Scalable Topic-Specific Influence Analysis on Microblogs

Yuanyuan Tian
IBM Almaden Research Center
Motivation

- Huge amount of textual and social information produced by popular microblogging sites.
  - Twitter had over 500 million users creating over 340 million tweets daily, reported in March 2012.

- A popular resource for marketing campaigns
  - Monitor opinions of consumers
  - Launch viral advertising

- Identifying key influencers is crucial for market analysis
Existing Social Influence Analysis

- Majority existing influence analyses are only based on network structures
  - E.g. influence maximization work by Kleinberg et al
  - Valuable *textual* content is ignored
  - Only *global* influencers are identified

- **Topic-specific** influence analysis
  - Network + Content
  - Differentiate influence in different aspects of life (topics)
Existing Topic-Specific Influence Analysis

- Separate analysis of content and analysis of networks
  - E.g. Topic-sensitive PageRank (TSPR), TwitterRank
  - Text analysis → topics
  - For each topic, apply influence analysis on the network structure
  - But in microblog networks, content and links are often related
    - A user tends to follow another who tweets in similar topics

- Analysis of content and network together
  - E.g. Link-LDA
  - But designed for citation and hyperlink networks
  - Assumption: content is the only cause of links!!
  - But in microblog networks, a user sometimes follows another for a content-independent reason
    - A user follows celebrities simply because of their fame and stardom
Overview

- Goal: Search for topic-specific key influencers on microblogs
- Model topic-specific influence in microblog networks
- Learn topic-specific influence efficiently
- Put it all together in a search framework

SKIT

healthy food
Michelle Obama
Jimmy Oliver
...
Road Map

- Followship-LDA (FLDA) Model
- Scalable Gibbs Sampling Algorithm for FLDA
- A General Search Framework for Topic-Specific Influencers
- Experimental Results
Intuition

- Each microblog user has a bag of words (from tweets) and a set of followees
  - **Content:** web, organic, veggie, cookie, cloud, …
  - **Followees:** Michelle Obama, Mark Zuckerberg, Barack Obama

- A user tweets in multiple topics
  - Alice tweets about technology and food

- A topic is a mixture of words
  - “web” and “cloud” are more likely to appear in the technology topic

- A user follows another for different reasons
  - Content-based: follow for topics
  - Content-independent: follow for popularity

- A topic has a mixture of followees
  - Mark Zuckerberg is more likely to be followed for the technology topic
  - **Topic-specific influence:** the probability of a user $u$ being followed for a topic $t$
Followship-LDA (FLDA)

**FLDA**: A Bayesian *generative* model that extends Latent Dirichlet Allocation (LDA)

- Specify a probabilistic procedure by which the content and links of each user are generated in a microblog network
- Latent variables (hidden structure) are introduced to explain the observed data
  - Topics, reasons of a user following another, the topic-specific influence, etc
- Inference: “reverse” the generative process
  - What hidden structure is most likely to have generated the observed data?
### Hidden Structure in FLDA

- Per-user topic distribution
- Per-topic word distribution
- Per-user followship preference
- Per-topic followee distribution
- Global followee distribution

#### Per-user topic distribution

<table>
<thead>
<tr>
<th>User</th>
<th>Tech</th>
<th>Food</th>
<th>Poli</th>
</tr>
</thead>
<tbody>
<tr>
<td>Alice</td>
<td>0.8</td>
<td>0.19</td>
<td>0.01</td>
</tr>
<tr>
<td>Bob</td>
<td>0.6</td>
<td>0.3</td>
<td>0.1</td>
</tr>
<tr>
<td>Mark</td>
<td>…</td>
<td>…</td>
<td>…</td>
</tr>
<tr>
<td>Michelle</td>
<td>…</td>
<td>…</td>
<td>…</td>
</tr>
<tr>
<td>Barack</td>
<td>…</td>
<td>…</td>
<td>…</td>
</tr>
</tbody>
</table>

