Streaming Variational Bayes

Tamara Broderick, Nick Boyd, Andre Wibisono, Ashia C. Wilson, Michael I. Jordan
Overview

• Big Data inference generally non-Bayesian
• Why Bayes? Complex models, coherent treatment of uncertainty, etc.
• We deliver: SDA-Bayes, a framework for Streaming, Distributed, Asynchronous Bayesian inference
• Experiments on streaming topic discovery (Wikipedia: 3.6M docs, Nature: 350K docs)
Overview

- Big Data inference generally non-Bayesian
Overview

• Big Data inference generally non-Bayesian

• Why Bayes? Complex models, coherent treatment of uncertainty, etc.
Overview

• Big Data inference generally non-Bayesian

• Why Bayes? Complex models, coherent treatment of uncertainty, etc.

• We deliver: **SDA-Bayes**, a framework for **Streaming, Distributed, Asynchronous** Bayesian inference

Experiments on streaming topic discovery (Wikipedia: 3.6M docs, Nature: 350K docs)
Overview

• Big Data inference generally non-Bayesian

• Why Bayes? Complex models, coherent treatment of uncertainty, etc.

• We deliver: **SDA-Bayes**, a framework for **Streaming, Distributed, Asynchronous** Bayesian inference

• Experiments on streaming topic discovery (Wikipedia: 3.6M docs, Nature: 350K docs)
Background
Background

- **Posterior**: belief about unobserved parameters $\theta$ after observing data $x$
Background

- **Posterior**: belief about unobserved parameters $\theta$ after observing data $x$

- **Variational Bayes (VB)**: approximate posterior by solving optimization problem (min KL divergence)
Background

- **Posterior**: belief about unobserved parameters $\theta$ after observing data $x$

- **Variational Bayes (VB)**: approximate posterior by solving optimization problem (min KL divergence)

- **Batch VB**: solves VB using coordinate descent
Background

- **Posterior**: belief about unobserved parameters $\theta$ after observing data $x$

- **Variational Bayes (VB)**: approximate posterior by solving optimization problem (min KL divergence)

- **Batch VB**: solves VB using coordinate descent

- **Stochastic Variational Inference (SVI)**: solves VB using stochastic gradient descent
Background

• **Posterior**: belief about unobserved parameters $\theta$ after observing data $x$

• **Variational Bayes (VB)**: approximate posterior by solving optimization problem (min KL divergence)

• **Batch VB**: solves VB using coordinate descent

• **Stochastic Variational Inference (SVI)**: solves VB using stochastic gradient descent
SDA-Bayes: Streaming
SDA-Bayes: Streaming

- Posterior update is iterative:
**SDA-Bayes: Streaming**

- Posterior update is iterative:

\[
p(\theta \mid x_{\text{old}}, x_{\text{new}}) \propto p(\theta \mid x_{\text{old}}) \cdot p(x_{\text{new}} \mid \theta)
\]
**SDA-Bayes: Streaming**

- Posterior update is iterative:
  \[ p(\theta \mid x_{\text{old}}, x_{\text{new}}) \propto p(\theta \mid x_{\text{old}}) \cdot p(x_{\text{new}} \mid \theta) \]

- Choose any posterior approximation \( A \):
SDA-Bayes: Streaming

- Posterior update is iterative:
  \[ p(\theta \mid x_{\text{old}}, x_{\text{new}}) \propto p(\theta \mid x_{\text{old}}) \cdot p(x_{\text{new}} \mid \theta) \]

- Choose any posterior approximation \( A \):

  ![Diagram](image)
  
  data \( x \)  
  prior \( p(\theta) \)  
  batch alg  
  \( q(\theta) \approx p(\theta \mid x) \)  
  posterior
SDA-Bayes: Streaming

- Posterior update is iterative:

\[ p(\theta | x_{\text{old}}, x_{\text{new}}) \propto p(\theta | x_{\text{old}}) \cdot p(x_{\text{new}} | \theta) \]

- Choose any posterior approximation \( A \):

  ![Diagram]

  - data \( x \)
  - prior \( p(\theta) \)
  - batch alg

- Iterate approximation if matches prior form:
**SDA-Bayes: Streaming**

- Posterior update is iterative:

\[ p(\theta \mid x_{\text{old}}, x_{\text{new}}) \propto p(\theta \mid x_{\text{old}}) \cdot p(x_{\text{new}} \mid \theta) \]

- Choose any posterior approximation \( \mathcal{A} \):

  - data \( x \) 
  - prior \( p(\theta) \) 
  - batch alg \( \mathcal{A} \) 
  - posterior \( q(\theta) \approx p(\theta \mid x) \)

- Iterate approximation if matches prior form:
Can calculate posteriors in parallel and combine with Bayes' Rule:

- Could substitute approximation found by $A$ instead

Update is just addition if prior and approximate posterior are in same exponential family:

$$\frac{S}{DA} - \text{Bayes}: \frac{\prod_{n=1}^{N} p(x_n | \theta)}{\prod_{n=1}^{N} p(x_n)}$$

$$\propto q(\theta) \exp(-\sum_{n=1}^{N} (\theta_n - \theta_0) \cdot T(\theta))$$
SDA-Bayes: Distributed

- Can calculate posteriors in parallel and combine with Bayes’ Rule:
Can calculate posteriors in parallel and combine with Bayes’ Rule:

\[
p(\theta \mid x_1, \ldots, x_N)
\]

\[
\propto \prod_{n=1}^{N} p(x_n \mid \theta) \ p(\theta)
\]
**SDA-Bayes: Distributed**

