Deep Latent Dirichlet Allocation with Topic-Layer-Adaptive Stochastic Gradient Riemannian MCMC

西安电子科技大学 XIDIAN UNIVERSITY

Yulai Cong*, Bo Chen*, Hongwei Liu*, Mingyuan Zhou#

* National Laboratory of Radar Signal Processing, Xidian University, Xi'an, Shaanxi, China

IROM Department, The University of Texas at Austin, Austin, TX, USA

THE UNIVERSITY OF TEXAS AT AUSTIN McCOMBS SCHOOL OF BUSINESS

Motivation

Big data prefer

- ➢ large-capacity models like deep latent variable models (LVMs)
- ➢ scalable inference methods like SG-MCMC

However, ***to make a deep LVM scalable is challenging***, because

- ➢ gradients of model parameters are difficult to compute
- ➢ ***different layers may be suitable for different learning rates***

Most ***existing methods*** adopt greedy layer-wise training, which

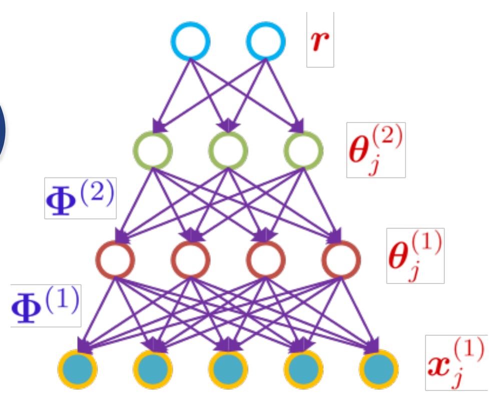
- ➢ usually uses a shared learning rates across all layers
- ➢ ***has no communication between different layers***

We develops a scalable SG-MCMC algorithm for deep latent Dirichlet allocation that jointly learns all hidden layers.

Background

❑ Poisson Gamma Belief Network (PGBN)

$$\boldsymbol{\theta}_j^{(L)} \sim \mathrm{Gam}\left(\boldsymbol{r}, 1/c_j^{(L+1)}\right),$$
$$\cdots$$
$$\boldsymbol{\theta}_j^{(l)} \sim \mathrm{Gam}\left(\boldsymbol{\Phi}^{(l+1)}\boldsymbol{\theta}_j^{(l+1)}, 1/c_j^{(l+1)}\right),$$
$$\cdots$$
$$\boldsymbol{x}_j^{(1)} \sim \mathrm{Pois}\left(\boldsymbol{\Phi}^{(1)}\boldsymbol{\theta}_j^{(1)}\right),\ \boldsymbol{\theta}_j^{(1)} \sim \mathrm{Gam}\left(\boldsymbol{\Phi}^{(2)}\boldsymbol{\theta}_j^{(2)}, \frac{p_j^{(2)}}{1-p_j^{(2)}}\right),$$



❑ Stochastic Gradient MCMC

For unknown $\boldsymbol{z}$ obeying $p(\boldsymbol{z}\,|X) \propto e^{-H(\boldsymbol{z})}$, one has a mini-batch update rule as

$$\boldsymbol{z}_{t+1} = \boldsymbol{z}_t + \varepsilon_t\Big\{-\big[\mathbf{D}(\boldsymbol{z}_t)+\mathbf{Q}(\boldsymbol{z}_t)\big]\nabla\tilde{H}(\boldsymbol{z}_t)+\Gamma(\boldsymbol{z}_t)\Big\}$$
$$+\mathcal{N}\Big(\mathbf{0}, \varepsilon_t\big[2\mathbf{D}(\boldsymbol{z}_t)-\varepsilon_t\hat{\mathbf{B}}_t\big]\Big),$$

where $\mathbf{D}(\boldsymbol{z})$ is a positive semidefinite matrix, $\mathbf{Q}(\boldsymbol{z})$ is a skew-symmetric matrix, $\Gamma_i(\boldsymbol{z}) = \sum_j \frac{\partial}{\partial z_j}\left[\mathbf{D}_{ij}(\boldsymbol{z}) + \mathbf{Q}_{ij}(\boldsymbol{z})\right]$, $\nabla\tilde{H}(\boldsymbol{z}) = \nabla\left[-\ln p(\boldsymbol{z}) - \rho\sum_{\boldsymbol{x}\in\tilde{X}} \ln p(\boldsymbol{x}\,|\boldsymbol{z})\right]$ is estimated with mini-batches, and $\hat{\mathbf{B}}_t$ is the SG noise variance estimate.

