ICML 2017

THE 34TH INTERNATIONAL CONFERENCE ON MACHINE LEARNING

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

ICML @ Sydney, August 9, 2017

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

* National Laboratory of Radar Signal Processing,

* Collaborative Innovation Center of Information Sensing and Understanding,

Xidian University, Xi'an, Shaanxi, China

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

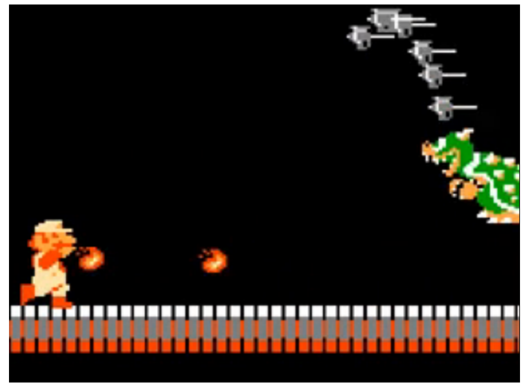
西安電子科技大學

XIDIAN UNIVERSITY

THE UNIVERSITY OF TEXAS AT AUSTIN

McCOMBS SCHOOL OF BUSINESS

Outline

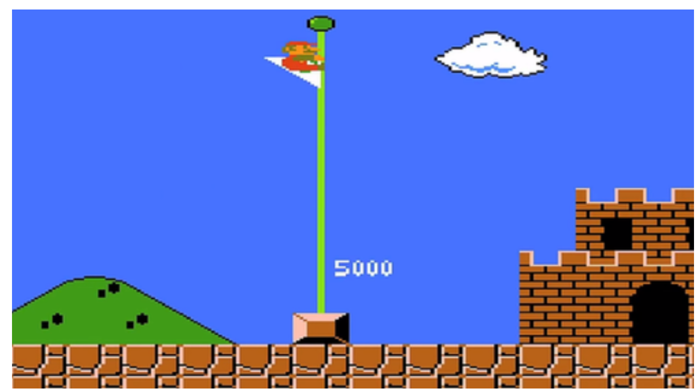


1.Motivation

3.Our Contribution

2. Barriers

4.Experiments



Motivation

Big data have

- ▶ Abundant information
- ▶ Huge volume
- ▶ …

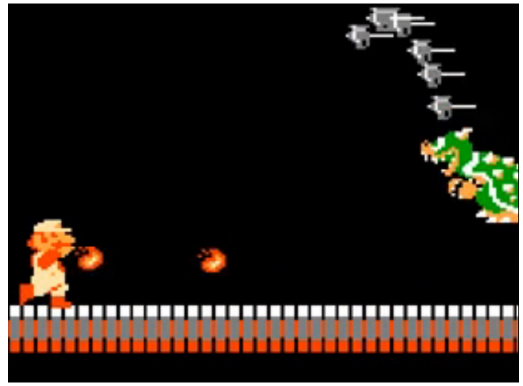
Thus, we prefer

- ▶ Large-capacity models
  - Deep latent variable models (LVMs)
  - …
- ▶ Scalable inference methods
  - Stochastic Gradient (SG) MCMC
  - …

