Parallel Gibbs Sampling for Hierarchical Dirichlet Processes via Gamma Processes Equivalence

Dehua Cheng and Yan Liu

Dehua Cheng

University of Southern California

August 25, 2014
Parallel HDP

Big data era

Parallel Computing

Topic Modeling

D. Cheng & Y. Liu (USC)
Sampling for LDA & HDP

There are several ways to improve the sampling algorithms for LDA & HDP.

- Collapsed Gibbs sampler [Griffiths and Steyvers, 2004]
  - “Remove” variables to improve mixing rate

- Parallel sampling algorithm:
  - Parallel Gibbs sampler with graph coloring [Gonzalez et al, 2011]
  - Hog-wild Gibbs sampler [Asuncion et al, 2008]
  - Introduce auxiliary variables to create conditional independence [Williamson et al, 2013]

- Our approach:
  - We propose a parallel sampling algorithm for an equivalent model, where the Dirichlet-Multinomial hierarchy is replaced by Gamma-Poisson hierarchy [Zhou et al, 2012].

- In other words, we replace the variables!
The Equivalent Model: Overview

\( \alpha \)

\( \pi \)

\( z \)

\( \theta \)

\( x \)

\( N \)

\( \theta_k \)

\( n_k \)

\( \infty \)

\( \pi_1, \pi_2, \ldots \) \sim \text{Dirichlet}(\alpha_1, \alpha_2, \ldots),

\( n_1, n_2, \ldots \) \sim \text{Multi}(N, \pi_1, \pi_2, \ldots).

\( \pi_k' \sim \text{Gamma}(\alpha_k, 1), \)

\( n_k \sim \text{Poisson}(C \cdot \pi_k') \).
The Equivalent Model: Overview

Core advantages:
- Rich conditional independence, great for parallel inference
- No inflation on the number of variables

Optimistically speaking, we get conditional independence for free!
The Equivalent Model: Bottlenecks

However, bad news awaits…

• Restriction from the observation
  • All $n_k$ are linked together, because they sum up to $N$, which is fixed given observation.

• Disconnection from the observation
  • The words generated by the equivalent model are pre-grouped by topics.
  • The words in observation are NOT pre-grouped by topics.
Our solution:

Sampling from the empirical distribution of the observation rather than the observation itself.

Based on this approach, we propose an approximate parallel sampling algorithm!
The Equivalent Model: Solution (cont’)

We keep the original observation \( X = \{x_1, x_2, ..., x_N\} \) in a stack \( S \), and we build the resampled observation \( X'_k = \{x'_{k,1}, x'_{k,2}, ..., x'_{k,n_k}\} \) while updating \( n_k \)...

- Add a word to \( X'_k \) and \( n_k \rightarrow n_k + 1 \):
  - Pop a word from stack \( S \), add it to \( X'_k \) with certain probability. If failed, push the word back to stack \( S \).
We keep the original observation $X = \{x_1, x_2, ..., x_N\}$ in a stack $S$, and we build the resampled observation $X'_k = \{x'_{k,1}, x'_{k,2}, ..., x'_{k,n_k}\}$ while updating $n_k$...

- Add a word to $X'_k$ and $n_k \rightarrow n_k + 1$:
  - Pop a word from stack $S$, add it to $X'_k$ with certain probability. If failed, push the word back to stack $S$.

- Delete a word from $X'_k$ and $n_k \rightarrow n_k - 1$:
  - Randomly choose a word from $X'_k$, delete it with certain probability. If succeed, push the word back to stack $S$.

Other variables are easy to update due to conjugate prior...
The Equivalent Model: Solution (cont’)

For each topic $k$, do asynchronously in parallel:
   For each document $d$, do in parallel:
      Do:
         For $X_{dk}$, add or delete a word;
      End for;
      Update $\theta_k, \pi'_{dk}$;
      Update $\alpha_k$;
   End for;
Return $\pi_{dk} \propto \pi'_{dk}, \theta_k, \alpha_k$, for all $d, k$;
Experimental Results

Dataset: NIPS 1-17 data\[1\]
#D=2,484 and #W=3,280,697
Baseline: Gibbs sampler [Teh et al, 2006],
Synch [Asuncion et al, 2008]
Number of Processors: 1,4,16.

