BigARTM: Open Source Library for Regularized Multimodal Topic Modeling of Large Collections

Konstantin Vorontsov[1], Oleksandr Frei[2], Murat Apishev[3], Peter Romov[4], and Marina Dudarenko[5]

[1] Yandex, Moscow Institute of Physics and Technology, voron@forecsys.ru
[2] Schlumberger Information Solutions, oleksandr.frei@gmail.com
[3] Lomonosov Moscow State University, great-mel@yandex.ru
[4] Yandex, Moscow Institute of Physics and Technology, peter@romov.ru
[5] Lomonosov Moscow State University, m.dudarenko@gmail.com

Abstract. Probabilistic topic modeling of text collections is a powerful tool for statistical text analysis. In this paper we announce the BigARTM open source project (http://bigartm.org), which provides the parallel online EM algorithm for learning additively regularized multimodal topic models of large collections. We show that BigARTM outperforms other popular packages in quality, runtime and multicriteria functionality.

Keywords: probabilistic topic modeling, Probabilistic Latent Sematic Analysis, Latent Dirichlet Allocation, Additive Regularization of Topic Models, stochastic matrix factorization, EM algorithm, BigARTM.

1 Introduction

Topic modeling is a rapidly developing branch of statistical text analysis [1]. Probabilistic topic model (PTM) reveals a hidden thematic structure of a text collection. It defines each topic by a discrete distribution over words, and then describes each document with a discrete distribution over topics. Practical applications of topic models include many areas, such as information retrieval for long-text queries, classification, categorization, summarization of texts. Modern literature on topic modeling offers hundreds of specialized models [4]. Nevertheless, most of these models are too difficult for practitioners to quickly understand, adapt and embed into applications. Most practical inconveniences are rooted in Bayesian inference, which requires a laborious mathematical work and prevents flexible unification, modification, selection, and combination of topic models.

In this paper we announce **the BigARTM open source project** for regularized multimodal topic modeling of large collections, http://bigartm.org. The theory behind BigARTM is based on a non-Bayesian multicriteria approach — *Additive Regularization of Topic Models*, ARTM [11]. Instead of building a purely probabilistic generative model of text we regularize an ill-posed problem of stochastic matrix factorization by maximizing a weighted sum of the log-likelihood and additional criteria. Many known Bayesian topic models were revisited in terms of ARTM in [12,13]. Compared to the Bayesian approach, ARTM makes it easier to design, infer and combine topic models, thus reducing the barrier for entering into topic modeling research field.

In section 2 we introduce a regularized multimodal topic model. In section 3 we describe the parallel architecture of the BigARTM library. In section 4 we show that BigARTM performs better than Vowpal Wabbit LDA and Gensim libraries in terms of perplexity and runtime on large Wikipedia corpus.

2 Multimodal regularized topic model

Let D denote a finite set (collection) of texts and W^1 denote a finite set (vocabulary) of all terms from these texts. A document can contain not only words or key phrases, but also terms of other modalities. Each modality is defined by a finite set (vocabulary) of terms W^m, $m = 1, \dots, M$. Examples of not-word modalities are: authors, class or category labels, date-time stamps, links, named entities, objects on images, users, advertising banners, etc.

Assume that each term occurrence in each document refers to some latent topic from a finite set of topics T. Text collection is considered to be a sample of triples (w_i, d_i, t_i), $i = 1, \dots, n$, drawn independently from a discrete distribution $p(w, d, t)$ over the finite space $W \times D \times T$, where $W = W^1 \sqcup \dots \sqcup W^m$ is a disjoint union of the vocabularies across all modalities. Terms w_i and documents d_i are observable variables, while topics t_i are latent variables.

Following the idea of Correspondence LDA [2] and Dependency LDA [9] we introduce a topic model for each modality:

$$p(w \,|\, d) = \sum_{t \in T} p(w \,|\, t)\, p(t \,|\, d) = \sum_{t \in T} \phi_{wt} \theta_{td}, \quad d \in D,\ w \in W^m,\ m = 1, \dots, M.$$

The parameters $\theta_{td} = p(t \,|\, d)$ and $\phi_{wt} = p(w \,|\, t)$ form matrices $\Theta = \big(\theta_{td}\big)_{T \times D}$ of *topic probabilities for the documents*, and $\Phi^m = \big(\phi_{wt}\big)_{W^m \times T}$ of *term probabilities for the topics.* The matrices Φ^m, if stacked vertically, form a $W \times T$-matrix Φ. Matrices Φ^m and Θ are *stochastic* with vector-columns representing discrete distributions. Usually $|T|$ is much smaller than $|D|$ and $|W|$.