Contributions

To develop principled SG-MCMC for the PGBN, with $\boldsymbol{z} = \left\{\boldsymbol{\Phi}^{(1)}, \cdots, \boldsymbol{\Phi}^{(L)}, \boldsymbol{r}\right\}$ and simple settings $\mathbf{D}(\boldsymbol{z}) = \mathbf{G}(\boldsymbol{z})^{-1}, \mathbf{Q}(\boldsymbol{z}) = \mathbf{0}$, and $\hat{\mathbf{B}}_t = \mathbf{0}$, all we need are (a) the Fisher information matrix (FIM) $\mathbf{G}(\boldsymbol{z})$ and (b) $\nabla\tilde{H}(\boldsymbol{z})$.

Directly computing the FIM of PGBN is intractable. But an alternative view of the PGBN makes it straightforward.

❑ Deep Latent Dirichlet Allocation (DLDA)

Exploiting data augmentation and marginalization techniques, one may re-express the hierarchical model of the PGBN as DLDA as

$$x_{k\cdot}^{(L+1)} \sim \mathrm{Log}(\tilde{p}), K_L \sim \mathrm{Pois}[-\gamma_0 \ln(1-\tilde{p})], X^{(L+1)} = \textstyle\sum_{k=1}^{K_L} x_{k\cdot}^{(L+1)}\delta_{\boldsymbol{\phi}_k^{(L)}},$$
$$\left(x_{vj}^{(L+1)}\right)_j \sim \mathrm{Mult}\left[x_{v\cdot}^{(L+1)}, \left(q_j^{(L+1)}\right)_j / q_{\cdot}^{(L+1)}\right], m_{vj}^{(L)(L+1)} \sim \mathrm{SumLog}(x_{vj}^{(L+1)}, p_j^{(L+1)}),$$
$$\cdots$$
$$x_{vj}^{(l)} = \textstyle\sum_{k=1}^{K_l} x_{vkj}^{(l)}, \left(x_{vkj}^{(l)}\right)_v \sim \mathrm{Mult}\left(m_{kj}^{(l)(l+1)}, \boldsymbol{\phi}_k^{(l)}\right), m_{vj}^{(l-1)(l)} \sim \mathrm{SumLog}(x_{vj}^{(l)}, p_j^{(l)}),$$
$$\cdots$$
$$x_{vj}^{(1)} = \textstyle\sum_{k=1}^{K_1} x_{vkj}^{(1)}, \left(x_{vkj}^{(1)}\right)_v \sim \mathrm{Mult}\left(m_{kj}^{(1)(2)}, \boldsymbol{\phi}_k^{(1)}\right).$$

where $q_j^{(1)} := 1$, $q_j^{(l+1)} = \ln\left(1 + q_j^{(l)}/c_j^{(l+1)}\right)$, $p_j^{(l)} := 1 - e^{-q_j^{(l)}}$, and $\tilde{p} := q_{\cdot}^{(L+1)}/(c_0 + q_{\cdot}^{(L+1)})$.

❑ Analytical and Practical Fisher Information Matrix

Under the alternative DLDA representation, one may readily derive a block-diagonal and thus easily-inversed Fisher information matrix (FIM) as

$$\mathbf{G}(\boldsymbol{z}) = \mathrm{diag}\left[\mathbf{I}\left(\boldsymbol{\phi}_1^{(1)}\right), \cdots, \mathbf{I}\left(\boldsymbol{\phi}_{K_L}^{(L)}\right), \mathbf{I}(\boldsymbol{r})\right]$$

- Note the FIM, playing a similar role as the Hessian matrix in optimization, enables principled joint inference of the PGBN (DLDA).