Outline

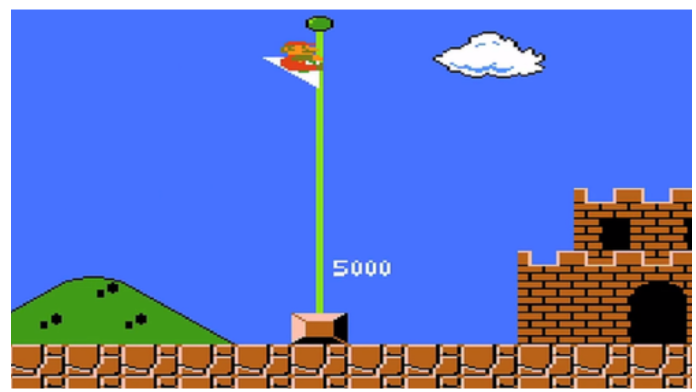
1.Motivation

2. Barriers



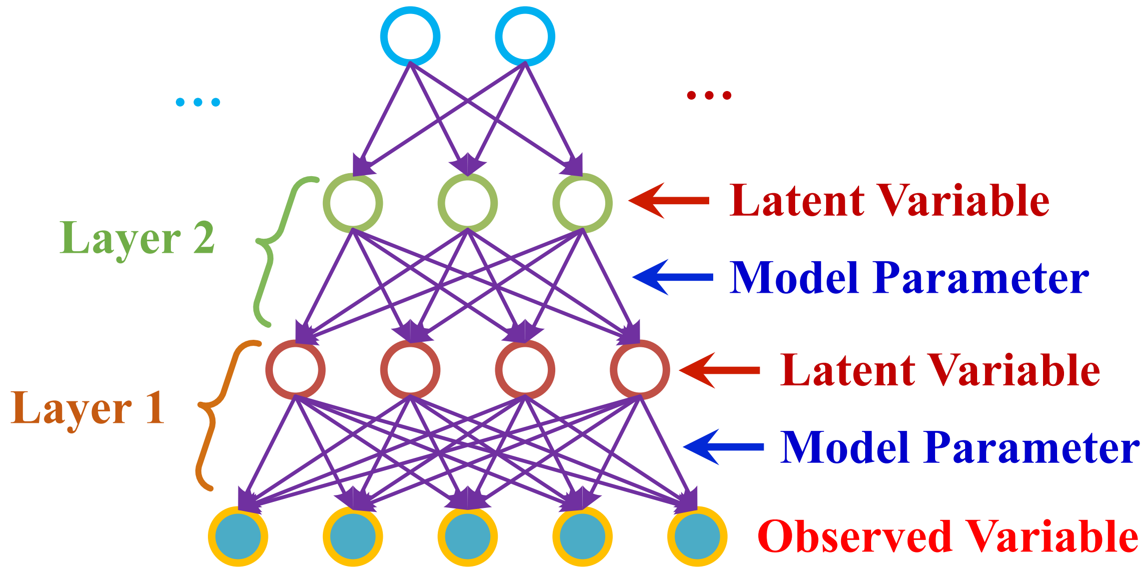
3.Our Contribution

4.Experiments



Barriers

To make a deep LVM scalable is challenging.

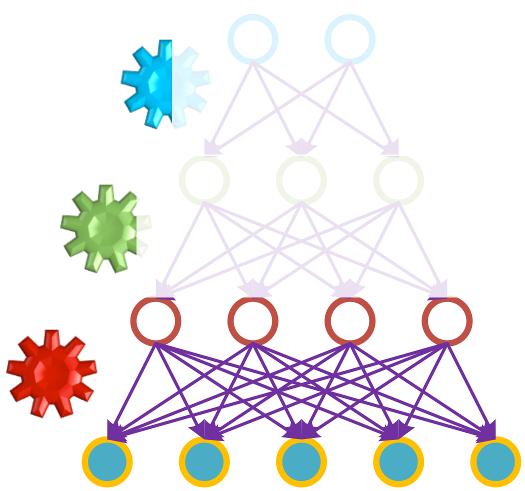


- ▶Gradients of model parameters are difficult to compute
- ▶Different layers, with different statistics, may require different learning rates

Barriers

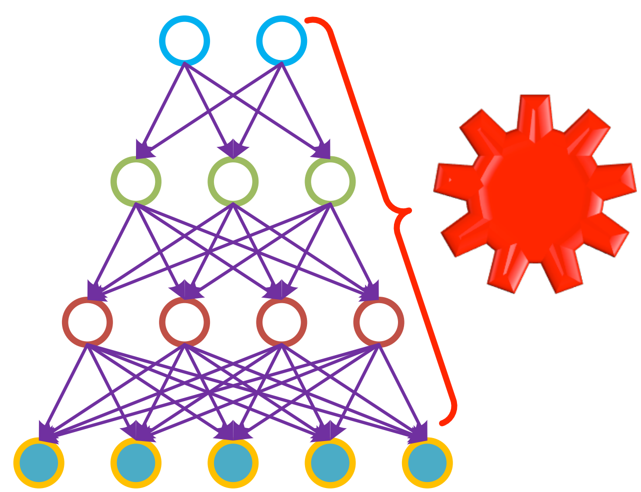
Most existing methods

- Scalable **but**
- Greedy layer-wise training
  - No communication between layers
- Shared SGD parameters

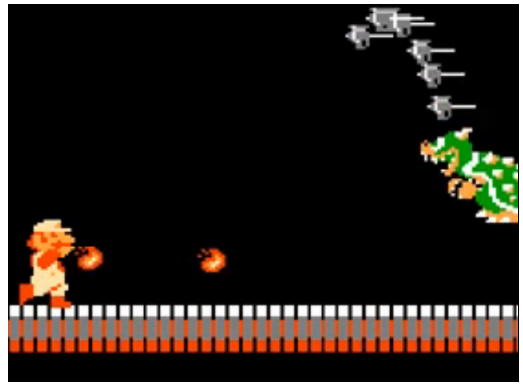


What we want

- Scalable **and**
- Principled joint learning
  - Communication & feedback between layers
- Adaptive SGD parameters