To learn parameters Φ^m, Θ from the multimodal text collection we maximize the log-likelihood for each m-th modality:

$$\mathscr{L}_m(\Phi^m, \Theta) = \sum_{d \in D} \sum_{w \in W^m} n_{dw} \ln p(w \,|\, d) \to \max_{\Phi^m, \Theta},$$

where n_{dw} is the number of occurrences of the term $w \in W^m$ in the document d. Following the ARTM approach, we add a regularization penalty term $R(\Phi, \Theta)$ and solve a constrained multicriteria optimization problem via scalarization:

$$\sum_{m=1}^{M} \tau_m \mathscr{L}_m(\Phi^m, \Theta) + R(\Phi, \Theta) \to \max_{\Phi, \Theta}; \tag{1}$$

$$\sum_{w \in W^m} \phi_{wt} = 1,\ \phi_{wt} \geq 0; \qquad \sum_{t \in T} \theta_{td} = 1,\ \theta_{td} \geq 0. \tag{2}$$

The local maximum (Φ, Θ) of the problem (1), (2) satisfies the following system of equations with auxiliary variables $p_{tdw} = p(t \,|\, d, w)$:

$$p_{tdw} = \operatorname*{norm}_{t \in T} \bigl(\phi_{wt}\theta_{td}\bigr); \tag{3}$$

$$\phi_{wt} = \operatorname*{norm}_{w \in W^m} \left(n_{wt} + \phi_{wt}\frac{\partial R}{\partial \phi_{wt}}\right); \quad n_{wt} = \sum_{d \in D} \tau_{m(w)} n_{dw} p_{tdw}; \tag{4}$$

$$\theta_{td} = \operatorname*{norm}_{t \in T} \left(n_{td} + \theta_{td}\frac{\partial R}{\partial \theta_{td}}\right); \quad n_{td} = \sum_{w \in d} \tau_{m(w)} n_{dw} p_{tdw}; \tag{5}$$

where operator $\operatorname*{norm}_{t \in T} x_t = \frac{\max\{x_t, 0\}}{\sum\limits_{s \in T} \max\{x_s, 0\}}$ transforms a vector $(x_t)_{t \in T}$ to a discrete distribution; $m(w)$ is the modality of the term w, so that $w \in W^{m(w)}$.

The system of equations (3)–(5) follows from Karush–Kuhn–Tucker conditions. Solving it by the simple-iteration method is equivalent to the EM algorithm, which repeats E-step (3) and M-step (4)–(5) in a loop. For single modality ($M = 1$) it corresponds to the regularized EM algorithm [11]. With no regularization ($R = 0$) it corresponds to *Probabilistic Latent Semantic Analysis*, PLSA [6]. Many Bayesian topic models including *Latent Dirichlet Allocation*, LDA [3], can be considered as special cases of ARTM with different regularizers R [12,13].

The most important feature of ARTM is that it allows to combine regularizers additively: $R(\Phi, \Theta) = \sum_{i=1}^{r} \lambda_i R_i(\Phi, \Theta)$, which leads to an easy modification of the M-step. BigARTM provides a build-in user extendable library of regularizers.

3 BigARTM architecture

BigARTM implements a fast online EM algorithm similar to the Online LDA [5]. We split the collection into batches, and run EM iterations so that each document vector θ_d is iterated until convergence at a constant matrix Φ. Matrix Φ is updated after all documents from the batch are processed. For a large collection matrix Φ often stabilizes after small initial part of the collection. Therefore a single pass through the collection might be sufficient to learn a topic model.

BigARTM processes a large collection without loading it entirely into the memory. This is achieved by storing each batch in a separate file on disk, and loading a limited number of batches into the main memory at any given time.