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>G2PP</td>
<td>1</td>
<td>1467 ± 345(1164)</td>
<td>2020</td>
</tr>
<tr>
<td>G2PP</td>
<td>4</td>
<td>481 ± 101(306)</td>
<td>2016</td>
</tr>
<tr>
<td>G2PP</td>
<td>16</td>
<td>102 ± 45(63)</td>
<td>2004</td>
</tr>
<tr>
<td>Gibbs</td>
<td>1</td>
<td>1920 ± 880(701)</td>
<td>2060</td>
</tr>
<tr>
<td>Synch</td>
<td>2</td>
<td>922 ± 419(368)*</td>
<td>2080</td>
</tr>
<tr>
<td>Synch</td>
<td>4</td>
<td>519 ± 331(188)*</td>
<td>2117</td>
</tr>
<tr>
<td>Synch</td>
<td>16</td>
<td>85.5 ± 42(49)*</td>
<td>2413</td>
</tr>
</tbody>
</table>

Experimental Results

Dataset: NY Times data\[2\]
\#D=300,000 and \#W=100,000,000 (approx)
Baseline: Synch [Asuncion et al, 2008]
Number of Processors: 16

On Bitcoin Blog: evaluation of interpretability
\#D=1,899 and \#W=554,508

\[2\]. http://archive.ics.uci.edu/ml/datasets/Bag+of+Words

<table>
<thead>
<tr>
<th>Topic 1</th>
<th>Topic 2</th>
<th>Topic 3</th>
<th>Topic 4</th>
<th>Topic 5</th>
</tr>
</thead>
<tbody>
<tr>
<td>BITCOIN</td>
<td>LIKE</td>
<td>BITCOIN</td>
<td>GOLD</td>
<td>SECURITY</td>
</tr>
<tr>
<td>CURRENCY</td>
<td>PEOPLE</td>
<td>CURRENCY</td>
<td>MARKET</td>
<td>MALWARE</td>
</tr>
<tr>
<td>ONLINE</td>
<td>BITCOIN</td>
<td>CHINA</td>
<td>PRICE</td>
<td>DATA</td>
</tr>
<tr>
<td>PAYMENT</td>
<td>ONE</td>
<td>BANK</td>
<td>FED</td>
<td>NSA</td>
</tr>
<tr>
<td>SAID</td>
<td>SEE</td>
<td>VIRTUAL</td>
<td>YEAR</td>
<td>NSA</td>
</tr>
<tr>
<td>DIGITAL</td>
<td>WOULD</td>
<td>CENTRAL</td>
<td>ECONOMY</td>
<td>NSA</td>
</tr>
<tr>
<td>FIRST</td>
<td>WAY</td>
<td>EXCHANGE</td>
<td>US</td>
<td>NSA</td>
</tr>
<tr>
<td>COMPANY</td>
<td>COULD</td>
<td>PRICE</td>
<td>INVESTORS</td>
<td>NSA</td>
</tr>
<tr>
<td>TRANSACTION</td>
<td>SEVEN</td>
<td>CHINESE</td>
<td>BUBBLE</td>
<td>NSA</td>
</tr>
<tr>
<td>ONE</td>
<td>RE</td>
<td>SYSTEM</td>
<td>RATE</td>
<td>NSA</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>NSA</td>
</tr>
</tbody>
</table>

D. Cheng & Y. Liu (USC)
Conclusion and Future Works

Conclusion:

We proposed an parallel sampling algorithm for an equivalent model of LDA&HDP, providing better trade-off between scalability, speed, and accuracy.

Future Works

• Design smarter scheduling to improve scalability.
• Modify the algorithm for online learning.
• Explore the proposed approach for other models.
References


Thank you!

See our poster at ???