Outline

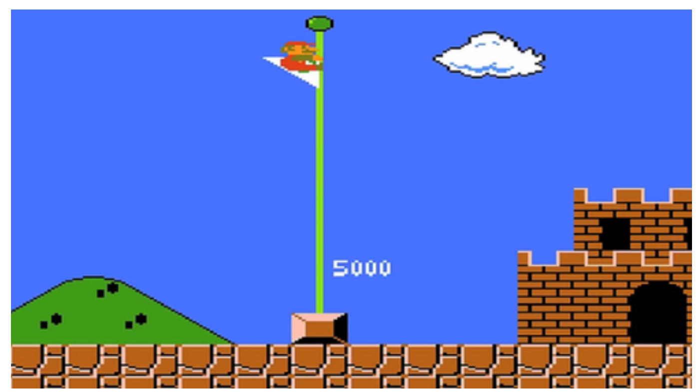


1.Motivation

2. Barriers

3.Our Contribution

4.Experiments

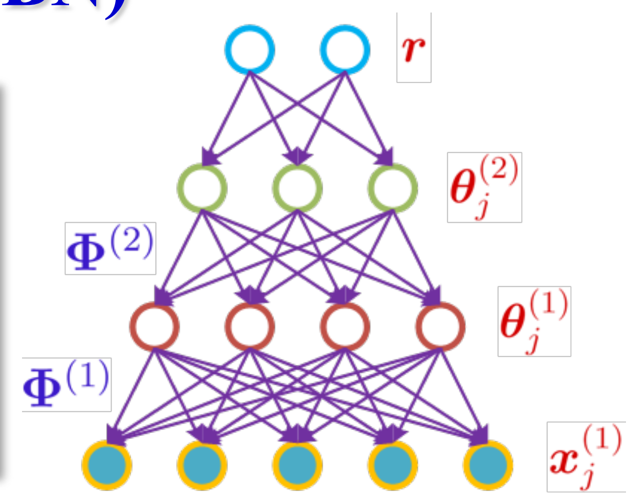


Our Contribution

We develop a principled joint SG-MCMC for a special deep LVM, namely the Poisson gamma belief network (PGBN).

❑ Poisson Gamma Belief Network (PGBN)

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



Priors: $\boxed{\phi_k^{(l)} \sim \mathrm{Dir}(\eta^{(l)}\mathbf{1}_{K_{l-1}})}$ $\quad \boldsymbol{r} \sim \mathrm{Gam}(\gamma_0/K_L, 1/c_0)$

$p_j^{(2)} \sim \mathrm{Beta}(a_0, b_0) \quad c_j^{(l)} \sim \mathrm{Gam}(e_0, 1/f_0)$

Gibbs

Not scalable

Preliminary for SG-MCMC

For the interested **batch posterior** $p(\boldsymbol{z}\,|X) \propto e^{-H(\boldsymbol{z})}$, one may get **posterior samples** using the **mini-batch** update rule as

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

$$\nabla \tilde{H}(\boldsymbol{z}) = \nabla \left[-\ln p(\boldsymbol{z}) - \rho \sum\nolimits_{\boldsymbol{x} \in \tilde{X}} \ln p(\boldsymbol{x}\,|\boldsymbol{z})\right]$$

$\mathbf{D}(\boldsymbol{z})$: positive semi-definite; $\mathbf{Q}(\boldsymbol{z})$: skew-symmetric;

$$\Gamma_i(\boldsymbol{z}) = \sum\nolimits_j \frac{\partial}{\partial z_j}\left[\mathbf{D}_{ij}(\boldsymbol{z}) + \mathbf{Q}_{ij}(\boldsymbol{z})\right]$$

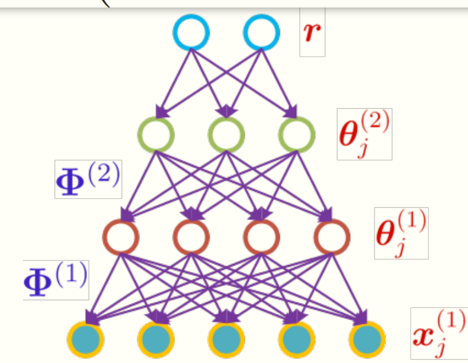
$\hat{\mathbf{B}}_t$: SG noise variance estimate.