To split collection into batches and process them concurrently is a common approach, introduced in AD-LDA algorithm [8], and then further developed in PLDA [15] and PLDA+ [7] algorithms. These algorithms require all concurrent workers to become idle before an update of the Φ matrix. Such synchronization step adds a large overhead in the online algorithm where Φ matrix is updated multiple times on each iteration. An alternative architecture without the synchronization step is described in [10], however it mostly targets a distributed cluster environment. In our work we develop an efficient single-node architecture where all workers benefit from the shared memory space.

The BigARTM out-of-core implementation avoids storing full Θ matrix in the memory by calculating θ_{td} on the fly during a batch processing. The inference of θ_{td} is executed in parallel in such a way that many batches are processed concurrently, however each batch is processed by simple single-threaded code. This gives almost linear speedup with the number of processors.

Processing each batch results in an n_{wt} increment that has to be applied to the Φ matrix. We merge n_{wt} in a background, and asynchronously build a new Φ matrix. This doubles the memory usage, but enables BigARTM to update Φ matrix without pausing all batch processing threads. All processor threads share the same Φ matrix, which means that memory usage stays at constant level regardless of how many cores are used for computation.

BigARTM uses dense single-precision matrices to represent Φ and Θ. Together with the Φ matrix we store a global dictionary of all terms $w \in W$. This dictionary can be extended automatically as more and more batches came through the system. To achieve this each batch contains a local dictionary, listing all terms that occur in the batch. The n_{dw} elements of the batch are stored as a sparse CSR matrix (Compressed Sparse Raw format), where rows correspond to documents from the batch, and terms run over the local batch dictionary.

The core of the library is written in C++ and is exposed via two equally rich APIs for C++ and Python (Java wrapper is planned in the near future). To input and output complex data structures the API uses Google Protocol Buffers. This approach makes it easy to integrate BigARTM into any research or production environment. The library is cross-platform and can be built for Linux, Windows and OS X in both 32 and 64 bit configuration.

BigARTM source code is released under the New BSD License, which permits free commercial and non-commercial usage.

4 Experiments

In the first experiment, we evaluate the BigARTM performance against two popular software packages — Gensim [14] and Vowpal Wabbit (VW)[6] on a collection of 3.7 million articles from the English Wikipedia[7]. Both use the Online Variational Bayes LDA [5]. VW.LDA is one of the fastest single-threaded implementation in C++. Gensim has two LDA implementations: LdaModel processes all batches sequentially, LdaMulticore processes several batches concurrently similar to BigARTM. The dictionary is formed by $|W| = 100\,000$ most frequent words. Each run performs one pass over the Wikipedia corpus and produces a model with $|T| = 100$ topics. Batch size is $10\,000$ documents. The perplexity measure is defined as $\mathcal{P}(D,p) = \exp\bigl(-\frac{1}{n}\sum_{d,w} n_{dw} \ln p(w \,|\, d)\bigr)$. Table 1 compares the performance results for an Intel-based CPU with 16 physical cores with hyper-threading. Fig. 1 shows BigARTM speedup and memory consumption depending on the number of CPU threads for Amazon AWS c3.8xlarge with 32 virtual cores.

[6] https://github.com/JohnLangford/vowpal_wabbit/

[7] http://dumps.wikimedia.org/enwiki/20141208/

Table 1. The comparison of BigARTM with VW.LDA and Gensim. *Train* is the time for model training, *inference* is the time for calculation of θ_d of 100 000 held-out documents, *perplexity* is calculated on held-out documents.

library	procs	train	inference	perplexity
BigARTM	1	35 min	72 sec	4000
LdaModel	1	369 min	395 sec	4161
VW.LDA	1	73 min	120 sec	4108
BigARTM	4	9 min	20 sec	4061
LdaMulticore	4	60 min	222 sec	4111
BigARTM	8	4.5 min	14 sec	4304
LdaMulticore	8	57 min	224 sec	4455

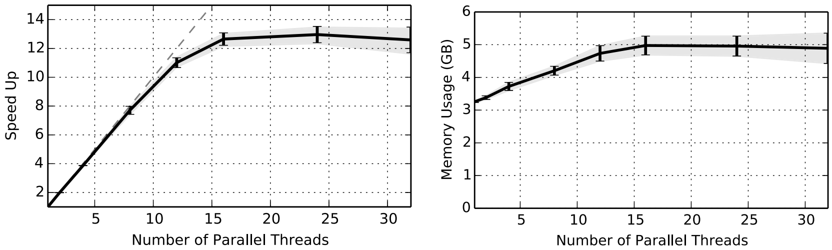

Fig. 1. Running BigARTM in parallel: speed up (left) and memory usage (right)

