# Scalable Inference for Logistic-Normal Topic Models

JIANFEI CHEN, JUN ZHU, ZI WANG, XUN ZHENG, BO ZHANG
chenjf10@mails.tsinghua.edu.cn, dcszj@mail.tsinghua.edu.cn, wangzi10@mails.tsinghua.edu.cn,
xunzheng@cs.cmu.edu, dcszb@mail.tsinghua.edu.cn

## Introduction

**Problem:** Discovering latent semantic structures from data efficiently, particularly **inference of non-conjugate logistic-normal topic models**.

**Topic Model Background:**

- Latent Dirichlet Allocation (LDA)
- Logistic-normal Topic Models (including CTM, DTM, infinite CTM)

**Existing methods for logistic-normal topic models:**

- variational inference, based on mean-field assumption
- **limitations:** subject to high computational cost, inaccuracy and inapplicability to real world for large scale of data

**Contribution:**

- A partially collapsed gibbs sampling algorithm;
- Efficient parallel implementation.

## Logistic-Normal Topic Models

**CTM Generating Process[1]:**

1. draw vector $\boldsymbol{\eta}_d \sim \mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\Sigma})$ and compute the topic mixing proportion $\boldsymbol{\theta}_d$: $\theta_d^k = \frac{e^{\eta_d^k}}{\sum_{j=1}^K e^{\eta_d^j}}$
2. for each word $n$ $(1 \le n \le N_d)$:
   (a) draw topic $z_{dn} \sim \text{Mult}(\boldsymbol{\theta}_d)$
   (b) draw word $w_{dn} \sim \text{Mult}(\boldsymbol{\Phi}_{z_{dn}})$, $\boldsymbol{\Phi}_k \sim \text{Dir}(\boldsymbol{\beta})$[a]

Non-conjugacy between the normal prior and the logistic transformation function makes it hard for CTM to infer the posterior distribution $p(\boldsymbol{\eta}, \mathbf{Z}, \boldsymbol{\Phi}|\mathbf{W}) \propto p_0(\boldsymbol{\eta}, \mathbf{Z}, \boldsymbol{\Phi}) p(\mathbf{W}|\mathbf{Z}, \boldsymbol{\Phi})$. Existing variational approximate methods is subject to strict factorization assumptions.

[a] $\text{Mult}(\cdot)$ denotes the multinomial distribution; $\text{Dir}(\cdot)$ is a Dirichlet distribution

## Conclusion

- We present a scalable Gibbs sampling algorithm for logistic-normal topic models.
- Expeirmental results show significant improvement in time efficiency over existing variational methods, with slightly better perplexity.
- The algorithm enjoys excellent scalability, suggesting the ability to extract large structures from massive data.

## Selected References

[1] D. Blei and J. Lafferty. Correlated topic models. In *Advances in Neural Information Processing Systems (NIPS)*, 2006.

[2] N. G. Polson, J. G. Scott, and J. Windle. Bayesian inference for logistic models using Polya-Gamma latent variables. *arXiv:1205.0310v2*, 2013.

[3] A. Ahmed, M. Aly, J. Gonzalez, S. Narayanamurthy, and A. Smola. Scalable inference in latent variable models. In *International Conference on Web Search and Data Mining (WSDM)*, 2012.

## Gibbs Sampling with Data Augmentation

Integrate out $\boldsymbol{\Phi}$ and perform Gibbs sampling for the marginalized distribution:

$$p(\boldsymbol{\eta}, \mathbf{Z}|\mathbf{W}) \propto p(\mathbf{W}|\mathbf{Z}) \prod_{d=1}^D \Big( \prod_{n=1}^{N_d} \frac{e^{\eta_d^{z_{dn}}}}{\sum_{j=1}^K e^{\eta_d^j}} \Big) \mathcal{N}(\boldsymbol{\eta}_d|\boldsymbol{\mu}, \boldsymbol{\Sigma}) \propto \prod_{k=1}^K \frac{\delta(\mathbf{C}_k + \boldsymbol{\beta})}{\delta(\boldsymbol{\beta})} \prod_{d=1}^D \Big( \prod_{n=1}^{N_d} \frac{e^{\eta_d^{z_{dn}}}}{\sum_{j=1}^K e^{\eta_d^j}} \Big) \mathcal{N}(\boldsymbol{\eta}_d|\boldsymbol{\mu}, \boldsymbol{\Sigma})^a$$