#### Per-topic word distribution

<table>
<thead>
<tr>
<th>Topic</th>
<th>web</th>
<th>cookie</th>
<th>veggie</th>
<th>organic</th>
<th>congress</th>
<th>…</th>
</tr>
</thead>
<tbody>
<tr>
<td>Tech</td>
<td>0.3</td>
<td>0.1</td>
<td>0.001</td>
<td>0.001</td>
<td>0.001</td>
<td>…</td>
</tr>
<tr>
<td>Food</td>
<td>0.001</td>
<td>0.15</td>
<td>0.3</td>
<td>0.1</td>
<td>0.001</td>
<td>…</td>
</tr>
<tr>
<td>Poli</td>
<td>0.005</td>
<td>0.001</td>
<td>0.001</td>
<td>0.002</td>
<td>0.25</td>
<td>…</td>
</tr>
</tbody>
</table>

#### Per-user followship preference

<table>
<thead>
<tr>
<th>User</th>
<th>Y</th>
<th>N</th>
</tr>
</thead>
<tbody>
<tr>
<td>Alice</td>
<td>0.75</td>
<td>0.25</td>
</tr>
<tr>
<td>Bob</td>
<td>0.5</td>
<td>0.5</td>
</tr>
<tr>
<td>Mark</td>
<td>…</td>
<td>…</td>
</tr>
<tr>
<td>Michelle</td>
<td>…</td>
<td>…</td>
</tr>
<tr>
<td>Barack</td>
<td>…</td>
<td>…</td>
</tr>
</tbody>
</table>

#### Per-topic followee distribution

<table>
<thead>
<tr>
<th>User</th>
<th>Tech</th>
<th>Food</th>
<th>Poli</th>
</tr>
</thead>
<tbody>
<tr>
<td>Alice</td>
<td>0.8</td>
<td>0.19</td>
<td>0.01</td>
</tr>
<tr>
<td>Bob</td>
<td>0.6</td>
<td>0.3</td>
<td>0.1</td>
</tr>
<tr>
<td>Mark</td>
<td>…</td>
<td>…</td>
<td>…</td>
</tr>
<tr>
<td>Michelle</td>
<td>…</td>
<td>…</td>
<td>…</td>
</tr>
<tr>
<td>Barack</td>
<td>…</td>
<td>…</td>
<td>…</td>
</tr>
</tbody>
</table>

#### Global followee distribution

<table>
<thead>
<tr>
<th>User</th>
<th>Alice</th>
<th>Bob</th>
<th>Mark</th>
<th>Michelle</th>
<th>Barack</th>
</tr>
</thead>
<tbody>
<tr>
<td>Global</td>
<td>0.005</td>
<td>0.001</td>
<td>0.244</td>
<td>0.25</td>
<td>0.5</td>
</tr>
</tbody>
</table>

---

**Topic-specific influence**

**Global popularity**
Generative Process of FLDA

For the $m^{th}$ user:
- Pick a per-user topic distribution $\theta_m$
- Pick a per-user followship preference $\mu_m$

For the $n^{th}$ word
- Pick a topic based on topic distribution $\theta_m$
- Pick a word based on per-topic word distribution $\phi_t$

For the $l^{th}$ followee
- Choose the cause based on followship preference $\mu_m$
- If content-related
  - Pick a topic based on topic distribution $\theta_m$
  - Pick a followee based on per-topic followee distribution $\sigma_t$
- Otherwise
  - Pick a followee based on global followee distribution $\pi$
Road Map

- Followship-LDA (FLDA) Model
- Scalable Gibbs Sampling Algorithm for FLDA
- A General Search Framework for Topic-Specific Influencers
- Experimental Results
Inference of FLDA Model

- **Gibbs Sampling**: A Markov chain Monte Carlo algorithm to approximate the distribution of latent variables based on the observed data.

- **Gibbs Sampling Process**
  - Begin with some initial value for each variable
  - Iteratively sample each variable conditioned on the current values of the other variables, then update the variable with its new value (100s of iterations)
  - The samples are used to approximate distributions of latent variables

- **Derived conditional distributions for FLDA Gibbs Sampling**

  \[ p(z_{m,n}=t|z_{m,n}) \]: prob. topic of \( n^{th} \) word of \( m^{th} \) user is \( t \) given the current values of all others

  \[ p(y_{m,l}=r|y_{m,l}) \]: prob. preference of \( l^{th} \) link of \( m^{th} \) user is \( r \) given the current values of all others

  \[ p(x_{m,l}=t|x_{m,l}, y_{m,l}=1) \]: prob. topic of \( l^{th} \) link of \( m^{th} \) user is \( t \) given the current values of all others
Gibbs Sampling for FLDA