SG-MCMC for PGBN

SG-MCMC & PGBN

- $\boldsymbol{z} = \left\{\boldsymbol{\Phi}^{(1)}, \cdots, \boldsymbol{\Phi}^{(L)}, \boldsymbol{r}\right\}$

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



For simplicity, we use

- $\mathbf{Q}(\boldsymbol{z}) = \mathbf{0}, \hat{\mathbf{B}}_t = \mathbf{0}$
- $\mathbf{D}(\boldsymbol{z}) = \mathbf{G}(\boldsymbol{z})^{-1}$
  - $\mathbf{G}(\boldsymbol{z})$: Fisher information matrix (FIM)

$$\boldsymbol{z}_{t+1} = \boldsymbol{z}_t + \varepsilon_t \left\{ -\mathbf{G}(\boldsymbol{z}_t)^{-1} \nabla \tilde{H}(\boldsymbol{z}_t) + \Gamma(\boldsymbol{z}_t) \right\} + \mathcal{N}\left(\mathbf{0}, 2\varepsilon_t \mathbf{G}(\boldsymbol{z}_t)^{-1}\right)$$

We need

- $\mathbf{G}(\boldsymbol{z})^{-1}$
- $\nabla \tilde{H}(\boldsymbol{z})$

However

- $\mathbf{G}(\boldsymbol{z})$ of PGBN is intractable
- $\mathbf{G}(\boldsymbol{z})^{-1}$ may be expensive

Calculate FIM

What we find: An alternative representation of the PGBN makes it straightforward to calculate $\mathbf{G}(\boldsymbol{z})$.

Deep Latent Dirichlet Allocation (DLDA)

Techniques:

- Data augmentation
- Marginalization

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

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

Calculate FIM

The alternative DLDA representation ➔

A *block-diagonal* and thus *easily inverted* FIM as

$$\mathbf{G}(\boldsymbol{z}) = \mathrm{diag}\left[\mathbf{I}\left(\phi_1^{(1)}\right), \cdots, \mathbf{I}\left(\phi_{K_L}^{(L)}\right), \mathbf{I}(\boldsymbol{r})\right]$$

Properties

- **Analytical & Practical**
- **Similar to the Hessian matrix in optimization**
- **Principled joint inference of the PGBN (DLDA)**
- **A separate puzzle for each $\phi_k^{(l)}$ and $\boldsymbol{r}$**

$$\boldsymbol{z}_{t+1} = \boldsymbol{z}_t + \varepsilon_t \left\{-\mathbf{G}(\boldsymbol{z}_t)^{-1}\nabla\tilde{H}(\boldsymbol{z}_t) + \Gamma(\boldsymbol{z}_t)\right\} + \mathcal{N}\left(\mathbf{0}, 2\varepsilon_t\mathbf{G}(\boldsymbol{z}_t)^{-1}\right)$$

Efficient Inference for Topics $\phi_k^{(l)}$

Reduced-Mean Inference on the Probability Simplex

For the batch posterior $(\phi_k|-) \sim \mathrm{Dir}(x_{1k\cdot}+\eta, \ldots, x_{Vk\cdot}+\eta)$ on the probability simplex, we derive a *mini-batch* update rule as

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

where $\tilde{x}_{vk\cdot}$ comes from mini-batches.

Techniques

- **Reduced-mean parameterization φ_k of ϕ_k, namely**

$$\phi_k = \left((\varphi_k)^T, 1 - \sum\nolimits_v \varphi_{vk}\right)^T$$

- **Fast simulation method in [1]**

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

The Whole Picture

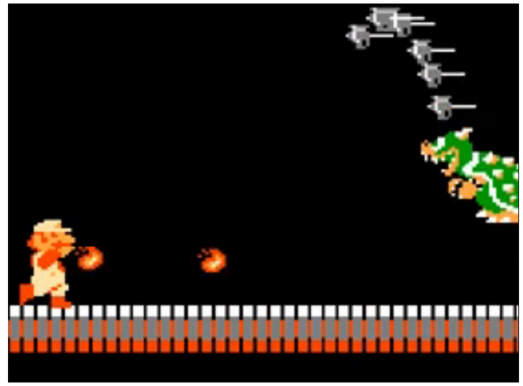
- **Topic-Layer-Adaptive Stochastic Gradient Riemannian (TLASGR) MCMC**

Algorithm 1 TLASGR MCMC for DLDA (PGBN).

