Finding Users/content of Interest in Micro-blogging Systems

Cédric du Mouza ... with the little help from my friends

CEDRIC Laboratory - CNAM - Paris, France

Russian-French Work. on Big Data, 2017
Introduction

Context

Growth of micro-blogging platforms: more and more users

**Issue:**

How to (efficiently) find users of interests?
Introduction

Context

Growth of micro-blogging platforms: more and more content posted

- about 700 millions of messages/day in 2016
- about 300 millions of messages/day in 2016
- 70 millions of publications/day in 2017
- 50 millions of pictures/day in 2017

Issue:

How to (efficiently) find content of interest?
## State of the art

<table>
<thead>
<tr>
<th>Method</th>
<th>+</th>
<th>-</th>
</tr>
</thead>
<tbody>
<tr>
<td>Content based</td>
<td>no need for interactions</td>
<td>difficult to describe short messages</td>
</tr>
<tr>
<td>Collaborative filtering</td>
<td>simple and good recommendations</td>
<td>size of the matrix and preprocessing time</td>
</tr>
<tr>
<td>Random-walks</td>
<td>reasonable cost</td>
<td>no need for important extra-storage</td>
</tr>
</tbody>
</table>
1 - User Recommendations
**Introduction**

**Method**

1. Which data is relevant?
   - Exploit the topological aspects of a social graph plus the semantic information

2. How can we process this information when the social network is huge in an "acceptable time"
   - Pre-computation approach (Landmark-based)
Model

Labeled Social Graph

- LSG

**B**
technology, news
- Microsoft announces a major bug windows 8 #bug
- Linux, now used by 23% of personal computers #stats
- Android KitKat in now live! bit.ly/uhias8z
- Is it the right moment to invest in #bitcoin?

**C**
bigdata, technology, science, news
- Data science is transforming the way we do business bit.ly/1rPwV6v
- #BigData is what happens when the cost of storing data falls below the cost of deciding to throw it away
- Intel launches its own distribution of Hadoop!
Model

Recommendation score

Our idea:
Exploit the topology plus the semantic information

Recommendation score

\[ \sigma(u, v, t) = \sum_{p \in P_{u,v}} \beta^{|p|} \omega_p(t) \]

\( P_{u,v} \) : Set of paths from \( u \) to \( v \)
\( \omega_p(t) \) : score of a path depending on the semantics and on authority
\( \beta^{|p|} \) : distance weight \(<1\) ("Decay")
Model

Recommendation score

Our idea:
Exploit the topology plus the semantic information

Recommendation score

\[ \sigma(u, v, t) = \sum_{p \in P_{u,v}} \beta^{|p|} \omega_p(t) \]

\( P_{u,v} \): Set of paths from u to v
\( \omega_p(t) \): score of a path depending on the semantics and on authority
\( \beta^{|p|} \): distance weight < 1 ("Decay")

Contains

- Connectivity
Model
Recommendation score

Our idea:
Exploit the topology plus the semantic information

Recommendation score

\[
\sigma(u, v, t) = \sum_{p \in P_{u,v}} \beta^{|p|} \omega_p(t)
\]

\(P_{u,v} : \) Set of paths from \(u\) to \(v\)

\(\omega_p(t) : \) score of a path depending on the semantics and on authority

\(\beta^{|p|} : \) distance weight \(< 1\) ("Decay")

Contains

- Connectivity
- Topological proximity
Model

Recommendation score

Our idea:
Exploit the topology plus the semantic information

Recommendation score

\[ \sigma(u, v, t) = \sum_{p \in P_{u,v}} \beta^{|p|} \omega_p(t) \]

\( P_{u,v} \) : Set of paths from \( u \) to \( v \)
\( \omega_p(t) \) : score of a path depending on the semantics and on authority
\( \beta^{|p|} \) : distance weight \(<1\) ("Decay")

Contains

- Connectivity
- Topological proximity
- Thematic relevance of the path
Model

How to scale: the landmark strategy
Model

How to scale: the landmark strategy

Landmarks selection
Model

How to scale: the landmark strategy

1. Landmarks selection
2. Calculation of recommendations for each Landmark
Model

How to scale: the landmark strategy

1. Landmarks selection
2. Calculation of recommendations for each Landmark
Model

How to scale: the landmark strategy

1. Landmarks selection
2. Calculation of recommendations for each Landmark
3. Calculation of recommendations to distance \( d \) from the request node
Model

How to scale: the landmark strategy

1. Landmarks selection
2. Calculation of recommendations for each Landmark
3. Calculation of recommendations to distance \( d \) from the request node
Model

How to scale: the landmark strategy

1. Landmarks selection
2. Calculation of recommendations for each Landmark
3. Calculation of recommendations to distance \( d \) from the request node

Composition property:
estimation of the recommendation score of \( v \) for \( u \) with composition of the scores between \((u, w)\) and \((w, v)\)
Experimentations

Results

**Figure :** Recall at N (Twitter)

**Figure :** User validation
2 - Content Recommendations
Data Analysis : Homophyly

We rely on similarity score proposed by Breese and al. based on retweets

<table>
<thead>
<tr>
<th>Distance</th>
<th># users</th>
<th>%</th>
<th>avg. similarity</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>3 229</td>
<td>02,65</td>
<td>0,0085</td>
</tr>
<tr>
<td>2</td>
<td>32 668</td>
<td>26,86</td>
<td>0,0014</td>
</tr>
<tr>
<td>3</td>
<td>81 645</td>
<td>67,13</td>
<td>0,0009</td>
</tr>
<tr>
<td>4</td>
<td>3 820</td>
<td>03,14</td>
<td>0,0010</td>
</tr>
<tr>
<td>5</td>
<td>43</td>
<td>00,03</td>
<td>0,0014</td>
</tr>
<tr>
<td>6</td>
<td>1</td>
<td>0</td>
<td>0,0008</td>
</tr>
<tr>
<td>Impossible</td>
<td>216</td>
<td>0,18</td>
<td>0,0017</td>
</tr>
</tbody>
</table>