- Sampling topic assignments

$$p(z_{dn}^k = 1|\mathbf{Z}_{\neg n}, w_{dn}, \mathbf{W}_{\neg dn}, \boldsymbol{\eta}) \propto p(w_{dn}|z_{dn}^k = 1, \mathbf{Z}_{\neg n}, \mathbf{W}_{\neg dn}) e^{\eta_d^k} \propto \frac{C_{k,\neg n}^{w_{dn}} + \beta_{w_{dn}}}{\sum_{j=1}^V C_{k,\neg n}^j + \sum_{j=1}^V \beta_j} e^{\eta_d^k}$$

- Sampling Logistic-Normal parameters with data augmentation technique[b]
  - **For** $\eta_d^k$: $p(\eta_d^k|\boldsymbol{\eta}_d^{\neg k}, \mathbf{Z}, \mathbf{W}, \lambda_d^k) \propto \exp\big(\kappa_d^k \eta_d^k - \frac{\lambda_d^k (\eta_d^k)^2}{2}\big) \mathcal{N}(\eta_d^k|\mu, \sigma^2) = \mathcal{N}(\gamma_d^k, (\tau_d^k)^2)$, given posterior mean and variance we can easily draw a sample from a univariate Gaussian distribution.
  - **For** $\lambda_d^k$: $p(\lambda_d^k|\mathbf{Z}, \mathbf{W}, \boldsymbol{\eta}) \propto \exp\big(-\frac{\lambda_d^k (\eta_d^k)^2}{2}\big) p(\lambda_d^k|N_d, 0) = \mathcal{PG}\big(\lambda_d^k; N_d, \eta_d^k\big)$, we draw the samples through drawing $N_d$ samples from $\mathcal{PG}(1, \eta_d^k)$
- Fully-Bayesian models

$$\boldsymbol{\Sigma}|\kappa, W \sim \mathcal{IW}(\boldsymbol{\Sigma}; \kappa, W^{-1}),\ \boldsymbol{\mu}|\boldsymbol{\Sigma}, \boldsymbol{\mu}_0, \rho \sim \mathcal{N}(\boldsymbol{\mu}; \boldsymbol{\mu}_0, \boldsymbol{\Sigma}/\rho)$$

$$p(\boldsymbol{\mu}, \boldsymbol{\Sigma}|\boldsymbol{\eta}, \mathbf{Z}, \mathbf{W}) \propto p_0(\boldsymbol{\mu}, \boldsymbol{\Sigma}) \prod_d p(\boldsymbol{\eta}_d|\boldsymbol{\mu}, \boldsymbol{\Sigma}) = \mathcal{NIW}(\boldsymbol{\mu}_0', \rho', \kappa', W')$$

[a] $C_k^t$ is the number of times topic $k$ being assigned to the term $t$ over the whole corpus.
[b] With $\kappa_d^k = C_d^k - N_d/2$ and $p(\lambda_d^k|N_d, 0)$ the Polya-Gamma distribution[2] $\mathcal{PG}(N_d, 0)$, likelihood is

$$\ell(\eta_d^k|\boldsymbol{\eta}_d^{\neg k}) = \prod_{n=1}^{N_d} \Big(\frac{e^{\rho_d^k}}{1+e^{\rho_d^k}}\Big)^{z_{dn}^k} \Big(\frac{1}{1+e^{\rho_d^k}}\Big)^{1-z_{dn}^k} = \frac{(e^{\rho_d^k})^{C_d^k}}{(1+e^{\rho_d^k})^{N_d}} = \frac{1}{2^{N_d}} e^{\kappa_d^k \rho_d^k} \int_0^\infty e^{-\frac{\lambda_d^k (\rho_d^k)^2}{2}} p(\lambda_d^k|N_d, 0) d\lambda_d^k,$$

