Fine Tuning Latent Dirichlet Allocation for Transfer Learning

Tommy Jones*

27 September 2021

Abstract

words.

1 Introduction

Recent progress in natural language processing (NLP) has been driven by adoption of pre-trained language models. These models belong to the *fine tuning paradigm of transfer learning* where researchers begin with a base mode, $f^{(t-1)}$, which has been pre-trained using data, $\boldsymbol{X}^{(t-1)}$. They then update (i.e., "fine tune") the base model with new data, $\boldsymbol{X}^{(t)}$, to produce a new model, $f^{(t)}$. Popular examples of pre-trained language models include BERT [1], XLM-R [2], GPT-2 [3], GPT-3 [4], and more. This generation of pre-trained language models are all deep neural networks. As a result, they excel at enabling supervised tasks such as summarization, named-entity recognition, classification, etc. but not at unsupervised topic detection where Latent Dirichlet Allocation (LDA) [5] and related models [6][7][8][9] remain popular.

This paper introduces *tLDA*, short for transfer-LDA. tLDA enables use cases for fine-tuning from a base model with a single incremental update (i.e., "fine tuning") or with many incremental updates—e.g., on-line learning, possibly in a time-series context—using Latent Dirichlet Allocation. tLDA uses collapsed Gibbs sampling [10] but its methods should extend to other MCMC methods [11][12][13][14]. tLDA is available for the R language for statistical computing [15] in the *tidylda* package [16].

*PhD Candidate, George Mason University Dept. of Computational and Data Sciences, tjones42@gmu.edu

1.1 Latent Dirichlet Allocation

Latent Dirichlet Allocation is a generative model for word frequencies. Let $\boldsymbol{X}$ be a $D \times V$ matrix of word frequencies where the d, v entries are counts of the v-th word (or more generally, "token") occurring in the d-th document (or more generally, "context"). Under the LDA model, $\boldsymbol{X}$ occurs by sampling from the following process:

1. Generate $\boldsymbol{B}$ by sampling K topics: $\boldsymbol{\beta}_k \sim \text{Dirichlet}(\boldsymbol{\eta}), \forall k \in \{1, 2, ..., K\}$
2. Generate $\boldsymbol{\Theta}$ by sampling D documents: $\boldsymbol{\theta}_d \sim \text{Dirichlet}(\boldsymbol{\alpha}), \forall d \in \{1, 2, ..., D\}$
3. Then for each document, d, and for each word in that document, n
    a. Draw topic $z_{d,n}$ from Categorical($\boldsymbol{\theta}_d$)
    b. Draw word $w_{d,n}$ from Categorical($\boldsymbol{\beta}_{z_{d,n}}$)

This process has a joint posterior of

$$P(\mathbf{W}, \mathbf{Z}, \boldsymbol{\Theta}, \boldsymbol{B} | \boldsymbol{\alpha}, \boldsymbol{\eta}) \propto \left[\prod_{d=1}^{D} \prod_{n=1}^{n_d} P(w_{d,n} | \boldsymbol{\beta}_{z_{d,n}}) P(z_{d,n} | \boldsymbol{\theta}_d) P(\boldsymbol{\theta}_d | \boldsymbol{\alpha}) \right] \left[\prod_{k=1}^{K} P(\boldsymbol{\beta}_k | \boldsymbol{\eta}) \right] \tag{1}$$

The above does not have an analytical closed form. However posterior estimates may be constructed using Gibbs sampling. Key are $D \times K$ matrix $\boldsymbol{Cd}$ and $K \times V$ matrix $\boldsymbol{Cv}$. $\boldsymbol{Cd}$ counts the number of times the k-th topic was sampled in the d-th document and $\boldsymbol{Cv}$ counts the number of times the k-th topic was sampled at the v-th word. Gibbs sampling is depicted in Algorithm 1.

