Community-Aware Prediction of Virality Timing Using Big Data of Social Cascades

By: Alvin Junus, Ming Cheung, James She, and Zhanming Jie

Apr-2015
Outline

- Introduction
- Methodology
- Experimental Results
- Conclusion
Outline

- Introduction
- Methodology
- Experimental Results
- Conclusion
Viral contents

- Content  popular  viral content
  rapid social sharing

- Popularity – likes, views, retweets

- Viral contents get popular fast
Virality timing

- Virality timing: the time at which content gets popular

- Predicting virality timing:
  - Model content’s popularity growth
  - Predict when its popularity will reach the target
Outline

- Introduction
- Methodology
- Experimental Results
- Conclusion
Modeling popularity growth

• Social cascade – a process of information diffusion in the social network

• Need to have social connections to form a social cascade

• Cascade size quantifies content virality
Modeling popularity growth

- **Basic reproduction number** $R_0$ – expected number of secondary infections resulting from an infected node in a cascade

- Models cascades’ behaviors

- $R_0(t)$: $R_0$ at time $t$
Community structure

- Simple contagion in social network for viral contents [1]

- $R_{0,c}(t): R_0$ for community $c$ at time $t$

Community-aware prediction algorithm

1. Extract social cascades & identify communities
2. Time prediction for each community
3. Time prediction for multiple communities

Stopping conditions fulfilled?
Yes: End prediction
No: Virality timing prediction from multiple communities

Stage A: Ongoing extraction
Community-aware prediction algorithm
Community-aware prediction algorithm
Community-aware prediction algorithm

\[ n_c'(t, t') = n_c(t) + \sum_{j=1}^{k(t,t')} \Delta n_c(t) \cdot (R_{0,c}(t))^j \]

\[ k(t, t') = \left[ \frac{t' - t}{T_0} \right] \]

- \( t = \) current time
- \( t' = \) future time
- \( T_0 = \) iteration interval

\( n_c(t) = \) no. of infected nodes in community \( c \) at time \( t \)

\( n_c'(t, t') = \) no. of infected nodes in community \( c \) at time \( t' \)
Community-aware prediction algorithm

\[
N'(t, t') = \sum_{c=1}^{L(t)} n'_c(t, t')
\]

\[
t(N) = \arg \min_{t'} N'(t, t') \geq N
\]

\[
L(t) = \text{no. of infected communities at time } t
\]

\[
t(N) = \text{predicted virality timing}
\]
Community-aware prediction algorithm

- Target $N$ has been reached
- Target $N$ can still not be reached after many iterations
Outline

- Introduction
- Methodology
- Experimental Results
- Conclusion
Evaluation

• Digg dataset:
  • 3 million user sharing records involving ~140,000 users
  • ~800 communities formed from social graph of user friendships

• Deviation from ground truth
  $$E = \frac{|t(N) - t_{GT}|}{t_{GT}}$$
Experimental results

- Virality timing prediction for the most popular story to reach 15000 ‘like’s at
  - (a) 40% data available

![Graph showing virality timing prediction](image)
Experimental results

- (b) 80% data available

- $E$ decreases with more data

- Community information improves prediction results
Experimental results

- Virality timing prediction for 10% most popular stories
- Community information improves average prediction results by 30%
- Performance improvement decreases with more data
Outline

- Introduction
- Methodology
- Experimental Results
- Conclusion
Conclusion

• Model popularity growth to predict virality timing

• Propose a self-correcting community-aware prediction algorithm

• Show that community information improves prediction accuracy in Digg dataset by 30%
Q&A

BigDataService2015