❑ Reduced-Mean Inference on the Probability Simplex

For the interested batch posterior $(\boldsymbol{\phi}_k\,|-) \sim \mathrm{Dir}(x_{1k\cdot} + \eta, \ldots, x_{Vk\cdot} + \eta)$ on the probability simplex, one may have the mini-batch-based inference as

$$(\boldsymbol{\phi}_k)_{t+1} = \left[(\boldsymbol{\phi}_k)_t + \frac{\varepsilon_t}{M_k}\left[(\rho\tilde{\boldsymbol{x}}_{:k\cdot} + \eta) - (\rho\tilde{x}_{\cdot k\cdot} + \eta V)(\boldsymbol{\phi}_k)_t\right] + \mathcal{N}\left(\mathbf{0}, \frac{2\varepsilon_t}{M_k}\mathrm{diag}(\boldsymbol{\phi}_k)_t\right)\right]_{\angle}$$

which is derived with the reduced-mean parameterization $\boldsymbol{\varphi}_k$ of $\boldsymbol{\phi}_k$, namely $\boldsymbol{\phi}_k = \left((\boldsymbol{\varphi}_k)^T, 1 - \sum_v \varphi_{vk}\right)^T$, and Theorem 2 of [1].

- [1] Cong, Y., Chen, B., and Zhou, M. Fast simulation of hyperplane-truncated multivariate normal distributions. Bayesian Analysis Advance Publication, 2017.

❑ Topic-Layer-Adaptive Stochastic Gradient Riemannian (TLASGR)

Algorithm 1 TLASGR MCMC for DLDA (PGBN).

Input: Data mini-batches;
Output: Global parameters of DLDA (PGBN).

1: **for** $t = 1, 2, \cdots$ **do**
2:   /* *Collect local information*
3:   Upward-downward Gibbs sampling (Zhou et al., 2016a) on the t^{th} mini-batch for $\tilde{\boldsymbol{x}}_{:k\cdot}$, $\tilde{x}_{\cdot k\cdot}$, $\tilde{\boldsymbol{x}}_{:\cdot}^{(L+1)}$, and $\tilde{q}_{\cdot}^{(L+1)}$;
4:   /* *Update global parameters*
5:   **for** $l = 1, \cdots, L$ and $k = 1, \cdots, K_l$ **do**
6:     Update $M_k^{(l)}$ with (18); then $\boldsymbol{\phi}_k^{(l)}$ with (15);
7:   **end for**
8:   Update $M^{(L+1)}$ with (19) and then $\boldsymbol{r}$ with (17).
9: **end for**