Algorithm 1: Allocate $\boldsymbol{Cd}$ and $\boldsymbol{Cv}$ by sampling

//Initialize $\boldsymbol{Cd}$, $\boldsymbol{Cv}$ as zero-valued matrices;

for *each document, d* **do**

  for *each word in the d-th context, n* **do**

    sample z such that $P(z = k) \sim \text{Uniform}(1, K)$;

    $Cd_{d,z}$ += 1;

    $Cv_{z,n}$ += 1;

  end

end

//Begin Gibbs sampling ;

for *each iteration, i* **do**

  for *each document, d* **do**

    for *each word in the d-th context, n* **do**

      $Cd_{d,z^{(i-1)}}$ -= 1, $Cv_{z^{(i-1)},n}$ -= 1;

      sample z such that $P(z = k) = \frac{Cv_{k,n}+\eta_n}{\sum_{v=1}^{V} Cv_{k,v}+\eta_v} \cdot \frac{Cd_{d,k}+\alpha_k}{\left(\sum_{k=1}^{K} Cd_{d,k}+\alpha_k\right)-1}$;

      $Cd_{d,z^{(i)}}$ += 1, $Cv_{z^{(i)},n}$ += 1;

    end

  end

end

Once sampling is complete, one can derive posterior estimates with

$$\hat{\theta}_{d,k} = \frac{Cd_{d,k} + \alpha_k}{\sum_{k=1}^{K} Cd_{d,k} + \alpha_k} \tag{2}$$

$$\hat{\beta}_{k,v} = \frac{Cv_{k,v} + \eta_v}{\sum_{v=1}^{V} Cv_{k,v} + \eta_v} \tag{3}$$

1.2 Related Work

Work on transfer lerning with LDA and other probabilistic topic models falls into three categories. The first category contains topic models that explicitly model a topic's evolution over time [7][17][18]. These models differ from true transfer learning in that time is explicitly part of the model, rather than being updated post-hoc. The second category contains models that allow external information to guide the development

of topics. External information may be in the form of supervised outcomes [19] [20] [21], seeded by model structure [22], seeded in the prior [23], or constructed interactively with subject matter experts [24]. The third category contains models designed for incremental and on-line learning [25][26][27].

1.3 Contribution

tLDA is a model for updating topics in an existing model with new data, enabling incremental updates and time-series use cases. tLDA has three characteristics differentiating it from previous work:

1. Flexibility - Most prior work can only address use cases from one of the above categories. In theory, tLDA can address all three. However, exploring use of tLDA to encode expert input into the $\boldsymbol{\eta}$ prior is left to future work.
2. Tunability - tLDA introduces only a single new tuning parameter, a. Its use is intuitive, balancing the ratio of tokens in $\boldsymbol{X}^{(t)}$ to the base model's data, $\boldsymbol{X}^{(t-1)}$.
3. Analyticsl - tLDA allows data sets and model updates to be chained together preserving the Markov property, enabling analytical study through incremental updates.

2 tLDA

2.1 The Model

Formally, tLDA can be stated as

$$z_{d,n}|\boldsymbol{\theta}_d \sim \text{Categorical}(\boldsymbol{\theta}_d) \tag{4}$$

$$w_{d,n}|z_k, \boldsymbol{\beta}_k^{(t)} \sim \text{Categorical}(\boldsymbol{\beta}_k^{(t)}) \tag{5}$$

$$\boldsymbol{\theta}_d \sim \text{Dirichlet}(\boldsymbol{\alpha}_d) \tag{6}$$

$$\boldsymbol{\beta}_k^{(t)} \sim \text{Dirichlet}(\omega_k^{(t)} \cdot \mathbb{E}\left[\boldsymbol{\beta}_k^{(t-1)}\right]) \tag{7}$$

The above indicates that tLDA places a matrix prior for words over topics where $\boldsymbol{\eta}_k^{(t)} = \omega_k^{(t)} \cdot \mathbb{E}\left[\boldsymbol{\beta}_k^{(t-1)}\right]$. Because the posterior at time t depends only on data at time t and the state of the model at time $t-1$, tLDA models retain the Markov property.

2.1.1 Selecting the prior weight

Each $\omega_k^{(t)}$ tunes the relative weight between the base model (as prior) and new data in the posterior for each topic. This specification introduces K new tuning parameters and setting $\omega_k^{(t)}$ directly is possible but not intuitive. Yet introducing a new parameter and performing some algebra collapses these K tuning parameters into a single parameter with several intutive critical values. This tuning parameter, $a^{(t)}$, is related to each ω_k^t as follows:

$$\omega_k^{(t)} = a^{(t)} \cdot \sum_{v=1}^{V} Cv_{k,v}^{(t-1)} + \eta_{k,v}^{(t-1)} \tag{8}$$

Appendix 1 shows the full derivation of the relationship between $a^{(t)}$ and $\omega_k^{(t)}$.