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

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

Outline

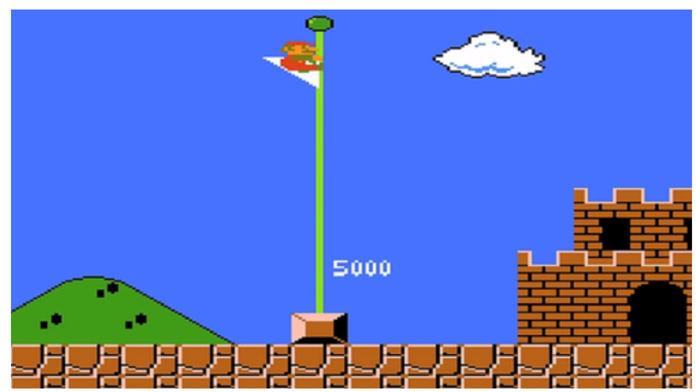


1.Motivation

3.Our Contribution

2. Barriers

4.Experiments



Experiments

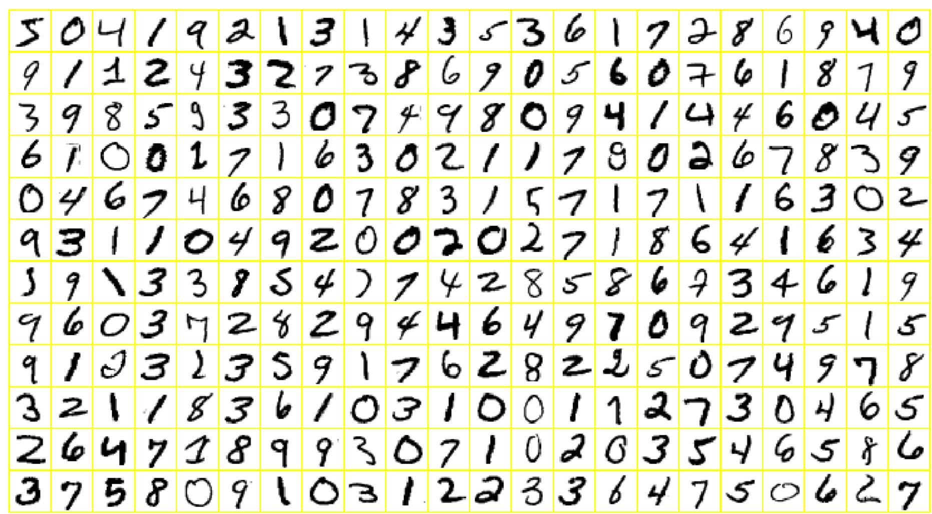
Three benchmark corpora

- ▶20Newsgroups (20News)
- ▶Reuters Corpus Volume I (RCV1)
- ▶Wikipedia (Wiki)

	20News	RCV1	**Wiki**
Vocabulary	2,000	10,000	**7,702**
Training	11,315	794,414	**≈10M**
Test	7,531	10,000	**1,000**

