Parallel Clustering of High-Dimensional Social Media Data Streams

NSF STREAM 2015 Workshop
Indianapolis, Indiana

October 27-28, 2015

Judy Qiu
School of Informatics and Computing
Indiana University
The Convergence of Cloud, Big Data and Mobile: What could happen in 5 years?
Big Trend: integrated batch and streaming analysis
Example: Google DataFlow
- Supporting non-trivial streaming algorithms requiring global synchronization
Parallel Tweet Online Clustering with Apache Storm

- IU DESPIC analysis pipeline for meme clustering and classification: Detecting Early Signatures of Persuasion in Information Cascades
- Implement with HBase + Hadoop (Batch) and HBase + Storm (Streaming) + ActiveMQ
- 2 million streaming tweets processed in 40 minutes; 35,000 clusters
- Storm Bolts coordinated by ActiveMQ to synchronize parallel cluster center updates – add loops/iterations to Apache Storm

Xiaoming Gao, Emilio Ferrara, Judy Qiu, Parallel Clustering of High-Dimensional Social Media Data Streams Proceedings of CCGrid, May 4-7, 2015
Social media data stream clustering

- Group social messages sharing similar social meaning
- Useful in meme detection, event detection, social bots detection, etc.
Social media data stream clustering

- Recent progress in learning **data representations** and **similarity metrics**
- High-quality clusters: leverage both textual and network information (high-dimensional vectors)
- Expensive similarity computation: 43.4 hours to cluster 1 hour’s worth of data with sequential algorithm
- **Goal:** meet real-time constraint through parallelization
- **Challenge:** efficient global synchronization in DAG-oriented parallel processing frameworks
Sequential algorithm for clustering tweet stream

- Online K-Means with sliding time window and outlier detection
- Group tweets as protomemes: hashtags, mentions, URLs, and phrases.
- Cluster protomemes using similarity measurement:

  - **Common user similarity**:
    \[ S_u(P_i, P_j) = \frac{\sum_{u \in U_i \cap U_j} P_{iu}P_{ju}}{\sqrt{\sum_{u \in U_i} P_{iu}^2} \sqrt{\sum_{u \in U_j} P_{ju}^2}} \]

  - **Common tweet ID similarity**:
    \[ S_t(P_i, P_j) = \frac{|P_i \cap P_j|}{\sqrt{|P_i|} \sqrt{|P_j|}} \]

  - **Content similarity**:
    \[ S_c(P_i, P_j) = \frac{\sum_{w \in W_i \cap W_j} P_{iw}P_{jw}}{\sqrt{\sum_{w \in W_i} P_{iw}^2} \sqrt{\sum_{w \in W_j} P_{jw}^2}} \]

  - **Diffusion similarity**:
    \[ S_d(P_i, P_j) = \frac{|N_i \cap N_j|}{\sqrt{|N_i|} \sqrt{|N_j|}} \quad N_\ell = U_\ell \cup M_\ell \cup R_\ell \quad \text{(Posting + mentioned + retweeting)} \]

  - **Combinations**:
    \[ \text{MAX}(P_i, P_j) = \max_k \{ S_k(P_i, P_j) \} \]
    \[ \mathcal{L}(P_i, P_j) = \sum_k \omega_k S_k(P_i, P_j) \]
Sequential Algorithm for Clustering Tweet Stream

- Online (streaming) K-Means clustering algorithm with *sliding time window* and *outlier detection*

- Group tweets in a time window as protomemes:
  - Label protomemes (points in space to be clustered) by “markers”, which are *Hashtags, User mentions, URLs*, and *phrases*
  - A phrase is defined as the textual content of a tweet that remains after removing the hashtags, mentions, URLs, and after stopping and stemming
    - Number of tweets in a protomeme: Min: 1, Max :206, Average 1.33

- Note a given tweet can be in more than one protomeme
  - One tweet on average appears in 2.37 protomemes
  - Number of protomemes is 1.8 times number of tweets
Online K-Means clustering

(1) Slide time window by one time step
(2) Delete old protomemes out of time window from their clusters
(3) Generate protomemes for tweets in this step
(4) For each new protomeme classify in old or new cluster (outlier)
Parallelization with Storm – Challenges

DAG organization of parallel workers: hard to synchronize cluster information

Synchronization initiation methods:
- Spout initiation by broadcasting INIT message
- Clustering bolt initiation by local counting
- Sync coordinator initiation by global counting (of #protomemes)

Suffer from variation of processing speed

Parallelize Similarity Calculation

Calculate Cluster Centers

Worker Process
- Clustering Bolt

Protomeme Generator Spout

tweet stream

ActiveMQ Broker

Synchronization Coordinator Bolt
## Scalability Comparison

### Full-centroids synchronization

<table>
<thead>
<tr>
<th>Number of clustering bolts</th>
<th>Total processing time (sec)</th>
<th>Compute time / sync time</th>
<th>Sync time per batch (sec)</th>
<th>Avg. length of sync message</th>
</tr>
</thead>
<tbody>
<tr>
<td>3</td>
<td>67603</td>
<td>30.3</td>
<td>6.71</td>
<td>22,113,520</td>
</tr>
<tr>
<td>6</td>
<td>35207</td>
<td>15.1</td>
<td>6.71</td>
<td>21,595,499</td>
</tr>
<tr>
<td>12</td>
<td>19295</td>
<td>7.0</td>
<td>7.32</td>
<td>22,066,473</td>
</tr>
<tr>
<td>24</td>
<td>11341</td>
<td>3.2</td>
<td>8.24</td>
<td>22,319,413</td>
</tr>
<tr>
<td>48</td>
<td>7395</td>
<td>1.5</td>
<td>9.15</td>
<td>21,489,950</td>
</tr>
<tr>
<td>96</td>
<td>6965</td>
<td>0.7</td>
<td>12.93</td>
<td>21,536,799</td>
</tr>
</tbody>
</table>

### Cluster-delta synchronization

<table>
<thead>
<tr>
<th>Number of clustering bolts</th>
<th>Total processing time (sec)</th>
<th>Compute time / sync time</th>
<th>Sync time per batch (sec)</th>
<th>Avg. length of sync message</th>
</tr>
</thead>
<tbody>
<tr>
<td>3</td>
<td>50381</td>
<td>252.6</td>
<td>0.62</td>
<td>2,525,896</td>
</tr>
<tr>
<td>6</td>
<td>22949</td>
<td>96.4</td>
<td>0.73</td>
<td>2,529,779</td>
</tr>
<tr>
<td>12</td>
<td>11560</td>
<td>42.2</td>
<td>0.81</td>
<td>2,532,349</td>
</tr>
<tr>
<td>24</td>
<td>6221</td>
<td>21.7</td>
<td>0.81</td>
<td>2,544,095</td>
</tr>
<tr>
<td>48</td>
<td>3490</td>
<td>8.4</td>
<td>1.08</td>
<td>2,559,221</td>
</tr>
<tr>
<td>96</td>
<td>2494</td>
<td>2.5</td>
<td>2.17</td>
<td>2,590,857</td>
</tr>
</tbody>
</table>

Reduce synchronization overhead by sending incremental changes to the centroid vector.
- Speedup on up to 96 bolts on two clusters, Moe and Madrid
- Red curve is old online K-means algorithm; green and blue are the new algorithm
- Full Twitter – 1000 way parallelism (expected)
Acknowledgements

Xiaoming Gao

Prof. Filippo Menczer & CNETS
Complex Networks and Systems

SALSA HPC Group
School of Informatics and Computing
Indiana University