When $a^{(t)} = 1$, fine tuning is equivalent to adding the data in $\boldsymbol{X}^{(t)}$ to $\boldsymbol{X}^{(t-1)}$. In other words, each word occurrence in $\boldsymbol{X}^{(t)}$ carries the same weight in the posterior as each word occurrence in $\boldsymbol{X}^{(t-1)}$. If $\boldsymbol{X}^{(t)}$ has more data than $\boldsymbol{X}^{(t-1)}$, then it will carry more weight. If it has less, it will carry less.

When $a^{(t)} < 1$, then the posterior has recency bias. Each word occurrence in $\boldsymbol{X}^{(t)}$ carries more weight than each word occurrence in $\boldsymbol{X}^{(t-1)}$. When When $a^{(t)} > 1$, then the posterior has precedent bias. Each word occurrence in $\boldsymbol{X}^{(t)}$ carries less weight than each word occurrence in $\boldsymbol{X}^{(t-1)}$.

Another pair of critical values are $a^{(t)} = \frac{N^{(t)}}{N^{(t-1)}}$ and $a^{(t)} = \frac{N^{(t)}}{N^{(t-1)} + \sum_{d,v} \eta_{d,v}}$, where $N^{(\cdot)} = \sum_{d,v} X_{d,v}^{(\cdot)}$. These put the total number of word occurrences in $\boldsymbol{X}^{(t)}$ and $\boldsymbol{X}^{(t-1)}$ on equal footing excluding and including $\boldsymbol{\eta}^{(t-1)}$, respectively. These values may be useful when comparing topical differences between a baseline group in $\boldsymbol{X}^{(t-1)}$ and "treatment" group in $\boldsymbol{X}^{(t)}$, though this use case is left to future work.

2.2 The Algorithm

The overall tLDA algorithm proceeds in 6 steps.

1. Construct $\boldsymbol{\eta}^{(t)}$
2. Predict $\boldsymbol{\Theta}^{(t)}$ using topics from $\boldsymbol{B}^{(t-1)}$
3. Align vocabulary
4. Add new topics
5. Initialize $\boldsymbol{Cd}^{(t)}$ and $\boldsymbol{Cv}^{(t)}$

6. Begin Gibbs sampling with $P(z = k) = \frac{Cv_{k,n}+\eta_{k,n}}{\sum_{v=1}^{V} Cv_{k,v}+\eta_{k,v}} \cdot \frac{Cd_{d,k}+\alpha_k}{\left(\sum_{k=1}^{K} Cd_{d,k}+\alpha_k\right)-1}$

Any real-world application of tLDA presents several practical issues which are addressed in steps 3 - 5, described in more detail below. These issues include: the vocabularies in $\boldsymbol{X}^{(t-1)}$ and $\boldsymbol{X}^{(t)}$ will not be identical; users may wish to add topics, expecting $\boldsymbol{X}^{(t)}$ to contain topics not in $\boldsymbol{X}^{(t-1)}$; and $\boldsymbol{Cd}^{(t)}$ and $\boldsymbol{Cv}^{(t)}$ should be initialized proportional to $\boldsymbol{Cd}^{(t-1)}$ and $\boldsymbol{Cv}^{(t-1)}$, respectively.

2.2.1 Aligning Vocabulary

tLDA implements an algorithm to fold in new words. This method slightly modifies the posterior probabilities in $\boldsymbol{B}^{(t-1)}$ and adds a non-zero prior by modifying $\boldsymbol{\eta}^{(t)}$. It involves three steps. First, append columns to $\boldsymbol{B}^{(t-1)}$ and $\boldsymbol{\eta}^{(t)}$ that correspond to out-of-vocabulary words. Next, set the new entries for these new words to some small value, $\epsilon > 0$ in both $\boldsymbol{B}^{(t-1)}$ and $\boldsymbol{\eta}^{(t)}$. Finally, re-normalize the rows of $\boldsymbol{B}^{(t-1)}$ so that they sum to one. For computational reasons, ϵ must be greater than zero. The *tidylda* implementation chooses ϵ to the lowest decile of all values in $\boldsymbol{B}^{(t-1)}$ or $\boldsymbol{\eta}^{(t)}$, respectively.