MNIST digits

- ▶**For illustration**
- ▶Size: 28×28
- ▶Training: 60000
- ▶Test: 10000



Performance: Perplexity

- ▶ Smaller is better
- ▶ Bold: Top two

DLDA & Gibbs

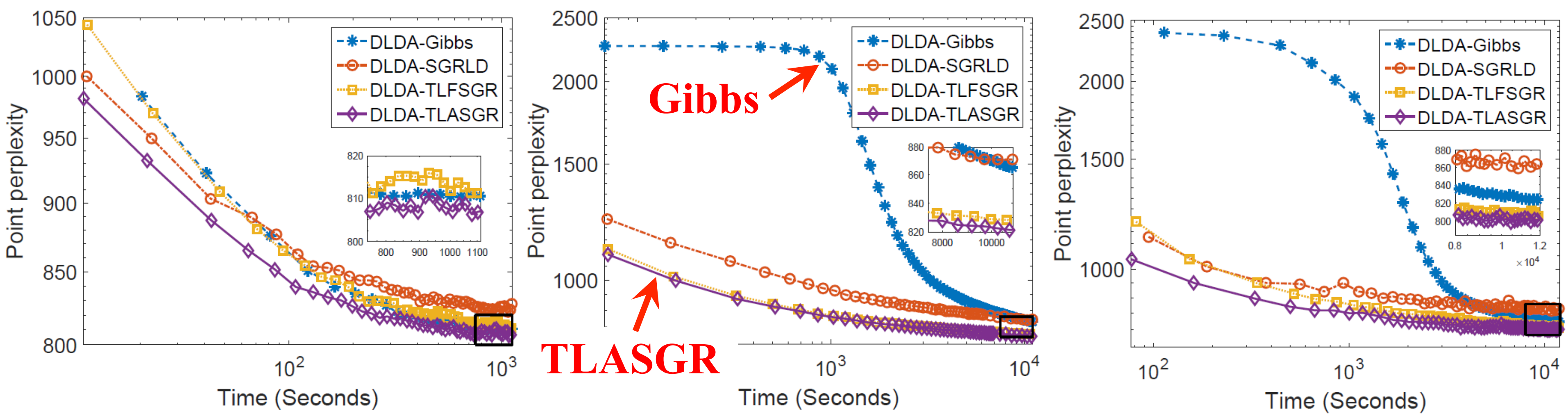
- ▶ Best results
- ▶ Not scalable

DLDA & TLASGR

- ▶ Second best
  - Comparable to Gibbs
- ▶ Scalable

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

Performance: Scalability



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

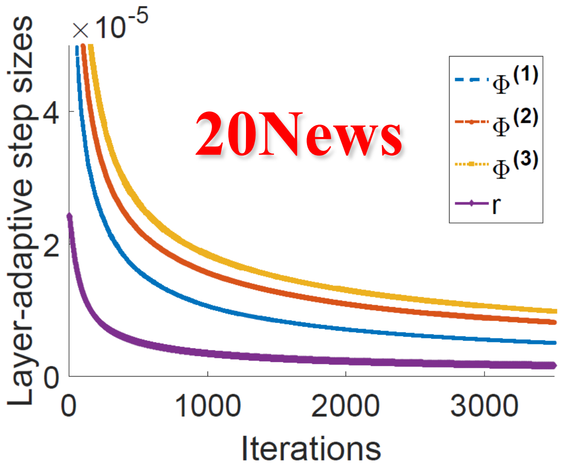
20News Train: **11K**