Table : Similarity scores w.r.t. the user distance

Conclusion

- few users have a high similarity score
- direct homophyly (dist. 1) does not capture all similar users
Similarity graph

Example

**Figure**: Twitter Graph
Similarity graph

Example

Figure: Twitter graph
Similarity graph

Example

Figure: Twitter graph
Similarity graph

Example

**Figure**: Twitter graph
Similarity graph

Example

Figure: Similarity graph
Similarity graph

Characteristics

<table>
<thead>
<tr>
<th></th>
<th>Twitter graph</th>
<th>Similarity Graph</th>
</tr>
</thead>
<tbody>
<tr>
<td>No of nodes</td>
<td>2,182,867</td>
<td>1,149,374</td>
</tr>
<tr>
<td>No of edges</td>
<td>325,451,980</td>
<td>4,950,417</td>
</tr>
<tr>
<td>Mean Similarity Score</td>
<td>57.8</td>
<td>0.008</td>
</tr>
<tr>
<td>avg. out-degree</td>
<td>5.9</td>
<td>5.9</td>
</tr>
</tbody>
</table>

**Table**: Similarity graph characteristics

**Speedup**

The 2-order magnitude gain for the number of edges allows us to propose ”on-line” recommendations
Propagation model

Example

Model principle

A propagation probability which relies on the similarity with neighbors and their probability to like the tweet.

\[ P(t1) = \sum_{v \in N(t1)} P(v) \times P(v \rightarrow t1) \]

\[ P(v) = \frac{1}{|N(v)|} \]

\[ P(v \rightarrow t1) = \max \left\{ 0, \min \left\{ 1, \frac{1}{|N(v)|} \right\} \right\} \]

**Figure**: A tweet \( t1 \) is published
Propagation model

Example

Figure: X shares t1

\[ p(x, t1) = 1 \]
Propagation model

Example

\[ p(w, t1) = \frac{\sum_{v \in Fw} p(w \leftarrow v, t)}{|Fw|} = \frac{0 + 1 \times 0.5}{2} = 0.25 \]
Propagation model

Example

\[
p(u, t1) = \frac{0.25 \times 0.5}{2} = 0.0625
\]

**Figure:** Propagation
Experiment

Hits

The diagram shows the hit number as a function of the recommendation limit per day (per user) for different algorithms: Bayes, CF, GraphJet, and SimGraph. The x-axis represents the recommendation limit per day, while the y-axis shows the hit number. The graph compares the performance of these algorithms in a micro-blogging context, indicating how they perform with varying recommendation limits.
Experiment
Runtime comparison

<table>
<thead>
<tr>
<th></th>
<th>initialization time</th>
<th>processing per message</th>
</tr>
</thead>
<tbody>
<tr>
<td>Bayes</td>
<td>0.0001s (per user)</td>
<td>0.9s</td>
</tr>
<tr>
<td>CF</td>
<td>60s (per user)</td>
<td>0.00005s</td>
</tr>
<tr>
<td>SimGraph</td>
<td>0.5s (per user)</td>
<td>0.03s</td>
</tr>
</tbody>
</table>

**Table:** Characteristics of the similarity graph

<table>
<thead>
<tr>
<th></th>
<th>initialization time</th>
<th>processing per user</th>
</tr>
</thead>
<tbody>
<tr>
<td>Graphjet</td>
<td>0</td>
<td>0.3s</td>
</tr>
</tbody>
</table>
3 - Graph partitioning
Graph partitioning

Problem

Large graph issues

- most graph algorithms hardly scale with very large graphs
- graph partitioning strategies moderately efficient for real large graphs (like social network ones)
- vertex partitioning leads to unbalanced computation and high communication cost [PowerGraph, Joseph et al., OSDI’12]

Our contribution

- an edge partitioning dedicated to random walk-based algorithms
- which ensures workload balance and limit the communication costs
- by taking into account the communities
Our proposal

1. Select k seeds (e.g. nodes with the highest degree)
2. Compute the distance from each vertex to the different seeds (potentially using a threshold)
3. Aggregate the 2 vertex-scores to get the edge score w.r.t. each seed
4. Allocate the edge to the closest seed (block)
5. When all edges are allocated, potentially split or merge blocks
Experimentations

Results

<table>
<thead>
<tr>
<th>#Partitions</th>
<th>Random-Vertex-Cut[18]</th>
<th>Block-based partitioning</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>VRF</td>
<td>real mess.</td>
</tr>
<tr>
<td>64</td>
<td>15.38</td>
<td>303.5m</td>
</tr>
<tr>
<td>100</td>
<td>17.61</td>
<td>381.9m</td>
</tr>
<tr>
<td>150</td>
<td>19.68</td>
<td>464.8m</td>
</tr>
<tr>
<td>200</td>
<td>21.12</td>
<td>525.5m</td>
</tr>
</tbody>
</table>

**Figure**: Messages for FMRW

**Figure**: Runtime for FMRW
4 - What else at CNAM?
Other topics at CNAM

- Big datawarehouse (Decision systems + big data)
- Data quality (for SSH, geographical data, artefacts, Web data)
- Semantics + open and linked data (different context including Web data, medicine, history, geography, etc)
- Security (data anonymization, early integration of security requirements, etc)
Conclusion

Thank you for your attention