In the second experiment, we combine three regularizers from the BigARTM built-in library: ϕ_t sparsing, θ_d sparsing, and ϕ_t decorrelation. This combination has improved several quality measures without significant loss of perplexity in previous experiments with offline implementation of ARTM [13]. Table 2 shows that this remains true for the BigARTM implementation. Figure 2 shows the convergence of quality measures on the number of processed documents.

In the third experiment, we show how BigARTM works with multimodal datasets. We prepared a text corpus containing 216 175 pairs of English and Russian Wikipedia articles with mutual interwiki links. We represent each linked pair of articles as a single document with two modalities, one modality for each language. The combined dictionary contains $|W| = 196\,749$ words (43% Russian, 57% English). We build a model with $|T| = 400$ topics. An independent assessor successfully interpreted all except four topics. Table 3 shows top 10 words for four randomly selected topics. Top words in these topics are clearly consistent between Russian and English languages.

5 Conclusions

BigARTM in an open source project for parallel online topic modeling of large text collections. It provides a high flexibility for various applications due to multimodality and additive combinations of regularizers.

Table 2. Comparison of LDA and ARTM models. Quality measures: $\mathcal{P}_{10k}$, $\mathcal{P}_{100k}$ — perplexity on 10K and 100K hold-out document sets, $\mathcal{S}_{\Phi}$, $\mathcal{S}_{\Theta}$ — sparsity of Φ and Θ matrices (in %), $\mathcal{K}_p$, $\mathcal{K}_c$ — average topic purity and contrast respectively.

Model	$\mathcal{P}_{10k}$	$\mathcal{P}_{100k}$	$\mathcal{S}_{\Phi}$	$\mathcal{S}_{\Theta}$	$\mathcal{K}_p$	$\mathcal{K}_c$
LDA	3436	3801	0.0	0.0	0.533	0.507
ARTM	3577	3947	96.3	80.9	0.785	0.731

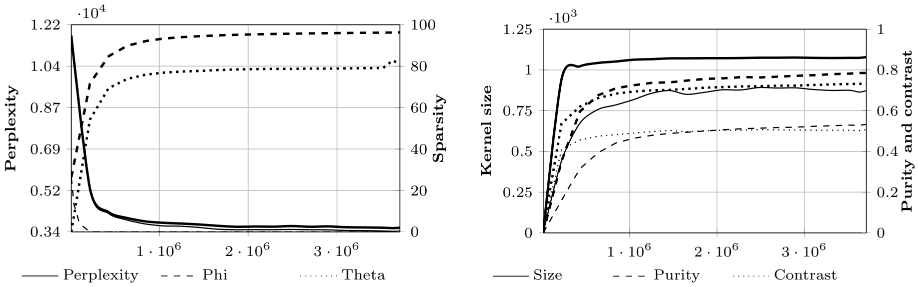

Fig. 2. Comparison of LDA (thin) and ARTM (bold) models. The number of processed documents is shown along the X axis. The left chart shows perplexity and sparsity of Φ, Θ matrices, and the right chart shows average lexical kernel measures.

Acknowledgements. The work was supported by the Russian Foundation for Basic Research grants 14-07-00847, 14-07-00908, 14-07-31176 and by Skolkovo Institute of Science and Technology (project 081-R).