- In each pass of data, for the \( m \)th user
  - For the \( n \)th word (observed value \( w \)),
    - Sample a new topic assignment \( t' \), based on \( p(z_{m,n} | -z_{m,n}) \)
  - For \( l \)th followee (observed value \( e \)),
    - Sample a new preference \( r' \), based on \( p(y_{m,l} | -y_{m,l}) \)
    - If \( r'=1 \) (content based), sample a new topic \( t' \), based on \( p(x_{m,l} | -x_{m,l}, y_{m,l}=1) \)
  - Keep counters while sampling
    - \( c_{m,w,t} \): # times \( w \) is assigned to \( t \) for \( m \)th user
    - \( d_{m,e,t} \): # times \( e \) is assigned to \( t \) for \( m \)th user

- Estimate posterior distributions for latent variables

\[
p(t|m) = \frac{c_{m}^{*,t} + d_{m}^{*,t} + \alpha_t}{c_{m}^{*,*} + d_{m}^{*,*} + \sum_{i=1}^{K} \alpha_i}
\]

per-user topic distribution

\[
p(e|t) = \frac{d_{e,t}^{*,t} + \gamma_e}{d_{e}^{*,*} + \sum_{i=1}^{M} \gamma_i}
\]

per-topic followee distribution (influence)
Distributed Gibbs Sampling for FLDA

- **Challenge:** Gibbs Sampling process is sequential
  - Each sample step relies on the most recent values of all other variables.
  - Sequential algorithm would run for **21 days**, on a high end server (192 GB RAM, 3.2GHz processor) for a Twitter data set with 1.8M users, 2.4B words and 183M links.

- **Observation:** dependency between variable assignments is relatively weak, given the abundance of words and links

- **Solution:** Distributed Gibbs Sampling for FLDA
  - Relax the sequential requirement of Gibbs Sampling
  - Implemented on Spark – a distributed processing framework for iterative machine learning workloads
Distributed Gibbs Sampling for FLDA

Global Counter: \( c_{w,t}^{w}, d_{e,t}^{e}, \ldots \)

<table>
<thead>
<tr>
<th>Word</th>
<th>W1</th>
<th>W2</th>
<th>W3</th>
</tr>
</thead>
<tbody>
<tr>
<td>Topic</td>
<td>T5</td>
<td>T4</td>
<td>T5</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Link</th>
<th>L1</th>
<th>L2</th>
</tr>
</thead>
<tbody>
<tr>
<td>Prefer</td>
<td>Y</td>
<td>N</td>
</tr>
<tr>
<td>Topic</td>
<td>T4</td>
<td>.</td>
</tr>
</tbody>
</table>

Local Counter: \( c_{m}^{w,t}, d_{m}^{e,t}, \ldots \)
Issues with Distributed Gibbs Sampling

- Check-pointing for fault tolerance
  - Spark used lineage for fault tolerance

- Frequency of global synchronization
  - Even every 10 iterations won’t affect the quality of result

- Random number generator in a distributed environment
  - Provably independent multiple streams of uniform numbers
  - A long-period jump ahead $F_2$-Linear Random Number Generator

- Efficient local computation
  - Aware of the local memory hierarchy
  - Avoid random memory access by sampling in order
Road Map

- Followship-LDA (FLDA) Model
- Scalable Gibbs Sampling Algorithm for FLDA
- A General Search Framework for Topic-Specific Influencers
- Experimental Results
Search Framework For Topic-Specific Influencers

- SKIT: search framework for topic-specific key influencers
  - Input: key words
  - Output: ranked list of key influencers
  - Can plug in various key influencer methods
    - FLDA, Link-LDA, Topic-Sensitive PageRank (TSPR), TwitterRank

- Step 1: Derivation of interested topics from key words
  - Treat the key words as the content of a new user
  - “Fold in” to the existing model
  - Result: a topic distribution of the input \(<p_1, p_2, ..., p_t, ...>\)

- Step 2: Compute influence score for each user
  - Per-topic influence score for each user from FLDA: \(inf_t^u\)
  - Combine the score \(inf_u = \sum p_t \times inf_t^u\)

