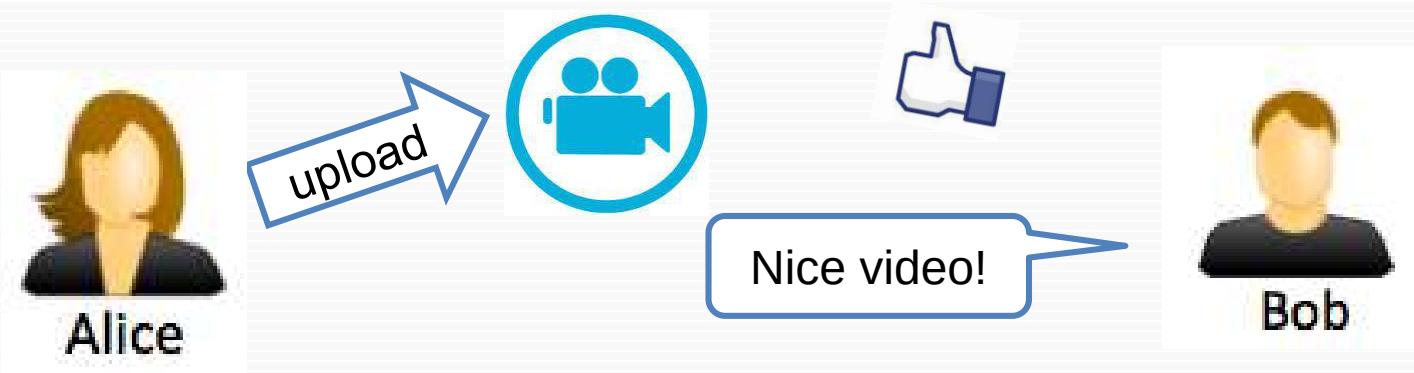
v_1	v_2	v_3	v_4	v_5
1	2	4	1	3

sorting

4	3	2	1	1

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=Top-50 similar
- **Social Interaction**
  $\boldsymbol{S_a} = \boldsymbol{S_a} \cup$ Top-50 users with most interactions

$|S_a| \leq 100!$

Selecting Users

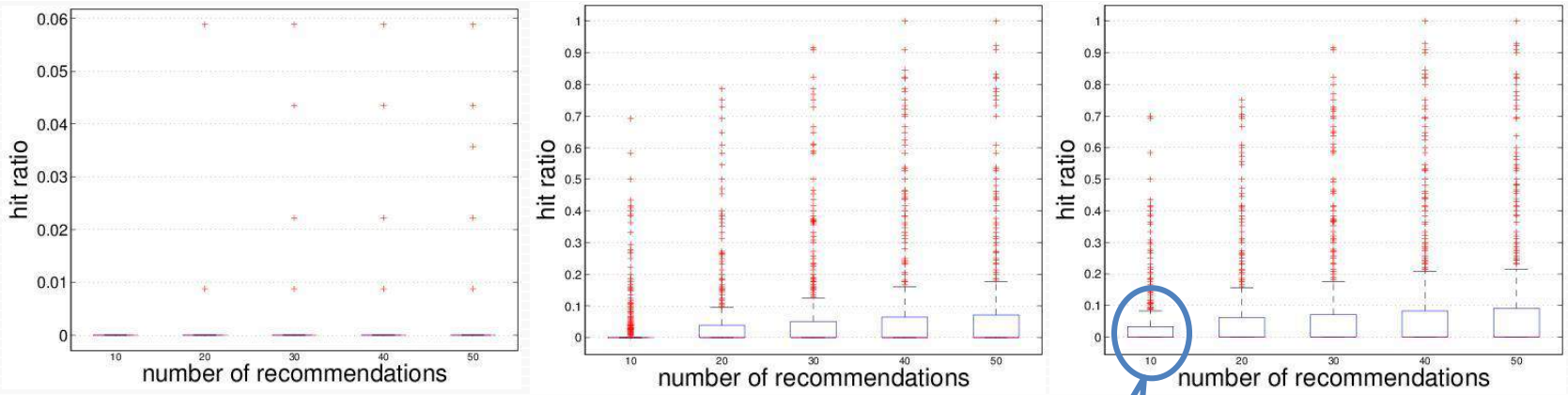
Scoring

Recommending

Evaluate Recommendation Accuracy

- **Top-k Hit-Ratio**

$$\mathrm{H}(\mathrm{a},\mathrm{k}) = (|V_{a,rec}^{k} \cap V_{a,test}|)/(|V_{a,test}|)$$