## Parallel Implementation and Fast Approximate Sampling

- Build upon state-of-the-art distributed sampler [3]
  - No communication is needed infering $\boldsymbol{\eta}_d$ and $\boldsymbol{\lambda}_d$.
  - Broadcasting the global variables $\boldsymbol{\mu}$ and $\boldsymbol{\Sigma}$ to every machine after each iteration.
- We propose a fast approximate sampling method to draw $\mathcal{PG}(n, \rho)$ samples, reducing the time complexity from $O(n)$ to $O(1)$.

## Experiments

**Data Sets**: Experiments are conducted on several benchmark data sets, including NIPS paper abstracts, 20Newsgroups, and NYTimes (New York Times) corpora and the Wikipedia corpus. All the data sets are randomly split into training and testing sets.

**Measurement**: Perplexity defined as $Perp(\mathcal{M}) = (\prod_{d=1}^D \prod_{i=L+1}^{N_d} p(w_{di}|\mathcal{M} w_{d,1:L}))^{\frac{-1}{\sum_{d=1}^D (N_d - L)}}$ where $\mathcal{M}$ denotes the model parameters, $w_{d,1:L}$ is the observed part of document $d$.

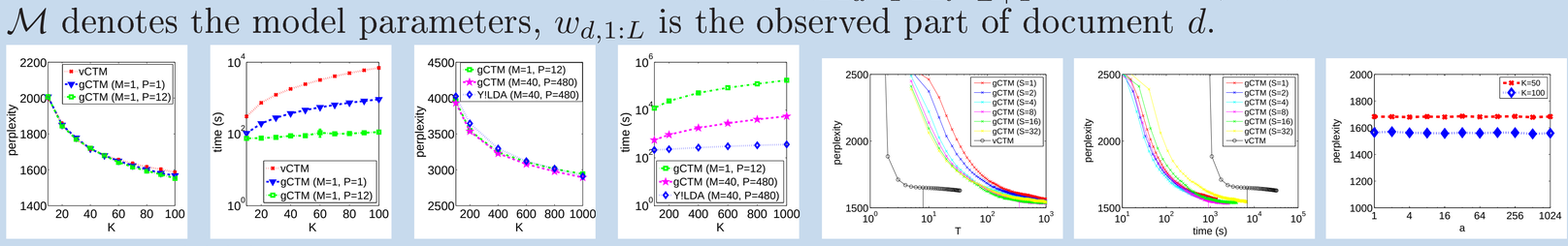


**Figure 1:** (a)(b): Perplexity and training time of vCTM, single-core gCTM, and multi-core gCTM on the NIPS data set; (c)(d): Perplexity and training time of single-machine gCTM, multi-machine gCTM, and multi-machine Y!LDA on the NYTimes data set. Sensitivity analysis with respect to key hyper-parameters: (e) perplexity at each iteration with different $S$; (f) convergence speed with different $S$; (g) perplexity tested with different prior.

The **efficiency** of vCTM and gCTM on different sized data sets was compared in Table 1. It can be observed that

- vCTM immediately becomes impractical when the data size reaches 285K.
- gCTM is still able to process larger data sets with considerable speed.

**Table 1:** Training time of vCTM and gCTM ($M = 40$) on various datasets.

| data set | $D$ | $K$ | vCTM | gCTM |
|---|---|---|---|---|
| NIPS | 1.2K | 100 | 1.9 hr | 8.9 min |
| 20NG | 11K | 200 | 16 hr | 9 min |
| NYTimes | 285K | 400 | N/A* | 0.5 hr |
| Wiki | 6M | 1000 | N/A* | 17 hr |

*not finished within 1 week.

**Scalability analysis**

- $M = 8, 16, 24, 32, 40$ each machine processes 150K documents.
- Wikipedia data set with $K = 500$, as a practical problem.
- As we pour in the same proportion of data and machines, the training time is almost kept constant. Parallel gCTM enjoys nice scalability.