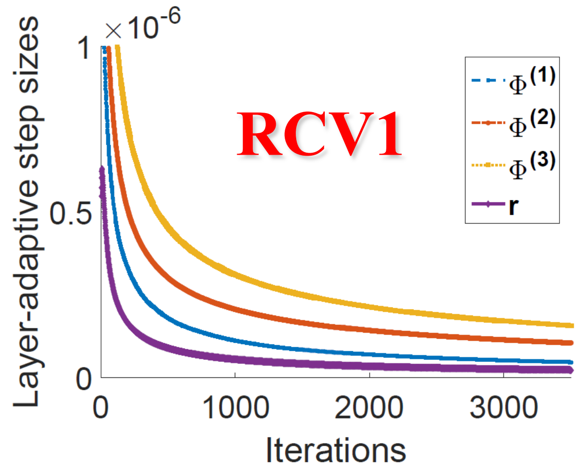
RCV1 Train: **0.8M**

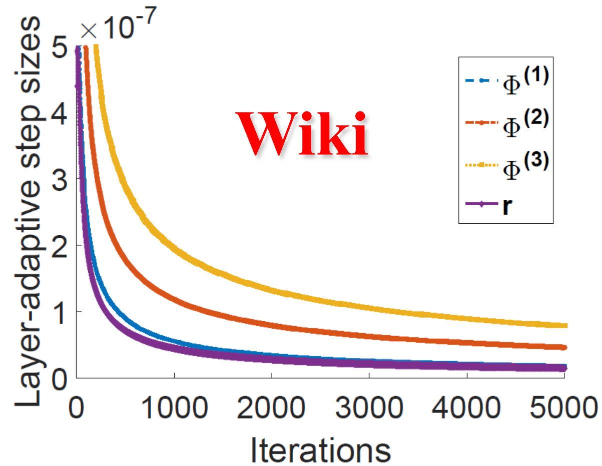
Wiki Train: **10M**

Benefits of Joint Learning

▶Topic-layer-adaptive learning rates





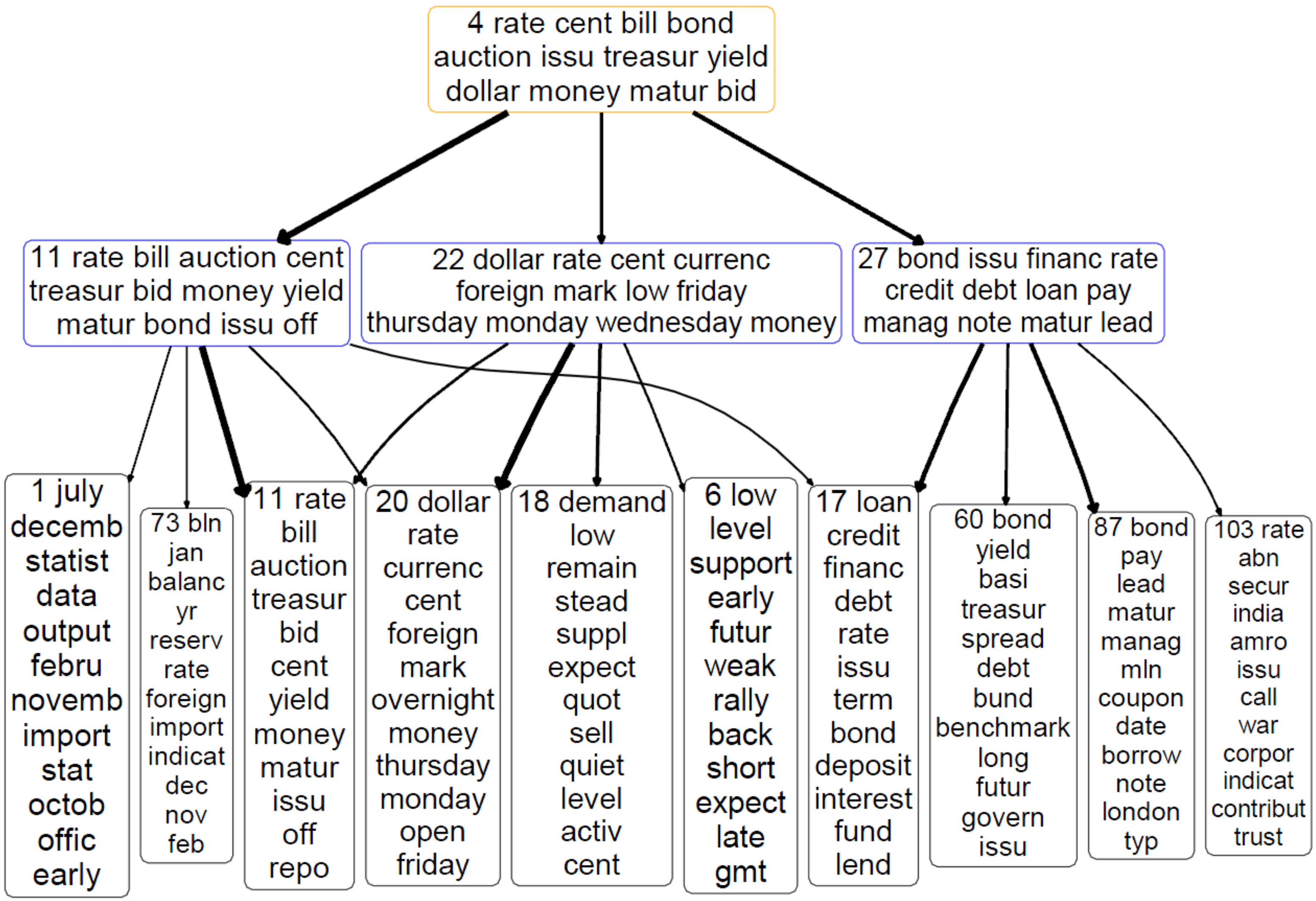


▶Better information propagation

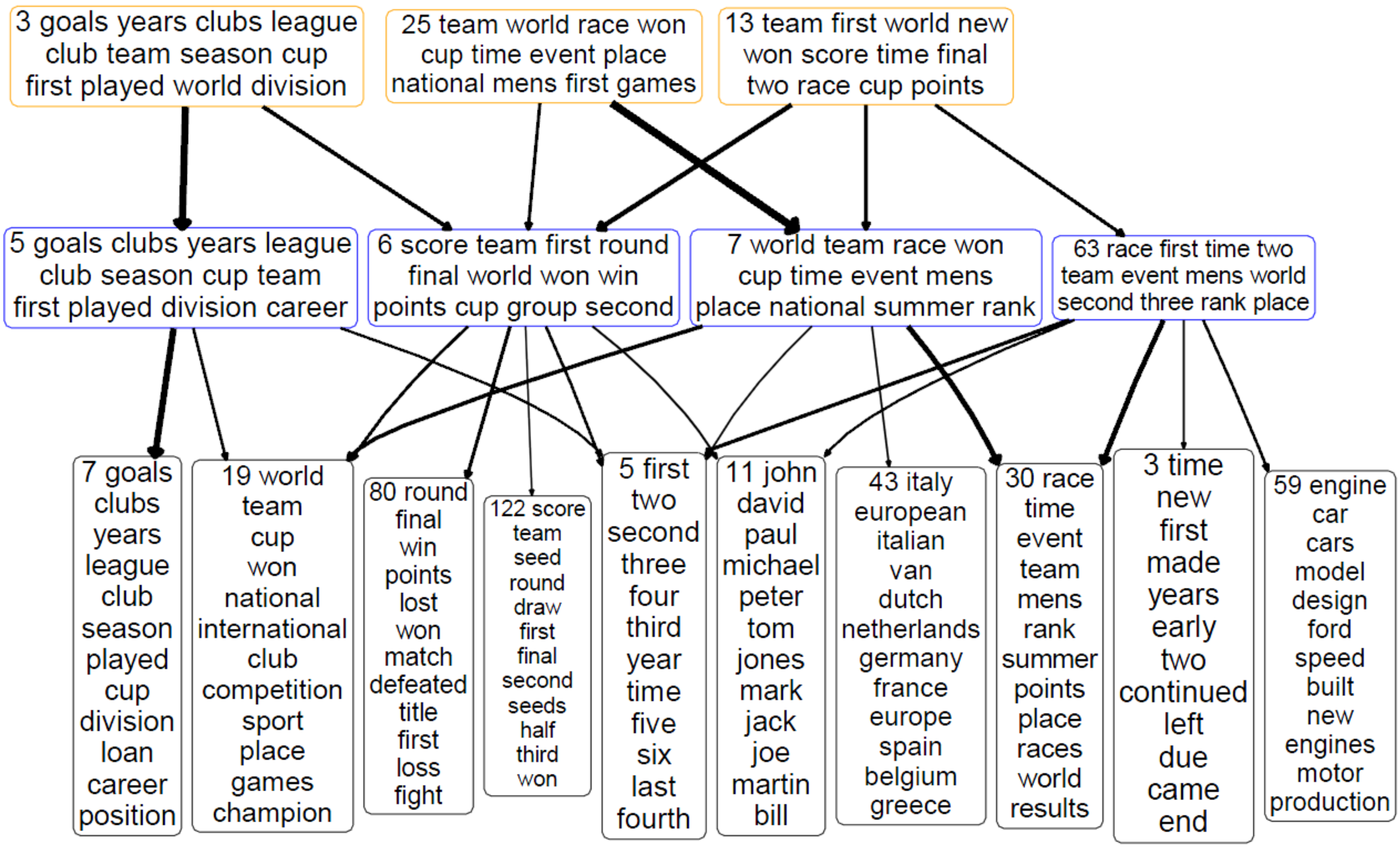
Topic-layer-FIXED learning rates

Topic-layer-ADAPTIVE learning rates

Example Topics: RCV1



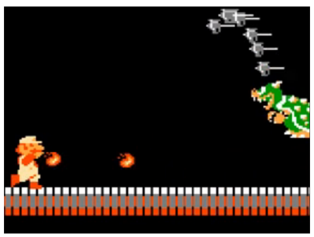
Example Topics: Wiki



Conclusions

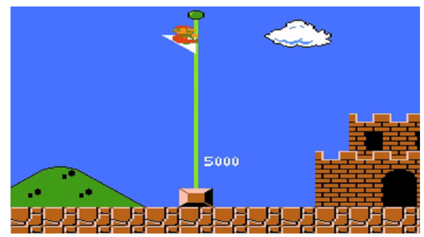
Scalable inference of deep LVMs for big data

- Gradients are difficult to calculate
- Joint learning of all layers of deep models



For the PGBN, we develop the principled joint TLASGR via newly introduced

- An alternative representation DLDA
- Analytical and block-diagonal FIM
- Reduced-mean parameterization on probability simplex



Experimentally, TLASGR

- Performance comparable to Gibbs
- Scalable to huge data
- Principle joint learning
  - Topic layer adaptive learning rates
  - Better information propagation
- Interpretable topics

THANK YOU!