- Can calculate posteriors in parallel and combine with Bayes’ Rule:

\[
p(\theta \mid x_1, \ldots, x_N) \\
\propto \left[ \prod_{n=1}^{N} p(x_n \mid \theta) \right] p(\theta) \\
\propto \left[ \prod_{n=1}^{N} p(\theta \mid x_n) p(\theta)^{-1} \right] p(\theta)
\]
SDA-Bayes: Distributed

• Can calculate posteriors in parallel and combine with Bayes’ Rule:

\[ p(\theta \mid x_1, \ldots, x_N) \]

\[ \propto \left[ \prod_{n=1}^{N} p(x_n \mid \theta) \right] p(\theta) \propto \left[ \prod_{n=1}^{N} p(\theta \mid x_n) p(\theta)^{-1} \right] p(\theta) \]

• Could substitute approximation found by $A$ instead
**SDA-Bayes: Distributed**

- Can calculate posteriors in parallel and combine with Bayes’ Rule:
  \[
p(\theta \mid x_1, \ldots, x_N)
  \propto \left[ \prod_{n=1}^{N} p(x_n \mid \theta) \right] p(\theta) \propto \left[ \prod_{n=1}^{N} p(\theta \mid x_n) p(\theta)^{-1} \right] p(\theta)
  \]

- Could substitute approximation found by \(A\) instead

- Update is just addition if prior and approximate posterior are in same exponential family:
**SDA-Bayes: Distributed**

- Can calculate posteriors in parallel and combine with Bayes’ Rule:

\[
p(\theta \mid x_1, \ldots, x_N)
\]

\[
\propto \left[ \prod_{n=1}^{N} p(x_n \mid \theta) \right] p(\theta) \propto \left[ \prod_{n=1}^{N} p(\theta \mid x_n) p(\theta)^{-1} \right] p(\theta)
\]

- Could substitute approximation found by \( A \) instead

- Update is just addition if prior and approximate posterior are in same exponential family:

\[
p(\theta \mid x_1, \ldots, x_N) \approx q(\theta) \propto \exp \left\{ \left[ \xi_0 + \sum_{n=1}^{N} (\xi_n - \xi_0) \right] \cdot T(\theta) \right\}
\]


**SDA-Bayes: Asynchronous**

- Each worker iterates:
  1. Collect a new data point $x$. 
  2. Copy the master posterior parameter locally: $\xi^{(\text{local})} \leftarrow \xi^{(\text{post})}$
  3. Compute the local approximate posterior parameter $\xi$ using $A$ with $\xi^{(\text{local})}$ as the prior parameter
  4. Return $\Delta\xi := \xi - \xi^{(\text{local})}$

- Each time the master receives $\Delta\xi$ from a worker, it updates synchronously:

  $\xi^{(\text{post})} \leftarrow \xi^{(\text{post})} + \Delta\xi$
Case Study: LDA
Case Study: LDA

- **Topic**: theme potentially shared by multiple documents
Case Study: LDA

- **Topic**: theme potentially shared by multiple documents
- **Latent Dirichlet Allocation** (LDA): a topic model
Case Study: LDA

• **Topic**: theme potentially shared by multiple documents

• **Latent Dirichlet Allocation** (LDA): a topic model

• (Unsupervised) inference problem: discover the topics and identify which topics occur in which documents
Experimental Setup

- SDA-Bayes with batch VB for $A$ vs. SVI (not designed for streaming)
- Training: 3.6M Wikipedia, 350K Nature
- Testing: 10K Wikipedia, 1K Nature
- Performance measure: log predictive probability on held-out words in held-out testing documents; higher is better
Experimental Setup

• **SDA-Bayes** with batch VB for *A vs. SVI* (not designed for streaming)
Experimental Setup

- **SDA-Bayes** with batch VB for **A vs. SVI** (not designed for streaming)

- Training: 3.6M Wikipedia, 350K Nature

- Testing: 10K Wikipedia, 1K Nature

Performance measure: log predictive probability on held-out words in held-out testing documents; higher is better.
Experimental Setup

- **SDA-Bayes** with batch VB for *A vs. SVI* (not designed for streaming)

- Training: 3.6M Wikipedia, 350K Nature

- Testing: 10K Wikipedia, 1K Nature

- Performance measure: **log predictive probability** on held-out words in held-out testing documents; higher is better
Results

- **SDA-Bayes** (streaming) as good as **SVI** (not streaming); 32 threads and 1 thread shown

<table>
<thead>
<tr>
<th></th>
<th>Wikipedia</th>
<th></th>
<th></th>
<th>Nature</th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>32-SDA</td>
<td>1-SDA</td>
<td>SVI</td>
<td>32-SDA</td>
<td>1-SDA</td>
<td>SVI</td>
</tr>
<tr>
<td>Log pred prob</td>
<td>-7.31</td>
<td>-7.43</td>
<td>-7.32</td>
<td>-7.11</td>
<td>-7.19</td>
<td>-7.08</td>
</tr>
<tr>
<td>Time (hours)</td>
<td>2.09</td>
<td>43.93</td>
<td>7.87</td>
<td><strong>0.55</strong></td>
<td>10.02</td>
<td>1.22</td>
</tr>
</tbody>
</table>
Results

• More threads in SDA improves runtime and performance
Results

• More threads in SDA improves runtime and performance
Results

- SVI is sensitive to the pre-specified number of documents $D$
Further information

• Streaming, distributed Bayesian learning without performance loss


• Code and slides at www.tamarabroderick.com