2.2.2 Adding New Topics

tLDA employs a similar method to add new, randomly initialized, topics if desired. This is achieved by appending rows to both $\boldsymbol{\eta}^{(t)}$ and $\boldsymbol{B}^{(t)}$, adding entries to $\boldsymbol{\alpha}$, and adding columns to $\boldsymbol{\Theta}^{(t)}$, obtained in step two above. The tLDA implementation in *tidylda* sets the rows of $\boldsymbol{\eta}^{(t)}$ equal to the column means across previous topics. Then new rows of $\boldsymbol{B}^{(t)}$ are the new rows of $\boldsymbol{\eta}^{(t)}$ but normalized to sum to one. This effectively sets the prior for new topics equal to the average of the weighted posteriors of pre-existing topics.

The choice of setting the prior to new topics as the average of pre-existing topics is admittedly subjective. A uniform prior over words is unrealistic, being inconsistent with Zipf's law [28]. (See also the appendix in [29].) The average over existing topics is only one viable choice. Another choice might be to choose the shape of new $\boldsymbol{\eta}_k$ from an estimated Zipf's coefficient of $\boldsymbol{X}^{(t)}$ and choose the magnitude by another means.

New entries to $\boldsymbol{\alpha}$ are initialized to be the median value of the pre-existing topics in $\boldsymbol{\alpha}$. Similarly, columns are appended to $\boldsymbol{\Theta}^{(t)}$. Entries for new topics are taken to be the median value for pre-existing topics on a per-document basis. This effectively places a uniform prior for new topics. This choice is also subjective. Other heuristic choices may be made, but it is not obvious that they would be any better or worse choices.

2.2.3 Initializing $\boldsymbol{Cd}^{(t)}$ and $\boldsymbol{Cv}^{(t)}$

Like most other LDA implementations, tLDA initializes tokens for $\boldsymbol{Cd}^{(t)}$ and $\boldsymbol{Cv}^{(t)}$ with a single Gibbs iteration. However, instead of sampling from a uniform random for this initial step, tLDA draws a topic for the n-th word of the d-th document from the following:

$$P(z_{d,n} = k) = \boldsymbol{\beta}_{k,n}^{(t)} \cdot \boldsymbol{\theta}_{d,k}^{(t)} \tag{9}$$

After a single iteration, the number of times each topic was sampled at each document and word occurrence is counted to produce $\boldsymbol{Cd}^{(t)}$ and $\boldsymbol{Cv}^{(t)}$. After initialization where topic-word distributions are fixed, tLDA continues in a standard fashion, recalculating $\boldsymbol{Cd}^{(t)}$ and $\boldsymbol{Cv}^{(t)}$ (and therefore $P(z_{d,n} = k)$) at each step.

3 Experiments

3.1 Simulation Analysis

3.1.1 Simulation Setup

3.1.2 Findings

3.2 Emperical Analysis

3.2.1 Data Description

3.2.2 Findings

4 Discussion

Note that you haven't explored adding expert input, but this could theoretically be encoded into the prior

5 Appendix 1

6 Appendix 2

References

[1] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding," *arXiv*, 2018.

[2] A. Conneau *et al.*, "Unsupervised Cross-lingual Representation Learning at Scale," *arXiv*, 2019.

[3] A. Radford, J. Wu, R. Child, D. Luan, D. Amodei, and I. Sutskever, "Language Models are Unsupervised Multitask Learners," *OpenAI blog*, vol. 1, no. 8, p. 9, 2019.

[4] T. B. Brown *et al.*, "Language Models are Few-Shot Learners," *arXiv*, 2020.

[5] D. M. Blei, A. Y. Ng, and M. I. Jordan, "Latent Dirichlet Allocation," *Journal of Machine Learning Research*, vol. 3, 2003.

[6] Y. W. Teh, M. I. Jordan, M. J. Beal, and D. M. Blei, "Hierarchical Dirichlet Processes," *Journal of the American Statistical Association*, vol. 101, no. 476, pp. 1566–1581, 2012, doi: 10.1198/016214506000000302.

[7] D. M. Blei and J. D. Lafferty, "Dynamic topic models," pp. 113–120, 2006, doi: 10.1145/1143844.1143859.

[8] D. M. Blei and J. D. Lafferty, "A correlated topic model of Science," *The Annals of Applied Statistics*, vol. 1, no. 1, pp. 17–35, 2007, doi: 10.1214/07-aoas114.

