POSTER: ELDA: LDA Made Efficient via Algorithm-System Codesign Submission

Shilong Wang
University of Massachusetts Lowell
shilong_wang@student.uml.edu

Da Li
Openmining Inc
dali@openmining.io

Hengyong Yu
University of Massachusetts Lowell
Hengyong_yu@uml.edu

Hang Liu*
Stevens Institute of Technology
hliu77@stevens.edu

Abstract

Latent Dirichlet Allocation (LDA) is a statistical approach for topic modeling with a wide range of applications. In spite of the significance, we observe very few attempts from *system* track to improve LDA, let alone the algorithm and system codesigned efforts. To this end, we propose eLDA with an algorithm-system codesigned optimization. Particularly, we introduce a novel three-branch sampling mechanism to taking advantage of the convergence heterogeneity of various tokens in order to reduce redundant sampling task. Our evaluation shows that eLDA outperforms the state-of-the-arts.

ACM Reference Format:
Shilong Wang, Da Li, Hengyong Yu, and Hang Liu. 2020. POSTER: ELDA: LDA Made Efficient via Algorithm-System Codesign Submission. In *25th ACM SIGPLAN Symposium on Principles and Practice of Parallel Programming (PPoPP '20), February 22–26, 2020, San Diego, CA, USA.* ACM, New York, NY, USA, 2 pages. https://doi.org/10.1145/3332466.3374517

1 Introduction

Topic modeling is a type of statistical approach that reveals the *latent* (i.e., unobserved) topics for a collection of documents (also referred to as corpus). LDA [1], which *carefully* chooses the Dirichlet distribution as the statistical model to formulate topic distributions, is one of the most popular topic modeling approach that finds applications in not only text analysis, but also computer vision [2], recommendation system[6] and network analysis [3] among many others. While LDA is widely studied in machine learning and algorithm community, very few researches have been done from *system* aspect, in part, due to the mathematical modeling complexity in the original work [1]. Let alone the algorithm-system codesigned efforts.

Related literature of LDA falls in the algorithm and system directions.

First, on the algorithm track, there exist many algorithms for fast LDA training. We summarize them into three basic categories: Variation Inference [1], Expectation Maximization [8] and Markov Chain Monte-Carlo [5]. In this paper, we make some algorithm improvements based on the Exponential Stochastic Cellular Automata (ESCA) method [11], which is an algorithm extended from the Expectation Maximization method. Compared with other LDA algorithms such as Collapsed Variational Bayes and Expectation Propagation, ESCA yields simpler expression, better parallelism and potentially less computations, in part, due to the sparsity aware design. Consequently, our work extends this direction.

Second, for GPU-based LDA, which is the interest of this work, we witness much fewer efforts. To the best of our knowledge, there only exists three such projects. Yan et al. [10] implement Collapsed Gibbs Sampling and collapsed Variational Bayesian on GPU. Afterwards, SaberLDA [7] advocates to store document-topic matrix in sparse format and introduces index tree for fast sampling. Note, although both document-topic and word-topic matrix are sparse, SaberLDA fails to overcome the challenges of storing both data structures in sparsity aware format. Further, CuLDA_CGS [9] scales LDA to multiple GPUs based on collapsed Gibbs sampling with similar optimizations as SaberLDA on each GPU. However, these methods can only support at most 10,000 topics because they have to store word-topic matrix in dense format - larger topics will exhaust the limited memory space of GPUs.

Contribution. This paper introduces eLDA, an algorithm-system codesgined GPU-based LDA project that can train LDA on PubMed dataset within 3,000 seconds while supporting the unprecedented 32,768 topics on one Nvidia Titan Xp GPU. Particularly, we introduce the three-branch sampling method which takes the advantage of the convergence

*Corresponding author

PPoPP '20, February 22–26, 2020, San Diego, CA, USA

ACM ISBN 978-1-4503-6818-6/20/02.
https://doi.org/10.1145/3332466.3374517

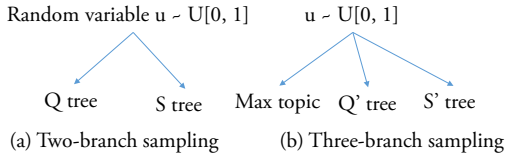

Figure 1. Two-branch vs three-branch sampling.

heterogeneity of various tokens to reduce the redundant sampling task. While the convergence heterogeneity is promising, the caveat is that one cannot simply avoid sampling a token because its topic remains unchanged for a consecutive number of iterations.

Inspired by our key observation that majority of the tokens often fall in the most popular topic, we single out the most popular topic as the third sampling branch in addition to the traditional two branches (detailed in Figure 1). During sampling, we introduce an algorithm to accurately estimate whether this token will remain in the most popular topic, thus avoid expensive sampling. Our evaluation shows three-branch sampling can avoid sampling 85% of the tokens in PubMed dataset.