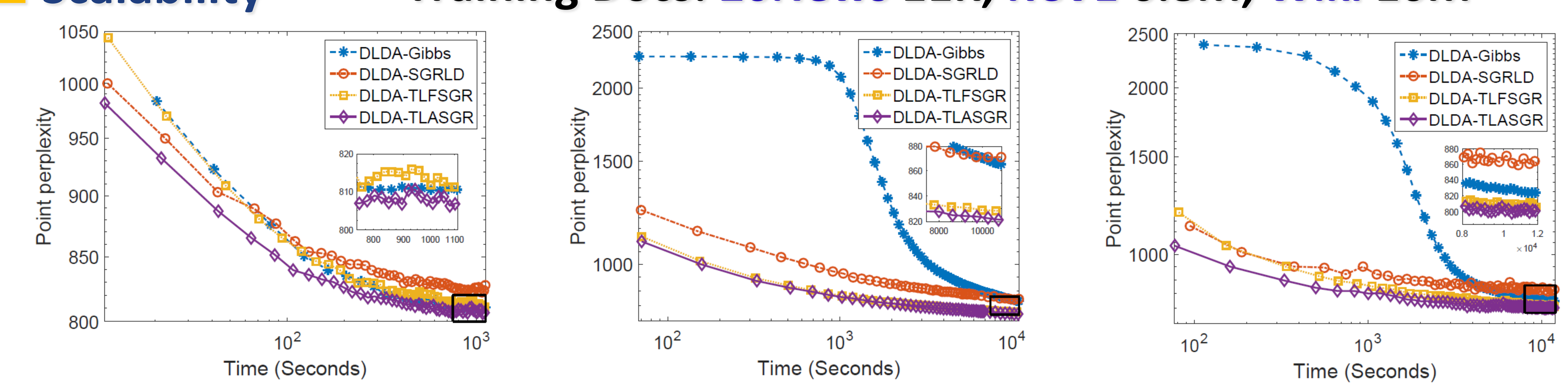
Experiment Results

❑ Perplexity Performance

Model	Method	Size	20 News	RCV1	Wiki
DLDA	TLASGR	128-64-32	757	815	**786**
DLDA	TLASGR	128-64	758	817	787
DLDA	TLASGR	128	770	823	802
DLDA	TLFSGR	128-64-32	760	817	789
DLDA	TLFSGR	128-64	759	819	791
DLDA	TLFSGR	128	772	829	804
DLDA	SGRLD	128-64-32	775	827	792
DLDA	SGRLD	128-64	770	823	792
DLDA	SGRLD	128	777	829	803
DLDA	Gibbs	128-64-32	**752**	**802**	—
DLDA	Gibbs	128-64	**754**	**804**	—
DLDA	Gibbs	128	768	818	—
DPFM	SVI	128-64	818	961	791
DPFM	MCMC	128-64	780	908	**783**
DPFA-SBN	SGNHT	128-64-32	827	1143	876
DPFA-RBM	SGNHT	128-64-32	896	920	942
nHDP	SVI	(10,10,5)	889	1041	932
LDA	Gibbs	128	893	1179	1059
FTM	Gibbs	128	887	1155	991
RSM	CD5	128	877	1171	1001