- Step 3: Sort users by influence scores and return the top influencers
Road Map

- Followship-LDA (FLDA) Model
- Scalable Gibbs Sampling Algorithm for FLDA
- A General Search Framework for Topic-Specific Influencers
- Experimental Results
Effectiveness on Twitter Dataset

- Twitter Dataset crawled in 2010
  - 1.76M users, 2.36B words, 183M links, 159K distinct words

<table>
<thead>
<tr>
<th>Topic</th>
<th>Top-10 key words</th>
<th>Top-5 Influencers</th>
</tr>
</thead>
<tbody>
<tr>
<td>“Information Technology”</td>
<td>data, web, cloud, software, open, windows, Microsoft, server, security, code</td>
<td>Tim O’ Reilly, Gartner Inc, Scott Hanselman (software blogger), Jeff Atwood (software blogger, co-founder of stackoverflow.com), Elijah Manor (software blogger)</td>
</tr>
<tr>
<td>“Cycling and Running”</td>
<td>Bike, ride, race, training, running, miles, team, workout, marathon, fitness</td>
<td>Lance Armstrong, Levi Leipheimer, Geroge Hincapie (all 3 US Postal pro cycling team members), Johan Bruyeel (US Postal team director), RSLT (a pro cycling team)</td>
</tr>
<tr>
<td>“Jobs”</td>
<td>Business, job, jobs, management, manger, sales, services, company, service, small, hiring</td>
<td>Job-hunt.org, jobsguy.com, integritystaffing.com/blog (job search Ninja), jobConcierge.com, careerealism.com</td>
</tr>
</tbody>
</table>

- Interesting findings
  - Overall ~ 15% of followships are content-independent
  - Top-5 global popular users: Pete Wentz (singer), Ashton Kutcher (actor), Greg Grunberg (actor), Britney Spears (singer), Ellen Degeneres (comedian)
Effectiveness on Tencent Weibo Dataset

- Tencent Weibo Dataset form KDD Cup 2012
  - 2.33M users, 492M words, 51M links, 714K distinct words
  - VIP users – used as “ground truth”
    - Organized in categories
    - Manually labeled “key influencers” in the corresponding categories
- Quality evaluation
  - Search Input: content of the VIP user
  - Precision: how many of the top-k results are the other VIP users of the same category
  - Measure: Mean Average Precision (MAP) for all the VIP users across all categories

- FLDA is significantly better than previous approaches
  - 2x better than TSPR and TwitterRank
  - 1.6x better than Link-LDA
- Distributed FLDA achieves similar quality results as sequential FLDA
Efficiency of Distributed FLDA

Setup
- Sequential: a high end server (192 GB RAM, 3.2GHz)
- Distributed: 27 servers (32GB RAM, 2.5GHz, 8 cores) connected by 1Gbit Ethernet, 1 master and 200 workers

Dataset
- Tencent Weibo: 2.33M users, 492M words, 51M links
- Twitter: 1.76M users, 2.36B words, 183M links

Sequential vs. Distributed
- 100 topics, 500 iterations

<table>
<thead>
<tr>
<th></th>
<th>Sequential</th>
<th>Distributed</th>
</tr>
</thead>
<tbody>
<tr>
<td>Weibo</td>
<td>4.6 days</td>
<td>8 hours</td>
</tr>
<tr>
<td>Twitter</td>
<td>21 days</td>
<td>1.5 days</td>
</tr>
</tbody>
</table>
Scalability of Distributed FLDA

- Scale up experiment on Twitter Dataset

![Graphs showing scalability of Distributed FLDA]

- Corpus size: 12.5%, 25%, 50%, 100%
- Wall clock time per iteration (sec)
- #concurrent workers
- nTopics: 25, 50, 100, 200
Conclusion

- FLDA model for topic-specific influence analysis
  - Content + Links

- Distributed Gibbs Sampling algorithm for FLDA
  - Comparable results as sequential Gibbs Sampling algorithm
  - Scales nicely with # workers and data sizes

- General search framework for topic-specific key influencers
  - Can plug in various key influencer methods

- Significantly higher quality results than previous work
  - 2x better than PageRank based methods
  - 1.6x better than Link-LDA