NaiveYouTube
(based on like count)

CollabRecommend

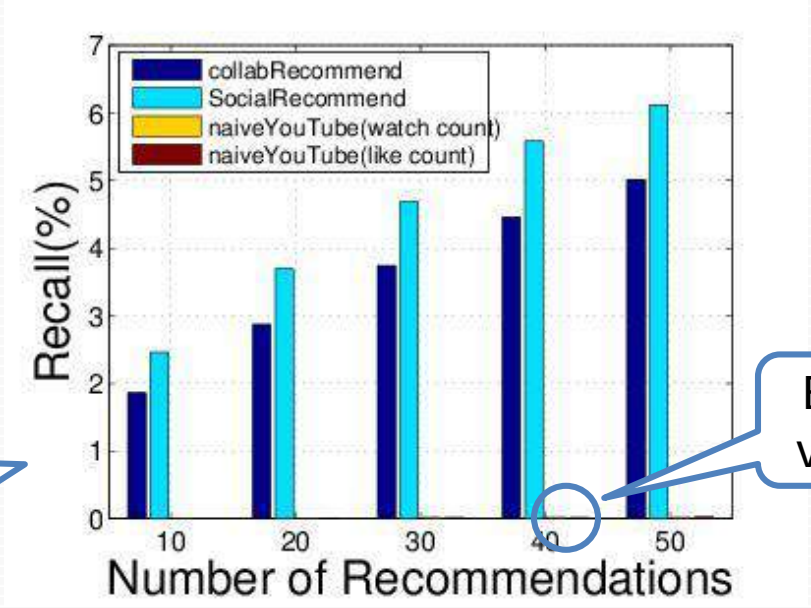
SocialRecommend

Most significant improvement

Evaluate Recommendation Algorithm

- **Recall**

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



About **20% ~ 25%** relative accuracy improvement

Barely visible!

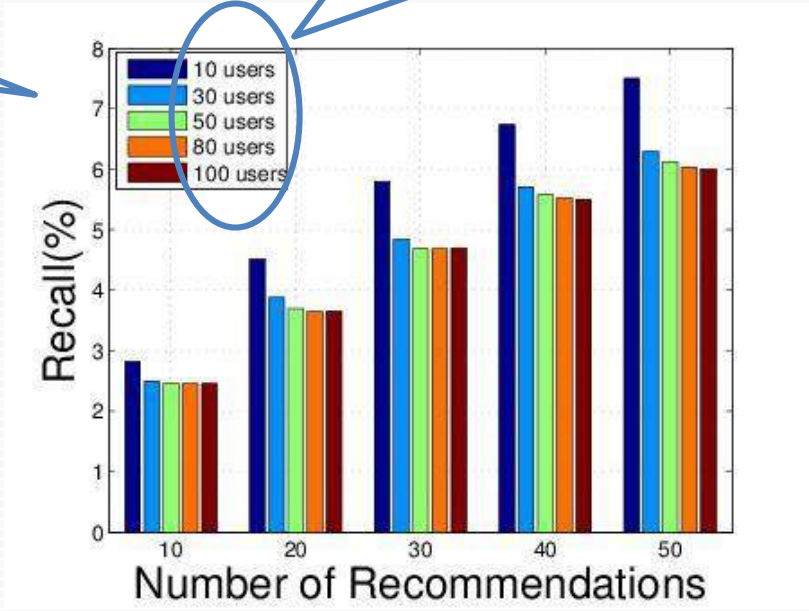
Recommendation Algorithm

- **Recall**

$$\text{recall} = \left(\sum_{i=1}^{|U_y|} |V_{i,rec}^{k}\right. \qquad \sum^{|U_y|} \left. |V_{i,test}|\right)$$

Vary number of candidate users

Social-Recommend



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!