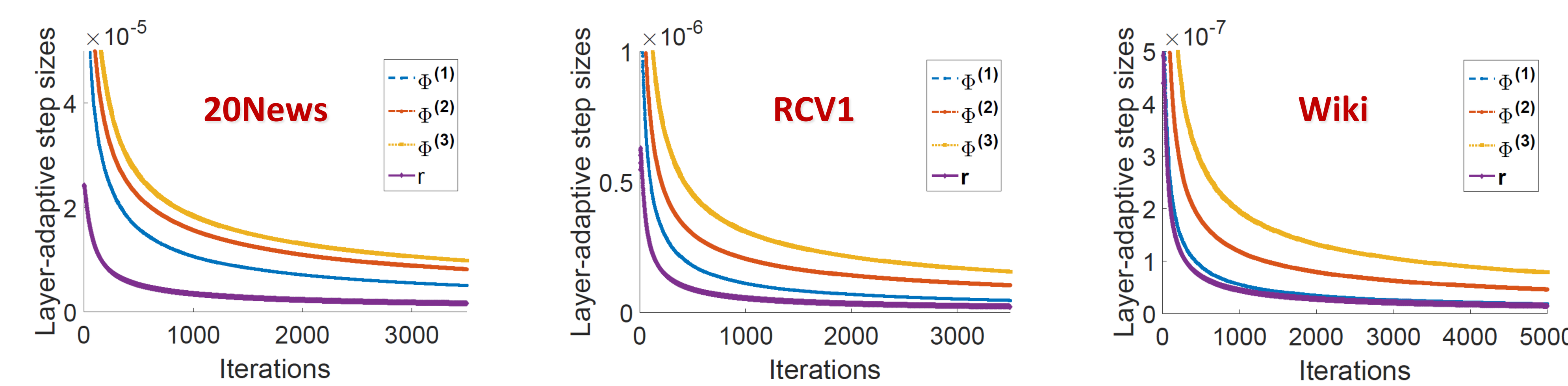
❑ Scalability

Training Docs: 20News 11K, RCV1 0.8M, Wiki 10M



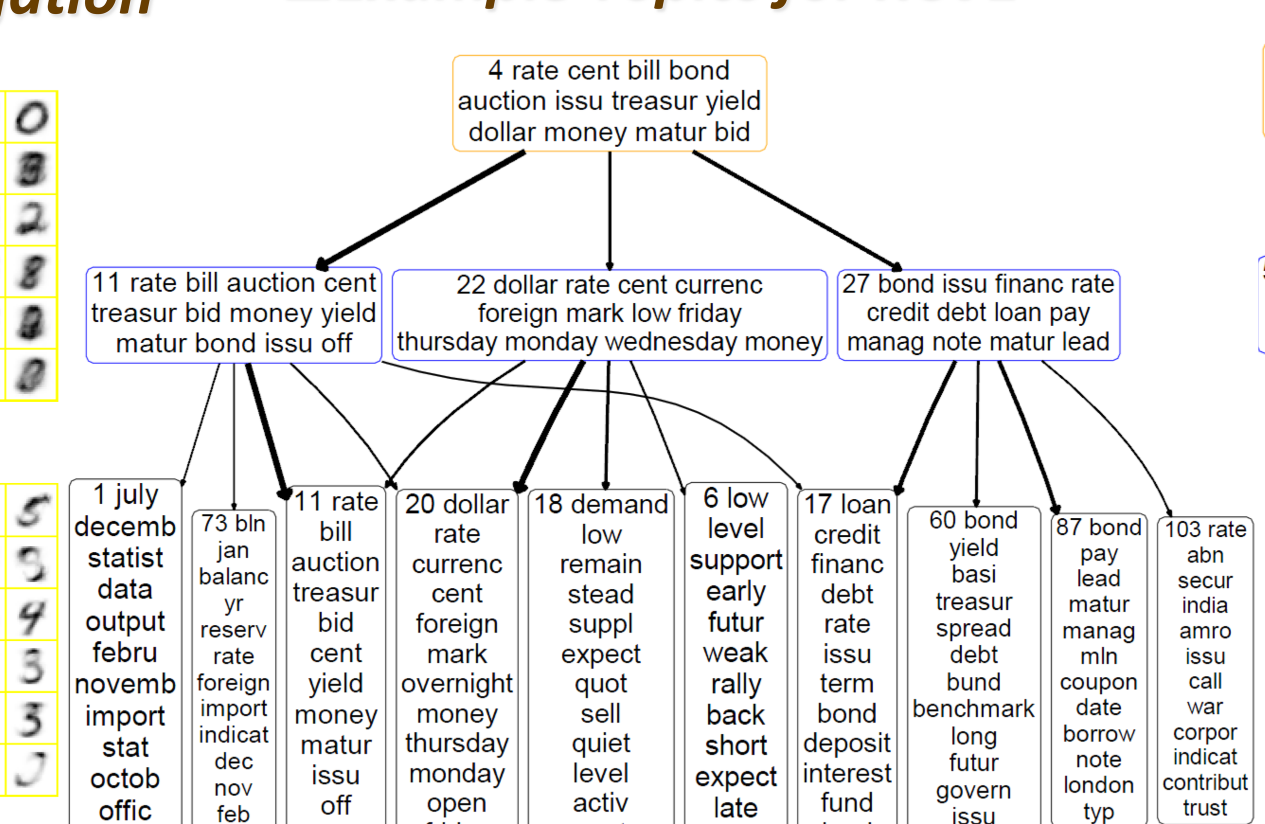
(a) A single-layer DLDA on 20News (b) DLDA of size 128-64 on RCV1 (c) DLDA of size 128-64 on Wiki

❑ *Scalable Joint Learning*



❑ *Better Information Propagation*

❑ *Example Topics for RCV1*



❑ *Example Topics for Wiki*