2 Experiments

We implement ɛLDA with ~4,000 lines of CUDA code and compile the source code with Nvidia CUDA 9.2 toolkit and -O3 optimization compilation flag. We use the Nvidia Titan Xp GPU, which runs on an Alienware with 24 GB memory and Intel(R) Core(TM) i7-8700 (3.20Hz) CPU to study the performance of ɛLDA. We evaluates ɛLDA with two popular datasets:

- `NYTimes` [4] : 299,752 documents, 101,636 unique words and 100M tokens.
- `PubMed` [4] : 8,200,000 documents, 141,043 unique words and 738M tokens.

ɛLDA vs. State-of-the-art Figure 2 further shows that ɛLDA climbs to higher perplexity with less training time. Note, at initial iterations, ɛLDA falls behind because the three-branch sampling takes overhead but yield very few benefits, given majority of the tokens have not converged at initial iterations.

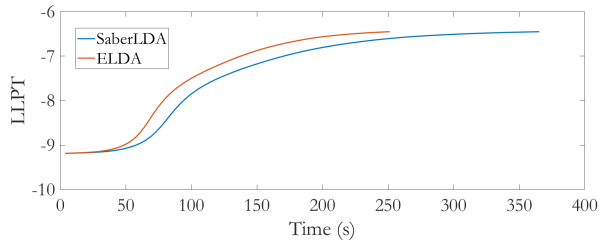

Figure 2. The convergence of ɛLDA and SaberLDA with 1,000 topics. Higher is better.

Three-branch sampling. Figure 3 shows the performance of the three-branch sampling. We can see large percentage of tokens are skipped by applying three-branch sampling and this trend will be enhanced with iterations.

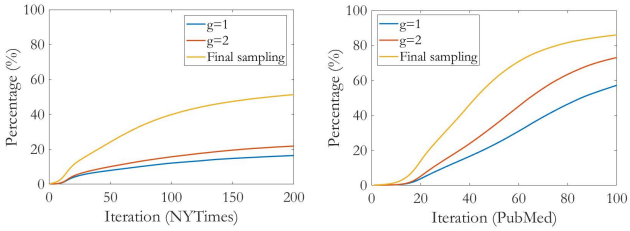

Figure 3. The percentage of tokens skipped by three-branch sampling for #topics = 1,000

3 Conclusion

In this paper, we present ɛLDA, an efficient LDA project with algorithm and system codesigned optimizations. Particularly, we introduce the novel three-branch sampling for LDA that yields superior performance over the state of the art projects.

Acknowledgment

We thank the anonymous reviewers for their constructive suggestions. We also would like to gracefully acknowledge the support from XSEDE supercomputers and Amazon AWS, as well as the NVIDIA Corporation for the donation of the Titan Xp and Quadro P6000 GPUs. This work was in part supported by NSF CRII Award No. 2000722.

References

[1] David M Blei and Others. 2003. Latent dirichlet allocation. *JMLR* 3, Jan (2003), 993–1022.
[2] Liangliang Cao and Fei Fei Li. 2007. Spatially Coherent Latent Topic Model for Concurrent Segmentation and Classification of Objects and Scenes. In *ICCV*. IEEE, Rio de Janeiro, Brazil, 1–8.
[3] Jonathan Chang and David Blei. 2009. Relational Topic Models for Document Networks. In *AISTATS (PMLR)*, Vol. 5. PMLR, Florida USA, 81–88.
[4] Dheeru Dua and Casey Graff. 2017. UCI Machine Learning Repository. http://archive.ics.uci.edu/ml
[5] Thomas L Griffiths and Mark Steyvers. 2004. Finding scientific topics. *PNAS* 101, suppl 1 (2004), 5228–5235.
[6] Ralf Krestel and Others. 2009. Latent dirichlet allocation for tag recommendation. In *RECSYS*. ACM, New York, USA, 61–68.
[7] Kaiwei Li, Jianfei Chen, Wenguang Chen, and Jun Zhu. 2017. Saberlda: Sparsity-aware learning of topic models on gpus. *ACM SIGOPS Operating Systems Review* 51, 2 (2017), 497–509.
[8] Thomas Minka and John Lafferty. 2002. Expectation-Propagation for the Generative Aspect Model *(UAI'02)*. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 352–359.
[9] Xiaolong Xie and Others. 2019. CuLDA: Solving Large-scale LDA Problems on GPUs. In *HPDC*. ACM, Phoenix, USA, 195–205.
[10] Feng Yan and Others. 2009. Parallel inference for latent dirichlet allocation on graphics processing units. In *NIPS*. Curran Associates, Inc., Vancouver, B.C., Canada, 2134–2142.
[11] Manzil Zaheer and Others. 2016. Exponential stochastic cellular automata for massively parallel inference. In *AISTATS*. JMLR, Cadiz, Spain, 966–975.