[9] M. E. Roberts, B. M. Stewart, and D. Tingley, "stm : An R Package for Structural Topic Models," *Journal of Statistical Software*, vol. 91, no. 2, 2019, doi: 10.18637/jss.v091.i02.

[10] T. L. Griffiths and M. Steyvers, "Finding scientific topics," *Proceedings of the National Academy of Sciences*, vol. 101, no. Supplement 1, pp. 5228–5235, 2004, doi: 10.1073/pnas.0307752101.

[11] L. Yao, D. Mimno, and A. McCallum, "Efficient methods for topic model inference on streaming document collections," pp. 937–946, 2009, doi: 10.1145/1557019.1557121.

[12] J. Yuan *et al.*, "LightLDA: Big Topic Models on Modest Computer Clusters," in *Proceedings of the 24th international conference on world wide web*, 2015, p. 1351—1361.

[13] A. Ahmed, M. Aly, J. Gonzalez, S. Narayanamurthy, and A. J. Smola, "Scalable inference in latent variable models," pp. 123–132, 2012, doi: 10.1145/2124295.2124312.

[14] J. Chen, K. Li, J. Zhu, and W. Chen, "WarpLDA: a Cache Efficient O(1) Algorithm for Latent Dirichlet Allocation," *arXiv*, 2015.

[15] R. C. Team, *R: A Language and Environment for Statistical Computing*. Vienna, Austria, 2013.

[16] T. Jones, "tidylda." 2021, [Online]. Available: https://CRAN.R-project.org/package=tidylda.

[17] C. Wang, D. Blei, and D. Heckerman, "Continuous Time Dynamic Topic Models," *arXiv*, 2012.

[18] Y. Wang, E. Agichtein, and M. Benzi, "TM-LDA: Efficient online modeling of latent topic transitions in social media," in *Proceedings of the 18th ACM SIGKDD international conference on knowledge discovery and data mining*, 2012, pp. 123–131.

[19] J. McAuliffe and D. Blei, "Supervised topic models," *Advances in neural information processing systems*, vol. 20, p. 121—128, 2007.

[20] D. Ramage, D. Hall, R. Nallapati, and C. D. Manning, "Labeled LDA: A supervised topic model for credit attribution in multi-labeled corpora," in *Proceedings of the 2009 conference on empirical methods in natural language processing*, 2009, p. 248—256.

[21] D. Andrzejewski and X. Zhu, "Latent Dirichlet Allocation with Topic-in-Set Knowledge," in *Proceedings of the NAACL HLT 2009 workshop on semi-supervised learning for natural language processing*, 2009, p. 43—48.

[22] J. Jagarlamudi, R. Udupa, and H. D. III, "Incorporating Lexical Priors into Topic Models," 2012.

[23] D. Andrzejewski, X. Zhu, and M. Craven, "Incorporating domain knowledge into topic modeling via Dirichlet Forest priors," pp. 25–32, 2009, doi: 10.1145/1553374.1553378.

[24] Y. Hu, J. Boyd-Graber, B. Satinoff, and A. Smith, "Interactive topic modeling," *Machine Learning*, vol. 95, no. 3, pp. 423–469, 2014, doi: 10.1007/s10994-013-5413-0.

[25] L. AlSumait, D. Barbará, and C. Domeniconi, "On-Line LDA: Adaptive Topic Models for Mining Text Streams with Applications to Topic Detection and Tracking," *2008 Eighth IEEE International Conference on Data Mining*, pp. 3–12, 2008, doi: 10.1109/icdm.2008.140.

[26] M. D. Hoffman, D. M. Blei, and F. Bach, "Online Learning for Latent Dirichlet Allocation," *advances in neural information processing systems*, vol. 23, p. 856—864, 2010.

[27] J. Rieger, C. Jentsch, and J. Rahnenfuhrer, "RollingLDA: An Update Algorithm of Latent Dirichlet Allocation to Construct Consistent Time Series from Textual Data," 2021, [Online]. Available: http://www.statistik.tu-dortmund.de/fileadmin/user/_upload/Lehrstuehle/IWuS/Forschung/rollinglda.pdf.

[28] G. K. Zipf, *Human behavior and the principle of least effort.* Oxford, England: Addison-Wesley Press, 1949.

[29] T. Jones, "A coefficient of determination for probabilistic topic models," *arXiv preprint arXiv:1911.11061*, 2019.