References

1. Blei, D.M.: Probabilistic topic models. Communications of the ACM 55(4), 77–84 (2012)
2. Blei, D.M., Jordan, M.I.: Modeling annotated data. In: Proceedings of the 26th Annual International ACM SIGIR Conference on Research and Development in Informaion Retrieval. pp. 127–134. ACM, New York, NY, USA (2003)
3. Blei, D.M., Ng, A.Y., Jordan, M.I.: Latent Dirichlet allocation. Journal of Machine Learning Research 3, 993–1022 (2003)
4. Daud, A., Li, J., Zhou, L., Muhammad, F.: Knowledge discovery through directed probabilistic topic models: a survey. Frontiers of Computer Science in China 4(2), 280–301 (2010)
5. Hoffman, M.D., Blei, D.M., Bach, F.R.: Online learning for latent Dirichlet allocation. In: NIPS. pp. 856–864. Curran Associates, Inc. (2010)
6. Hofmann, T.: Probabilistic latent semantic indexing. In: Proceedings of the 22nd annual international ACM SIGIR Conference on Research and Development in Information Retrieval. pp. 50–57. ACM, New York, NY, USA (1999)
7. Liu, Z., Zhang, Y., Chang, E.Y., Sun, M.: PLDA+: parallel latent Dirichlet allocation with data placement and pipeline processing. ACM Trans. Intell. Syst. Technol. 2(3), 26:1–26:18 (May 2011)

Table 3. Top 10 words with $p(w \mid t)$ probabilities (in %) from two-language topic model, based on Russian and English Wikipedia articles with mutual interlanguage links.

Topic 68				**Topic 79**			
research	4.56	институт	6.03	goals	4.48	матч	6.02
technology	3.14	университет	3.35	league	3.99	игрок	5.56
engineering	2.63	программа	3.17	club	3.76	сборная	4.51
institute	2.37	учебный	2.75	season	3.49	фк	3.25
science	1.97	технический	2.70	scored	2.72	против	3.20
program	1.60	технология	2.30	cup	2.57	клуб	3.14
education	1.44	научный	1.76	goal	2.48	футболист	2.67
campus	1.43	исследование	1.67	apps	1.74	гол	2.65
management	1.38	наука	1.64	debut	1.69	забивать	2.53
programs	1.36	образование	1.47	match	1.67	команда	2.14
Topic 88				**Topic 251**			
opera	7.36	опера	7.82	windows	8.00	windows	6.05
conductor	1.69	оперный	3.13	microsoft	4.03	microsoft	3.76
orchestra	1.14	дирижер	2.82	server	2.93	версия	1.86
wagner	0.97	певец	1.65	software	1.38	приложение	1.86
soprano	0.78	певица	1.51	user	1.03	сервер	1.63
performance	0.78	театр	1.14	security	0.92	server	1.54
mozart	0.74	партия	1.05	mitchell	0.82	программный	1.08
sang	0.70	сопрано	0.97	oracle	0.82	пользователь	1.04
singing	0.69	вагнер	0.90	enterprise	0.78	обеспечение	1.02
operas	0.68	оркестр	0.82	users	0.78	система	0.96

8. Newman, D., Asuncion, A., Smyth, P., Welling, M.: Distributed algorithms for topic models. J. Mach. Learn. Res. 10, 1801–1828 (Dec 2009)
9. Rubin, T.N., Chambers, A., Smyth, P., Steyvers, M.: Statistical topic models for multi-label document classification. Machine Learning 88(1–2), 157–208 (2012)
10. Smola, A., Narayanamurthy, S.: An architecture for parallel topic models. Proc. VLDB Endow. 3(1–2), 703–710 (Sep 2010)
11. Vorontsov, K.V.: Additive regularization for topic models of text collections. Doklady Mathematics 89(3), 301–304 (2014)
12. Vorontsov, K.V., Potapenko, A.A.: Additive regularization of topic models. Machine Learning, Special Issue on Data Analysis and Intelligent Optimization (2014)
13. Vorontsov, K.V., Potapenko, A.A.: Tutorial on probabilistic topic modeling: Additive regularization for stochastic matrix factorization. In: AIST'2014, Analysis of Images, Social networks and Texts. vol. 436, pp. 29–46. Springer, CCIS (2014)
14. Řehůřek, R., Sojka, P.: Software framework for topic modelling with large corpora. In: Proceedings of the LREC 2010 Workshop on New Challenges for NLP Frameworks. pp. 45–50. ELRA, Valletta, Malta (May 2010)
15. Wang, Y., Bai, H., Stanton, M., Chen, W.Y., Chang, E.Y.: PLDA: Parallel latent Dirichlet allocation for large-scale applications. In: Proceedings of the 5th International Conference on Algorithmic Aspects in Information and Management. pp. 301–314. AAIM '09, Springer-Verlag, Berlin, Heidelberg (2009)